Prof. Florian Gallwitz, Sommersemester 2023

# Medienverarbeitung

# Organisatorisches

# Über mich

Prof. Florian Gallwitz

Schwerpunkte:
- Medieninformatik
- Mustererkennung
- Bildverarbeitung und Spracherkennung
- Deep Learning

florian.gallwitz@ohm-hochschule.de
Büro: HQ518
Tel. 0911/5880-1667

Home Page: http://th-nuernberg.de
➔ Fakultäten
➔ Informatik
➔ Professoren
➔ Gallwitz

Sprechstunde: nach Vereinbarung per E-Mail

# Medieninformatik an der Ohm-Hochschule

# Modulbeschreibung „Medienverarbeitung“

## Medienverarbeitung

| | |
|---|---|
| Studiengang | Bachelor Medieninformatik<br><br>2. Studienabschnitt (3. – 7. Studiensemester)<br>Modulgruppe Digitale Medien<br>Medienanalyse |
| Modul: | Medienverarbeitung |
| Modulverantwortliche(r): | Prof. Dr. Gallwitz |
| Vorkenntnisse: | Mathematik III |
| Arbeitsaufwand: | 210 Stunden, davon:<br>95 Stunden Präsenzzeit,<br>115 Stunden Vor- und Nachbereitung des Lehrstoffs, Übungsaufgaben und Prüfungsvorbereitung |
| Leistungspunkte: | 7 |
| Semesterwochenstunden: | 6 |
| Veranstaltungstyp: | 4 SWS Vorlesung, 2 SWS Übung |
| Semesterturnus: | Sommersemester |
| Unterrichtssprache: | Kurs nur in Deutsch |
| Beitrag zu den Zielen des Studiengangs: | Die Fähigkeiten zur spezialisierten Anwendungsentwicklung werden maßgeblich entwickelt. |
| Lehrziel: | Kenntnis der Grundlagen und Methoden der Bild-, Video- und Audioverarbeitung, Fähigkeit zur Entwicklung von Anwendungen zur Aufnahme, Verarbeitung und Analyse von Mediendaten. |
| Schlüsselqualifikationen: | Analyse und Klassifikation von Problemen, kreatives Problemlösen. |
| Lehrinhalte: | Bild-, Video-, Audioverarbeitung (Erfassung und Verbesserung von Medien).<br>Mustererkennung (automatische Klassifikation mit extrahierten Merkmalen).<br>Bild- und Sprachverstehen (Computer Vision, rechnergestütztes Erkennen von Bildinhalten). |
| Bemerkungen: | |
| Leistungsnachweis: | Schriftliche Prüfung (90 min) |



© 2009 Fakultät Informatik

# Medienverarbeitung 2023

Vorlesung: **(erster Termin: 16.3.2023)**

- Mittwochs 09:45 – 11:15, HQ.007
- Donnerstags 09:45 – 11:15, HQ.007

Übungen: **(erster Termin: 22.3.2023)**

- Mittwochs 11:30 – 13:00 , HQ.211
- Freitags 09:45 – 11:15, HQ.205

- schriftliche Prüfung gegen Ende des Semesters
- zugelassene Hilfsmittel: „Taschenrechner, nicht programmierbar“

# Übungen

- Übungsblätter online (i.d.R. am Abend vorher)
- Bearbeitung der Aufgaben in 3er-Gruppen
- Gruppeneinteilung erfolgt für jede Übung neu
- Aufgaben haben praktischen Charakter
  - Umgang mit Software zur Medienverarbeitung und Deep Learning (u.a. OpenCV, Keras/Tensorflow)
  - Programmieraufgaben (Programmiersprache i.d.R. nicht vorgegeben, Python oder C++ für OpenCV, Python für Keras)
  - Nutzung von ChatGPT?

- Während der Übungen stehe ich für Fragen zur Verfügung.

# Literatur

A. Nischwitz, M. Fischer, P. Haberäcker: *Computergrafik und Bildverarbeitung*, Vieweg-Teubner, 2011

873 S., 54,95 €

# Literatur

A. Kaehler and G. Bradsky: *Learning OpenCV 3 – Computer Vision in C++ with the OpenCV Library*, O'Reilly, 2016

992 S., ca. 50.- €

Die 2., aktualisierte Auflage war schon jahrelang angekündigt, ist Anfang 2017 endlich erschienen. Die 1. Auflage beschreibt noch das alte C-Interface, nicht das neuere C++-Interface.

# Vorlesung

- Foliensatz stelle ich online zur Verfügung
- Die Folien sollen vom Mitschreiben während der Vorlesung entlasten.
- Das Mitschreiben wird dadurch nicht überflüssig.
- Die Folien sind kein Lehrbuch.
- Die Folien sind daher im allgemeinen nur mit den Erläuterungen während der Vorlesung und entsprechenden eigenen Notizen verständlich

# 1. Einführung

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - **Grundbegriffe**
   - Anwendungen
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Begriff „Medien“

Zu unterscheiden sind:

**Gesellschaftliche Medien**

Sichtweise der Kommunikationswissenschaft, Soziologie etc.

Ganzheitliche Betrachtung komplexer Kommunikations-formen, z.B. „Medium Zeitung“ oder „Medium Rundfunk“

Analog „Medium Internet“ oder „Medium WWW“

**Technische Medien**

Sichtweise der Informatik und der Nachrichtentechnik

Betrachtung von (integrierbaren) Einzelmedien, z.B. „Medium Text“ oder „Medium Ton“

Spezieller: „Medium MPEG-Strom“ oder „Medium JPEG-Bild“

# Medienanalyse?

Artikel | Diskussion | Lesen | Bearbeiten | Versionsgeschichte | Suche

## Medienanalyse

Die **Medienanalyse** ist ein Forschungsfeld der Kommunikations- und Medienwissenschaft und befasst si
dem Medium an sich aus verschiedenen Perspektiven (z. B. ausgehend vom Rezipienten).

Um einen groben Überblick über die Möglichkeiten zu geben, die **medienanalytische Verfahren** bieten,
einige methodologische Vorgehensweisen vorgestellt. Dazu sollen verschiedene Theorien kurz umrissen
die zum Verständnis der auf ihnen aufbauenden Analysemethoden unerlässlich sind. Die verschiedenen
zur Medienanalyse unterscheiden sich nicht nur hinsichtlich ihrer methodischen Vorgehensweise sondern
allem durch ihre erkenntnistheoretischen Hintergründe. Ihr Verständnis soll als Grundlage für die Differen
der unterschiedlichen methodischen Herangehensweisen dienen.

**Inhaltsverzeichnis** [Verbergen]


1 Hermeneutik
2 Handlungsorientierte Medienanalyse
3 Rezipientenorientierte Medienanalyse
4 Diskursanalyse
5 Literatur


**ist hier nicht gemeint!**

# Begriffsbestimmung „Medienanalyse“

- „Eine **Analyse** ist eine systematische Untersuchung, bei der das untersuchte Objekt oder Subjekt in seine Bestandteile zerlegt wird und diese anschließend geordnet, untersucht und ausgewertet werden. Dabei dürfen die Vernetzung der einzelnen Elemente und deren Integration nicht außer Acht gelassen werden.“
- „Das Gegenteil der Analyse – unter dem Aspekt des „Auflösens in Einzelbestandteile" – ist die **Synthese** (‚Zusammensetzen‘).“

Quelle: Wikipedia

# Begriffsbestimmung „Medienverarbeitung“

- „**Datenverarbeitung** (DV) bezeichnet den organisierten Umgang mit Datenmengen mit dem Ziel, Informationen über diese Datenmengen zu gewinnen oder diese Datenmengen zu verändern.“

Quelle: Wikipedia

# Definition „Medienverarbeitung“

- **Medienverarbeitung** ist der systematische Umgang mit Mediendaten (insb. Bildern und Audiosignalen) mit dem Ziel, diese zu *verändern* oder *Informationen aus diesen Daten zu gewinnen*. Medienverarbeitung umfasst insbesondere die Bereiche **Bildverarbeitung** und **Sprachverarbeitung**.

**Veränderung** von Medien, z.B.:

- Bildverbesserung (*Image Enhancement*)
- Rauschunterdrückung
- *Time-Stretching* bzw. *Pitch-Shifting*

Gewinnen von **Informationen** aus Mediendaten, z.B.

- Gesichtserkennung
- Handschrifterkennung
- Spracherkennung

Das Gewinnen von Informationen aus Mediendaten bezeichnet man als **Mustererkennung** (*Pattern Recognition*).

# Veränderung von Medien: Bsp. Bildverbesserung

Hier:
Medianfilterung
zur Rausch-
unterdrückung

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Handschrifterkennung (Windows 8)

# Autonome Steuerung eines Quadrocopters

# Smartphone-Bedienung (Apple Siri)

# Sprachsteuerung von Google Glass

# Analyse von medizinischen Bilddaten

# Autonomes Fahren

https://www.youtube.com/watch?v=PjH_1hEoIDs

# Objekterkennung- und –segmentierung mit Mask R-CNN

https://www.youtube.com/watch?v=hzR-wHuM0dE

# Umwandlung von Skizzen in fotorealistische Bilder mit GANs

https://www.youtube.com/watch?v=p5U4NgVGAwg

# Definition „Mustererkennung“

- **Mustererkennung** (*Pattern Recognition*) ist die automatische Transformation eines Sensorsignals in eine aufgabenspezifische symbolische Beschreibung. Der Begriff Mustererkennung umfasst
  - die **Klassifikation** einfacher Muster, z.B. die Erkennung einzelner Buchstaben, Wörter oder Gesichter → einziges Label reicht um das Muster zu beschreiben
  - die **Musteranalyse,** d.h. die Analyse komplexer Muster. Hierzu zählen z.B. das Verstehen gesprochener Sprache und das „Maschinelle Sehen“ (*Computer Vision*), z.B. zur Robotersteuerung.

# Beispielanwendungen - Übersicht

| **Anwendung** | **Sensor** | **Muster** | **Symbolische Beschreibung** |
|---|---|---|---|
| Schriftzeichenerkennung | Scanner/Kamera | f(x,y) (z.B. Grauwerte) | z.B. ASCII-Zeichen |
| Erkennung von Werkstücken | Kamera | f(x,y) bzw. f(x,y,t) (z.B. Grauwerte) | Typ bzw. Lage des Werkstücks |
| Verkehrszeichen-Erkennung | Kamera | f(x,y) bzw. f(x,y,t) (z.B. RGB-Farbwerte) | Art des Verkehrszeichens |
| Handschrifterkennung | Scanner/Kamera, Stift mit Drucksensor oder Touch-Display | f(x,y) bzw. f(t) (Grauwerte bzw. Koordinaten und ggf. Anpressdruck) | Text (z.B. in ASCII) |
| Erkennung gesprochener Worte | Mikrophon | f(t) (Amplitude) | Wort (in ASCII) bzw. Wortnummer |
| Sprecher-Erkennng | Mikrophon | f(t) (Amplitude) | Sprecher-ID |
| Sprachverstehen | Mikrophon | f(t) (Amplitude) | Semantische Repräsentation |
| Autonomes Fahren | Kamera, Radar | f(x,y,t) bzw. f(x,y,t) (RGB-Farbwerte + Tiefenwerte) | z.B. Lenk- und Bremseingriffe |
| Analyse von medizinischen Bilddaten | Röntgenbilder, CT-Bilder, MRT-Bilder, CT-Bildfolgen etc. | f(x,y), f(x,y,z,), f(x,y,z,t) (z.B. Grauwerte) | z.B. Umrisse von Organen oder Tumoren |

# Klassifikation einfacher Muster

# Klassifikation einfacher Muster

- Das Muster repräsentiert ein Objekt dieser Welt
- Jedes Objekt gehört (genau) einer Klasse an
- Es stehen endlich viele Klassen (Kategorien) zur Auswahl
- Beispiele:
  - Schriftzeichenerkennung
  - Sprecheridentifikation
  - Einzelworterkernnung

# Musteranalyse

- Das Muster ist *komplex strukturiert*
- Die *Einzelteile des Musters* werden klassifiziert
- Die *Relationen zwischen den Komponenten* werden erfasst
- Beispiele:
  - Automatische Erkennung kontinuierlicher Sprache, Sprachverstehen
  - Auswertung eines Satellitenbilds, einer Straßenszene o.ä.

**Symbolische Beschreibung:**
Hierarchische Struktur, die kompatibel mit dem gespeicherten **Wissen** ist und optimal zu den **Sensordaten** des Musters passt.

# Musteranalyse

- Musteranalyse umfasst i.d.R. die Teilprobleme **Klassifikation** und **Segmentierung**, die beim heutigen Stand der Technik meist gemeinsam gelöst werden.
- **Segmentierung**: Zusammenfassung inhaltlich zusammenhängender benachbarter Bildpunkte (oder Audio-Samples etc.) zu Regionen (nach einem anwendungsspezifischen Homogenitätskriterium)

Bild: https://medium.freecodecamp.org/mask-r-cnn-explained-7f82bec890e3

# Was ist ein Muster?

**Norbert Wiener** (amerik. Mathematiker und Philosoph)**:**

*One of the most interesting aspects of the world is that it can be considered to be made up of* ***patterns****.*

*A pattern is essentially an* ***arrangement****.*

*It is characterized by the* ***order*** *of the elements of which it is made rather than by the intrinsic nature of these elements.*

DNA-Sequenzierung

# Benachbarte Disziplinen

# Benachbarte Disziplinen

# Benachbarte Disziplinen

# Benachbarte Disziplinen

# Benachbarte Disziplinen

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - Anwendungen
   - **Überblick**
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# 2. Vorverarbeitung

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - **Abtastung und PCM**
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Mikrofon

Beispiel Kondensatormikrofon: Kondensator -> speichert Ladung

- Spannungsquelle erzeugt Potentialgefälle zwischen Membran und Gegenelektrode => Plattenkondensator
- Kapazität schwankt bei auftreffender Schallwelle
- Daraus resultiert eine Spannungsschwankung (elektr. Signal)

# Analoge Signale

- Ein **Signal**
  - ist die deterministische Änderung einer physikalischen Größe (über Raum und/oder Zeit)
  - trägt Information durch Raum und Zeit.
- Im allgemeinen sind physikalische Größen *kontinuierlich* (d.h. durch *stetige* Funktionen darstellbar).
  - Ausnahme: z.B. Quantenphysik
- Signale mit kontinuierlichem Verlauf (d.h. als stetige Funktionen modellierbar) heißen **analog**.
- In analogen Signalen sind prinzipiell beliebig genaue Beobachtungen möglich.
- Analoge Signale sind anfällig gegen Störungen und damit Informationsverluste (z.B. beim Kopieren).

# Elektrische Signale und Digitalisierung

- Die digitale Verarbeitung basiert in der Regel auf analogen Signalen elektrischen Stroms (bzw. elektr. Spannung)
- andere Signalarten werden umgewandelt
- Beispiele
  - Mikrofon und Lautsprecher
  - Fotodiode und LED
  - Halbleiter-Temperatursensoren

# Abtastung

# Digitalisierung

- die Abtastung (**Sampling**) in einem festen Raster bezeichnet man auch als **Diskretisierung**

  messen an diskreten Punkten

- **Digitalisierung** = Diskretisierung + Quantisierung

  beides ist

- die Folge von quantisierten Abtastwerten wird im Rechner bzw. auf digitalen Speichermedien als Folge von Binärzahlen repräsentiert, d.h. als eine **Folge von Bits**

# Abtastung

- elektrisches Signal (z.B. aus dem Mikrofon) muss für die Verarbeitung durch einen Computer zunächst digitalisiert werden

# Signale als Überlagerung von Sinusschwingungen

# Abtasttheorem

Ein kontinuierliches, bandbegrenztes Signal, mit einer Minimalfrequenz von 0 Hz und einer Maximalfrequenz $f_{max}$, muss mit einer Frequenz größer als $2 \cdot f_{max}$ abgetastet werden, damit man aus dem so erhaltenen zeitdiskreten Signal das Ursprungssignal rekonstruieren kann.

min. 2 * fmax abtasten -> in Periode der höchsten Frequenz 2 Abtastwerte

kurz:

Die **Abtastfrequenz** muss **mindestens doppelt so groß** sein wie der höchste im Signal enthaltene Frequenzanteil.

# Frequenzgang eines realen Tiefpassfilters

lässt nur Tiefe Frequenzen durch
-> 20 kHz => Abtastrate 40 kHz man will nur für die CD Frequenzen bis 20 kHz haben (Tiefpassfilter)
Für CDs meistens bisschen höher gesetzt 22,5 kHz, da die Filter nicht genau auf der Grenze sind

# Abtasttheorem: Aliasing (untere Bildhälfte)

Bild: Peterpall, CC by-sa

# Abtasttheorem (Audiobeispiel)

Ton, dessen Frequenz von ca. 100 Hz bis 8000 Hz linear zunimmt

1. Abtastung mit 16kHz:

2. Abtastung des gleichen Signals mit 8 kHz (Unterabtastung):

1. Tiefpassfilter 4 kHz, dann Abtastung mit 8 kHz:

# Abtasttheorem – Moiré-Effekt

akte durch Verletzung des Abtasttheorems

abtastung des roten Signals und Rekonstruktion durch

fert

al:

aufnahme für ein IVR-System liegt als 16 kHz-Aufnahme vor

R-System wird aber ein 8 kHz-Signal benötigt (Telefon)

atz: Jeden zweiten Abtastwert übernehmen

etzung des Abtasttheorems! Folge: Klirren oder Pfeifen

mit $f_{max}$ = 4 kHz

# Abtasttheorem (Video)

# Abtasttheorem und PowerPoint

# Abtasttheorem und PowerPoint (Ausschnitt)

# Quantisierung (I)

# Quantisierung (II)

- Die Abtastwerte sind nicht kontinuierlich, sondern diskret
- Die Rundung führt zum sog. Quantisierungsrauschen
- Mehr Quantisierungsstufen => weniger Quantisierungsrauschen
- Die Abtastung mit konstanter Abtastrate, Quantisierung der Abtastwerte und anschließende Codierung (i.d.R. im Binärcode) bezeichnet man auch als **Pulse Code Modulation (PCM)**.

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - **Digitale Audiodaten**
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Audio-Signale: Physikalische Grundlagen

- Schall: Druckwelle in einem physikalischen Medium (z.B. Luft oder Wasser)

- Ausbreitungsgeschwindigkeit $c$ in der Luft: ca. 343 m/s (bei 20° C und 1013 hPa), in Wasser: ca. 1484 m/s, in Diamant: ca. 18000 m/s

- Viele akustische Signale sind (in kurzen Zeitabschnitten) annähernd periodisch, z.B.
  - der Ton einer Flöte
  - ein Vokal in einem gesprochenen Wort

- Wesentlich bei periodischen Signalen: **Amplitude** und **Periodenfrequenz** $f$ (bzw. **Periodendauer** $1/f$ bzw. **Wellenlänge** $\lambda$**)**

$$f = \frac{c}{\lambda}$$

# Hörbarer Frequenzbereich

Hörbereich des Menschen 20Hz - 20 kHz

**Abbildung 4.2:** Hörbarer Frequenzbereich: Sprache nutzt nur einen Teil der hörbaren Frequenzen aus.

80 Hz 160 Hz 320 Hz 640 Hz 1280 Hz 2560 Hz 5120 Hz 10240 Hz

- Die Wellenlängen liegen in der Größenordnung von Alltagsgegenständen
- Schallwellen unterschiedlicher Frequenz werden in unterschiedlichem Maße von Objekten reflektiert oder abgeschattet

# Verlustfreie Audio-Codierung

- Klassisches Verfahren: Pulse Code Modulation (PCM)
- Beispiele:
  - G.711, 8000 Hz x 8 Bit = 64 kBit/s,
    - Codierung der Abtastwerte mit log. Kennlinie (μ-law bzw. a-law)
    - Verwendung: ISDN, VoIP
  - Audio-CD: 44,1 kHz x 16 Bit x 2 Kanäle (Stereo) = 1.411.200 Bit/s
    - Codierung der Abtastwerte mit linearer Kennlinie
- DPCM (Differential PCM): Speicherung der Differenz zweier aufeinanderfolgender Abtastwerte
- ADPCM (Adaptive DPCM): Vorhersage des zukünftigen Signalverlaufs und Speicherung der Differenz zur Vorhersage.
  - Anwendungsbeispiel: DECT-Telefonie

# Logarithmische Kennlinie (μ-law)

- Werte sind in der Sprache exponentiell verteilt ⇒ logarithmische Verzerrung ist effizienter:
  - Auflösung kleiner Amplituden erhöht
  - Bereiche darüber werden komprimiert
  - Gleichverteilung der Werte

# MP3

- MP3: Entwickelt in Erlangen (Fraunhofer Institut für Integrierte Schaltungen + Friedrich-Alexander-Universität) ab 1982
- Seit 1992 Teil des MPEG-1-Video-Standards (MPEG = Movie Picture Encoding Group)
- „MP3“: zu MPEG-1 Audio Layer III
- Verfahren nutzt psychoakustische Effekte (Maskierung)
- Seit Mitte der 90er Jahr ermöglicht MP3 den breiten Austausch von Musikdateien über das Internet.
- Seit 1998 portable MP3-Player auf dem Markt
- Seit 1998 verlangen die Patentinhaber Lizenzgebühren von den Hard- und Softwareherstellern => verstärkte Suche nach Alternativen

# MP3-Codierung

**Abbildung 4.14:** Prinzipschaltbild eines MP3-Encoders (vereinfacht nach: (Watkinson 2004))

# Alternativen zu MP3

- AAC (Advanced Audio Coding)
  - Weiterentwicklung von MP3, Standard der MPEG-Group
  - populär durch iTunes-Store und iPod (Apple), aber auch Unterstützung durch Geräte von Nokia, Samsung, Sony-Ericsson, Nintendo
  - höhere Kompressionsrate als MP3
  - Patentsituation: anders als bei MP3 keine Lizenzzahlungen für Inhalte, aber für die Implementierung von AAC
- Windows Media Audio (WMA)
  - proprietärer Audio-Codec von Microsoft, Teil der Windows Media-Plattform
- (Ogg) Vorbis
  - freier Codec ohne Patente und Lizenzzahlungen, u.a. Nutzung durch Wikipedia
- (Ogg) Opus
  - neuer, universell einsetzbarer, freier und offener Internet-Standard seit 09/2012
  - anders als MP3, AAC und Vorbis auch für Echtzeitanwendungen wie Telefonie geeignet, da niedrige **Latenz** (Verzögerungszeit)
  - anderen aktuellen Formaten bei fast allen Bitraten qualitativ überlegen

# Opus

Wieso wird MP3 nicht zum telefonieren benutzt?:
MP3 arbeitet erst, wenn schon ein längeres Audio Signal vorhanden ist
-> 1/5 der Sekunde wird verschluckt (Latenz); MP3 nicht für Echtzeit geeignet

Bilder: opus-codec.org

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - **Digitale Bilddaten**
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Lichtspektrum und Elektromagnetisches Spektrum

- Wellenlänge und Frequenz, Beispiel:
  - grünes Licht mit Wellenlänge $550\ nm$

Lichtgeschw.

Frequenz

- $$f = \frac{c}{\lambda} = \frac{3 \cdot 10^8 \frac{m}{s}}{550 \cdot 10^{-9} m} = 5,77 \cdot 10^{14} \frac{1}{s} = 5,45 \cdot 10^{14} Hz$$

Wellenlänge in m umgrechnet

# Farbwarnehmung: Stäbchen und Zapfen

- Antwortspektren von Stäbchen (S) und Zapfen (R, G, B):

Abbildung 3.3

# Farben als „optische Täuschung“

- Farbeindruck ist identisch, wenn die Erregung der drei Zapfentypen übereinstimmt
- Sehr unterschiedliche Spektren können zu einem identischen Farbeindruck führen

**Abbildung 3.4:** Zwei Signale (a) und (b) führen zu gleicher Farbwahrnehmung.

# Additive und subtraktive Farbmischung

RGB-Farben (z.B. Monitor, Beamer)

CMY-Farben (z.B. Farbdrucker)

# RGB-Farbwürfel

**Abbildung 3.5**

# Weitere Farbmodelle: HSL und HSV

# Digitalisierung und Codierung von Bildern

- Digitale Bilder setzen sich aus **Pixeln** (Bildpunkten) zusammen, die in einem Raster angeordnet sind.
- Pro Bildpunkt: Helligkeits- und Farbinformationen
- Zahl der Bit pro Pixel wird **Farbtiefe** genannt
  - 2 Farben (1 Bit): Schwarz/Weiß-Bild
  - 16 Farben (4 Bit)
  - 256 Graustufen (8 Bit)
  - 256 Farben (8 Bit), z.B. bei GIF (*Graphics Interface Format*), als Farbpalette zu jedem Bild
  - 16,7 Millionen Farben (24 Bit) – „True Color“, z.B. im JPEG und PNG-Format (Portable Network Graphics)
  - auch 16 und 32 Bit je Farbkanal („High Dynamic Range“)
- Zusätzlich ggf. sog. **Alphakanal** (Transparenz)
  - in GIF nur eine „Transparenzfarbe“ (keine „Halbtransparenz“ möglich)
  - in PNG viele unterschiedliche Grade an Transparenz möglich

# Alphakanal

- Farbige Pixel mit nach unten hin zunehmender Transparenz über grau/weißem Schachbrettmuster:

# GIF-Format mit 256 verschiedenen Farben

Original (True Color JPEG)

GIF-Format

Deutliche Artefakte: GIF für Fotos ungeeignet

# Bildkompression

- **GIF** (1987 vom US-Online-Dienst CompuServe eingeführt): verlustfrei, aber geringe Farbtiefe (8 Bit), verwendet den LZW-Algorithmus (Lempel-Ziv-Welch) Farbpalette von 256 Farben frei wählbar
- starke Kompression auch bei großer Farbtiefe i.d.R. nur mit verlustbehafteten Verfahren möglich
- Beispiel: **JPEG**-Verfahren
  - Standard existiert seit 1988, festgelegt durch die „Joint Photographic Expert Group“
  - Verfahren beinhaltet vier Schritte:
    1. **Chroma-Subsampling** Trennung Farbkanäle (Reduktion um Faktor 2) von Helligkeit Kanal
    2. **Umcodierung** in den **Frequenzraum**
    3. **Quantisierung** Runden von hohen Frequenzen stärker als niedrigere
    4. **Kompression**

# Artefakte bei starker JPEG-Kompression

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - **Vektorquantisierung**
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Vektorquantisierung: Bsp. Farbpalette

Punkte die weit weg von Prototypvektor sind, sind nicht ideal

Foto: Dcoetzee, CC-by-sa

# Vektorquantisierung

- Transformation von kontinuierlichen Eingabevektoren auf ein endliches **Klassenalphabet** (Menge von Kodebuchklassen)
- Jedem Vektor $\mathbf{c}$ wird der Index $k_t$ seiner Kodebuchklasse (oder Quantisiererzelle) zugeordnet
- Kodebücher können **unüberwacht** aus einer Stichprobe gelernt werden
- Verfahren hierfür z.B. k-Means-Algorithmus

Zufällige Wahl d. Clusterzentren
Zuordnung der Punkte zum Clusterzentrum
Arithmet. Mittel errechnen aus Zentrum & Vektoren im Cluster
Zuordnung d. Punkte zu neuen Clusterzentrum -> Mittel rechnen

-> Solange Wiederholen bis Änderungen gering oder (z.B nach 20 Durchläufen)

# k-Means-Algorithmus

Ablauf:

1. Initialisierung: (zufällige) Auswahl von $k$ Clusterzentren,
   z.B. durch zufällige Auswahl von $k$ Eingabevektoren als Clusterzentren

2. Zuordnung: Jedem Eingabevektor wird dem ihm am nächsten liegenden Clusterzentrum zugeordnet
   (Abstandsmaß: z.B. Euklidischer Abstand)

3. Neuberechnung: Es werden für jeden Cluster die Clusterzentren durch Schwerpunktbildung neu berechnet

4. Wiederholung:
   Falls sich nun die Zuordnung der Objekte ändert, weiter mit Schritt 2, sonst Abbruch

# k-Means- bzw. LBG-Algorithmus

- Ergebnis ist stark von der Initialisierung abhängig (Henne-Ei-Problem)
- Findet lokales Optimum, nicht notwendigerweise das globale
- Möglicher Ausweg:
  - Mehrere Durchläufe mit unterschiedlichen Initialisierungen
  - Entscheidung für diejenige Lösung, bei der ein Gütekriterium den günstigsten Wert aufweist (z.B. die Summe der mittleren quadratischen Abstände der Eingabevektoren zu ihren Klassenzentren) abnimmt
- Gezielte Berechnung geeigneter Startwerte in der Praxis kaum besser als rein zufällige Initialisierung

# Vektorquantisierung: Anwendungen

Quantisierung = Rundung
Vektorquantisierung = Vektor auf einen Gitter legen

- verlustbehaftete Datenkompression
  - Grundidee: Übertrage zunächst das Codebuch, dann nur noch die Indizes der einzelnen Vektoren
  - Audio- und Video-Kompression (z.B. Ogg Vorbis, H.264/MPEG-4)
  - Bestimmung von Farbpaletten (*Color Quantization*, z.B. für die Codierung von Bildern im GIF-Format)
- zahlreiche Anwendungen in der Mustererkennung
  - Suche nach einheitliche Regionen in Bildern (z.B. im Hinblick auf Farbe oder Textur)
  - Bestimmung von Laut-Unterklassen für die Spracherkennung
  - k-Means-Algorithmus ist Vorstufe des EM-Algorithmus, des wichtigsten Lernverfahrens in der statistischen Mustererkennung (folgt in Kapitel 4)

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - **Schwellwertoperationen & Histogramme**
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Punkt-Operationen

- Die Werte von einzelnen Pixeln werden verändert, ohne dabei die Nachbarpixel zu betrachten
- Zu den Punktoperationen zählen
  - Schwellwertoperationen, z.B. zur Transformation eines Grauwertbildes in ein Schwarzweißbild (**Binärisierung**)

$$f'(x,y) = \begin{cases} 1, & \text{falls } f(x,y) > w \\ 0, & sonst \end{cases}$$

  - Histogramm-basierte Modifikationen von Farbe oder Grauwerten, z.B. zur Verbesserung des **Kontrasts**
    - Ein **Histogramm** liefert zu jedem Grauwert (bzw. zu jeder Quantisierungsstufe der einzelnen Farbkanäle) dessen relative Häufigkeit

# Grauwerthistogramm - Bsp. Pupillenerkennung

Quelle: Daniil Moerman, „Implementierung eines echtzeitfähigen Feature-Extraction-Algorithmus aus Livebilddaten auf der ARM-i.MX6-Architektur für ein Medizingerät“, Bachelorarbeit, TH Nürnberg, 2013

# Binärisierung - Bsp. Pupillenerkennung

**Original**

**Spot Regions**

**Pupil Regions**

**Background Regions**

Quelle: Daniil Moerman, „Implementierung eines echtzeitfähigen Feature-Extraction-Algorithmus aus Livebilddaten auf der ARM-i.MX6-Architektur für ein Medizingerät“, Bachelorarbeit, TH Nürnberg, 2013

# Farbhistogramme

# Histogramm und Kontrast

**Abbildung 3.12:** Grauwertbilder mit Histogrammen. Linkes Bild: Alle Grauwerte kommen vor. Mittleres Bild: Nur Grauwerte zwischen 50 und 150 kommen vor. Rechtes Bild: Spreizung der Grauwerte auf den ganzen Bereich, wobei das Ausgangsbild das mittlere Bild ist.

Lineare Spreizung des Histogramms (Schritt vom mittleren Bild zum rechten Bild):

$$f'(x,y) = \frac{f(x,y) - h_{\min}}{h_{\max} - h_{\min}} w_{\max}$$

# Histogrammlinearisierung (*histogram equalization*)

maximaler Kontrast -> jeder Grauwert gleich oft

Foto: Phillip Capper, modifiziert durch „Konstable“, CC-by-sa

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - **Lineare Filter**
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Filter und lineare Filter

- Filter berücksichtigen - anders als Punkt-Operationen - auch die Werte von Bildpunkten in der Umgebung des betrachteten Bildpunktes, um für einen Pixel im Originalbild einen neuen Wert zu berechnen.
- Beispiel: Mittelwertfilter (einfacher Weichzeichner)

$$f'(x,y) = \frac{1}{9} \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} f(i,j) \qquad (1)$$

- $$f'(x,y) = \underbrace{\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}}_{\text{Filterkern}} * \overbrace{\begin{pmatrix} f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ f(x-1,y) & \underbrace{f(x,y)}_{\text{Mittelpunkt des Filters}} & f(x+1,y) \\ f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \end{pmatrix}}^{\text{alle Werte aufsummiert}} \qquad (2)$$
- Faltungsoperator ´ * ´ : paarweise Multiplikation der Wertepaare, die an derselben Position stehen, und Aufsummieren der Ergebnisse
- Filter, die wie hier durch eine Faltungsoperation darstellbar sind, bezeichnet man als **lineare Filter**

# Filter und lineare Filter

- Filter berücksichtigen - anders als Punkt-Operationen - auch die Werte von Bildpunkten in der Umgebung des betrachteten Bildpunktes, um für einen Pixel im Originalbild einen neuen Wert zu berechnen.
- Beispiel: Mittelwertfilter (einfacher Weichzeichner)

$$f'(x,y) = \frac{1}{9} \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} f \qquad (1)$$

Diese Matrix bezeichnet man als **Filterkern.**

- $$f'(x,y) = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ f(x-1,y) & f(x,y) & f(x+1,y) \\ f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \end{pmatrix} \qquad (2)$$
- Faltungsoperator $*'$ : paarweise Multiplikation der Wertepaare, die an derselben Position stehen, und Aufsummieren der Ergebnisse
- Filter, die wie hier durch eine Faltungsoperation darstellbar sind, bezeichnet man als **lineare Filter.**

# Gauß-Filter (Gaußscher Weichzeichner)

3x3 Filterkern:

$$\frac{1}{8}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

5x5 Filterkern:

$$\frac{1}{273}\begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}$$

# Gaußscher Weichzeichner

# Filterkerne weiterer Filtertypen

Laplace-Filter zur Kantenhervorhebung (2 Varianten):

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \qquad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Boost-Filter (Scharfzeichner):

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

# Laplace-Filter

# Kantenhervorhebung mit dem Sobel-Operator

- Sei das Originalbild die Matrix $\mathbf{A}$.
- Die gefalteten Resultate $\mathbf{G}_x$ und $\mathbf{G}_y$ berechnen sich mittels der Sobel-Operatoren $\mathbf{S}_x$ und $\mathbf{S}_y$ folgendermaßen:

vertikale Kanten -> $$\mathbf{G}_x = \mathbf{S}_x * \mathbf{A} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * \mathbf{A}$$

horizontale Kanten -> $$\mathbf{G}_y = \mathbf{S}_y * \mathbf{A} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \mathbf{A}$$

Bsp:

(Randwerte wenn A weiter gehen würde)

- Richtungsunabhängige Information:

Gesamtstärke der Kante -> $$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

aus der Matrix

$|\vec{G}_x| = G_x$

$|\vec{G}_y| = G_y$

$|\vec{G}| = G$

-> Richtungsberechnung d. Steigung: für diesen Pixel

Vektoren $\vec{G}_x, \vec{G}_y, \vec{G}$

$\tan\alpha = \frac{G_y}{G_x}$ -> $\alpha = \arctan \frac{G_y}{G_x} = \arctan \frac{28}{14} = \arctan 2 \approx 63°$

# Sobel-Operator, Beispiel

# Filter

in der Bildverarbeitung
z.B. Laplace-, Sobel-Filter
(Kantenhervorhebung)

in der Bildverarbeitung
z.B. Gauß-Filter
(Weichzeichner)

hohe Frequenzen beim Bild
-> scharfe Kanten, feine Strukturen

tiefe Frequenzen beim Bild
-> homogene Flächen, Farben/Grauwerte von Flächen

# Filterung im Frequenzraum

- Rechenaufwand für Faltungsfilter steigt überproportional bei Vergrößerung der Filtermaske
- Alternative: Überführung des Bildes in den Frequenzraum (2D-Fourier-transformation) -> Spektrum ausrechnen & manipulieren
- Manipulation des Bildes im Frequenzraum
- Anschließend Rücktransformation in den Ortsraum (Bildraum)
- Beispiel rechts:
  - Bild ist überlagert durch diagonales Störsignal
  - Transformation in den Frequenzraum (oben rechts)
  - Störfrequenzen werden im Frequenzraum gelöscht (schwarze Streifen im Bild unten links)
  - Rücktransformation in den Ortsraum (unten rechts)

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - **Nichtlineare Operationen**
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Rangordnungsfilter

- Sortiere die Pixel einer definierten Umgebung (z.B. 3x3, 11x11) aufsteigend nach ihrem Grauwert
- ersetze den Grauwert des aktuellen Pixels (in der Mitte der Umgebung) durch den Wert an einer definierten Position in der Liste der Grauwerte:
  - **Minimumfilter**: 1. Position der Liste (dunkle Strukturen werden größer)
  - **Medianfilter**: mittlere Position der Liste Rauschen verschwindet, Ausreißer verschwinden
  - **Maximumfilter**: letzte Position der Liste (helle Strukturen werden größer)
- Besonders häufig wird der Medianfilter eingesetzt:
  - kann einzelne fehlerhafte Pixel in Flächen vollständig korrigieren
  - keine Kantenglättung wie bei Weichzeichner (Gauß- oder Mittelwert-Filter)
  - eignet sich besonders zur Korrektur von vereinzelten, starken „Ausreißern“

# 3x3-Medianfilter

# Morphologische Operationen

Wie zählt man automatisch die Bakterien auf dem Bild?

# Morphologische Operationen

Wie zählt man automatisch die Bakterien auf dem Bild?

Operation: Erosion angewendet auf des Vordergrundes (grüne Farbe)

Leichter geht es auf dem erodierten Bild

# Erosion

sobald der Mittelpunkt des Strukturelement
auf Vordergrund trifft wird ein Vordergrundpunkt
gezeichnet in der Mitte des Strukturelements
--> Ränder werden nicht gezeichnet (abfräßen)

abfräßen des Randes

# Dilatation (engl. *Dilation*)

sobald der Mittelpunkt des Strukturelement
auf Vordergrund trifft wird das komplette Strukturelement
mit Vordergrundfarbe gezeichnet
--> Ränder werden ausgedehnt

ausdehnen der Ränder

# Opening: Erst Erosion, dann Dilatation

# Closing: Erst Dilatation, dann Erosion

# Morphologische Operationen im Überblick

FIGURE 25-10
Morphological operations. Four basic morphological operations are used in the processing of binary images: *erosion*, *dilation*, *opening*, and *closing*. Figure (a) shows an example binary image. Figures (b) to (e) show the result of applying these operations to the image in (a).

Erosion -> Dilatation

Dilatation -> Erosion

*The Scientist and Engineer's Guide to Digital Signal Processing*, Steven W. Smith

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - **Normierungsmaßnahmen**
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Normierungsmaßnahmen

- Muster mit gleicher Bedeutung sollen einer Klasse zugeordnet werden
- Viele Unterschiede können ohne Einfluss auf die Bedeutung sein, z.B.
  - Größe von Buchstaben
  - Lautstärke von Wörtern
  - Drehung und Lage eines Werkstücks
- „Die **Normierung** von Mustern soll den Wertebereich von Parametern, die für die Klassifikation irrelevant sind, reduzieren, um bei gegebenem Aufwand für die Klassifikation eine geringere Fehlerwahrscheinlichkeit zu erreichen.“ (Niemann 1983)

# Normierungsmaßnahmen bei der Handschrifterkennung

Gefahr beim Normieren -> Verlust von Information durch weglassen von Eigenschaften

Normierung der Schreibrichtung

Aufrichtung von schräggestellten Zeichen

Bild links: Niemann 2003

# Lagenormierung zur Auswertung eines Formulars

Id. no. ADA052502360 Pg. 01
14 / 03 / 2001

Application for the Post of Scientist/Engineers

(Please fill the form with black pen after reading Instruction-cum-Info sheet (IIS))

Post-Field Code BBBAFDD

1. Name MANISH KUMAR

2. Date of Birth (dd/mm/yyyy) 24 / 05 / 1974

3. Gender (MALE/FEMALE) MALE

4. Category (Refer to Part B1(b) of IIS) GEN

5. Communciation address

C 36 CHURCH STREET
MARKET K R
City/Dist BANGALORE
State KARNATAKA
Pincode 560001

Telephone No. (STD & Local No) 080 - 2269416

E-Mail address (if any) manish@yahoo.com

6. Educational Qualification Details (Refer to Part B1 (c), (d), (e), 2, 3 & 4 of IIS)

6.1 Graduation level (BE/BTECH/AMIE/AMAS/MSC/MCA/BSCE)

6.11 Course Name BTECH

6.12 Branch MECH

6.13 Year of passing 1999

6.14 University/Institute BANGALORE UNIV

6.15 Class/Div (FIRST/SECOND/THIRD/PASS/UNAWARDED) FIRST

6.16 Duration of the course : 08 semesters years

6.17 Semester/Yearwise % of marks (as the case may be)

6.171 First 67.4 6.172 Second 74.3 6.173 Third 69.2 6.174 Fourth 72.7 6.175 Fifth 68.5

6.176 Sixth 70.5 6.177 Seventh 72.5 6.178 Eighth 65.3 6.179 Ninth 6.180 Tenth

6.181 Aggregate % marks of all semesters/years as the case may be (Average of 6.171 to 6.180)

In Figures 73.3

In words SEVEN THREE. THREE

Id. no. ADA052502360 Pg. 01
14 / 03 / 2001

Application for the Post of Scientist/Engineers

(Please fill the form with black pen after reading Instruction-cum-Info sheet (IIS))

Post-Field Code BBBAFDD

1. Name MANISH KUMAR

2. Date of Birth (dd/mm/yyyy) 24 / 05 / 1974

3. Gender (MALE/FEMALE) MALE

4. Category (Refer to Part B1(b) of IIS) GEN

5. Communciation address

C 36 CHURCH STREET
MARKET K R
City/Dist BANGALORE
State KARNATAKA
Pincode 560001

Telephone No. (STD & Local No) 080 - 2269416

E-Mail address (if any) manish@yahoo.com

6. Educational Qualification Details (Refer to Part B1 (c), (d), (e), 2, 3 & 4 of IIS)

6.1 Graduation level (BE/BTECH/AMIE/AMAS/MSC/MCA/BSCE)

6.11 Course Name BTECH

6.12 Branch MECH

6.13 Year of passing 1999

6.14 University/Institute BANGALORE UNIV

6.15 Class/Div (FIRST/SECOND/THIRD/PASS/UNAWARDED) FIRST

6.16 Duration of the course : 08 semesters years

6.17 Semester/Yearwise % of marks (as the case may be)

6.171 First 67.4 6.172 Second 74.3 6.173 Third 69.2 6.174 Fourth 72.7 6.175 Fifth 68.5

6.176 Sixth 70.5 6.177 Seventh 72.5 6.178 Eighth 65.3 6.179 Ninth 6.180 Tenth

6.181 Aggregate % marks of all semesters/years as the case may be (Average of 6.171 to 6.180)

In Figures 73.3

In words SEVEN THREE. THREE

D. Deodhare, R. Amit, 2005

# Normierungsmaßnahmen bei der Gesichtserkennung (1)

Un-normalised faces



Mean and variance normalisation

Histogram equalisation

Egdiness filtering

Bild: ACG Group, AIT

# Normierungsmaßnahmen bei der Gesichtserkennung (2)

Alignment pipeline. (a) The detected face, with 6 initial fiducial points. (b) The induced 2D-aligned crop. (c) 67 fiducial points on the 2D-aligned crop with their corresponding Delaunay triangulation, we added triangles on the contour to avoid discontinuities. (d) The reference 3D shape transformed to the 2D-aligned crop image-plane. (e) Triangle visibility w.r.t. to the fitted 3D-2D camera; black triangles are less visible. (f) The 67 fiducial points induced by the 3D model that are using to direct the piece-wise affine warpping. (g) The final frontalized crop. (h) A new view generated by the 3D model (not used in this paper).

Quelle: Yaniv Taigman et. al., „DeepFace: Closing the Gap to Human-Level Performance in Face Verification“, CVPR 2014

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Merkmale: Beispiel 1

- Merkmale zur Bestimmung der Vogelart: Größe, Farbe, Schnabelform, Schnabellänge, Stimme, Beinlänge, Lebensraum usw.
- Ornithologe kann anhand solcher Merkmale die Vogelart mit einer gewissen Zuverlässigkeit bestimmen (klassifizieren), ohne den Vogel selbst gesehen zu haben.
- Schlechte Merkmale für Bestimmung der Vogelart: z.B. Anzahl Beine, Federn (j/n)
- Je weniger Merkmale, desto geringer die Zuverlässigkeit
- Ein einzelnes Merkmal reicht selten aus

Merkmals-Vektor erstellt und Klassifikator übergeben. Klassifikator filtert Klasse heraus

# Merkmale: Beispiel 2

Bild: Eidenberger, iX 02/2009

# Klassifikation (Lineare Trennebene)

Bild: Eidenberger, iX 02/2009

# Merkmale: Beispiel 3 (Iris-Datensatz)

$\Omega_1$ = Iris setosa

$\Omega_2$ = Iris versicolor

$\Omega_3$ = Iris virginica

| Sepal length | Sepal width | Petal length | Petal width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.4 | 0.4 | setosa |
| 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 6.6 | 2.9 | 4.6 | 1.3 | versicolor |
| 5.2 | 2.7 | 3.9 | 1.4 | versicolor |
| 6.0 | 2.2 | 4.0 | 1.0 | versicolor |
| 6.4 | 2.7 | 5.3 | 1.9 | virginica |
| 6.1 | 2.6 | 5.6 | 1.4 | virginica |
| 5.9 | 3.0 | 5.1 | 1.8 | virginica |

Fotos: Wikipedia

# Merkmale: Beispiel 3 (Iris-Datensatz)

Bild: Wikipedia

# Aufbau eines Klassifikationssystems

# Definitionen (I)

Die **Umwelt** ist die Gesamtheit der mit physikalischen Geräten messbaren Größen. Sie wird repräsentiert durch die Menge

$$U = \{ {}^{\rho}b(x) | \rho = 1,2,\ldots \}$$

von Funktionen ${}^{\rho}b(x)$

bestimmter Ausschnitt

Ein **Problemkreis** wird mit Ω bezeichnet und enthält nur Objekte oder Funktionen, die zu einer strikt begrenzten Anwendung oder einem Ausschnitt der Umwelt gehören. Er ist definiert durch die Menge

$$\Omega = \{ {}^{\rho}f(x) \mid \rho = 1,2,\ldots \} \subset U$$

von Funktionen ${}^{\rho}f(x)$ und ist eine Teilmenge der Umwelt U.

# Definitionen (II)

Die Elemente aus der Menge Ω, dem Problemkreis, heißen **Muster**. Daher ist ein Muster eine Funktion

$$f(x) = \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \ldots \\ f_m(x_1, x_2, \ldots, x_n) \end{pmatrix}$$

# Definitionen (III)

**Mustererkennung** beschäftigt sich mit

- der *Klassifikation einfacher Muster* sowie
- der *Analyse und dem Verstehen komplexer Muster*.

Bei der **Klassifikation einfacher Muster** wird jedes Muster als ein Ganzes betrachtet und unabhängig von anderen Mustern einer Klasse $\Omega_\kappa$ aus $k$ möglichen Klassen $\Omega_\lambda$, $\lambda = 1, \ldots, k$ zugeordnet. Es ist möglich, ein Muster zurückzuweisen, das heißt einer $(k + 1)$ten Klasse $\Omega_0$, der Rückweisungsklasse, zuzuordnen.

# Klassen (I)

Partition (Aufteilung)

$\Omega_\kappa$ erhält man aus einer Partition der Menge $\Omega$ in $k$ (oder $k+1$) Teilmengen $\Omega_\kappa$, $\kappa = 1,\ldots,k$ oder $\kappa = 0,1,\ldots,k$. Es wird gefordert, dass

$$\Omega_\kappa \neq \emptyset,\ \kappa = 1,\ldots,k,$$

darf nicht die leere Menge sein -> für jede Klasse gibt es Muster die auf diese Klasse abgebildet werden können

$$\Omega_\kappa \cap \Omega_\lambda = \emptyset,\ \kappa \neq \lambda,$$

Klassengebiete (Menge der Muster in einer Klasse) überschneiden sich nicht -> eindeutige Zuordnung

$$\bigcup_{\kappa=1}^{k} \Omega_\kappa = \Omega \ \text{ oder } \ \bigcup_{\kappa=0}^{k} \Omega_\kappa = \Omega$$

Jedes Muster was auftreten kann im Problemkreis kann auf einer Klasse abgebildet werden

**Anmerkungen**

- Klassen sind nach obiger Definition disjunkt
- nicht-disjunkte Klassen können für manche Anwendungen sinnvoll sein
- obige Definition gibt keine Kriterien, um Klassen zu finden

# Klassen (II)

- Muster einer Klasse sollten
  - untereinander **ähnlich** sein,
  - **verschieden** von Mustern **anderer** Klassen sein.
- Wenn ein Muster nicht zuverlässig klassifizierbar ist, wird es
  - entweder zurückgewiesen, d.h. nach $\Omega_0$ klassifiziert,
  - oder es werden mehrere Alternativen ausgegeben, d.h. nach $\{\Omega_{\kappa,1}, \Omega_{\kappa,2}, \ldots, \Omega_{\kappa,n\kappa}\}$ klassifiziert. hier die ersten 5 Treffer ausgegeben
- Die Auswahl von Klassen ist häufig durch die Anwendung vorgegeben. Beispiele:
  - „ja“ vs. „nein“ (IVR-System)
  - defekte vs. intakte Werkstücke (Qualitätskontrolle)
  - die zehn Ziffern (Handschrifterkennung bei Postleitzahlen)

# Postulate (I)

Ein Postulat ist ein grundlegendes Annahme- oder Behauptungssatz in der Mathematik oder in anderen Wissenschaftsbereichen

**Stichprobe**:

Es steht eine *repräsentative Stichprobe* $\omega \subset \Omega$ von Mustern $f(x) \in \Omega$ zur Verfügung.

*„There‘s is no data like more data“*

Bob Mercer, 1985

*„It never pays to think until you’ve run out of data“*

Eric Brill, 2001

*„Fire everybody and spend the money on data.“*

Unbekannt

# Postulate (II)

**Merkmale**:

Ein (einfaches) Muster hat Merkmale $c_v$ , $v = 1, \ldots, n$, die charakteristisch für seine Klassenzugehörigkeit sind.

es muss was messbares geben für die Mustererkennung

**Kompaktheit**:

Merkmale von Mustern einer Klasse $\Omega_K$ nehmen einen kompakten Bereich im Merkmalsraum ein.

Cluster die zu einer Klasse gehören -> Beispiel Koala / Panda (Häufung mehrerer Bereiche im Merkmalsraum für die Mustererkennung)

**Ähnlichkeit**:

Zwei Repräsentationen (von Mustern) sind ähnlich, wenn ein geeignet definiertes Abstandsmaß klein ist.

niedriges Wert für Elemente gleicher Klasse

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - **Orthogonale Reihenentwicklung** DCT/DFT
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Orthogonale Reihenentwicklung

Die **diskrete Fouriertransformation (DFT)** und die **diskrete Kosinustransformation (DCT)** sind von fundamentaler Bedeutung für die Medienverarbeitung. Beispiele:

MFCC = Merkmalerkennung/Vektoren in der Spracherkennung

- Zur Berechnung der MFCCs kommt sowohl die DFT als auch die DCT zum Einsatz (siehe Abschnitt „Merkmale für die Spracherkennung“).
- Die (zweidimensionale) DFT erlaubt die effiziente Implementierung von linearen Filtern, z.B. zur Bildvorverarbeitung (siehe Kapitel 2).
- Die DCT ist wesentlicher Bestandteil sowohl der JPEG-Bildkompression als auch der MP3-Audiokompression (hier als modifizierte DCT, MDCT).
- Die DFT dient zur digitalen Berechnung von Spektren und Spektrogrammen (siehe vorangehender Abschnitt).

DFT und DCT (wie auch ihre kontinuierlichen Entsprechungen) sind Spezialfälle einer **orthogonalen Reihenentwicklung bzw. Transformation.**

# Orthogonale Transformation, Beispiel 1

- Ein eindimensionale Signal z.B. der Länge 256 lässt sich als ein Vektor im 256-dimensionalen Raum auffassen. Man betrachtet dazu den 1. Abtastwert als die 1. Koordinate, den 2. Abtastwert als die 2. Koordinate usw.
- Wie betrachten zunächst ein Audiosignal mit nur 2 Abtastwerten:

3, -5

- Dieses Signal entspricht dem Vektor

$$\begin{pmatrix} 3 \\ -5 \end{pmatrix}$$

- Dieser Vektor lässt sich als Linearkombination der Einheitsvektoren darstellen:

$$\begin{pmatrix} 3 \\ -5 \end{pmatrix} = 3 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + (-5) \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Die Einheitsvektoren stellen jedoch nur eine mögliche Kombination von Basisvektoren dar, durch deren Linearkombinationen sich beliebige Vektoren erzeugen lassen.

# Orthogonale Transformation, Beispiel 1

- Alternative Basisvektoren für den zweidimensionalen Raum wären z.B.

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

- Unser Beispielsignal lässt sich so folgendermaßen darstellen:

$$\begin{pmatrix} 3 \\ -5 \end{pmatrix} = (-1) \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 4 \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

- Nach einer solchen **Transformation** der Basis wird das Beispielsignal also dargestellt durch die beiden **Koeffizienten**: (-1), 4 Sinn: neue Koeffizienten haben andere Eigenschaften als vorherige
- Die hier verwendete Basis ist **orthogonal** (rechtwinklig), weil das Skalarprodukt verschiedener Basisvektoren jeweils 0 ergibt.
- Sie ist nicht **orthonormal**, weil hierzu zusätzlich noch die Länge der Basisvektoren 1 sein müsste. Durch geeignete Normierung der Basisvektoren lässt sich aber leicht Orthonormalität herstellen:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{und} \quad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

# Orthogonale Transformation, Beispiel 2

ein 2x2-Bild

| | |
|---|---|
| 30 | 90 |
| 150 | 210 |

zugehörige Grauwerte

$$\begin{pmatrix} 30 & 90 \\ 150 & 210 \end{pmatrix}$$

Matrixschreibweise

Konstruktion des 2x2-Bildes aus den 4 „Einheits-Basisbildern“:

$$\begin{pmatrix} 30 & 90 \\ 150 & 210 \end{pmatrix} = 30 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + 90 \cdot \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + 150 \cdot \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + 210 \cdot \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

(weiß:1, grau: 0, schwarz: -1)

# Orthogonale Transformation, Beispiel 2

ein 2x2-Bild

| | |
|---|---|
| 30 | 90 |
| 150 | 210 |

zugehörige Grauwerte

$$\begin{pmatrix} 30 & 90 \\ 150 & 210 \end{pmatrix}$$

Matrixschreibweise

Konstruktion des Bildes aus den 4 Basisbildern der diskreten Kosinustransformation (DCT):

alle 4 Matrizen/Vektoren sind orthogonal und normal

$$\begin{pmatrix} 30 & 90 \\ 150 & 210 \end{pmatrix} = 240 \cdot \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} - 120 \cdot \begin{pmatrix} 0.5 & 0.5 \\ -0.5 & -0.5 \end{pmatrix} - 60 \cdot \begin{pmatrix} 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix} + 0 \cdot \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$$

(weiß:0.5, grau: 0, schwarz: -0.5)

# Orthogonale Transformation, Beispiel 2

ein 2x2-Bild

| | |
|---|---|
| 30 | 90 |
| 150 | 210 |

zugehörige Grauwerte

$$\begin{pmatrix} 30 & 90 \\ 150 & 210 \end{pmatrix}$$

Matrixschreibweise

**DCT-Koeffizienten**

Konstruktion des Bildes aus den 4 Basisbildern der diskreten Kosinustransformation (DCT):

$$\begin{pmatrix} 30 & 90 \\ 150 & 210 \end{pmatrix} = 240 \cdot \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} - 120 \cdot \begin{pmatrix} 0.5 & 0.5 \\ -0.5 & -0.5 \end{pmatrix} - 60 \cdot \begin{pmatrix} 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix} + 0 \cdot \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$$

DCT -> Bild falten mit den Basisbildern und Ergebnis werden als Koeffizienten verwendet

(weiß:0.5, grau: 0, schwarz: -0.5)

# DCT von 8x8-Blöcken in JPEG

hier 64 Basisbilder

Statt aus 64 Einzelpunkten wird jeder $8 \times 8$-Block als Linearkombination dieser 64 Blöcke dargestellt.

# Basiskosinusfunktionen der DCT für N=8,

- Die 8 Basiskosinusfunktionen für *N*=8, abgetastet an 8 Stellen:
  - u=0: „DC-Koeffizient“ („Gleichstromanteil“)
  - u=1: ½ Periode
  - u=2: 1 Periode
  - u=3: 1½ Perioden
  - u=4: 2 Perioden
  - u=5: 2½ Perioden
  - u=6: 3 Perioden
  - u=7: 3 ½ Perioden

- Die 8 Basiskosinusfunktionen sind orthogonal, bilden daher eine Orthogonalbasis für den 8-dimensionalen Vektorraum.

Bild: H. Wons, Universität Mannheim, 2000

# DCT von 8x8-Blöcken in JPEG

- Die 64 Koeffizienten $F_{xy}$ geben die Gewichtung der einzelnen Basis-Bilder an.
- Sie lassen sich aus einem gegebenen 8x8-Bild mit den Pixelwerten $f_{mn}$ nach folgender Formel bestimmen:

$$F_{xy} = \frac{1}{4} C_x C_y \sum_{m=0}^{7} \sum_{n=0}^{7} f_{mn} \cos\left[\frac{(2m+1)x\pi}{16}\right] \cos\left[\frac{(2n+1)y\pi}{16}\right]$$

$$C_x, C_y = \begin{cases} \frac{1}{\sqrt{2}} & \text{wenn } x, y = 0 \\ 1 & \text{sonst} \end{cases}$$

# Diskrete Fourier-Transformation (DFT)

- Die DCT kann als ein Spezialfall der DFT angesehen werden
- Unterschiede:
  - Die Koeffizienten der DFT sind komplexe Zahlen, d.h. es findet eine Transformation vom reellen in den komplexen Zahlenraum statt. Die Koeffizienten der DCT sind dagegen immer reelle Zahlen.
  - Die Berechnung des Realteils der Koeffizienten erfolgt bei der DFT unter Verwendung von Basis-Kosinusfunktionen (ähnlich wie bei der DCT).
  - Die Berechnung des Imaginärteils der Koeffizienten erfolgt ähnlich wie bei der DCT, jedoch unter der Verwendung von Basis-Sinusfunktionen.

# Die Basisfunktionen der DFT für N=32

- 8 der 34 Basisfunktionen der DFT für *N*=32, abgetastet an 32 Stellen, sind im Bild rechts dargestellt.
- Realteil (links): $c_0, \ldots, c_{16}$
- Imaginärteil (rechts): $s_0, \ldots, s_{16}$
- 2 der 34 Funktionen tragen keine Information bei: $s_0$ und $s_{16}$
- Die übrigen 32 Funktionen bilden eine Orthogonalbasis für den 32-dimensionalen Vektorraum.
- Basisfunktionen Realteil:

$$c_u(k) = \cos\left(\frac{2\pi u}{N} \cdot k\right)$$

- Basisfunktionen Imaginärteil:

$$s_u(k) = \sin\left(\frac{2\pi u}{N} \cdot k\right)$$

Bild: S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 1997

# Berechnung der Koeffizienten der DFT

- Basisfunktionen: $c_u(k) = \cos\left(\frac{2\pi u}{N} \cdot k\right) \qquad s_u(k) = \sin\left(\frac{2\pi u}{N} \cdot k\right)$
- Realteil der DFT-Koeffizienten (für *u* = 0, ... , N-1):

$$\mathrm{Re}\big(F(u)\big) = \sum_{k=0}^{N-1} f(k) \cdot c_u(k) = \sum_{k=0}^{N-1} f(k) \cdot \cos\left(\frac{2\pi u}{N} \cdot k\right)$$

- Imaginärteil der DFT Koeffizienten (für *u* = 0, ... , N-1):

$$\mathrm{Im}\big(F(u)\big) = \sum_{k=0}^{N-1} -f(k) \cdot s_u(k) = \sum_{k=0}^{N-1} -f(k) \cdot \sin\left(\frac{2\pi u}{N} \cdot k\right)$$

- Die Rücktransformation aus dem Frequenzbereich in den Zeit-/Ortsbereich erfolgt analog ($F(u)$ und $f(k)$ werden in den Formeln vertauscht).

# Die Basismatrizen der 2D-DFT (8x8)

Realteil

Imaginärteil

# Nichtperiodische Signale

- Sowohl die DFT als auch die DCT setzen implizit voraus, dass das Eingangssignal periodisch ist.
- Verletzung dieser Voraussetzung bei der DCT meist etwas weniger problematisch als bei der DFT, da die DCT anders als die DFT nur gerade Basisfunktionen hat (siehe folgende Folie, DCT-I und DCT-II).
- Daher bei der DCT keine impliziten Sprünge an den Rändern wie bei der DFT (siehe „Leck-Effekt“ im Abschnitt „Merkmale für die Spracherkennung“).
- Bei Einsatz der DFT auf kurzen Abschnitten nichtperiodischer Signale, wie bei der Merkmalberechnung für die Spracherkennung, ist eine geeignete Fensterfunktion erforderlich (z.B. Hamming-Fenster). Man spricht hier auch von einer **STFT** (**S**hort **T**ime **F**ourier **T**ransform, Kurzzeit-Fourier-Transformation).
- Bei der STFT gilt die folgende **Unschärferelation**: Je höher die zeitliche Auflösung, desto geringer die Frequenzauflösung, und umgekehrt.

# Implizite Fortsetzungen der DCT

DCT nimmt an, dass der Signalverlauf gespiegelt ist -> kein Sprung

- Darstellung der impliziten Fortsetzung am Beispiel einer Eingangsdatenfolge mit 11 Werten (in rot) und deren Möglichkeiten zur geraden bzw. ungeraden Fortsetzungen im Rahmen der vier üblichen DCT-Varianten (DCT Typ I bis IV)
- Für die JPG-Kompression wird beispielsweise die DCT-II verwendet.
- Die DCT-III dient zur Rücktransformation der DCT-II in den Ortsbereich.

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung DCT/DFT
   - **Wavelet-Transformation**
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Wavelet-Transformation

- Kurzzeit-Fouriertransformation (STFT) erfordert Kompromiss zwischen Zeit- und Frequenzauflösung.
- Abstufung der Frequenzen bei der STFT über den gesamten Frequenzbereich linear
- **Aber**: Bei hohen Frequenzen wäre eine gute Zeitauflösung wichtiger, da eine vollständige Schwingung hier weniger Zeit beansprucht und sich die Momentanfrequenz daher schneller ändern kann. Und bei z.B. 100 Hz bedeutet eine Abweichung von 10 Hz eine wesentlich größere relative Abweichung als bei 10 kHz.
- Häufig wünschenswert: bessere zeitliche Auflösung für hohe Frequenzen.
- Genau das leistet die **Wavelet-Transformation** (zu engl. *Wavelet* = kleine Welle).
- Das einfachste Wavelet ist das **Haar-Wavelet** (nach dem ungar. Mathematiker Alfréd Haar, veröffentlicht 1909).

# Wavelet-Transformation: Haar-Wavelets

H0 = 8 x 1

H1 = 4 x 1 & 4 x -1

- H0 bis H7 sind die ersten 8 Haar-Wavelets. Sie bilden (wie die Basisfunktionen der DCT oder der DFT) die Orthogonalbasis für einen 8x8-Vektorraum (und können durch Normierung leicht in eine Orthonormalbasis überführt werden).
- H1 zeigt das Haar-Wavelet in seiner Grundform („Mutter-Wavelet“)
- H2 bis H7 entstehen durch **Verschiebung** und **Skalierung (Stauchung)** des „Mutter-Wavelets“.

# 2D-Haar-Wavelet-Transformation (4x4)

Bild: B.Selig, Tilman Walther, FU Berlin

Die 16 Basisfunktionen der 4x4-Haar-Wavelet-Transformation

- jedes 4x4-Bild lässt sich durch Überlagerung (Linearkombination) dieser 16 Basis-Bilder darstellen (analog zur DCT)
- Die Koeffizienten zu einem gegebenen Bild lassen sich effizient bestimmen.

Anwendungen z.B.

- Merkmale für die Objekterkennung
- Kompression von Sensordaten

# Morlet-Wavelets

- Alternativ zum Haar-Wavelet kann z.B. das Morlet-Wavelet verwendet werden
- Es entsteht durch Modulation der Kosinusfunktion mit einer Gauß-Glocke

Bild: C.T. Russell, UCLA

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - **Heuristische Verfahren**
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Heuristische Verfahren

- Eine **Heuristik** bezeichnet in der Informatik eine Vorgehensweise, bei der man versucht, ein Problem zu Lösen mithilfe von
  - Schätzungen
  - „Faustregeln‘
  - intuitiv-intelligentem Raten.

- Optimale Eigenschaften sind dabei nicht garantiert.
- Alle bisher besprochenen Merkmalsberechnungsverfahren (z.B. MFCCs, DFT-Koeffizienten) sind in dem Sinne heuristisch, dass sie zwar eine fundierte mathematische Grundlage besitzen, aber ihre Anwendung als Merkmale nur intuitiv und experimentell begründet ist.
- Es gibt analytische Methoden, um unter bestimmten Voraussetzungen optimale Merkmale für Klassifikationsaufgaben zu berechnen. Diese sind i.d.R. sehr rechenaufwändig und nicht Gegenstand dieser Vorlesung.
- Im Folgenden weitere Beispiele für heuristische Ansätze, die sich in der Praxis bewährt haben.

# Nulldurchgangsrate

Die **Nulldurchgangsrate** berechnet sich als die Anzahl der Vorzeichen-wechsel des Signals in einem Zeitfenster konstanter Länge, das über das Signal geschoben wird.

# Heuristische Merkmale z.B. zur Erkennung von Blättern

Vorverarbeitung:

- Binärisierung
- Ausschneiden der *Region of Interest* (ROI)

Merkmale (Auswahl):

$$Aspektverhältnis = \frac{Bildbreite}{Bildhöhe}$$

$$Projektion_{x_i} = \frac{\sum_{\forall Punkte \in ROI_x} 1}{Bildbreite}$$

Streifen/col zB 3 und Verhältnis schwarzer Pixel zu anderen Streifen

$$Projektion_{y_i} = \frac{\sum_{\forall Punkte \in ROI_y} 1}{Bildhöhe}$$

gleiches Prinzip bloß row

$$Kompaktheit = \frac{\left(\sum_{\forall Randpunkte \in ROI} 1\right)^2}{\sum_{\forall Punkte \in ROI} 1}$$

Kompaktheit = $U^2/A$

zur Kompaktheit:

Quelle: O. Schmidt, M. Freisinger, G.Sovak, IT-Projekt „Leaves“, Fak. Informatik, Ohm-Hochschule Nürnberg, 2011

# Heuristische Merkmale z.B. zur Erkennung von Blättern

Vorverarbeitung:

- Binärisierung
- Ausschneiden der *Region of Interest* (ROI)

Merkmale (Auswahl):

$$As$$

$$P$$

$$Projektion_{y_i} = \frac{}{Bildhöhe}$$

$$Kompaktheit = \frac{(\sum_{\forall Randpunkte \in ROI} 1)^2}{\sum_{\forall Punkte \in ROI} 1}$$

- Was ist die kompakteste geometrische Form (minimale Kompaktheit)?
- Welche Kompaktheit besitzt sie?

zur Kompaktheit:

Quelle: O. Schmidt, M. Freisinger, G. Sovak, IT-Projekt „Leaves“, Fak. Informatik, Ohm-Hochschule Nürnberg, 2011

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren Nulldurchgangsrate
   - **Merkmale für die Spracherkennung**
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Sprachproduktion

# Querschnitt durch die Stimmlippen (Animation)

# Anregungssignal

Signal von Stimmlippen (Öffnen und Schließen)

# Formung des Spektrums durch Resonator

Spracherzeugung aus der Sicht der Signalverarbeitung

Anregungssignal

Filter
(Vergleich zB Zungenposition)

gefiltertes Signal

# Zungenstellung und Resonanzraum

# Akustische Wahrnehmung

**Abbildung 4.7**

# Gehörschnecke (Cochlea)

# Gehörschnecke (Cochlea)

# Schallwahrnehmung (I)

- Hörbereich: ca. 20 Hz – 20 kHz, höchste Schallempfindlichkeit bei 3–4 kHz
- Ohrmuschel fängt Schall auf – durch äußeren Gehörgang zum Trommelfell über Gehörknöchelchen an ovales Fensters des Innenohrs (mit inkompressibler Lymphflüssigkeit gefüllt)
- dabei Druckverstärkung um den Faktor 20
- Schwingungen laufen durch die Cochlea (Gehörschnecke)
- spiralförmiger, sich verjüngender Kanal, ca. 30 mm lang + 2.5 Windungen
- Auslenkung der Basilarmembran durch Wanderwelle
- Amplitude ist stärkenabhängig, Ort der stärksten Auslenkung von Frequenzkomponenten abhängig
- Membranpunkte für unterschiedliche Frequenzen unterschiedlich empfindlich
- Haarzellen auf Membran lösen Nervenimpulse aus

# H()/77D/)$,#)&6,8%B>>F

! 7;)*H98)'(C#'(9*Y(:*#)$*#8*F9:*X;<92*V'(;CC%9CC98*#8*#(:9*?:9qB98W;8$9#C9*
W9:C9<98

! IB)%9:$B8<*#H*9:M&C<$*#8*';5*>/*?:9qB98W<:BF=B9)8#E%&"+:J.-"6 K

! 7#9)9*%9:F98*<9H9#8);H*;B)<9%9:$9$2*A9#*L9)$#HHB8<*F9:*X;B$)$D:S92*F9
[C;8<)*B8F*F9:*G#'($B8<

! IB)8;(H9-*]&8(I(92*(#9:*%9)98$C#'(*(I(9:9*?:9qB98W;BMCI)B8<
! :9#89:*]&8*`*<:&k9*7;B9:*! ]&8(I(98B8$9:)'(#9F*J&8*+51y* 0 +5=y*
%;(:89(HA;:2*W5L5*A9#*1*S3W-*U8$9:)'(#9F*J&8*=*3W*(I:A;:

# Akustische Merkmale: Überblick

= Fenster

Fenster über das komplette Signal oben

Spektrum

# Kurzzeitanalyse (I)

- Diskrete Fourier Analyse (DFT) berechnet **Frequenzspektrum** von Abschnitt des Signals
- Ziel: Beschreibung der spektralen Zusammensetzung des Sprachsignals (vgl. Funktion der Gehörschnecke)
- Problem: Diskrete Fourier Transformation ist nur für **periodische** Signale sinnvoll

# Kurzzeitanalyse (II)

- Ein Sprachsignal ist grundsätzlich nicht periodisch, da es sich im Zeitverlauf ändert
- Beobachtung: für sehr kurze Zeitabschnitte sind Sprachsignale näherungsweise stationär, d.h. die spektrale Zusammensetzung ist für ein kleines Zeitintervall konstant:

# Kurzzeitanalyse (III)

Vorgehensweise:

- schneide für jeden Zeitpunkt *m* ein Fenster aus dem Signal aus
- analysiere dieses Fenster mit der DFT
- dabei wird implizit angenommen, dass sich der ausgeschnittene Bereich periodisch wiederholt: Periodische Fortsetzung des Sprachsignalfensters
- Offene Fragen:
  - Ausschneiden entspricht der Multiplikation des Sprachsignals mit einer Fensterfunktion, die genau in dem interessanten Intervall $\neq 0$ ist: Welchen Einfluss hat das auf das Spektrum?
  - Welche Fensterfunktion sollte gewählt werden?
  - Wie groß soll das Fenster sein?
  - Sollen sich die Fenster überlappen und wenn ja, wie weit?

# Leck-Effekt bei Rechteckfenster

= Aufeinanderlaufen des Spektrum

Fenster bisschen größer und passt nicht mehr zur Frequenz

# Verminderter Leck-Effekt durch Hamming-Fenster

leakage -> Leck-Effekt (auslaufen des Spektrums)

# Wahl der Fenstergröße

- Je größer das Fenster im Zeitbereich , desto höher ist die Frequenzauflösung der DFT:
  - z.B: für ein Fenster der Größe 256 Abtastwerte beträgt die Frequenzauflösung der DFT bei $f_A$ = 16 kHz 16000/256 = 62.5 Hz
  - dagegen beträgt die Frequenzauflösung für ein Fenster der Größe 64 Abtastwerte nur 16000/64 = 250 Hz
  - macht man das Fenster zu groß, ist das Sprachsignal innerhalb des Fensters nicht mehr stationär
  - **Unschärfeprinzip**: Je besser die Zeitauflösung, desto schlechter die Frequenzauflösung und umgekehrt
  - **Kompromiss**: Typische Fortschaltzeit 10ms, Fensterbreite 25ms (Überlappung!)

# Spektrogramme

- ignoriere Phasenbeziehungen → **Betragsquadrat- oder Leistungsspektrum**
- die Zeit wird auf der x-Achse, die Frequenz in Hz auf der y-Achse dargestellt
- die Intensität wird am Grad der Schwärzung abgelesen
- Breitbandspektrogramm:
  - geringe Frequenzauflösung & hohe Zeitauflösung
  - senkrechte Balken im Abstand der Grundperiode
  - Detektion kurzdauernder Plosivphasen
- Schmalbandspektrogramm:
  - hohe Frequenzauflösung & geringe Zeitauflösung
  - waagerechte Balken im Abstand der Grundfrequenz
  - Unterscheidung nahe beieinanderliegender Formanten

# Spektrum des Phonems /e:/ (in Audacity)

# Spektrogramme

# Was sieht man im Spektrogramm?

- Formanten:
  - Bereiche im Spektrum mit starker Intensität
  - besonders im Bereich von Vokalen
  - $F_1$ und $F_2$ sind charakteristisch für bestimmte Vokale
- Harmonische der Anregungsfrequenz (Grundfrequenz)
  - ganzzahlige Vielfache der Grundfrequenz
  - bei einer Grundfrequenz von z.B. 150 Hz sind weitere Harmonische bei 300, 450, 600, ... zu finden

# Vokalviereck

u hat niedrige F1 und F2 Werte

# Bandspektren

- gehörorientierte Energieintegration innerhalb kritischer Frequenzbänder (Frequenzgruppen, Bark- bzw. Mel-Skala) wird durch Bank von Bandpassfiltern modelliert

- Ergebnis: Aus i.d.R. 128 oder 256 Koeffizienten des Leistungsspektrums erhält man ca. 25 **Mel-Spektrumskoeffizienten** (Datenreduktion)

# Berechnung der Mel-Spektrumskoeffizienten

Band spectrum coefficient $BSC(j) =$

$$.15 * A(i) + .5 * A(i+1) + .85 * A(i+2) + .85 * A(i+3) + .5 * A(i+4) + .15 * A(i+5)$$

# Mel-Spektrum

- 7 Dreieckfilter um die Mittenfrequenzen 150, 200, . . . , 450 Hz
- 3 Oktaven von 500 Hz bis 4000 Hz mit je 6 Bändern
- jedes Band endet bei den Mittenfrequenzen seiner Nachbarbänder

- Spektralverlauf wird geglättet
- harmonische Struktur verschwindet
- Vokaltraktresonanzen treten hervor
- Mel-Spektrum des neutralen Vokals (z.B. bei unbetontem „e“)

Ausfilterung der Anregungsfrequenz

# Berechnung der Mel-Cepstrumskoeffizienten

MFCC =
Mel-Frequency-Cepstral-Coefficients (Mel-Cepstrumskoeffizienten)

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - **Merkmale für die Objekterkennung**
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Merkmale für die Objekterkennung

- Bei der Objekterkennung sind häufig Merkmale von Teilbereichen des Bildes interessant, z.B. wenn das gesuchte Objekt nur einen Teil des Bildes überdeckt und dessen Position vorab nicht bekannt ist.

FFT

- Analog zur Kurzzeit-Fouriertransformation auf eindimensionalen Signalen kann auf Bildern eine **gefensterte Fouriertransformation** eingesetzt werden, um die Eigenschaften in lokalen Bereichen des Bildes zu ermitteln.

- Alternativ können **Wavelets** oder **lineare Filteroperationen** zur Merkmalberechnung verwendet werden (z.B. Gauß-Filter oder der Laplace-Operator)

Weichzeichner

Kanten hervorheben

- Eine wichtige Rolle bei der Objekterkennung spielen generell Merkmale aus **Farbe, Form (Kontur)** und **Textur (Oberflächenstruktur)** sowie **Punktmerkmale,** häufig auch in Kombination.

# Farbe und Farbhistogramme

Farbe verändert sich mit der Beleuchtung

Quelle: O. Schmidt, M. Freisinger, G. Sovak, IT-Projekt „Leaves“, Fak. Informatik, Ohm-Hochschule Nürnberg, 2011

# Textur (Oberflächenstruktur)

Bild: A. Drimbarean, P.F. Whelan, Experiments in colour texture analysis, Pattern Recognition Letters, 2001

# Regionensegmentierung mit Texturmerkmalen

- Grenzen zwischen den Regionen sind weiß markiert, Farben entsprechen dem Regionen-Label

Bilder: *Modeling Smoothly Varying Texture*, Jason Chang, John W. Fisher III, CSAIL Computer Vision Group, MIT, http://groups.csail.mit.edu

# Punktmerkmale

- Grundidee: **Markante Punkte** sind entscheidend die Objekterkennung.
- Die Eigenschaften dieser Punkte lassen sich durch **Merkmalvektoren** beschreiben.
- Die Merkmalvektoren sollten nach Möglichkeit **invariant** sein gegenüber:
  - Translation
  - Rotation
  - Skalierung
  - Änderung der Beleuchtung
  - Affiner Verzerrung (nur teilweise möglich)
- **Objekterkennung** durch **Zuordnung** der markanten Punkte im gesuchten Objekt zu den markanten Punkten im Bild anhand ihrer Ähnlichkeit (z.B. Euklidischer Abstand) => Lage des Objekts im Bild
- Alternative Anwendung: „**Stitching**“ mehrerer Bilder zu einem Panoramabild.
- Bekannte Verfahren u.a.: **SIFT** (*Scale-Invariant Feature Transfom*), **SURF** (*Speeded Up Robust Features*), **ORB** (*Oriented FAST and Rotated BRIEF*)

# Objekterkennung mit Punktmerkmalen (hier: SURF)

Quelle: Christian Ullrich, „Evaluierung und Implementierung eines Konzeptes für die autonome Orientierung eines humanoiden Roboters anhand potentieller Warnzeichen oder Gefahrensituationen“, Masterarbeit, TH Nürnberg, 2013

# Objekterkennung mit Punktmerkmalen (hier: SURF)

Quelle: Christian Ullrich, „Evaluierung und Implementierung eines Konzeptes für die autonome Orientierung eines humanoiden Roboters anhand potentieller Warnzeichen oder Gefahrensituationen“, Masterarbeit, TH Nürnberg, 2013

# Punktmerkmale

Man unterscheidet

- Keypoint-**Detektoren**, die markante Punkte *identifizieren*, und
- Keypoint-**Deskriptoren**, die markante Punkte als Merkmalsvektor *beschreiben* -> Dieser Merkmal gehört zu diesem Punkt

Verfahren wie SIFT und SURF können für beide Zwecke eingesetzt werden, aber auch in unterschiedlicher Weise miteinander kombiniert werden.

# Stitching mit Punktmerkmalen (hier: SIFT)

Quelle: Gizem Erdogan, Image Stitching, West Virginia University, 2011

# Panoramafoto mit Stitching (Ausschnitt)

Quelle: reddit/lukeallen1

# Visual SLAM auf Basis von Punktmerkmalen

- **SLAM**: Simultaneous Localization And Mapping (Simultane Lokalisierung und Kartenerstellung

Quelle: https://www.youtube.com/watch?v=Q3EMgGI6E5s

# Berechnung von Punktmerkmalen (hier: SIFT)

Quelle: Carsten Fries, „*Objekterkennung mit SIFT-Merkmalen*„, HAW Hamburg, 2010

# 4. Numerische Klassifikation

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Aufbau eines Klassifikationssystems

# Klassifikator

- **Klassifikationsverfahren** sind Methoden und Kriterien zur Einteilung von Mustern in Klassen, das heißt zur **Klassifizierung**. Ein solches Verfahren wird auch als **Klassifikator** bezeichnet.

# Numerische Klassifikation

- Aufgabe: Ordne den Merkmalvektor $\mathbf{c}$ einer Klasse $\Omega_\kappa$ zu, mit

$$\kappa \in \{1,\ldots,k\} \text{ oder } \kappa \in \{0,\ldots,k\}$$

- Die Komponenten des Vektors c sind reelle Zahlen, deshalb spricht man von **numerischer Klassifikation**

Ansätze:

- **Verteilungsfreie Klassifikatoren**: Zerlegung des Merkmalraumes durch Trennfunktionen

- **Nichtparametrische Klassifikatoren**: Speicherung der gesamten Stichprobe (z.B. Nächster-Nachbar-Klassifikator) Abstand zur Beobachtung

- **Statistische Klassifikatoren**: Verwendung parametrischer Dichtefunktionen

# Aufbau eines Klassifikationssystems

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - **Nichtparametrische** Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Nichtparametrische Klassifikatoren

- Nächster-Nachbar-Klassifikator (NN-Regel) Beispiel Kmeans: Zuordnung zum nächsten Cluster
  - Speicherung der gesamten Stichprobe
  - Zuweisung der Klasse des dem Merkmalvektor im Merkmalsraum am nächsten gelegenen Stichprobenelements

  - Nachteil: Rechenaufwand steigt mit Größe der Trainingsstichprobe
  - Variante: Mehrheitsentscheidung aus Betrachtung der $m$ nächsten Nachbarn (mNN-Regel) Nächsten Nachbarn anschauen -> Einzelner Ausreißer nicht mehr schlimm
  - Erfolgreich eingesetzt in einem der leistungsfähigsten aktuellen Gesichtserkennungssysteme, FaceNet (vgl. Abschnitt 6)

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - **Verteilungsfreie Klassifikatoren**
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Verteilungsfreie Klassifikatoren

- Bestimmung der **Parameter** von geeigneten **Trennfunktionen** aus der Lernstichprobe
- Es werden **keine Annahmen** über die statistische **Verteilung** der Merkmale getroffen

linear trennbar

nicht linear trennbar

Support Vector zum berechnen der Trennlinie (gestrichelte in der Mitte)

-> höhere Dimension führen/transformieren für lineare Trennbarkeit

- Aktuell leistungsfähigstes Verfahren: **Support Vector Machine**
- Grundidee:
  - Bestimme die breitestmögliche gerade „Straße“ zwischen den Klassengebieten. Der „Mittelstreifen“ bildet die Klassengrenze.
  - Nur diejenigen Strichprobenelemente, die auf dem „Straßenrand“ liegen, beeinflussen den Verlauf der Klassengrenze. Sie werden **Support Vectors** (Stützvektoren) genannt.
  - Durch **Transformation** des Merkmalraumes in einen Raum mit höherer Dimension werden auch verschachtelte Klassen linear trennbar.

# Support Vector Machine (SVM)

Animation: João Abrantes, https://www.youtube.com/watch?v=BE1M1xxr03s

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren Nächster Nachbar
   - Verteilungsfreie Klassifikatoren Support Vector Machine
   - **Statistische Klassifikatoren**
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Statistische Klassifikation

- Voraussetzung:
  - Kenntnisse über die statistischen Eigenschaften der Merkmalvektoren zu einer Klasse $\Omega_K$ sind gegeben
- Die Dichtefunktion $p(\mathbf{c} \mid \Omega_K)$ wird für die Klassifikation benötigt; diese wird dadurch bestimmt, dass deren unbekannten Parameter $\mathbf{a}_K$ aus einer geeigneten Stickprobe geschätzt werden.
- Beispiel Normalverteilung: Parameter sind
  - Mittelwert $\mu$
  - Standardabweichung $\sigma$

# Dichtefunktion

Beispiel Normalverteilung („Gauß-Glocke“):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Maximum-Likelihood-Schätzwerte für Mittelwert und Standardabweichung:

$$\hat{\mu} = \frac{1}{N} \sum_i x^{(i)}$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_i (x^{(i)} - \hat{\mu})^2$$

# Gauß-Verteilung (2D)

# Dichtefunktion (II)

- Gibt an, welche Wertebereiche der Zufallsvariablen $X$ wahrscheinlicher sind als andere
- Pendant zur Wahrscheinlichkeitsfunktion für diskrete Zufallsvariablen, aber
- aus der Dichtefunktion $f(x)$ lässt sich nicht unmittelbar die Wahrscheinlichkeit konkreter Werte (z.B. $X=1,3627$) ablesen. Diese ist bei kontinuierlichen Zufallsvariablen immer 0.
- Nur für einen Wertebereich ($a < X < b$) lässt sich eine Wahrscheinlichkeit angeben. Sie entspricht dem Integral der Dichtefunktion in den Grenzen von a bis b:

$$P(a \leq X \leq b) = \int_a^b f(x)\,dx$$

# Bedingte Wahrscheinlichkeiten

- Beispiel: Wir werfen einen Würfel
  - sei A das Ereignis „gerade Augenzahl“
  - sei B das Ereignis „Augenzahl kleiner oder gleich 3“
- Wie groß ist die Wahrscheinlichkeit für eine gerade Augenzahl?
- $P(A) = 3/6 = 1/2$ (mögliche Ergebnisse: 1, **2**, 3, **4**, 5, **6**)
- Wir erhalten nun einen Hinweis, dass das Ergebnis kleiner oder gleich drei ist. Wie groß ist nun die Wahrscheinlichkeit für eine gerade Augenzahl?
- $P(A|B)$ („*P von A unter der Bedingung B*“) ist 1/3 (mögliche Ergebnisse: 1, **2**, 3)

# Statistische Klassifikation

- Beobachtete Muster werden als das **Ergebnis eines Zufallsprozesses** aufgefasst:

  - Innerhalb des Problemkreises $\Omega$ wird zufällig eine Klasse ausgewählt, wobei die Klasse $\Omega_\varkappa$ mit der *a priori* Wahrscheinlichkeit $p(\Omega_\varkappa)$ gewählt wird.

  - Nach Wahl von $\Omega_\varkappa$ wird eine Beobachtung der Zufallsvariablen $\mathbf{c}$ gemacht (der Merkmalvektor eines Musters), wobei $\mathbf{c}$ die bedingte Dichte $p(\mathrm{c} \mid \Omega_\varkappa)$ hat

  - Ergebnis des Zufallsprozesses ist ein Paar aus einer Klassennummer $\kappa$ und einer Beobachtung $\mathbf{c}$

# Bayes-Formel

- Gesucht ist diejenige Klasse $\Omega_\varkappa$ mit maximaler *a posteriori* Wahrscheinlichkeit $p(\Omega_\varkappa \mid \mathbf{c})$ Wahrscheinlichkeit, dass eine bestimmte Klasse vorliegt
- Problem: $p(\Omega_\varkappa \mid \mathbf{c})$ lässt sich i.d.R. nicht direkt bestimmen
- Dagegen lässt sich die bedingte Dichtefunktion $p(\mathbf{c} \mid \Omega_\varkappa)$ in viele Fällen gut aus Stichproben abschätzen, ebenso wie die *a priori* Wahrscheinlichkeit $p(\Omega_\varkappa)$
- Die Bayes-Formel löst das Problem:

(Müssen wir können)

$$p(\Omega_\kappa \mid \mathbf{c}) = \frac{\overset{\text{a priori}}{p(\Omega_\kappa)}\,\overset{\text{a posteriori}}{p(\mathbf{c} \mid \Omega_\kappa)}}{p(\mathbf{c})}$$

# Beispiel: Screening-Untersuchung

- Es gibt einen neuen Schnelltest auf eine gefährliche Krankheit K
- Die Bevölkerung wird zu einem Screening aufgefordert
- Der Schnelltest wird mit **„99 % Genauigkeit“** beworben, denn er hat
  - eine Falsch-Positiv-Rate von nur einem Prozent:
    $p(\textit{positiv}|\textit{nicht K}) = 0.01$ (d.h. die sog. ***Spezifität*** beträgt 99%)
  - eine Falsch-Negativ-Rate von nur einem Prozent:
    $p(\textit{negativ}|K) = 0.01$ (d.h. die sog. ***Sensitivität*** beträgt 99%)
- Die Erkrankungshäufigkeit in der Bevölkerung ist eins zu tausend, d.h. $p(K) = 0.001$ (*a-priori*-Wahrscheinlichkeit)
- Herr Müller erhält ein positives Testergebnis
- Wie groß ist die Wahrscheinlichkeit, dass Herr Müller tatsächlich an der Krankheit K erkrankt ist?

# Beispiel: Screening-Untersuchung

- Anwendung der Bayes-Formel:

$$p(K|positiv) = p(K)\cdot p(positiv|K) \,/\, p(positiv) =$$
$$= 0.001\cdot 0.99/(0.999 \cdot 0.01 + 0.001 \cdot 0.99)$$
$$\approx 9{,}02\%$$

$$p(K|positiv) = \frac{p(K)\cdot p(positiv|K)}{\underbrace{p(\bar{K})\cdot p(positiv|\bar{K}) + p(K)\cdot p(positiv|K)}_{p(positiv)}}$$

# Beispiel: Screening-Untersuchung

- Anwendung der Bayes-Formel:

$$p(K|positiv) = p(K)\cdot p(positiv|K) / p(positiv) =$$
$$= 0.001\cdot 0.99/(0.999 \cdot 0.01 + 0.001 \cdot 0.99)$$
$$\approx 9{,}02\%$$

Herr Müller ist trotz positiver Diagnose mit fast 91 Prozent Wahrscheinlichkeit **nicht** an K erkrankt!

# Beispiel Screening-Untersuchung

Bayes-Formel Rechnung anhand einer Population

An K erkrankt

| | | p | n | Summe |
|---|---|---|---|---|
| Schnelltest positiv | p` | 99 | 999 | 1098 |
| | n` | 1 | 98901 | 98902 |
| | Summe | 100 | 99900 | 100.000 |

Einwohner

# Beispiel Screening-Untersuchung

An K erkrankt

| | | p | n | Summe |
|---|---|---|---|---|
| Schnelltest positiv | p` | 99 | 999 | 1098 |
| | n` | 1 | 98901 | 98902 |
| | Summe | 100 | 99900 | 100.000 |

$$recall = sensitivity = \frac{99}{100} = 99\%$$

$$precision = \frac{99}{1098} = 9,02\%$$

p(positiv|K)

$$specificity = \frac{98901}{99900} = 99\%$$

# Beispiel Screening-Untersuchung

An K erkrankt

| | | p | n | Summe |
|---|---|---|---|---|
| Schnelltest positiv | p` | 99 | 999 | 1098 |
| | | 1 | 98901 | 98902 |
| | | | 99900 | |

Die Wahrscheinlichkeit, dass Herr Müller nach positivem Test (!) tatsächlich erkrankt ist.

Spezifität: Vorsicht, Marketing!

$$recall = sensitivity = \frac{99}{100} = 99\%$$

$$precision = \frac{99}{1098} = 9,02\% \qquad specificity = \frac{98901}{99900} = 99\%$$

# Bayes-Klassifikator

- Der Nenner der Bayes-Formel

$$p(\Omega_\kappa \mid \mathbf{c}) = \frac{p(\Omega_\kappa)\, p(\mathbf{c} \mid \Omega_\kappa)}{p(\mathbf{c})}$$

  ist unabhängig vom gesuchten Klassenindex κ und spielt deshalb bei der Bestimmung des Maximalwerts keine Rolle

- **Entscheidungsregel**:
  Gegeben ein Merkmalvektor **c**, dann entscheide dich für diejenige Klasse $\Omega_\kappa$, für die gilt:

Die Klasse, wo der Term max ist
ist die gesuchte Klasse

$$p(\Omega_\kappa)\, p(\mathbf{c} \mid \Omega_\kappa) = \max_\lambda p(\Omega_\lambda)\, p(\mathbf{c} \mid \Omega_\lambda)$$

Zähler = max (Über den Term)

- Es lässt sich zeigen, dass der **Bayes-Klassifikator** der Klassifikator mit der **geringsten Fehlerwahrscheinlichkeit** ist

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - **Neuronale Netze**
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Biologische Neuronale Netze

- Neuronen kombinieren viele Eingangsimpulse (über die **Dendriten**) zu einem Ausgangsimpuls (auf dem **Axon**).
- Bestimmte Kombinationen von Eingangsimpulsen führen zu einem Ausgangsimpuls, andere nicht.
- Der Ausgangsimpuls wird über die **Synapsen** auf Dendriten anderer Neuronen weitergeleitet.
- Das Verhalten eines einzelnen Neurons lässt sich vereinfacht über eine einfache mathematische Funktion simulieren.

Bilder: Carry Rhode, Lower Columbia College, http.//lowercolumbia.edu

# Perzeptron (Frank Rosenblatt, 1957)

- einfache Simulation eines biologischen Neurons
- Funktionsweise:

- Als „Aktivierungsfunktion“ f dient eine sog. Schwellenwertfunktion:

$$f(x) = \begin{cases} 1 \text{ falls } x \geq 0 \\ 0 \text{ sonst} \end{cases}$$

# Training eines Perzeptrons

- Ziel: Bestimme die Gewichte des Perzeptrons so, dass für eine gegebene Lernstichprobe die gewünschte Ausgabe erzeugt wird (z.B. 1 für die Klasse „Koala“, 0 für die Klasse „Panda“).
- Geeignete Gewichte lassen sich **iterativ** bestimmen:
  - Beginnend mit einer zufälligen (oder uniformen) Initialisierung der Gewichte wird wiederholt
    - die Ausgabe $y$ zu jedem Element der Lernstichprobe bestimmt
    - anhand der Differenz zu dem erwünschten Ergebnis eine Korrektur der Gewichte vorgenommen
- In günstigen Fällen ist der Fehler am Ende des Trainings für alle Elemente der Lernstichprobe 0.
- Das gelingt aber nur bei **linear separierbaren** Merkmalsgebieten, da das Perzeptron lediglich eine **lineare Trennebene** modelliert.

# Training eines Perzeptrons (Beispiel)

# Mehrschichtperzeptron *(Multilayer Perceptron,* MLP*)*

- Künstliche Neuronale Netze entstehen durch Verknüpfung von mehreren Perzeptrons zu einem Netzwerk (*artificial neural network*, ANN)
- Häufigste Topologie: 3 (oder mehr) Schichten von Perzeptrons (Eingabeschicht, verborgene Schicht, Ausgabeschicht)
- Mögliche Topologie für einen 3-dimensionalen Merkmalvektor und 2 Klassen:

- Training erfolgt i.d.R. über den sog. *Backpropagation*-Algorithmus
- Selbst MLPs mit nur einer verborgenen Schicht können beliebig komplexe Funktionen und nichtlineare Klassengrenzen modellieren *(Universal Approximation Theorem)*
- Künstliche neuronale Netze mit **rekurrenten** (= rückgekoppelten Kanten) sind sogar **Turing-vollständig**
- **Deep Neural Network (DNN), Deep Learning** *(*und zunehmend auch „Künstliche Intelligenz“) sind Buzzwords für neuronale Netze mit mehr als nur ein paar Schichten (ca. 4 bis >1000)

# But what *is* a Neural Network?

https://www.youtube.com/watch?v=aircAruvnKk

# Aktivierungsfunktionen (Auswahl)

- Nichtlineare Aktivierungsfunktionen sind zwingend notwendig, weil sich jedes Netz sonst durch Ausmultiplizieren der linearen Transformationen durch eine einzige Schicht darstellen ließe und damit nur noch lineare Funktionen modellieren könnte.
- **Schwellwertfunktion**: Biologisch plausibel (Neuronen feuern oder sie feuern nicht), aber mathematisch unhandlich weil kein Gradientenabstieg möglich.
- Die **Sigmoidfunktion** „quetscht“ alle reellen Zahlen in das Intervall ]0;1[ und war lange Zeit in MLPs sehr beliebt. In tiefen Netzen führt sie zum **Vanishing Gradient Problem**.

  je mehr Schichten, desto weniger bleibt vom Gradienten übrig
- Die **ReLU**-Funktion wird in tiefen neuronalen Netzen häufig verwendet und trägt dazu bei, das Vanishing Gradient Problem zu vermeiden
- In der letztem Schicht eines NNs für die Klassifikation wird i.d.R. die **Softmax**-Aktivierungsfunktion verwendet. Sie liefert Schätzwerte für die **a posteriori-Wkten** aller Klassen:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Summe ganzer letzter Schicht ist 1

Schwellwertfunktion

Sigmoidfunktion

ReLU (Rectified Linear Unit)

# Definition eines MLP in Keras

- **Keras** ist aktuell die populärste Deep-Learning-Library
- High-level-API für Python, die auf einem darunter liegenden Deep-Learning-Framework (**TensorFlow**, CNTK oder Theano) aufsetzt und dieses abstrahiert
- Beispiel-MLP für den MNIST-Ziffernerkennungs-Datensatz: 784 Eingabewerte (Pixel), 2 verborgene Schichten mit je 512 Knoten, num_classes (=10) Ausgabeknoten („Wahrscheinlichkeiten“).

```
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
```

- **Dropout** ist eine Maßnahme gegen Überadaption (**overfitting)**. Ein Teil der Knoten jeder Schicht wird während des Trainings zeitweise zufällig deaktiviert, hier 20 Prozent.

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - **Unüberwachtes Lernen**
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Unübewachtes Lernen

- Beispiel: Transformation von kontinuierlichen Eingabevektoren auf ein endliches **Klassenalphabet** (Menge von Kodebuchklassen)
- Jedem Vektor $\mathbf{c}$ wird der Index $k_t$ seiner Kodebuchklasse (oder Quantisiererzelle) zugeordnet (Vektorquantisierung)
- Kodebücher können **unüberwacht** aus einer Stichprobe gelernt werden
- Verfahren hierfür z.B.:
  - k-Means-Algorithmus
  - EM-Algorithmus
- Grundidee dieser Verfahren wird auch zur Schätzung der Parameter komplexerer Modellierungsverfahren eingesetzt (z.B. Hidden-Markov-Modelle).

# k-Means-Algorithmus (Wiederholung)

Ablauf:

1. Initialisierung: (zufällige) Auswahl von $k$ Clusterzentren,
   z.B. durch zufällige Auswahl von $k$ Eingabevektoren als Clusterzentren

2. Zuordnung: Jedem Eingabevektor wird dem ihm am nächsten liegenden Clusterzentrum zugeordnet
   (Abstandsmaß: z.B. Euklidischer Abstand)

3. Neuberechnung: Es werden für jeden Cluster die Clusterzentren durch Schwerpunktbildung neu berechnet Mittelwert von Vektoren in einem Cluster

4. Wiederholung:
   Falls sich nun die Zuordnung der Objekte ändert, weiter mit Schritt 2, sonst Abbruch

# Gaußsche Mischverteilung (GMM)

- mehrere Gauß-Dichten werden gewichtet und aufsummiert
- auch multimodale Dichten können so repräsentiert werden
- Definition der **GMM-Dichte (**mehrdimensional)**:**

$$P(\mathbf{x} \mid p(1), \mu_1, \Sigma_1, \cdots, p(M), \mu_M, \Sigma_M) = \sum_{m=1}^{M} p(m) \cdot \mathcal{N}\left(\mathbf{x} \mid \mu_m, \Sigma_m\right)$$

  - N bezeichnet die Dichtefunktion der mehrdimensionalen (oder multivariaten) Normalverteilung (Gaußsche Glockenkurve)
  - *p(m)* : *a-priori*-Wahrscheinlichkeiten der einzelnen Dichten
- Es gilt:

$$\sum_{m=1}^{M} p(m) = 1$$

- Problem: die Anzahl M der Normalverteilungen festzulegen (wie bei k-Means-Algorithmus)
- **Maximum-Likelihood-Schätzung** der Parameter mit dem **EM-Algorithmus**

# Gaußsche Mischverteilung (3-modal, 1D)

# EM-Algorithmus (Expectation-Maximization)

Zufällig erzeugte Vektoren einer Gaußschen 4–Mischverteilung im $IR^2$

# EM-Algorithmus

Beobachtung im $\mathbb{R}^2$

# EM-Algorithmus

Mit EM berechnete $\mu$, $\sigma$

# EM-Algorithmus

- Problem bei der Schätzung der Parameter eines GMM: die Zuordnung eines Merkmalvektors $\boldsymbol{x}$ zu einer der $\mathrm{M}$ Dichten ist nicht bekannt.

- Der EM-Algorithmus löst das Problem durch iterative Optimierung (analog zum k-Means-Algorithmus).

- Es ist nicht garantiert, dass so die besten Parameter gefunden werden, die die Wahrscheinlichkeit der Trainingsstichprobe maximieren.

- Aber: es wird immer ein lokales Optimum gefunden, d.h. Parameter, die mindestens so gut sind wie die Startwerte.

- Erfahrung: es werden sehr gute Parameter gefunden, wenn die Startwerte gut sind und genug Trainingsdaten vorliegen

# EM-Algorithmus

1. Bestimme Startparameter

$$\hat{B}^0 = \{p(m), \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, \forall m\}$$

2. Für $i$ = 1,2,3,...:

**(1)** **Expectation**-Schritt: Bestimme für jede Klasse $\Omega_m$ und jeden Merkmalvektor $x_j$ die a-posteriori Wahrscheinlichkeit:

$$\gamma_{j,m}^{i} = P(\Omega_m \mid \boldsymbol{x}_j, \hat{B}^{i-1}) = \frac{\mathcal{N}(\boldsymbol{x}_j \mid \mu_m^{i-1}, \boldsymbol{\Sigma}_m^{i-1}) \cdot p(m)}{\sum_k \mathcal{N}(\boldsymbol{x}_j \mid \mu_k^{i-1}, \boldsymbol{\Sigma}_k^{i-1}) \cdot p(k)}$$

Bayes-Formel

# EM-Algorithmus

(2) **Maximization**-Schritt: Berechne die neuen Parameter $\hat{B}^i$ :

$$p(m) = \frac{1}{J} \sum_{j=1}^{J} \gamma_{j,m}^{i}$$

$$\boldsymbol{\mu}_m = \frac{1}{\sum_j \gamma_{j,m}^{i}} \sum_{j=1}^{J} \gamma_{j,m}^{i} \cdot \boldsymbol{x}_j$$

Gewichtungsfaktor a-priori-Wahrscheinlichkeit der Klasse m

$$\boldsymbol{\Sigma}_m = \frac{1}{\sum_j \gamma_{j,m}^{i}} \sum_{j=1}^{J} \gamma_{j,m}^{i} \cdot (\boldsymbol{x}_j - \boldsymbol{\mu}_m) \cdot (\boldsymbol{x}_j - \boldsymbol{\mu}_m)^\top$$

(3) Prüfe eine Abbruchbedingung.

# EM-Algorithmus

$$\boldsymbol{\mu}_1 = \begin{pmatrix} 0.2 \\ 0.8 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix},$$

$$\boldsymbol{\mu}_2 = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

# EM-Algorithmus

„Weiche“ Zuordnung zu den Klassen führt auch zu lila Punkten.

# EM-Algorithmus

# EM-Algorithmus

$$\boldsymbol{\mu}_1 = \begin{pmatrix} 0.31 \\ 0.32 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.008 & 0 \\ 0 & 0.008 \end{bmatrix},$$

$$\boldsymbol{\mu}_2 = \begin{pmatrix} 0.68 \\ 0.71 \end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.008 & 0 \\ 0 & 0.008 \end{bmatrix}$$

# EM-Algorithmus

- Schätzung der GMM-Parameter nur eine von vielen Anwendungen

- GMMs stellen einen zweistufigen Zufallsprozess dar, der eine **verborgene** Zufallsvariable beinhaltet: Man kann zwar den vom GMM erzeugten Merkmalvektor beobachten, es bleibt jedoch unsichtbar, welche der Merkmalkomponenten ihn erzeugt hat.

- EM-Algorithmus auch auf andere zweistufige Zufallsprozesse mit verborgenen Zufallsvariablen anwendbar

- Wichtige weitere Anwendung: Schätzung der Modellparameter von **Hidden-Markov-Modellen**

# 5. Spracherkennung

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - **Dynamic Time Warping**
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# DTW: Problemstellung

- Aufgabe: **Einzelworterkennung**
- **Idee**: **Nächster-Nachbar-Klassifikator** für Wörter
  - Anlernphase ("Trainingsphase")
    - Aufnahme eines Referenzmusters zu jedem Wort im Wortschatz
    - Berechne zu jedem Referenzmuster die Folge von Merkmalvektoren und speichere diese
  - Laufender Betrieb
    - Aufnahme eines unbekannten Sprachmusters
    - Berechne zu diesem Sprachmuster die Folge von Merkmalvektoren
    - Bestimme nun den Abstand dieser Folge von allen Referenz-Folgen
    - Entscheidung für das Wort mit dem geringsten Abstand
- **Problem**: Wie berechnet man den Abstand von zwei Folgen von Merkmalvektoren?
  - Summe der Abstände der einzelnen Merkmalvektoren? Aber:
    - Folgen können unterschiedlich lang sein
    - Dehnungen und Stauchungen müssen nicht linear erfolgen, z.B. können einzelne Vokale stark gedehnt sein

# DTW: Grundidee

Zuordnung von jedem Signalpunkt A zu jedem Signalpunkt B

Reihenfolge bleibt gleich halloooo -> haaaallo

h auf h
a auf aaaa
...
Zeitliche Streckung/Stauchung beachten

Quelle:http://cst.tu-plovdiv.bg/bi/DTWimpute/DTWalgorithm.html

# DTW: Algorithmus

Abstand von den Zuordnungen -> Summe = 6

Wort 1 mit n Merkmalen

Wort 2 mit m Merkmalen

```
int DTWDistance(char s[1..n], char t[1..m]) {
    declare int DTW[0..n, 0..m]
    declare int i, j, cost

    for i := 1 to m
        DTW[0, i] := infinity
    for i := 1 to n
        DTW[i, 0] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := 1 to m
            cost:= d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                        DTW[i  , j-1],    // deletion
                                        DTW[i-1, j-1])    // match

    return DTW[n, m]
}
```

Abstand der Merkmalvektoren

Bsp:

Merkmals-vektoren

Entscheidender Abstand für Zuordnung

i := 0 ; j := 1

6 - 1 = 5

↳ min (∞, ∞, 1)

5 + 1 = 6

i := 0 , j := 0

2 - 1 = 1 -> min(∞, 0, ∞) = 0

↳ 1 + 0 = 1

Quelle: Wikipedia

# DTW: Rückverzeigerung und Backtracking

- DTW-Algorithmus liefert in der beschriebenen Form nur den Abstand, nicht jedoch die optimale zeitliche Zuordnung

- Zeitliche Zuordnung kann ggf. durch **Rückverzeigerung** und **Backtracking** ermittelt werden

- Bei jeder Minimierungsoperation speichert man in einem zusätzlichen *n* x *m*-Array einen Zeiger (bzw. einen geeigneten Hinweis) auf das Vorgängerfeld, das den minimalen Kostenbeitrag geliefert hat (**Rückverzeigerung**)

- Nach Abschluss der Iteration lässt sich der optimale Pfad ausgehend vom Feld [n,m] rückwärts rekonstruieren (**Backtracking**)

# DTW in der Spracherkennung

- DTW ist ein klassischer Algorithmus zur Einzelworterkennung
- Besonders geeignet für sprecherabhängige Erkennung von Wörtern, zu denen nur ein einziges akustisches Referenzmuster vorliegt
- Einsatz z.B. in älteren Handys, aber auch für Sprachsteuerungen in Kampfflugzeugen
- In leistungsfähigeren Spracherkennungssystemen heute fast ausschließlich HMMs (Hidden-Markov-Modelle)
- Auch diese sind in der Lage, die dynamische Zeitverzerrung ähnlich dem DTW-Algorithmus zu modellieren.
- Der für die Erkennung mit HMMs eingesetzte Viterbi-Algorithmus ähnelt sehr stark dem DTW-Algorithmus (siehe nächster Abschnitt)

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - **Hidden-Markov-Modelle**
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Einzelworterkennung mit Bayes-Klassifikator

- gegeben: Menge von Wörtern $\mathcal{W} = \{W_1,\ldots,W_L\}$
- beobachtet wird eine Merkmalvektorfolge $\mathbf{X} = \mathbf{x}_1,\ldots,\mathbf{x}_T$ (Äußerung)
- welches Wort wurde gesprochen?

→ Entscheidung gemäß der Bayes-Formel

$$l^* = \arg\max_l P(W_l \mid \mathbf{X}) \text{ mit } P(W_l \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid W_l) \cdot P(W_l)}{P(\mathbf{X})}$$

- *a-priori*-Wkt. der Wörter $P(W_l)$ werden z.B. durch Auszählen einer Stichprobe geschätzt
- $P(\mathbf{X} \mid W_l)$ kann nicht z.B. durch eine Normalverteilungsdichte approximiert werden, da die Länge $T$ der Äußerung variabel ist
- aber: $P(\mathbf{X} \mid W_l)$ kann durch Hidden-Markov-Modelle (HMM) geschätzt werden

# Diskretes HMM: Beispiel

- HMM für das Wort „haben“ mit gängigen Verschleifungen und dialektalen Variationen

Model mit Reihen von Zuständen, wo man weiß wie das Wort ausgesprochen wird

# Diskretes HMM: Beispiel

Summe = 1.0

- **Frage**: Wie wahrscheinlich wird die Beobachtungssequenz „h h a a b m m“ von dem Modell produziert?

$1{,}0 \cdot 0{,}8 \cdot 1{,}0 \cdot 0{,}2 \cdot 0{,}9 \cdot 0{,}7 \cdot 0{,}9 \cdot 0{,}2 \cdot 1{,}0 \cdot 0{,}1 \cdot 0{,}3 \cdot 0{,}8 \cdot 0{,}3 \cdot 0{,}2$

h  bleiben in 1

# Diskretes HMM: Beispiel

**Antwort**:

- Die einzige mögliche Zustandssequenz zur Beobachtungssequenz „h h a a b m m“ ist 1-1-2-2-3-5-5.
- Die Wahrscheinlichkeit errechnet sich damit so:
  1,0·0,8·1,0·0,2·0,9·0,7·0,9·0,2·1,0·0,1·0,3·0,8·0,3·0.2

  ↳ siehe oben

# Diskretes HMM: Beispiel

- In diesem Beispiel gibt es nur eine einzige Zustandssequenz, die die Beobachtungssequenz erzeugen kann
- Be mehreren möglichen Pfaden werden die Wahrscheinlichkeiten der einzelnen Pfade aufsummiert.

# Hidden-Markov-Modell: Definition

Hidden -> man beobachtet nur die Merkmalsvektoren

$\lambda = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, mit

$\boldsymbol{\pi} = (\pi_i) : P(q_1)$ Startwahrscheinlichkeiten

$\mathbf{A} = \left[ a_{ij} \right] : P(q_t = s_j \mid q_{t-1} = s_i)$ Übergangswahrscheinlichkeiten Wechsel in verschiedene Zustände

$\mathbf{B} = (b_j) : P(x_t \mid q_t = s_j)$ Emissionswahrscheinlichkeiten, abhängig vom Modelltyp:

Wskt für den Buchstaben

- Diskretes HMM (DHMM): endliches Alphabet, z.B. Phonemsymbole
  Erkennung auf Lauten
- Kontinuierliches HMM (CDHMM / GMM-HMM): $b_j(\mathbf{x}) = \sum_{k=1}^{K_j} \omega_{jk} \cdot \mathcal{N}\left(\mathbf{x} \mid \mu_{jk}, \Sigma_{jk}\right)$
  Gaußverteilung für die Wskt des Merkmalsvektor in einem Zustand
- Deep-Learning-HMM-Hybrid (DNN-HMM): Werte von **B** werden durch die Ausgabeschicht eines tiefen Neuronalen Netzes bereitgestellt (modifiziert entsprechend der Bayes-Formel)

# Kontinuierliche HMMs

- Um diskrete HMMs für Spracherkennung verwenden zu können, müssen die Merkmalvektoren zunächst auf Symbole abgebildet werden, d.h. klassifiziert (Vektorquantisierung).
- Dieser Schritt geht mit einem Informationsverlust einher, der zu einer subobtimalen Erkennungsrate führt.
- CDHMMs / GMM-HMMs vermeiden dieses Problem, indem sie kontinuierliche Dichtefunktionen (pdfs) direkt mit den HMM-Zuständen verknüpfen.

# HMM: Vier Fragestellungen

Struktur

1. Wie soll die **Topologie** des HMM festgelegt werden, d.h. welche Zustandsübergänge werden erlaubt, $P(s_i \mid s_j) > 0$, und welche nicht, $P(s_i \mid s_j) = 0$?

2. Wie kann die **Produktionswahrscheinlichkeit** $P(\mathbf{x}_1, .., \mathbf{x}_T \mid \lambda)$ effizient berechnet werden? Alle verschiedene Varianten

3. Dekodierung: Wie kann **die wahrscheinlichste Zustandsfolge** ermittelt werden wenn eine Beobachtung $\mathbf{x}_1, .., \mathbf{x}_T$ gegeben ist?

   Genau eine Reihenfolge

4. Wie können die **Parameter** des HMM aus einer Trainingsstichprobe **geschätzt** werden?

# HMM-Topologien für die Spracherkennung

Keine Rückschritte, da Vertauschungsgefahr

Anzahl der Zustände
=> typischerweise 3 Mal so viel
wie Laute: haben --> 15 Zustände

Übergangsmatrix

# Berechnung der Produktionswahrscheinlichkeit

- Wir suchen $P(\mathbf{X}|\,\boldsymbol{\lambda})$, also die Wahrscheinlichkeit, dass $\mathbf{X}=\mathbf{x}_1, .., \mathbf{x}_T$ vom HMM $\lambda$ erzeugt wurde.
- Brute-Force-Ansatz: Summe aller möglichen Zustandsreihenfolgen

$$P(\mathbf{X}\,|\,\lambda)=\sum_{\mathbf{q}\in Q^T} P(\mathbf{X},q\,|\,\lambda)=\sum_{\mathbf{q}\in Q^T} \pi_{q_1} b_{q_1}(\mathbf{x}_1)\cdot\prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{x}_t)$$

(etwa $2T\cdot N^T$ Multiplikationen), wobei Q die Menge der HMM-Zustände in $\boldsymbol{\lambda}$ ist und $N$=|$Q$| deren Anzahl

- Exponentielle Komplexität in Bezug auf die Signaldauer T
- Effizienter: berechne **Vorwärts-** und **Rückwärtswahrscheinlichkeiten** α und β

Wie wahrscheinlich bin ich zum Zeitpunkt t im Zustand J und habe bis dahin die Merkmalsvektoren x_1 bis x_t produziert

- Vorwärtswahrscheinlichkeit: $\alpha_t(j)=P(\mathbf{x}_1...\mathbf{x}_t,q_t=j\,|\,\lambda)$
- Rückwärtswahrscheinlichkeit: $\beta_t(i)=P(\mathbf{x}_{t+1}...\mathbf{x}_T,q_t=i\,|\,\lambda)$

Idee: Zusammenfassen und Aufsummieren der verschiedenen Varianten wie man in einen Zustand landet

- Zu jedem Zeitpunkt *t* gilt: $\alpha_t(j)\cdot\beta_t(i)=P(\mathbf{X},q_t=j\,|\,\lambda)$ und

$$P(\mathbf{X}\,|\,\lambda)=\sum_{j=1}^{N}\alpha_t(j)\cdot\beta_t(j)$$

# Vorwärtsalgorithmus and Rückwärtsalgorithmus

Für Erkennung benötigt man Vorwärtsalgorithmus; Für Training zusätzlich auch noch Rückwärts

Zwei gleichwertige Algorithmen (auf Basis der Vorwärts- bzw. Rückwärtswahrscheinlichkeiten):

- **Initialisierung**:

für $j = 1,\ldots,N$

$$\alpha_1(j) = \pi_j b_j(\mathbf{x}_1)$$

für $j = 1,\ldots,N$

$$\beta_T(j) = 1$$

- **Rekursion:**

für $t > 1$ und $j = 1,\ldots,N$

$$\alpha_t(j) = \left( \sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij} \right) b_j(\mathbf{x}_t)$$

für $t < T$ und $I = 1,\ldots,N$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(x_{t+1}) \cdot \beta_{t+1}(j)$$

- **Terminierung**: Berechne

$$P(\mathbf{X} \mid \lambda) = \sum_{j=1}^{N} \alpha_T(j)$$

Vorwärts

$$P(\mathbf{X} \mid \lambda) = \sum_{j=1}^{N} \pi_j b_j(x_1) \beta_1(j)$$

Rückwärts

# Viterbi-Algorithmus

- Wir suchen die Zustandsfolge **q*** = $q_1, \ldots, q_T$ welche die Wahrscheinlichkeit $P(q_1, \ldots, q_T \mid \mathbf{X}, \lambda)$ maximiert, also die wahrscheinlichste Zustandsfolge, die unsere Beobachtung **X** erzeugt hat.

- **q*** ist die **optimale Zustandsfolge,** wenn

$$P\left(\mathbf{X},\mathbf{q}^* \mid \lambda\right) = \max_{\mathbf{q}\in Q^T} P\left(\mathbf{X},\mathbf{q} \mid \lambda\right) =: P^*\left(\mathbf{X} \mid \lambda\right)$$

- Der **Viterbi-Algorithmus** ist eine Modifikation des Vorwärtsalgorithmus, der im Rekursionsschritt anstelle der Summe eine Maximierung durchführt.

- Zur Bestimmung der besten Zustandsfolge werden wird (wie beim Dynamic Time Warping) eine Rückverzeigerung und *Backtracking* verwendet

# Viterbi-Algorithmus

- **Initialisierung:**

  für $j = 1,\ldots,N$

  $$\vartheta_1(j) = \pi_j b_j(\mathbf{x}_1) \quad \text{and} \quad \psi_1(j) = 0$$

- **Rekursion:**

  für $t > 1$ und $j = 1,\ldots,N$

  $$\vartheta_t(j) = \max_i \left(\vartheta_{t-1}(i) \cdot a_{ij}\right) b_j(\mathbf{x}_t) \quad \text{und} \quad \psi_t(j) = \arg\max_i \vartheta_{t-1}(i) \cdot a_{ij}$$

- **Terminierung**: Berechne

  $$P^*(\mathbf{X} \mid \lambda) = \max_j \vartheta_T(j) \quad \text{und} \quad q_T^* = \arg\max_j \vartheta_T(j)$$

- **Backtracking**: für $t = T-1, \ldots, 1$

  $$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

Bsp: ja/nein Erkennung

jaIn ?

-> 6 Zustände = 6 Zeilen

-> j a I n = 4 Spalten

$\vartheta$

| | | | | |
|---|---|---|---|---|
| $S_1$ | $0{,}5 \cdot 0{,}1$ | $0{,}5 \cdot 0$ | 0 | 0 |
| $S_2$ | 0 | $(0{,}5 \cdot 0{,}1) \cdot 0{,}5 \cdot 0{,}8$ | $0{,}5^2 \cdot 0{,}1^2 \cdot 0{,}8$ | $0{,}5^2 \cdot 0{,}1^3 \cdot 0{,}8$ |
| $S_3$ | $0{,}5 \cdot 0{,}8$ | $0{,}5 \cdot 0$ | 0 | 0 |
| $S_4$ | 0 | $0{,}5 \cdot 0{,}8 \cdot 0{,}5 \cdot 0{,}8$ | $0{,}5^3 \cdot 0{,}8^2 \cdot 0{,}1$ | 0 |
| $S_5$ | 0 | 0 | $0{,}5^3 \cdot 0{,}8^3$ | $0{,}5^4 \cdot 0{,}8^3 \cdot 0{,}1$ |
| $S_6$ | 0 | 0 | 0 | $0{,}5^4 \cdot 0{,}8^4$ |

Vorgänger über a

max

0,0002

0,052

j a I n

=> max = 0,052 => nein

# Schätzung der Modellparameter

- Die Schätzung der HMM-Parameter erfolgt meist mittels des **Baum-Welch-Algorithmus** (Vorwärts-Rückwärts-Algorithmus, *Forward Backward Algorithm*), einer Variante des EM-Algorithmus.
- Der Baum-Welch-Algorithmus maximiert die Parameter des HMMs so, dass die Wahrscheinlichkeit der Lernstichprobe maximiert wird (Maximum-Likelihood- bzw. ML-Schätzung).
- Wie der EM-Algorithmus zur Schätzung der Parameter Gaußscher Mischverteilungen (GMMs) findet auch der Baum-Welch-Algorithmus nur ein lokales Optimum, nicht notwendigerweise das globale.

# Baum-Welch-Algorithmus

Maximierungsschritt im Fall eines diskreten HMM:

$$\hat{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N} \alpha_1(j)\beta_1(j)}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}$$

$$\hat{b}_{jk} = \frac{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)\chi_{[\mathbf{x}_t = v_k]}}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)}$$

Die χ-Funktion liefert 1, falls die Bedingung wahr ist, sonst 0.

# 6. Objekterkennung

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - **Kantendetektion**
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Kantendetektion

- Wir haben bereits im Kapitel 2 mehrere Verfahren zur Hervorhebung von Kanten in Bildern kennengelernt:
  - Laplace-Operator
  - Sobel-Operator
  - Hochpassfilterung des Bildes mittels Fourier-Transformation
- Auf diese Weise entsteht ein sogenanntes Kantenbild (bzw. 2 Kantenbilder beim Sobel-Operator), in dem die Stärke der Kante durch den Grauwert repräsentiert wird.
- Für die Objekterkennung ist es häufig sinnvoll, die Umrisse eines Objekts als Linie der Breite 1 Pixel aus dem Bild zu extrahieren (auch: *Kantenextraktion*).
- Man kann das Problem auch als Klassifikationsaufgabe betrachten: Jeder Pixel des Bildes ist **entweder Kantenpixel oder nicht**.

# Canny-Algorithmus

- Klassischer Algorithmus der Bildverarbeitung, veröffentlicht von *John Canny* (1986)
- Canny definiert zunächst drei mathematische Kriterien für optimale Kantenextraktion:
  1. **Erkennung**: alle tatsächlichen Kanten sollen gefunden werden, aber keine falschen für jeden Pixel einen Wert der sagt, ob Kante
  2. **Lokalisierung**: Abstand zwischen tatsächlicher und erkannter Kante soll minimal sein
  3. **Ansprechverhalten**: Kanten sollen nicht mehrfach erkannt werden, d.h. insbesondere soll ihre Breite nicht mehr als einen Pixel betragen.
- Auf Basis dieser Kriterien stellt er seinen Algorithmus vor, der im Sinne dieser Kriterien optimal sein soll.
- Verfahren wird auch **Canny-Operator** genannt.

# Canny-Algorithmus (Beispiel)

Foto: Alexander Schneider

# Canny-Algorithmus (Beispiel schrittweise)

# Canny-Algorithmus (Beispiel schrittweise)

Erläuterung zur vorhergehenden Folie:

- Bild 1: Ausgangsbild (Farbe)
- Bild 2: **Grauwertbild**
- Bild 3: Ergebnis einer **Gaußfilterung**, vgl. Folie 151.
- Bild 4: Ergebnis des **Sobel-Operators**, vgl. Folie 156.
  Dargestellt ist die richtungsunabhängige Information $\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$.

  Daneben kommt im folgenden Schritt auch noch die Richtung des Gradienten in jedem Bildpunkt zum Einsatz (nicht dargestellt).
- Bild 5: Ergebnis der **Nicht-Maximum-Unterdrückung**: Bereiche um lokale Maxima im Gradientenbetrag (d.h. um Kanten) werden reduziert auf scharfe, ein Pixel breite Kanten (nächste Folie).
- Bild 6: Ergebnis des **Hysterese-Schwellwertverfahrens** (übernächste Folie).

# Canny-Alg.: Nicht-Maximum-Unterdrückung

- Ermittlung der Richtung des Gradienten zu jedem Pixel: $\theta = \arctan\left(\frac{G_y}{G_x}\right)$

  Anschauen der zwei Nächsten Nachbarn

  G_x G_y wird durch Sobel geliefert

- Quantisierung der Gradientenrichtung in 4 Stufen: nord-süd, ost-west, nordost-südwest und nordwest-südost

$\tan\theta = \frac{G_x}{G_x}$

| | | | | |
|---|---|---|---|---|
| -45 | -72 | -90 | -90 | -90 |
| -18 | -45 | -81 | -90 | -90 |
| 0 | -9 | -45 | -90 | -90 |
| 0 | 0 | 0 | / | / |
| 0 | 0 | 0 | / | / |

- $\theta = 0$ Grad bedeutet z.B., dass die zugehörige Kante in **nord-süd**-Richtung verläuft

- Entsprechend wird in diesem Fall geprüft, ob der Betrag des Gradienten von einem oder beiden direkten Nachbarpixel in **ost-west**-Richtung größer ist als der Betrag des Gradienten des aktuellen Pixels.

- Ist dies der Fall, wird das aktuelle Pixel aus dem Kantenbild gelöscht (d.h. der Gradient wird auf 0 gesetzt).

- Auf diese Weise bleiben nur Kanten der Breite 1 Pixel erhalten (Beispielbild 5).

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 3 | 3 | 3 | 3 |
| 1 | 1 | 2 | 3 | 3 | 3 | 3 |
| 1 | 1 | 2 | 3 | 3 | 3 | 3 |
| 1 | 1 | 2 | 3 | 3 | 3 | 3 |

$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$ $S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$

$* S_x \longrightarrow$ $G_x$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 |
| 3 | 4 | 1 | 0 | 0 |
| 4 | 6 | 3 | 0 | 0 |
| 4 | 6 | 4 | 0 | 0 |
| 4 | 6 | 4 | 0 | 0 |

$G_y$

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 4 | 4 | 4 |
| -1 | -4 | -6 | -6 | -6 |
| 0 | -1 | -3 | -4 | -4 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

$* S_y$

×gelöscht
1 Winkel tan θ

$G$

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 4 | 4 | 4 |
| 3 | 6 | 6 | 6 | 6 |
| 4 | 6 | 4 | 4 | 4 |
| 4 | 6 | 4 | 0 | 0 |
| 4 | 6 | 4 | 0 | 0 |

$G = \sqrt{G_x^2 + G_y^2}$

# Canny-Alg.: Hysterese-Schwellwert-Verfahren

- Nun sollen schwache Kanten (d.h. Pixel mit kleinem Gradienten) unterdrückt werden, aber zusammenhängende Kanten möglichst nicht fragmentiert werden.
- Ein einfacher Schwellenwert führt zu Fragmentierung.
- Deshalb zwei Schwellenwerte $T_1$ und $T_2$ mit $T_1 \leq T_2$:
  - Pixel mit einem Gradienten unterhalb von $T_1$ werden auf schwarz gesetzt (d.h. ignoriert).
  - Pixel mit einem Gradienten oberhalb von $T_2$ werden als Kantenpixel markiert.
  - Pixel mit einem Gradienten zwischen den beiden Schwellwerten sind nur dann Kantenpixel, wenn mindestens einer der Nachbarn in Kantenrichtung ebenfalls Kantenpixel ist. Dieser Schritt wird rekursiv wiederholt, bis sich keine Änderung mehr ergibt.
- Die Wahl der Schwellwerte hat einen deutlichen Einfluss auf das Ergebnis (folgendes Beispiel aus der Originalveröffentlichung von Canny).

# Canny-Alg.: Hysterese-Schwellwert-Verfahren

690 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986

Fig. 7. (a) Parts image, 576 by 454 pixels. (b) Image thesholded at $T_1$. (c) Image thresholded at 2 $T_1$. (d) Image thresholded with hysteresis using both the thresholds in (a) and (b).

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - **Hough-Transformation**
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Beispiel 1

**Aufgabe**: Erkennung der Linien eines Fußballfeldes

**Anwendung**: Berechnung von 3D-Animationen von Spielszenen

# Beispiel 1 (Video)

# Beispiel 1 (Erinnerung)

**Aufgabe**: Erkennung der Linien eines Fußballfeldes

**Anwendung**: Berechnung von 3D-Animationen von Spielszenen

# Beispiel 1 (Ausschnitte)

Robustheit erforderlich gegen

- ➔ Verdeckung
- ➔ Unterbrechungen von Linien

# Beispiel 2

**Aufgabe**: Erkennung des Fahrbahnrandes

**Anwendung**: Spurassistenz-Systeme

# Beispiel 2 (Video)

# Beispiel 2 (Video)

# Beispiel 2 (Erinnerung)

**Aufgabe**: Erkennung des Fahrbahnrandes

**Anwendung**: Spurassistenz-Systeme

# Beispiel 2 (Ausschnitte)

Robustheit erforderlich gegen

- ➔ schwierige Beleuchtungsverhältnisse
- ➔ kontrastarme, verrauschte und nur annähernd gerade Linien

# Vorverarbeitung

- Kantendetektion (z.B. Sobel- oder Laplace-Filter)
- Hier: Laplace-Filter + anschließende Schwellwertoperation
- Ergebnis: Binärbild mit Kantenpunkten (schwarz dargestellt)

# Zielstellung

Ziel ist die **automatische** Bestimmung der **Parameter** aller Geradengleichungen, die den Verlauf von (längeren) im Bild verlaufenden **geraden Linien** beschreiben.

**Teilprobleme**:

- Welche parametrische Darstellung ist geeignet?
- Algorithmus zur Bestimmung der Parameter

# Die Gerade als lineare Funktion

$$y = mx + b$$

$m$: Steigung
$b$: $y$-Achsenabschnitt

# Die Gerade als lineare Funktion

$$y = mx + b$$

Senkrechte Geraden können durch die Parameter *(m,b)* nicht beschrieben werden.

# Hesse'sche Normalform der Geraden

$$x \cdot \cos\boldsymbol{\theta} + y \cdot \sin\boldsymbol{\theta} - r = 0$$

Teta -> Winkel der Normalen zur x-Achse
r -> Abstand Gerade Ursprung

Jede Gerade im Koordinaten-system lässt sich durch das Parameterpaar *(r,θ)* eindeutig beschreiben.

$\vec{n}$ : Einheits-Normalenvektor

# Hough-Transformation

nach Paul V. C. Hough, Patent 1962

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

Jeder Geraden im Bildraum entspricht ein Punkt im Hough-Raum.

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

$$\theta = -\frac{\pi}{4}$$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

$$r = x_i \cdot \cos\theta + y_i \cdot \sin\theta$$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

Jedem Punkt im Bildraum entspricht eine sinusförmig verlaufende Kurve im Hough-Raum.

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

$$r = 8.5 \cdot \cos\theta + 4.5 \cdot \sin\theta$$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

Jedem Kurven-Schnittpunkt im Hough-Raum entspricht eine Gerade durch mehrere Punkte im Bildraum.

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

Inverse Hough-Transformation:

$$m = -\frac{1}{\tan\theta} \quad b = \frac{r}{\sin\theta} \quad \text{für } \theta \neq 0$$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

Inverse Hough-Transformation:

$$m = -\frac{1}{\tan\theta} \quad b = \frac{r}{\sin\theta} \quad \text{für } \theta \neq 0$$

# Hough-Transformation

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

# Diskretisierung des Hough-Raumes

Bildraum $(x,y)$

Hough-Raum $(r,\theta)$

Hough-Akkumulatoren,

Wert: 0 1 2 3

Jede Zelle wird hochgezählt -> Zellen die mehrfach geplottet werden haben einen größeren Wert

# Rücktransformation in den Bildraum

Geraden im Bildraum zeigen sich als Maxima in den Hough-Akkumulatoren.

# Hough-Algorithmus (vereinfacht)

KantenPixel:= { (1;2), (2,5;2,5), (4;5), (8,5;4,5) }

Ergebnis der Kantendetektion, hier: Punkte aus Beispiel

HoughRaum[-π/2...π/2][-6...6]:=0

Initialisierung

**foreach** (x,y) **in** KantenPixel **do**

  **for** Θ:= -π/2 **to** π/2 **do**

    r:= x * **cos**(Θ) + y * **sin**(Θ)

    HoughRaum[Θ][r]++

Hough-Akkumulation

Erhöhung der Zelle um 1

  **end**

**end**

(r,Θ):=**argmax** (HoughRaum)

Bestimmung des Maximums

# Beispiel 1

# Beispiel 2

# Variante: Erkennung von Kreisen

- Hough-Raum $(x,y,r)$ mit Kreismittelpunkten und Radien
- Falls Radius $r$ bekannt, nur minimale Änderungen am Algorithmus
- Erweiterung möglich auf
  - Ellipsen
  - beliebig geformte Objekte
  - **Rechenaufwand** und **Speicherbedarf** kann sehr groß werden!

# Erkennung von Kreisen

# Generalsierte Hough-Transformation

# Generalisierte Hough-Transformation

# Generalisierte Hough-Transformation

# Effizienzsteigerung

- Nur **wenige, gute Kantenpunkte** in der Vorverarbeitung extrahieren
- **Richtungsinformation** nutzen
  - Sobel-Filter liefert zu jedem Kantenpunkt den Gradienten
  - Nur Hough-Akkumulatoren hochzählen, die etwa der Richtung des Gradienten entsprechen

# Hough-Transformation: Zusammenfassung

- Die **Hough-Transformation** ist ein robustes Verfahren zur **Objekterkennung in Bildern**.
- Anwendbar auf **Geraden**, **Kreise** und **beliebige Objekte**
- Hoher **Rechenaufwand** und **Speicherbedarf**
- Grundidee: **Transformation** aller **Kantenpunkte** in den Raum, der durch die gesuchten Lage- und Objektparameter aufgespannt wird (**Hough-Raum**)

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - **Histogrammbasierte Verfahren**
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Histogrammbasierte Objekterkennung

Prüfen ob 2 Bilder das gleiche Objekt zeigen

- einfaches und leistungsfähiges Verfahren nach Swain & Ballard (1991)
- Objekterkennung und -lokalisierung ausschließlich auf Basis des Farbhistogramms, die Form wird nicht berücksichtigt.
- 3-dimensionaler Farbraum wird in *n* Bereiche aufgeteilt, genannt *Bin* (zu engl. „Behälter“). Durch lineare Zerlegung der R-, G- und B-Dimension des RGB-Farbraumes in jeweils 6 Abschnitte ergeben sich z.B. $n = 6 \cdot 6 \cdot 6 = 216$ Bins.
- Aus allen Referenzbildern und aus dem Bild des unbekannten Objekts werden die Farbhistogramme berechnet.
- Vergleich zweier Histogramme $I$ (unbekanntes Bild) und $M$ (Modell) erfolgt durch Berechnung des **Histogrammschnitts** (*histogram intersection*):

$$s(I,M) = \frac{\sum_{j=0}^{n} \min(I_j, M_j)}{\sum_{j=1}^{n} M_j}$$

- Bei vollständiger Übereinstimmung ergibt sich der Wert 1, bei vollkommen verschiedenen Farbhistogrammen der Wert 0.

# Histogrammschnitt (*Histogram Intersection*)

Bild: Hwang A D et al., J Vis 2009;9:25

# Ergebnisse von Swain & Ballard, 1991

*Fig. 7.* The results of matching all combinations of image and database histograms displayed pictorially where the size of the squares are proportional to match values. The dominance of the diagonal values shows that the correct match is almost always selected. Twenty-nine of thirty-two matches are correct; in three cases the correct model received second-highest score. Models are along the horizontal axis; unknown objects along the vertical axis.

# Histogrammbasierte Objekterkennung

- **Hohe Robustheit** gegen
  - unterschiedliche Auflösungen
  - Rotation
  - moderate Unterschiede in der Entfernung
  - unterschiedliche Blickwinkel
  - teilweise Verdeckung
  - Änderungen des Hintergrunds
- die gewählte Auflösung des Histogramms hat nur einen geringen Einfluss auf die Erkennungsgenauigkeit
- Bei dreidimensionalen Objekten sollen ca. 6 Ansichten jedes Referenzobjekts für eine robuste Erkennung genügen.
- **Empfindlich** bei Änderungen der **Beleuchtung**; mögliche Beleuchtungs-änderungen sollten vorab separat normiert werden.
- Neuere Verfahren berücksichtigen z.B. noch die räumliche Nähe von Punkten oder verwenden lokale Filterantworten anstelle der Farbwerte.

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - **Viola-Jones-Algorithmus**
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Viola-Jones-Algorithmus

- schnelles und robustes Verfahren zur **Detektion von Gesichtern**, vorgestellt von Paul Viola und Michael Jones (2001)
- später auch erfolgreich zur Detektion **anderer Objekte** oder zur Erkennung von **Personen** (ganzer Körper) eingesetzt
- in OpenCV vorhanden und für verschiedene Problemstellungen vortrainiert, ebenso ein Trainingsskript für neue Objektklassen (*HaarTraining*)
- Ähnliche Verfahren kommen heute in vielen Kompaktkameras zum Einsatz, um Schärfe (*auto focus*, AF), Belichtung (*auto exposure*, AE) und Blitzsteuerung (*flash exposure*, FE) für den Bereich der detektierten Gesichter zu optimieren, teilweise auch das Kompressionsverfahren.

*Without Face Detection AF/AE/FE*

*With Face Detection AF/AE/FE*

Bild: Canon

# Viola-Jones-Algorithmus

- Grundidee des **Trainings**:
  - Wir benötigen eine große Stichprobe von Bildern einheitlicher Größe, die das gesuchte Objekt **formatfüllend** zeigen (Positivbeispiele) oder nicht zeigen (Negativbeispiele).
  - Berechne zu jedem Bild eine Vielzahl (i. d. Praxis Zehntausende) von Merkmalen, ähnlich den Haar-Wavelets (Kapitel 3). Man nennt sie ***Haar-like features*.**
  - Mittels des **AdaBoost**-Lernverfahrens (*adaptive boosting*) werden automatisch Teilmengen von Merkmalen identifiziert, die in Kombination am besten geeignet sind, Positivbeispiele von Negativbeispielen zu trennen (insgesamt i.d.R. noch wenige Tausend Merkmale). Diese Merkmale werden in Gruppen nach ihrer Trennschärfe sortiert (die besten Merkmalsgruppe zuerst).
- Grundidee der **Erkennung**:
  - Ein rechteckiges Fenster wird in unterschiedlichen Größen über das zu durchsuchende Bild geschoben, jeder Bildausschnitt wird einzeln betrachtet.
  - Eine **Kaskade von Klassifikatoren** weist Bildausschnitte zurück, die anhand der vordersten, besten Merkmalsgruppe mit hoher Sicherheit als Negativbeispiele identifiziert wurden. Andernfalls werden weitere Merkmale für die nächste Klassifikatorstufe berechnet. Die volle Zahl von Merkmalen muss deshalb nur dann berechnet werden, wenn mit hoher Wahrscheinlichkeit ein Treffer vorliegt.

# V.-J.-Algorithmus: *Haar-like features*

- Grau bedeutet im Bild oben „+“ (Addiere die Summe aller Bildpunkte unterhalb der Fläche), schwarz bedeutet „-“ (Subtrahiere die Summe aller Bildpunkte unterhalb der Fläche).

- (a) und (b) sind Zwei-Blockmerkmale,
  (c) und (d) Drei-Blockmerkmale
  (e) ist ein Vier-Blockmerkmal

- Die enorme Zahl an Merkmalen entsteht aus den 5 Grundtypen (a) bis (e) durch **Verschiebung** und **Skalierung**.

zum Vergleich: Haar-Wavelets:

Bild oben: Sergiu Dotenco, Bild rechts: B.Selig, Tilman Walther, FU Berlin

# V.-J.-Algorithmus: *Haar-like features*

- Beispielhafte Platzierung der Blockmerkmale in Suchfenstern.
- Blockmerkmale werden in unterschiedlichen Skalierungen $w'$, $h'$ an verschiedenen Stellen im Suchfenster der Größe $w$, $h$ positioniert.
- Die Summe der Pixel, die in den helleren Regionen liegen, wird von der Pixelsumme in den dunklen Rechtecken subtrahiert. Das Ergebnis des Subtraktion bildet der Merkmalswert.

aus: Sergiu Dotenco, *Ein Personenverfolgungsmodul für humanoide Roboter*, Masterarbeit, 2011

# V.-J.-Algorithmus: Einige relevante Merkmale

Bild: Adam Harvey

# V.-J.-Algorithmus: Effiziente Merkmalberechnung

- Das **Integralbild** $ii(x, y)$ entspricht der Summe der Pixel-Intensitäten $I$ links und oberhalb von (x,y):

$$ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y')$$

- $ii$(Punkt 1) entspricht der Summe der Pixel im Rechteck A
- $ii$(Punkt 2) entspricht der Summe der Pixel im Rechteck A+B
- $ii$(Punkt 3) entspricht der Summe der Pixel im Rechteck A+C
- $ii$(Punkt 4) entspricht der Summe der Pixel im Rechteck A+B+C+D
- Die Summe der Pixel im Rechteck D ergibt sich somit als
  $ii$(Punkt 4) + $ii$(Punkt 1) – $ii$(Punkt 2) – $ii$(Punkt 3), da D=(A+B+C+D)+A-(A+B)-(A+C)

Bild: Sergiu Dotenco

# V.-J.-Algorithmus: Effiziente Merkmalberechnung

- Das zur Berechnung der Merkmale verwendete Integralbild

$$ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y')$$

lässt sich mithilfe einer kumulativen Spaltensumme $s$ effizient in einem Durchlauf über das Bild berechnen:

Spaltensummen

nächster Wert in der Zeile von Spalte

$$s(x, y) = s(x, y-1) + I(x, y)$$

$$ii(x, y) = ii(x-1, y) + s(x, y)$$

Links nach rechts über Zeilen -> Integralbild

wobei

$$s(x, -1) = 0$$

und

$$ii(-1, y) = 0$$

I

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 3 | 3 | 1 | 2 |
| 2 | 1 | 1 | 2 | 1 | 1 | 3 |
| 3 | 4 | 1 | 2 | 2 | 1 | 4 |
| 1 | 3 | 2 | 1 | 1 | 1 | 3 |
| 3 | 1 | 1 | 2 | 2 | 3 | 4 |
| 1 | 1 | 1 | 2 | 3 | 4 | 4 |

S

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 3 | 3 | 1 | 2 |
| 3 | 3 | 2 | 5 | 4 | 2 | 5 |
| 6 | 7 | 3 | 7 | 6 | 3 | 9 |
| 7 | 10 | 5 | 8 | 7 | 4 | 12 |
| 10 | 11 | 6 | 10 | 9 | 7 | 16 |
| 11 | 12 | 7 | 12 | 12 | 11 | 20 |

ii

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 7 | 10 | 11 | 13 |
| 3 | 6 | 8 | 13 | 17 | 19 | 24 |
| 6 | 13 | 16 | 23 | 29 | 32 | 41 |
| 7 | 17 | 22 | 30 | 37 | 41 | 53 |
| 10 | 21 | 27 | 37 | 46 | 53 | 69 |
| 11 | 23 | 30 | 42 | 54 | 65 | 85 |

$53 + 6 - 21 - 19 = \underline{\underline{19}}$

# V.-J.-Algorithmus: Die Klassifikator-Kaskade

- Hintergrund-Bildausschnitte sind viel häufiger als Gesichter
- Ziel ist es daher, Hintergrund mit möglichst wenigen Merkmalen und so früh wie möglich als solchen zu erkennen.
- Das Training der späteren Klassifikatorstufen erfolgt jeweils nur mit den Bildern aus der Trainingsstichprobe, die die vorherigen Stufen durchlaufen haben.
- Viola & Jones verwendeten eine Kaskade aus 32 Klassifikatoren. Der erste Klassifikator in der Kaskade wertet dabei nur zwei Merkmale aus, der zweite fünf und die letzten 20 Klassifikatoren je 200 Merkmale.

Bild: Bucknall, J.M., *How face detection works*, PC Plus Nr. 296, Juli 2010

# Viola-Jones-Algorithmus

Animation: Adam Harvey, http://vimeo.com/12774628

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - **Convolutional Neural Networks**
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Convolutional Neural Networks (ConvNets, CNNs)

Shaoqing Ren, *et al*, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2015

# Convolutional Neural Networks (ConvNets, CNNs)

- Seit etwa 2012 **das dominierende Verfahren** zur Objekterkennung/Klassifikation von Bildern
- Für bestimmte Aufgabenstellungen **dem Menschen** (bzw. menschlichen Experten) bereits heute **gleichwertig oder überlegen**, z.B. bei der
  - Erkennung von Personen anhand ihrer Gesichter (Taigman et. al., 2014)

  - Erkennung von Hautkrebs in Fotos von Hautveränderungen (Esteva et. al., Februar 2017)

  - Erkennung von Brustkrebs in Gigapixel-Mikroskopie-Aufnahmen von histopathologischen Präparaten (Liu et. al., März 2017)

# Convolutional Neural Networks (ConvNets, CNNs)

T. Ranzenberger, D. Stadelmann, T. Strotmann, C. von Kleinsorgen, IT-Projekt „Smoothie Assistant“, Fakultät IN, TH Nürnberg, 2016

# Convolutional Neural Networks (ConvNets, CNNs)

Trainingsdaten für die Klasse „Banane“ (Auswahl):

# Convolutional Neural Networks (ConvNets, CNNs)

- Das ConvNet erlernt neben den **Klassengrenzen** auch eine geeignete **Merkmalberechnung** anhand der Trainingsstichprobe.
- Es kann daher mit rohen Bilddaten trainiert werden.

# Convolutional Neural Networks (ConvNets, CNNs)

- Die obersten Schichten eines ConvNet führen Faltungsoperationen (engl. *Convolution*) durch, analog zu klassischen Sobel-, Laplace-, Gauß-Filtern etc.
- Daneben gibt es drei weitere wichtige Operationen in einem ConvNet, jeweils meist mehrfach:
  - Nichtlinearität (Schwellwertfunktion, oft ReLU)
  - *Pooling*- oder *Sub-Sampling*-Schicht -> große Feature map zu kleinere (zB: Zusammenfassen von 4 Pixel zu einem)
  - *Fully Connected Layer* (zur Klassifikation, MLP, vgl. Kapitel 4)
- Typisches Architekturbild eines einfachen ConvNet:

Bild: https://www.clarifai.com/technology

# *Convolution Layer* (Faltungsschicht)

- Beim Entwurf des ConvNets wird die Zahl der verschiedenen Filterkerne und ihre Größe festgelegt, nicht jedoch die Zahlenwerte in den Matrizen der Filterkerne.
- Die Größe der resultierenden „*Feature Map*“ hängt darüberhinaus von der verwendeten Schrittweite ab („*stride*“) und von der Behandlung des Randes („*zero padding*“ ja oder nein)

Bild: http://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/

# ReLu-Operation

- Anstelle der Schwellwertfunktion (oder der Sigmoid-Funktion o.ä.) wie beim klassischen Perzeptron wird in ConvNets meist die ReLU-Funktion als Nichtlinearität eingesetzt.

**Output = Max(zero, Input)**

negative Werte = 0
positive = dargestellt

Bild: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

# *Pooling*-Schritt (*Subsampling*)

- Meist Max-Pooling, aber z.B. auch Summe oder Mittelwert möglich, reduziert die Größe der *Feature Map*:

Bild: http://cs231n.github.io/convolutional-networks/

# *Pooling*-Schritt (Subsampling)

- Reduziert Zahl der Parameter, vermeidet dadurch *Overfitting* (Überadaption)
  gezwungen die wesentlichen Parameter zu lernen -> keine Überadaption
- Erzeugt Invarianz gegenüber leichten Translationen, Verzerrungen und Skalierungen

Bild: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

# Fully Connected Layers (Linear Layers)

- Die untersten Schichten des ConvNets sind *fully connected*, d.h. jedes Neuron einer Schicht ist mit jedem Neuron der darüber liegenden Schicht verknüpft.
- Häufig *Softmax*-Aktivierungsfunktion, ähnlich Sigmoid-Funktion
- Dieser Teil des ConvNets entspricht einem klassischen *Multi Layer Perceptron*.
- Für jede Klasse existiert ein Ausgabeknoten, der so etwas wie eine *a-posteriori*-Wahrscheinlichkeit liefert.

# Training des ConvNets

- Training des gesamten Netzes mit Backpropagation-Algorithmus
- Zielvektor im Beispiel (0, 0, 1, 0) für die Klasse „Boat“
- Nach zufälliger Initialisierung aller Gewichte des ConvNets ist der Output für das Bild z.B. (0.2, 0.4, 0.3, 0.1)
- Anpassung der Gewichte so, dass der Fehler zwischen dem Zielvektor und dem vom Netz gelieferten Vektor kleiner wird
- neues Ergebnis z.B. (0.1, 0.1, 0.7, 0.1)
- Wiederhole dies immer wieder für alle Bilder in der Trainingsstichprobe

https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

# Hardware und Frameworks

- Leistungsfähige (CUDA-fähige) Grafikkarten beschleunigen das Training von neuronalen Netzen enorm.
- Es existiert eine Reihe von Frameworks, mit denen sich ConvNets und andere neuronale Netze einfach konfigurieren und trainieren lassen, u.a.
  - TensorFlow
  - Torch
  - Theano
  - Caffe
  - Keras
  - CNTK

```
conv_1 = Convolution2D(96, 11, 11,subsample=(4,4),activation='relu',
                       name='conv_1')(inputs)

conv_2 = MaxPooling2D((3, 3), strides=(2,2))(conv_1)
conv_2 = crosschannelnormalization(name="convpool_1")(conv_2)
conv_2 = ZeroPadding2D((2,2))(conv_2)
conv_2 = merge([
    Convolution2D(128,5,5,activation="relu",name='conv_2_'+str(i+1))(
        splittensor(ratio_split=2,id_split=i)(conv_2)
    ) for i in range(2)], mode='concat',concat_axis=1,name="conv_2")

conv_3 = MaxPooling2D((3, 3), strides=(2, 2))(conv_2)
conv_3 = crosschannelnormalization()(conv_3)
conv_3 = ZeroPadding2D((1,1))(conv_3)
conv_3 = Convolution2D(384,3,3,activation='relu',name='conv_3')(conv_3)

conv_4 = ZeroPadding2D((1,1))(conv_3)
```

Beispiel: Definition einer ConvNet-Architektur mit Keras

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - **Gesichtserkennung mit CNNs**
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Gesichtserkennung (Erkennung der Identität)

- Anwendungsgebiete z.B. Zugangskontrolle, Fahndung, Videoüberwachung, Foto-Verwaltungsprogramme
- Problemstellung etwas anders als bei üblichen Objekterkennungsproblemen (Ziffern, Autos, Katzen etc.)
- Grund: Im Einsatz werden Gesichter von Menschen erkannt und verglichen, die **nicht in den Trainingsdaten** enthalten waren (neue Klassen!).

- Es wir kein Klassifikator für *n* vorab feststehende Klassen benötigt, sondern eine Merkmalberechnung, die für Bilder verschiedener Personen möglichst unterschiedliche und für verschiedene Bilder der gleichen Person möglichst ähnliche **Merkmalsvektoren** erzeugt.
- Die Klassifikation kann dann z.B. über den Euklidischen Abstand erfolgen (Nächster-Nachbar-Klassifikator). Hierfür genügt schon **ein einziges Referenzbild** (z.B. Fahndungsfoto).

Eigenes Biild vom Bahnhof Südkreuz in Berlin, Februar 2018

# Warum ist Gesichtserkennung schwierig? (I)

**Fig. 1.** Current passport photos (Left), staff card photos (Middle), and personal photos (Right) for authors RJ (top) and AMB (bottom). Consider image similarity by rows and by columns.

Jenkins, R.,et al. Variability in photos of the same face. Cognition (2011)

# Warum ist Gesichtserkennung schwierig (II)

Jenkins, R.,et al. Variability in photos of the same face. Cognition (2011)

# ConvNets für die Gesichtserkennung

- Das ConvNet erlernt eine geeignete **Merkmalsberechnung** anhand der Trainingsstichprobe, bei FaceNet **260 Mio. Bilder** von > **8 Mio. Personen**
- Der Merkmalvektor zu dem Bild eines Gesichts wird oft als **Face Embedding** bezeichnet.
- Zu erkennende Klassen (Personen) werden erst in der Arbeitsphase vorgegeben.

# Architektur von DeepFace (Facebook, 2014)

- Training des Netzes erfolgt auf 4,4 Mio. Bildern von 4030 Personen

  => 4030 Knoten in der Ausgabeschicht (F8)
- Optimierungsziel: diese 4030 Personen möglichst gut zu erkennen
- Fehlerfunktion (Loss), die für alle Trainingsbilder mittels Backpropagation minimiert wird: $L = -\log p_k$ (entspricht ML-Training)
- Für die Anwendung wird **die letzte Schicht des Netzes entfernt**
- Statt dessen werden die Aktivierungen der vorletzten Schicht (F7) **normiert** und als 4096-dimensionaler **Merkmalvektor** (Face Embedding) verwendet.
- Zur Klassifikation z.B. **Skalarprodukt** zweier Merkmalvektoren + Schwellwert

# Architektur von FaceNet bzw. OpenFace

FaceNet von Google (2015), nachgebildet als Open-Source-Projekt OpenFace:

Bild: https://cmusatyalab.github.io/openface/

# FaceNet (2015)

- Bis zu 260 Mio. Trainingsbilder von **8 Mio. Personen**!
- Andere Trainingsstrategie: gewünschte Eigenschaft der Merkmalvektoren ist **direkt** das **Optimierungsziel**

  Eigenschaft: gleiche Person möglichst ähnliche Merkmalsvektoren -> verschiedene Personen möglichst unterschiedliche Merkmalsvektoren
- Hierzu wird der „Triplet Loss“ minimiert. Es werden je 3 Bilder betrachtet, davon 2 von der gleichen Person (*Anchor* und *Positive)* sowie eins von einer anderen Person (*Negative*)
- Ist der Abstand zwischen *Anchor* und *Positive* größer als der zwischen *Anchor* und *Negative*, dann erfolgt mittels Backpropagation eine Anpassung der Gewichte:

- Klassifikation über Euklidischen Abstand und Schwellwert; Fehlerrate um den Faktor 7 geringer als die von DeepFace (und damit als die von Menschen)

Bilder: Schroff et. al, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

# FaceNet (2015)

- Face Embedding mit 128 Dimensionen optimal
- Ein Byte pro Dimension genügt.
- Für alle Menschen auf der Erde genügt 1 TB Speicherplatz.
- Ähnlicher Ansatz mit noch besseren Ergebnissen kurz danach von Baidu (2015)

http://bamos.github.io/2016/01/19/openface-0.2.0/

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - **Objektverfolgung**
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Objektverfolgung: Anwendungen (Auswahl)

immer dann wenn Videoströme automatisch verarbeitet werden

- **Robotik**
  - Augenkontakt mit Gegenüber, Verfolgung von Personen, Erkennung von Gesten, Imitation von Körperbewegungen
  - Robocup (kontinuierliche Erkennung von Ball und Gegenspielern)
- **Überwachungstechnik**
- **Augmented Reality** (Erweiterte Realität)
  - Ergänzung von Videos mit Zusatzinformationen oder virtuellen Objekten mittels Einblendung/Überlagerung, z.B.
    - Einblendung von Linien- und Entfernungen bei Fußball-Freistößen
    - Anzeige von 3D-Fußballspielern auf EM-Sammelkarten
- **Mensch-Maschine-Interaktion**
  - Augen- bzw. Gesichtsverfolgung (z.B. *Head-coupled perspective, 3D-Fernseher mit head-tracking*)
  - Gestenerkennung
- **Fahrerunterstützende Systeme**
- **autonomes Fahren/Fliegen**
- **Editieren und Manipulieren von Videosequenzen**

# Objektverfolgung (Beispiele)

Quelle: http://www.youtube.com/watch?v=RG5uV_h50b0

# Objektverfolgung (Beispiel)

https://www.youtube.com/watch?v=9nlSeVRqJDQ

# Objektverfolgung

- **Objektverfolgung** *(visual object tracking,* meist einfach **Tracking***)* ist die kontinuierliche Bestimmung der Position eines Objekts in einer Bildfolge, bei sich **das Objekt und/oder die Kamera in Bewegung** befindet.
- Schwierigkeitsgrad hängt u.a. ab von
  - Freiheitsgraden der Bewegung von Objekt und Kamera
  - Art des Objekts (u.a. starr/beweglich)
  - Art des Hintergrunds
  - Geschwindigkeit des Objekts relativ zur *Frame Rate*
- Beispiele:
  - **Einfach**: Verfolgung eines roten Balls vor einer schwarzen Wand, der sich unbeschleunigt durch das Bild bewegt, wobei die Aufnahme mit einer auf einem Stativ befestigten Kamera erfolgt.
  - **Schwierig**: Verfolgung eines bestimmten Zebras in der verwackelten Luftaufnahme einer vor dem Kamerahubschrauber flüchtenden Zebraherde.

# Objektverfolgung

- Merkmale für die zu verfolgende Region: häufig **Farbhistogramme**, aber auch Texturmerkmale etc.
- Meist Kernel-Tracking mit rechteckiger Kernelfunktion (d.h. ein rechteckiger Bildausschnitt wird verfolgt)
- Klassisches Verfahren zum Kernel-Tracking: **Mean-Shift-Algorithmus**

- Ausgehend von der Position des Zielobjekts (hier: Hautton-Farbhistogramm) im vorherigen Frame wird die Bildregion so lange verschoben, bis ein Abstandsmaß zum Zielobjekt minimal ist (z.B. basierend auf Histogrammschnitt)
- Dabei wird die Ableitung der Abstandsfunktion zur Bestimmung der Richtung verwendet
- Erweiterungen erlauben, auch die Größe des Bildausschnitts anzupassen.

Bild: http://docs.opencv.org/3.2.0/db/df8/tutorial_py_meanshift.html

# Objektverfolgung: TLD-Tracking

- Verfolgung über einen längeren Zeitraum erfordert Möglichkeit, verloren gegangenes Zielobjekt wiederzufinden („Detection“) und das Tracking **neu zu initialisieren**
- Das „Detection“-Modell wird laufend aktualisiert
- vgl. Beispielvideo (Quadrocopter)

während des Tracking lernt das Model, wie das Objekt aussieht angepasst an die Umgebungen

Z Kalal, K Mikolajczyk, J Matas: Tracking-learning-detection, IEEE trans. on pattern analysis and machine intelligence 34 (7), 1409-1422, 2010

# Objektverfolgung

Nützlich ist oft eine **Prognose der zukünftigen Objektposition** auf der Grundlage **vergangener Beobachtungen**.

- **Kalman-Filter** Eine Hypothese wo sich das Objekt befindet
  - Klassischer Ansatz zur Lösung des Tracking-Problems (Bild nächste Folie)
  - Modellbasiertes Verfahren, das Zustand eines **linearen Systems** anhand bisheriger, möglicherweise fehlerbehafteter Beobachtungen bzw. Messwerte vorherzusagen versucht.
  - Systemzustand kann Informationen über Position, Größe, Beschleunigung und Geschwindigkeit des Zielobjekts enthalten.
  - Ausgehend von einem **Bewegungs**- und einem **Beobachtungsmodell** läuft der Filterungsprozess iterativ, in zwei Schritten ab:
    - **Prädiktion** des Systemzustands anhand vorangehender Beobachtungen
    - **Korrektur** anhand der fehlerbehafteten Messung
  - **Voraussetzung**: Störung (Rauschen) muss annähernd normalverteilt sein.
- **Partikelfilter** Parallele Verfolgung mehrerer Hypothesen über das Objekt
  - gehört zu den aktuell leistungsfähigsten und robustesten Tracking-Verfahren
  - probabilistisches Modell, erlaubt auch die Modellierung nichtlinearer und nicht normalverteilter Zustände
  - Als **Partikel** werden dabei die (gewichteten) Einzelhypothesen über den Systemzustand bezeichnet.

# Kalman-Filter: Beispiel

schwarz: Flugbahn des Objekts
rot: Beobachtungen (Messungen)
grün: Ausgabe des Kalman-Filters

Bild: Tobias Pietzsch, TU Dresden

# 7. Experimentelle Evaluation

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße

# Stichproben

- **Lernstichprobe**
  - Grundlage für automatische Parameterschätzung
- **Validierungsstichprobe**
  - Grundlage für das Optimieren von einstellbaren Parametern (z.B. Schwellwerte u. Gewichte)
- **Teststichprobe**
  - darf erst dann betrachtet werden, wenn die Optimierung abgeschlossen ist
  - dient der Ermittlung von Erkennungsraten, die realistisch sind für ungesehenen Daten
- Dieses Prinzip wird in der Praxis häufig verletzt!

# Inhalt der Vorlesung „Medienverarbeitung“

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - **Gütemaße**

# Gesichtserkennung

| | |
|---|---|
| W/O 5 | 0. |
| W/O 6 | 0.2896 |
| W/O 7 | 0.291 |
| | W/O |

Fig. 13. An
(row 1) and
person. Bot

g. 12. Top: ROC Curve of LBP+AdaBoost method for face verification
he training set. Bottom: ROC curves for various compared methods,
curve for testing the eyeglass effect and an ROC for te

# Erkennung von Epitopen (Bereich Biomedizin)

Quelle: Wikipedia

# Aus dem „Deutschen Ärzteblatt“

Prozent beschrieben (15).

## Verbesserung der Spezifität bei PSA < 4 ng/mL

Circa 13 bis 20 Prozent aller Männer mit PSA-Werten in Normbereich (2,5 bis < 4 ng/mL) weisen ein klinisch erkennbares Prostatakarzinom auf (19). Vashi et al. untersuchten Männer mit PSA-Werten zwischen 3 bis 4 ng/mL. Bei 19 Prozent-f-PSA wurden als Cut-off-Wert 90 Prozent aller Prostatakarzinome erkannt (23). Ähnliche Daten wurden von Catalona et al. anhand einer Screeninguntersuchung von mehr als 900 Männern mit PSA-Werten zwischen 2,6 und 4 ng/mL und normalem rektalen Tastbefund berichtet. Ein klinisches Problem stellt die geri

gena xierten PSA (18). In viel ungen wurde die Sensitivität und Spezifität von t-PSA, f-PSA, f/t-PSA und c-PSA sowie weiteren Ratios verglichen. Gemessen wurde hierbei die AUC („area under curve“) der ROC -Kurven („receiver operator curve“) (2, 7, 9, 14). Sokoll et al. verglichen c-PSA mit Gesamt-PSA und demonstrierten, dass die Spezifität der Karzinomerkennung bei Männern mit PSA-Werten zwischen 4 und 10 ng/mL bei c-PSA verbessert war (21). Bei einer Sensitivität von 95 Prozent war die Spezifität mit 25 Prozent vergleichbar der Spezifität der Messung von Prozent-f-PSA. Vergleichbare Ergebnisse wurden von Brawer et al. und anderen publiziert (2).

Die Autoren schlossen aus diesen Daten, dass die Bestimmung von

Sensitivi
die ents
signifika
(P <0,000
tistisch
PSA (60,
*le)*. Ob
men zu ei
Prostatal
lich dur
en bean

Manuskript
angenommen

Die Autore
ne der Ric
cal Journal

Zit

# Schlüsselwort-Erkennung in handschriftl. Dokumenten

**Fig. 7.** Precision-recall curves of the four keyword retrieval tests on 342 PCR forms.

the four keyword retrieval tests on 342 PCR

$S^{WR}$ $S^{WSR}$

## 5. Conclusion

In this paper we pres a no

the handwritt

# Fehlermaße in 2-Klassen-Szenarien

- Wir betrachten Zweiklassenszenarien der Art
  - Diagnose einer Krankheit (erkrankt/nicht erkrankt)
  - Sprecheridentifikation (angegebene Identität korrekt/nicht korrekt)
  - Rauchmelder (es brennt/es brennt nicht)
  - Qualitätskontrolle in der Fertigung (Werkstück fehlerhaft/nicht fehlerhaft)
- Es existiert hierzu ein wahrer Zoo von Fehlermaßen
- Uneinheitliche Benennung der einzelnen Maße

# Verwechslungsmatrix (Vierfeldertafel)

| | | tatsächliche Klasse | | |
|---|---|---|---|---|
| | | **p** | **n** | Summe |
| hypothetisierte Klasse | **p`** | $TP$ | $FP$ | $P`$ |
| | **n`** | $FN$ | $TN$ | $N`$ |
| | Summe | $P$ | $N$ | |

Bsp: Hautkrebserkennung

| | p | n | |
|---|---|---|---|
| p' | 150 | 100 | 250 |
| n' | 50 | 700 | 750 |
| | 200 | 800 | 1050 |

$TP$: *True positives*, korrekt als „positiv“ erkannt, richtig positiv

$FP$: *False positives*, fälschlicherweise als „positiv“ erkannt, falsch positiv, Fehler II. Art

$FN$: *False negatives*, fälschlicherweise als „negativ“ erkannt, falsch negativ, Fehler I. Art

$TN$: *True negatives*, korrekt als „negativ“ erkannt , richtig negativ

# Recall und Precision

Bsp: Hautkrebs

recall $= \frac{150}{200} = 75\%$

precision $= \frac{150}{250} = 60\%$

tatsächliche Klasse

hypothetisierte Klasse

| | p | n | Summe |
|---|---|---|---|
| p` | $TP$ | $FP$ | $P`$ |
| n` | $FN$ | $TN$ | $N`$ |
| Summe | $P$ | $N$ | |

Recall = Sensitivität = Trefferquote
= *true positive rate* = *hit rate*
= *detection power* korrekt als positiv erkannt
= *verification rate*

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

Precision = Genauigkeit = Positiver Vorhersagewert = *positive predictive value* = PPV = Relevanz = Wirksamkeit

$$precision = \frac{TP}{TP + FP} = \frac{TP}{P'}$$

# Sensitivität und Spezifität

tatsächliche Klasse

| hypothetisierte Klasse | **p** | **n** | Summe |
|---|---|---|---|
| **p`** | $TP$ | $FP$ | $P`$ |
| **n`** | $FN$ | $TN$ | $N`$ |
| Summe | $P$ | $N$ | |

Sensitivität = Recall = Trefferquote
= *true positive rate* = *hit rate*
= *detection power*
= *verification rate*

$$sensitivity = \frac{TP}{TP + FN} = \frac{TP}{P}$$

Spezifität = *true negative rate*

$$specificity = \frac{TN}{FP + TN} = \frac{TN}{N}$$

# Negativer und positiver Vorhersagewert

tatsächliche Klasse

| hypothetisierte Klasse | p | n | Summe |
|---|---|---|---|
| p` | $TP$ | $FP$ | $P`$ |
| n` | $FN$ | $TN$ | $N`$ |
| Summe | $P$ | $N$ | |

Negativer Vorhersagewert = Segreganz = Trennfähigkeit = *negative predictive value* = NPV

$$npv = \frac{TN}{TN + FN} = \frac{TN}{N'}$$

Positiver Vorhersagwert = Precision = Genauigkeit = *positive predictive value* = PPV = Relevanz = Wirksamkeit

$$ppv = \frac{TP}{TP + FP} = \frac{TP}{P'}$$

# Falsch-Positiv- und Falsch-Negativ-Rate

tatsächliche Klasse

| | | p | n | Summe |
|---|---|---|---|---|
| hypothetisierte Klasse | p` | $TP$ | $FP$ | $P`$ |
| | n` | $FN$ | $TN$ | $N`$ |
| | Summe | $P$ | $N$ | |

Falsch-Positiv-Rate = *false alarm rate*
= *false accept rate*
= *false positive rate*

$$fp_rate = \frac{FP}{FP + TN} = \frac{FP}{N}$$

Falsch-Negativ-Rate = *false reject rate* = FRR

$$fn_rate = \frac{FN}{TP + FN} = \frac{FN}{P}$$

# Accuracy und F-Wert

tatsächliche Klasse

| | | p | n | Summe |
|---|---|---|---|---|
| hypothetisierte Klasse | p` | $TP$ | $FP$ | $P`$ |
| | n` | $FN$ | $TN$ | $N`$ |
| | Summe | $P$ | $N$ | |

*Accuracy* = Erkennungsrate

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{P + N}$$

F-Wert = *F1 score* = *F-score*
= *F-measure*

$$F - measure = \frac{2}{1/precision + 1/recall} = \frac{2 \cdot TP}{P + P'}$$

# Beispiel Qualitätskontrolle in der Fertigung

Werkstück defekt

| | | p | n | Summe |
|---|---|---|---|---|
| Werkstück als defekt erkannt | p` | 24 | 18 | 42 |
| | n` | 5 | 2493 | 2498 |
| Summe | | 29 | 2511 | 2540 |

$$recall = sensitivity = \frac{24}{29} = 82,8\%$$

$$precision = \frac{24}{42} = 57,1\% \qquad specificity = \frac{2493}{2511} = 99,3\%$$

# Beispiel Qualitätskontrolle in der Fertigung

Werkstück defekt

| | | p | n | Summe |
|---|---|---|---|---|
| Werkstück als defekt erkannt | p` | 24 | 18 | 42 |
| | n` | 5 | 2493 | 2498 |
| | Summe | 29 | 2511 | |

Spezifität:
Vorsicht, Marketing!

$$recall = sensitivity = \frac{24}{29} = 82,8\%$$

$$precision = \frac{24}{42} = 57,1\%$$

$$specificity = \frac{2493}{2511} = 99,3\%$$

# Beispiel Screening-Untersuchung

An K erkrankt

| | | p | n | Summe |
|---|---|---|---|---|
| Schnelltest positiv | p` | 9 | 100 | 109 |
| | n` | 1 | 9890 | 9891 |
| | Summe | 10 | 9990 | 10.000 |

$$recall = sensitivity = \frac{9}{10} = 90\%$$

$$precision = \frac{9}{109} = 8,2\% \qquad specificity = \frac{9890}{9990} = 99\%$$

# Beispiel Screening-Untersuchung

An K erkrankt

| | | p | n | Summe |
|---|---|---|---|---|
| Schnelltest positiv | p` | 9 | 100 | 109 |
| | | 1 | 9890 | 9891 |
| | | | 9990 | |

Die Wahrscheinlichkeit, dass Herr Müller nach positivem Test (!) tatsächlich erkrankt ist.

Spezifität: Vorsicht, Marketing!

$$recall = sensitivity = \frac{9}{10} = 90\%$$

$$precision = \frac{9}{109} = 8,2\%$$

$$specificity = \frac{9890}{9990} = 99\%$$

# Einstellung des Arbeitspunkts

Arbeitspunkt -> Empfindlichkeit des Systems

- Die einzelnen Maße sind nicht voneinander unabhängig
  - höherer Recall => niedrigere Precision + höhere Falsch-Positiv-Rate
  - höhere Sensitivität => niedrigere Spezifität
- Ein einzelner dieser Werte ist für sich genommen **ohne Aussagekraft!** Bsp:
  - „Klassifikator“ A entscheidet immer auf „positiv“ => $recall$ = 100%
  - „Klassifikator“ B entscheidet immer auf „negativ“ => $fp_rate$ = 0%
- *Accuracy* führt bei 2-Klassenproblemen meist in die Irre:
  - Der Beispiel-Schnelltest auf die Krankheit K hat eine *Accuracy* von 99%.
  - Ein „Schnelltest“, der immer auf „negativ“ entscheidet, hat hier sogar eine *Accuracy* von 99,9%!
- F-Wert besitzt eine höhere Aussagekraft, ist aber beim Vergleich von Klassifikatoren mit unterschiedlichen Arbeitspunkten dennoch mit Vorsicht zu genießen.

# Arbeitspunkt und Kosten

- Kosten einer Fehlentscheidung in die eine oder andere Richtung hängen stark von der Anwendung ab:
  - „Kosten“ eines Feuer-Fehlalarms vs. „Kosten“ eines Großbrandes
  - „Kosten“ eines Fahrzeug-Defekts in der Garantiezeit vs. Kosten eines aussortierten (nicht defekten) Werkstücks
- Ziel in praktischen Anwendungen ist i.d.R. nicht die Minimierung der Fehlerrate, sondern die Minimierung der „Kosten“.
- Zur Bestimmung eines geeigneten Arbeitspunktes ist i.d.R. eine Stellgröße / ein Schwellwert vorgesehen

# ROC-Kurve (*Receiver operator characteristic*)

Quelle: Wikipedia

# ROC-Kurve (*Receiver operator characteristic*)

| A | | | B | | | C | | | C' | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TP=63 | FP=28 | 91 | TP=77 | FP=77 | 154 | TP=24 | FP=88 | 112 | TP=76 | FP=12 | 88 |
| FN=37 | TN=72 | 109 | FN=23 | TN=23 | 46 | FN=76 | TN=12 | 88 | FN=24 | TN=88 | 112 |
| 100 | 100 | 200 | 100 | 100 | 200 | 100 | 100 | 200 | 100 | 100 | 200 |
| TPR = 0.63 | | | TPR = 0.77 | | | TPR = 0.24 | | | TPR = 0.76 | | |
| FPR = 0.28 | | | FPR = 0.77 | | | FPR = 0.88 | | | FPR = 0.12 | | |
| ACC = 0.68 | | | ACC = 0.50 | | | ACC = 0.18 | | | ACC = 0.82 | | |

Quelle: Wikipedia

# AUC: Area under ROC Curve

Fläche unter der Kurve: Wahrscheinlichkeit, dass die Klassifikation richtig ist

Ausgangsbild: Wikipedia

# EER: Equal Error Rate

Quelle: Du, Chang 2007

# Fehlermaße für mehr als 2 Klassen

- Erkennungsrate = recognition rate = *(recognition) accuracy, word correct rate*

$$rr = \frac{\#\,\text{korrekt erkannte Muster}}{\#\,\text{korrekt erkannte Muster} + \#\text{falsch erkannte Muster}}$$

- Bei Symbolfolgen unbekannter, variabler Länge (Wortfolgen, Folgen von Lautsymbolen, Folgen von einzelnen Buchstaben etc.) ist es sinnvoll, Einfügungs- und Löschungsfehler zu berücksichtigen.

- In einem ersten Schritt wird mittels **dynamischer Programmierung** eine optimale Zuordnung zwischen Erkennungsergebnis und Referenz berechnet. Dann bestimmt man:

  Wortakkuratheit = *word accuracy,* analog *phoneme accuracy, letter accuracy*

  Wortfehlerrate = *word error rate* = WER

$$WAcc = 1 - WER$$

$$WER = \frac{\#\text{Substitutions} + \#\text{Deletions} + \#\text{Insertions}}{\#\,\text{Words in the reference}}$$

# => Anzahl

# Fehler bei der Objekterkennung/-verfolgung

Schwellwert $\gamma$ legt minimalen **Jaccard-Index** fest (i.d.R. $\gamma$=0.5), siehe nächste Folie

(a) True Positive

(b) False Negative

(c) False Positive

(d) False Negative and False Positive

(e) True Negative

GT: Ground Truth (Referenz)
ALG: vom Algorithmus generierte Hypothese

Nebehay, Georg: *Robust Object Tracking Based on Tracking-Learning-Detection*, 2012. http://www.gnebehay.com/publications/master_thesis/master_ thesis.pdf

# Jaccard-Index (Intersection over Unit, IoU)

- Kennzahl, mit der sich ganz allgemein die Ähnlichkeit zweier Mengen bestimmen lässt
- Berechnet sich als die Größe der Schnittmenge geteilt durch die Größe der Vereinigungsmenge:

$$\mathrm{IoU}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

- In der Objekterkennung erfolgt die Berechnung über die Größe der Flächen (Anzahl der Pixel):

Bilder: Adrian Rosebrock, http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection, CC-BY-SA 4.0

# Überblick und Zusammenfassung

1. **Einführung**
   - Grundbegriffe
   - **Anwendungen**
   - Überblick
2. **Vorverarbeitung**
   - Abtastung und PCM
   - Digitale Audiodaten
   - Digitale Bilddaten
   - Vektorquantisierung
   - Schwellwertoperationen & Histogramme
   - Lineare Filter
   - Nichtlineare Operationen
   - Normierungsmaßnahmen
3. **Merkmale**
   - Orthogonale Reihenentwicklung
   - Wavelet-Transformation
   - Heuristische Verfahren
   - Merkmale für die Spracherkennung
   - Merkmale für die Objekterkennung
4. **Numerische Klassifikation**
   - Nichtparametrische Klassifikatoren
   - Verteilungsfreie Klassifikatoren
   - Statistische Klassifikatoren
   - Neuronale Netze
   - Unüberwachtes Lernen
5. **Spracherkennung**
   - Dynamic Time Warping
   - Hidden-Markov-Modelle
6. **Objekterkennung**
   - Kantendetektion
   - Hough-Transformation
   - Histogrammbasierte Verfahren
   - Viola-Jones-Algorithmus
   - Convolutional Neural Networks
   - Gesichtserkennung mit CNNs
   - Objektverfolgung
7. **Experimentelle Evaluation**
   - Stichproben
   - Gütemaße