

# Early, intermediate and late fusion heterogener Sensordaten mittels Deep Learning



*Anhand von Beispielen aus der Robotik*

**Dr. Ing. Cristian Axenie**

# Vorgeschlagene Wahlpflichtfächer/Lehrbeitrag (benötigte Vorkenntnisse)

## Maschinelles Lernen

SWS	4
ECTS	5
Sprache(n)	Deutsch (Standard) Englisch
Lehrform	SU mit Praktikum
Angebot	im Wechsel mit anderen Fächern der gleichen Fachgruppe
Aufwand	30 Präsenzstunden Vorlesung, 30 Präsenzstunden Praktikum, 45 Stunden Vor-/Nachbereitung des Praktikums, 45 Stunden Nachbereitung der Vorlesung und Prüfungsvorbereitung
Voraussetzungen	Kenntnisse in linearer Algebra und Analysis, Grundlegende Programmierkenntnisse. BA Inf: Mathematik I, Mathematik II, Prozedurale Programmierung, Objektorientierte Programmierung
Ziele	<p>Lernziele:</p> <p>Die Studierenden lernen verschiedene Modelltypen und passende Lernverfahren aus dem Bereich des maschinellen Lernen kennen und anwenden, um sie in ihrer späteren beruflichen Tätigkeit bei der Analyse von Daten verschiedenster Modalitäten hinsichtlich Erkenntnisgewinn und Vorhersage sinnvoll einsetzen zu können.</p> <p>Fach- &amp; Methodenkompetenz:</p> <p>Die Studierenden sind in der Lage</p> <ul style="list-style-type: none"><li>• grundlegende und komplexere Konzepte hinter maschinellen Lernverfahren zu erläutern,</li><li>• einfachere maschinelle Lernverfahren selbst zu implementieren,</li><li>• grundlegende und komplexere Machine-Learning-Modelle in verschiedenen Problemstellungen mit Hilfe moderner Frameworks anzuwenden und zu evaluieren</li><li>• sich anhand dieser Grundlagen selbstständig in weiterführende und komplexere Themengebiete einzuarbeiten</li></ul> <p>Überfachliche Kompetenz:</p> <ul style="list-style-type: none"><li>• Teamarbeit: Die Studierenden bearbeiten Problemstellungen in Kleingruppen</li></ul>
Inhalt	<ul style="list-style-type: none"><li>• Wiederholung der mathematischen Grundlagen des maschinellen Lernens (Lineare Algebra und Multivariate Analysis)</li><li>• Überblick Grundbegriffe des maschinellen Lernens</li><li>• Lineare Regression und erweiterte Lineare Regression mit Basiswechsel, nichtlinearen Basisfunktionen und Norm-Penalties</li><li>• Logistische Regression mit Maximum Likelihood Parameterschätzung</li><li>• K-Nearest Neighbors</li><li>• Entscheidungsbäume</li><li>• Unsupervised Methoden: PCA und Clustering</li><li>• Support Vector Machines für Klassifikation und Regression</li><li>• Einführung in die Neuronale Netze mit Perceptron und Adaline</li><li>• Ausblick Multilayer Perzeptron Netze und Deep Learning</li></ul>
Medien und Methoden	Beamer, Tafel, Jupyter Notebooks. Praktische Programmierungsaufgaben: Moodle CodeRunner, DOMJudge
Literatur	Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.  Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.  Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. Springer.  A. Aldo Faisal, Cheng Soon Ong, and Marc Peter Deisenroth (2020). Mathematics for Machine Learning, MIT press.

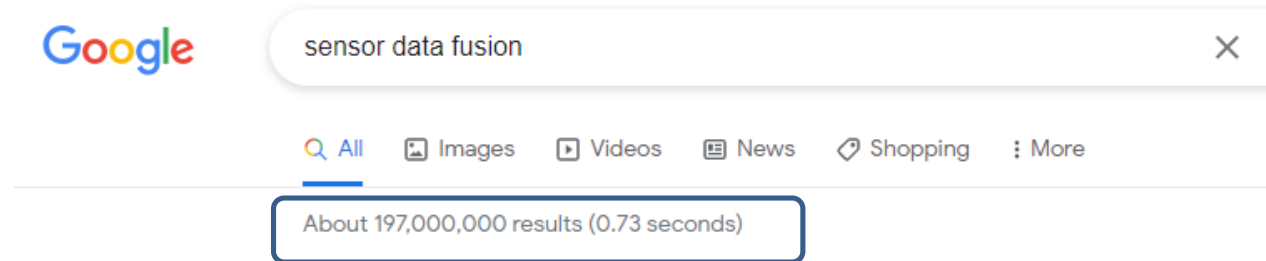
## Deep Learning

SWS	4
ECTS	5
Sprache(n)	Deutsch (Standard) Englisch
Lehrform	SU mit Praktikum
Angebot	nach Ankündigung
Aufwand	30 Präsenzstunden Vorlesung, 30 Präsenzstunden Praktikum, 45 Stunden Vor-/Nachbereitung des Praktikums, 45 Stunden Nachbereitung der Vorlesung und Prüfungsvorbereitung
Voraussetzungen	Kenntnisse in linearer Algebra und Analysis, Grundlegende Programmierkenntnisse. BA Inf: Mathematik I, Mathematik II, Prozedurale Programmierung, Objektorientierte Programmierung
Ziele	Kennenlernen und Verstehen grundlegender theoretischer Prinzipien des Deep Learning sowie die praktische Anwendung der Algorithmen auf unterschiedlichste Probleme. Erlernen von Fähigkeiten, um die Algorithmen in Python (eine der führenden Programmiersprachen im Bereich des maschinellen Lernens) zu implementieren und anzuwenden. Fähigkeiten, verschiedene Deep Learning Architekturen zu verstehen, zu implementieren und anzuwenden.
Inhalt	Deep Learning hat sich zu einem rasant wachsenden Gebiet im Bereich des maschinellen Lernens entwickelt und wird in einer Vielzahl unterschiedlichster Technologien erfolgreich eingesetzt, z. B. in der natürlichen Spracherkennung, Bild- und Objekterkennung oder autonomen Systemen und Robotern. In diesem Kurs beschäftigen wir uns zuerst mit den Grundlagen neuronaler Netze und lernen dann komplexere Systeme und Architekturen kennen. * Einführung in neuronale Netze (Perceptron) * Adaptive Linear Neurons, Gradient Descent, Stochastic Gradient Descent, Mini-batch Gradient Descent * Multilayer Neural Networks und Trainieren mittels Backpropagation * Aktivierungsfunktionen und Lossfunktionen * Normalisierung und Regularisierung * Moderne Verfahren der Hyperparameteroptimierung * Komplexere Optimierungsverfahren (AdaGrad, RMSProp, Adam) * Convolutional Neural Networks * Unterschiedliche Netzwerkarchitekturen * Transfer Learning und Neural Style Transfer * Object Recognition, Object Detection * Recurrent Neural Networks * Restricted Boltzmann Machines, AutoEncoders
Medien und Methoden	Beamer, Tafel, Jupyter Notebooks. Praktische Programmierungsaufgaben: Moodle CodeRunner, DOMJudge
Literatur	Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer. A. Aldo Faisal, Cheng Soon Ong, and Marc Peter Deisenroth (2020). Mathematics for Machine Learning, MIT press. C. Aggarwal (2018), Neural Networks and Deep Learning: A Textbook. Springer

# Inhalt

- Multimodale Sensorfusion: was und warum?
- Sensorfusion in der Praxis
- Deep Learning für multimodale Sensorfusion
  - Early Sensorfusion
  - Intermediate Sensorfusion
  - Late Sensorfusion
- Tipps und Tricks in der Praxis
- Fazit

# Multimodale Sensorfusion: was?



[27] Jalobeanu und Gutierrez (2006). „Das Datenfusionsproblem kann als die Berechnung der posterioren pdf [Wahrscheinlichkeitsverteilungsfunktion] des unbekannten einzelnen Objekts bei allen Beobachtungen angegeben werden.“

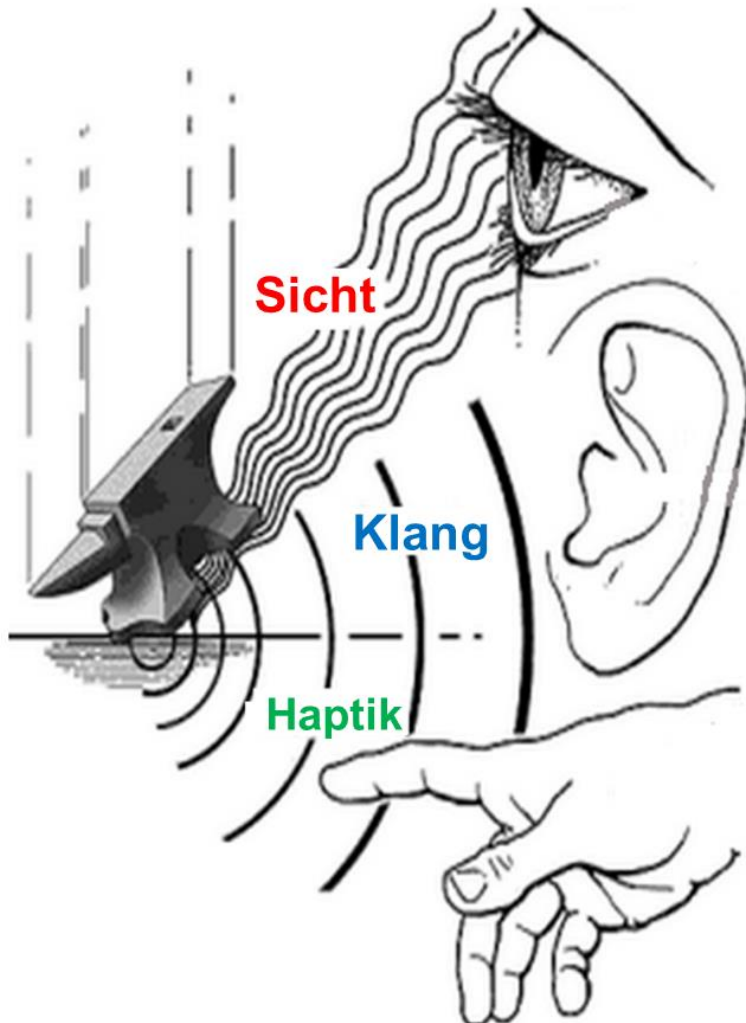
[28] Mastrogiovanni et al. (2007). „Das Ziel eines Datenfusionsprozesses ist es, den nützlichen Informationsgehalt aus heterogenen Quellen zu maximieren, um auf relevante Situationen und Ereignisse in Bezug auf die beobachtete Umgebung zu schließen.“

[29] Wikipedia (2007). „Informationsintegration ist ein Studienggebiet, das unter verschiedenen Begriffen bekannt ist: Informationsfusion, Deduplizierung, referentielle Integrität und so weiter. Es bezieht sich auf das Studienggebiet von Techniken, die versuchen, Informationen aus unterschiedlichen Quellen trotz unterschiedlicher konzeptioneller, kontextueller und typografischer Darstellungen zusammenzuführen. Dies wird beim Data Mining und der Konsolidierung von Daten aus halb- oder unstrukturierten Ressourcen verwendet.“

[30] Wikipedia (2007). „Sensorfusion ist die Kombination von sensorischen Daten oder Daten, die von sensorischen Daten aus unterschiedlichen Quellen abgeleitet werden, so dass die resultierenden Informationen in gewissem Sinne besser sind, als dies möglich wäre, wenn diese Quellen einzeln verwendet würden. Der Begriff „besser“ kann in diesem Fall „genauer“, „vollständiger“ oder „zuverlässiger“ bedeuten oder sich auf das Ergebnis einer entstehenden Ansicht beziehen, wie z ). Die Datenquellen für einen Fusionsprozess sind nicht so spezifiziert, dass sie von identischen Sensoren stammen. Man kann direkte Fusion, indirekte Fusion und Fusion der Ausgänge der ersteren zwei unterscheiden. Direkte Fusion ist die Fusion von Sensordaten aus einem Satz heterogener oder homogener Sensoren, Softsensoren, und Verlaufswerte von Sensordaten, während die indirekte Fusion Informationsquellen wie A-priori-Wissen über die Umwelt und menschliche Eingaben nutzt. Sensorfusion wird auch als (Multisensor-)Datenfusion bezeichnet und ist eine Teilmenge der Informationsfusion.“

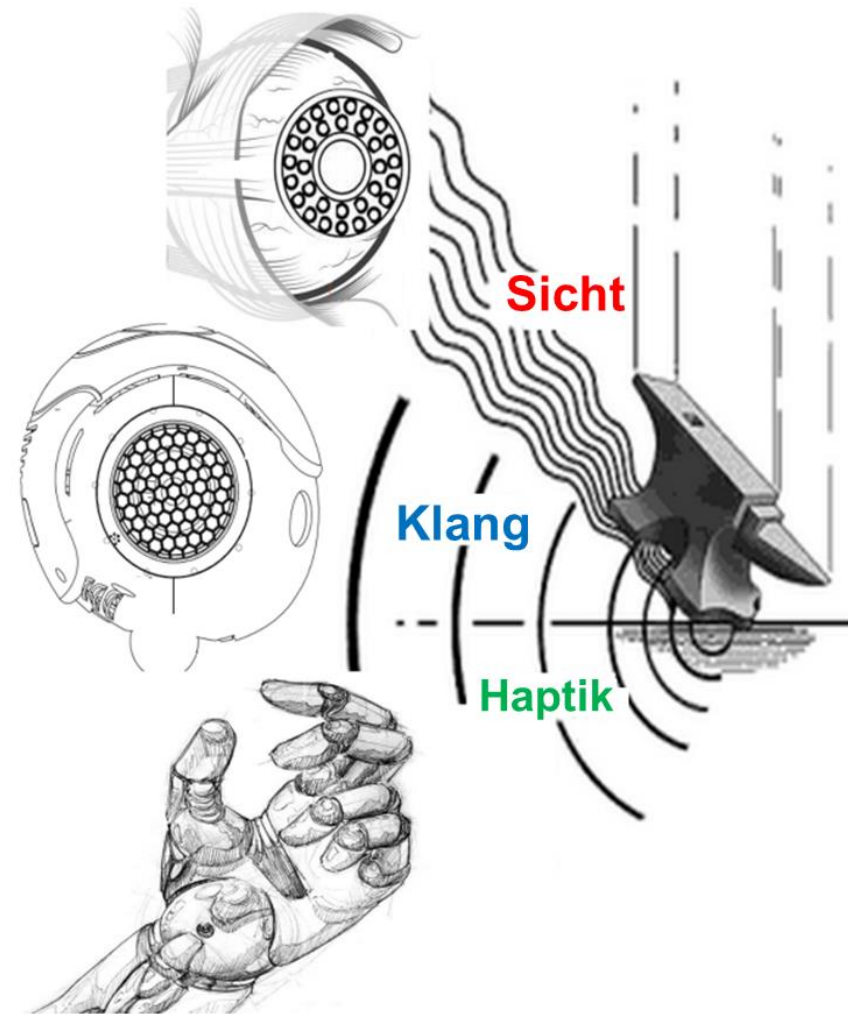
# Multimodale Sensorfusion: **warum?**

Biologische Systeme



Übertragung

Technische Systeme





# Sensorfusion in der Praxis

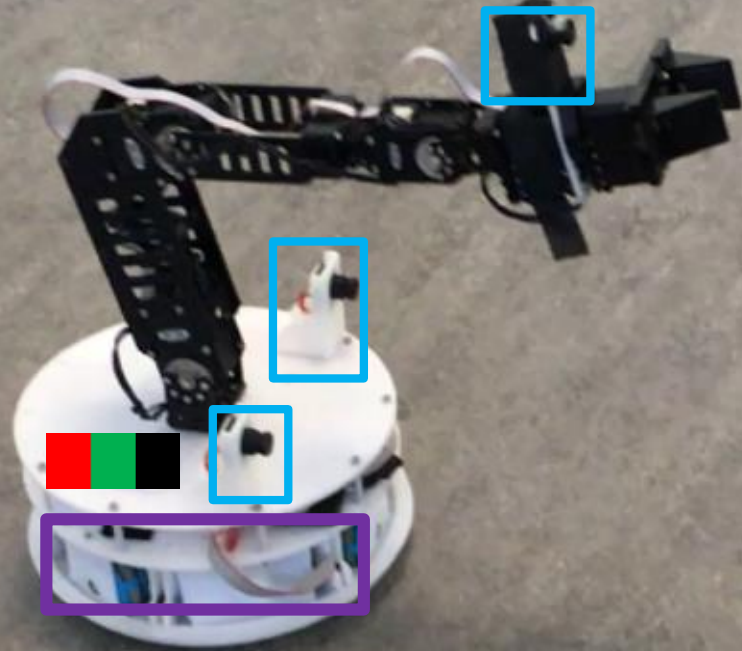
DEMO VIDEO



Aufgabe: Annäherung und Manipulation von Objekten mit LED-Reize, die mit unterschiedlichen Frequenzen pulsieren, um sie in Ordnung zu bringen (siehe [1])



# Sensorfusion in der Praxis



Pulsierende LEDs



Roboter-Sensoren:

3x Kameras

1x Gyroskop

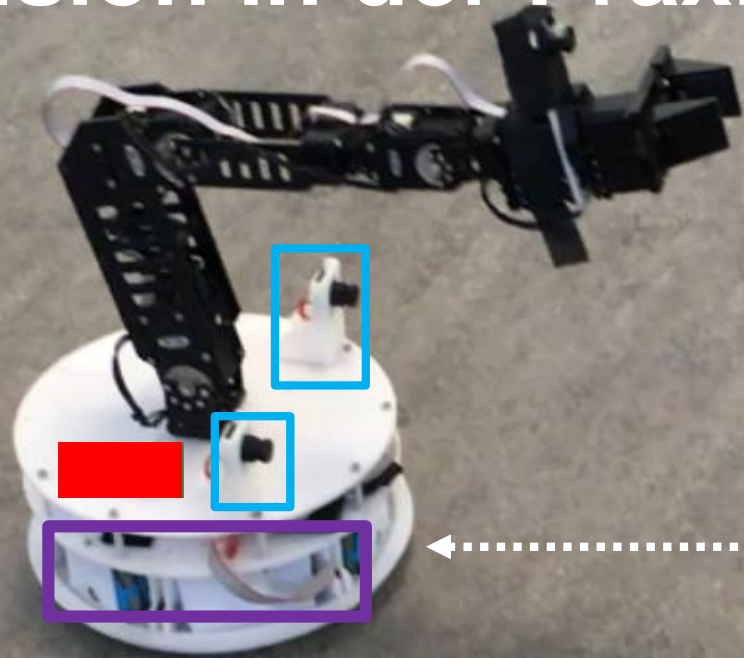
1x Beschleunigungssensor

3x Rad-Encoder

1x Magnetometer



# Sensorfusion in der Praxis



Pulsierende LEDs

Abstand zum Objekt

2x Kameras

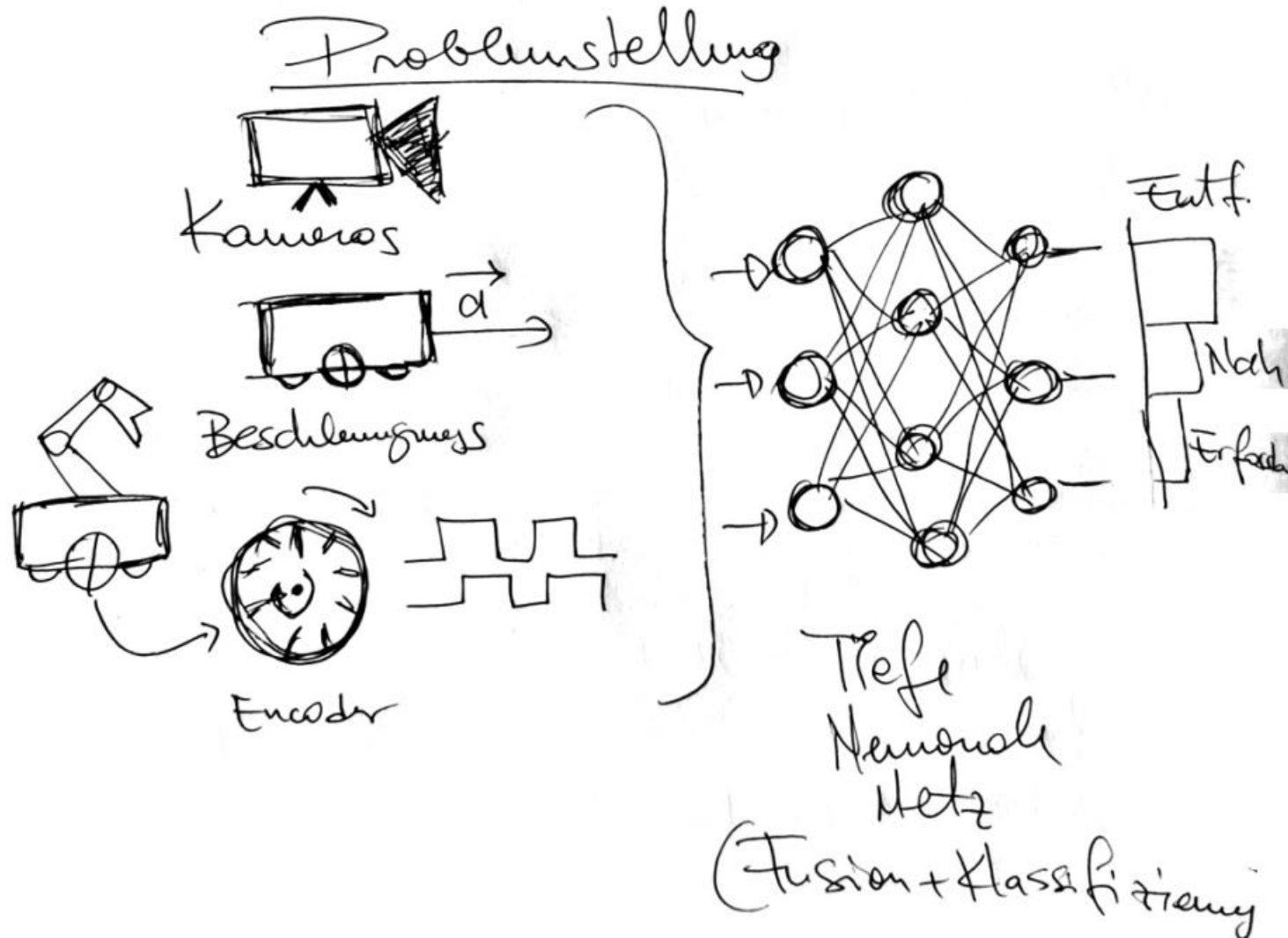
1x Beschleunigungssensor

3x Rad-Encoder (Odometrie)

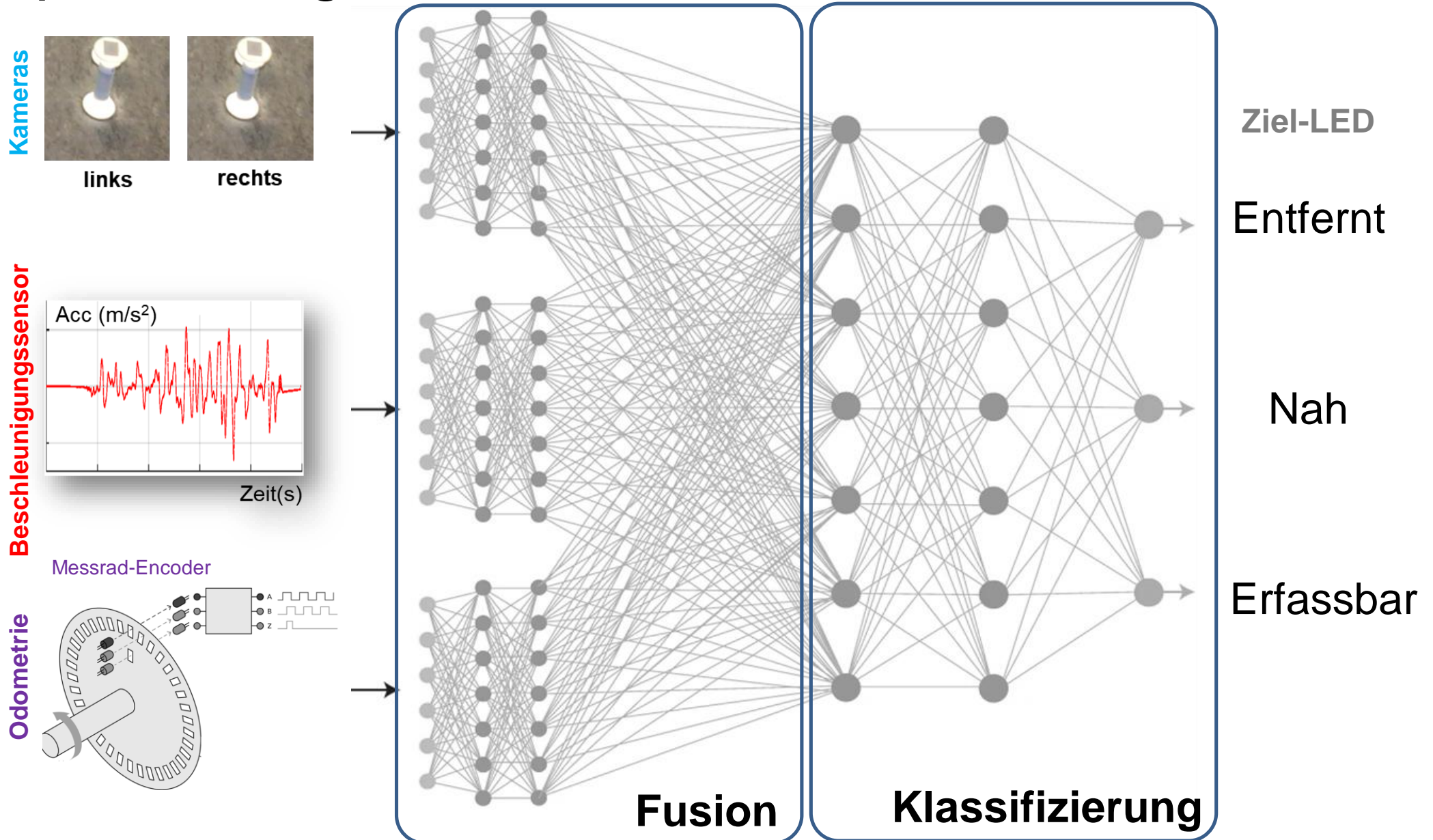


# Deep Learning für multimodale Sensorfusion

Schwarzbrett:

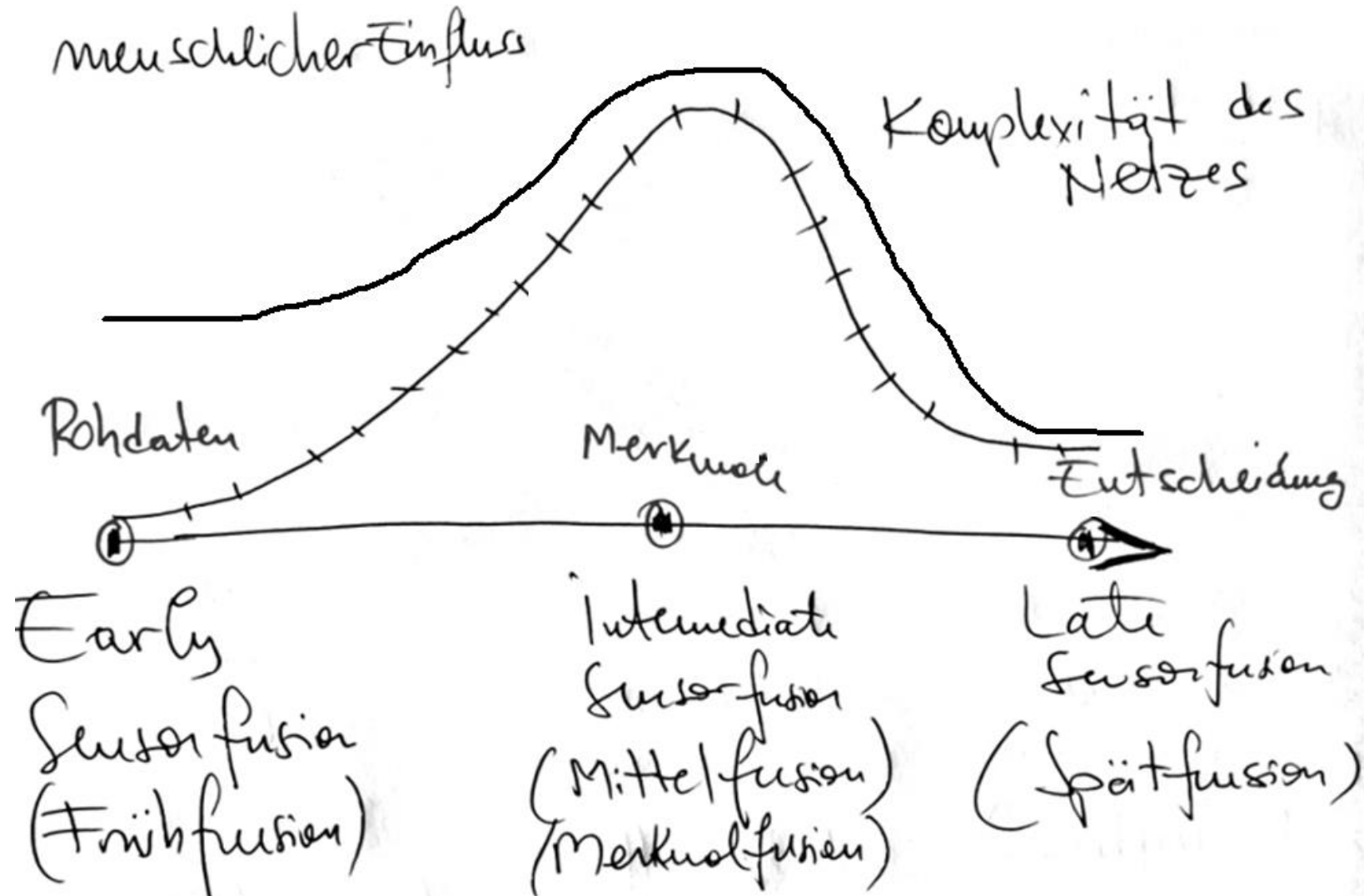


# Deep Learning für multimodale Sensorfusion



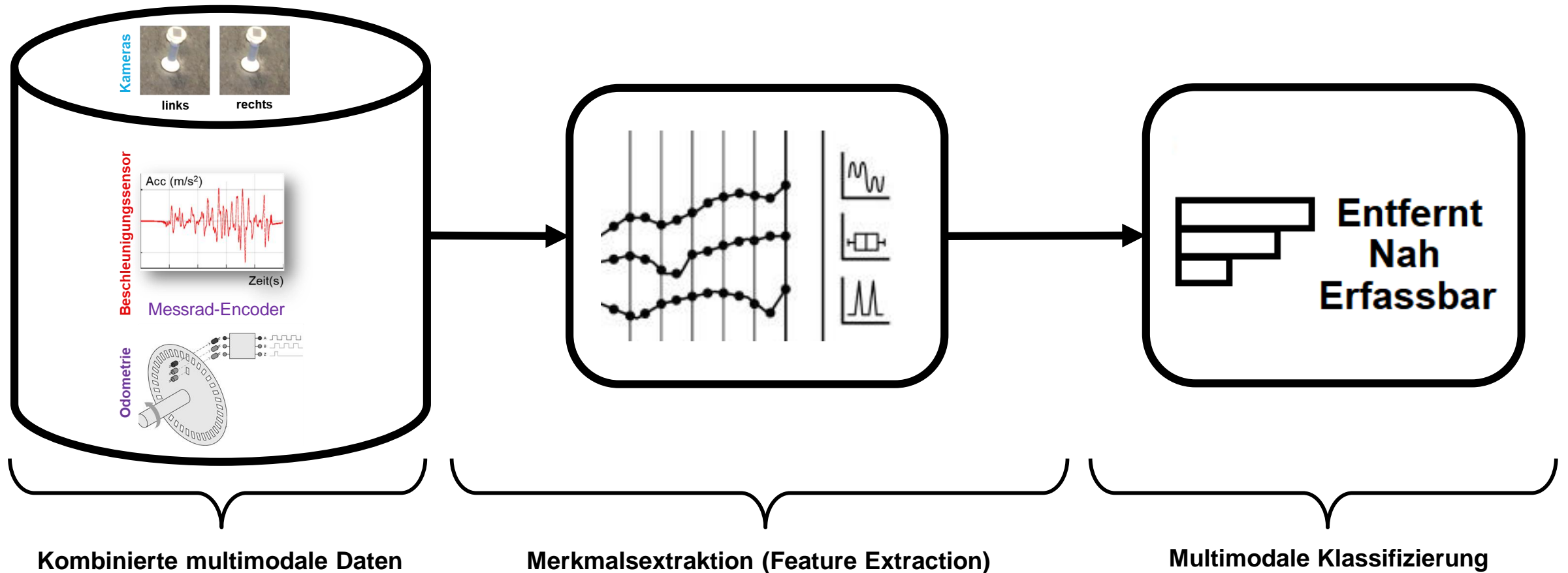
# Deep Learning für multimodale Sensorfusion

Schwarzbrett:



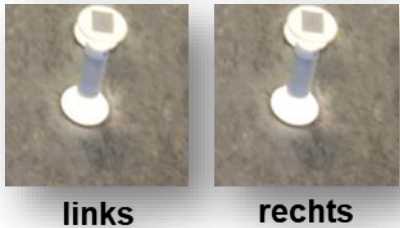


# Deep Learning für Early Sensorfusion

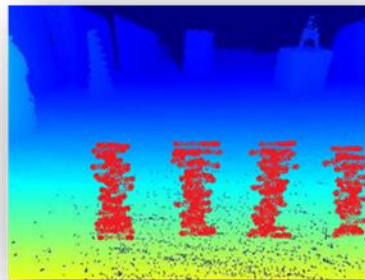


# Deep Learning für Early Sensorfusion

Schwarzbrett:

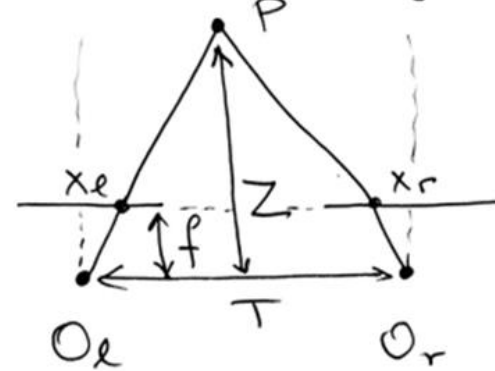


Kameras



Nah Fern

Tiefenentfernung Berechnung durch Triangulation  
- geometrische Abstandsmessung



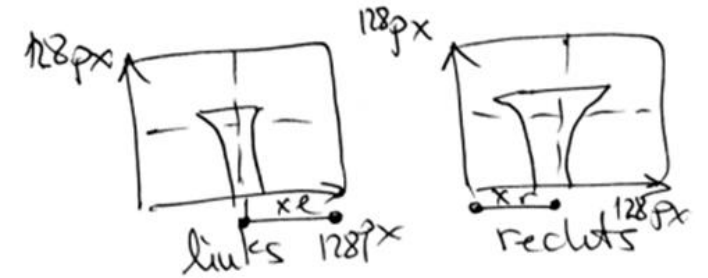
Kamera links

Kamera rechts

$$\frac{T}{Z} = \frac{T-d}{Z-f}$$

$\Downarrow$

$$Z = \frac{f \cdot T}{d}$$

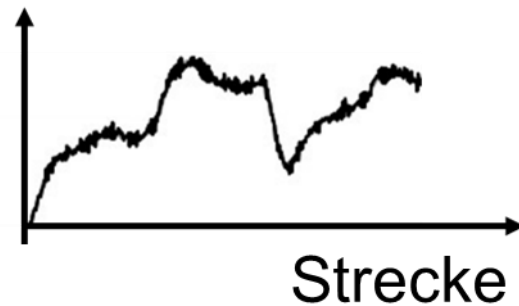
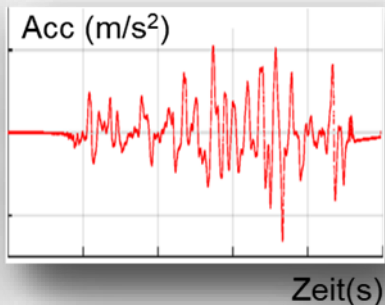


Korrespondenzanalyse  
berechnete Disparität

$$d = x_l - x_r$$

# Deep Learning für Early Sensorfusion

Schwarzbrett:



Positionsbeziehung aus Beschleunigung

Die Beschleunigung ist die Änderungsrate der Geschwindigkeit eines Objekts. Gleichzeitig ist die Geschwindigkeit die Änderungsrate der Position desselben Objekts.

$$\left. \begin{aligned} \vec{a} &= \frac{d\vec{v}}{dt} \\ \vec{v} &= \frac{d\vec{s}}{dt} \end{aligned} \right\} \Rightarrow \vec{a} = \frac{d(\frac{d\vec{s}}{dt})}{dt^2}$$

Wenn  $\vec{a}$  bekannt ist, können wir  $\vec{s}$  erhalten durch doppelte Integration

$$\vec{v} = \int \vec{a} dt$$

$$\vec{s} = \int \vec{v}(dt) = \iint \vec{a} dt dt$$



# Deep Learning für Early Sensorfusion

Schwarzbrett:

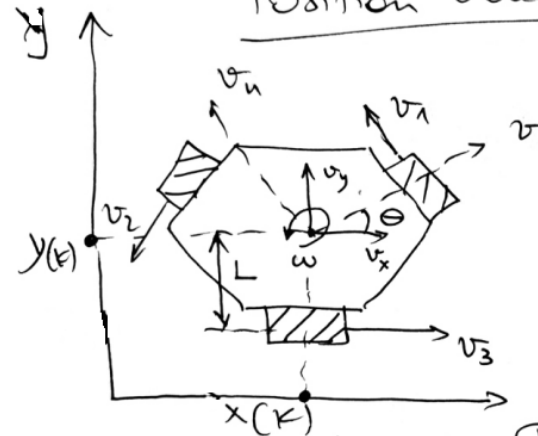


Messrad-Encoder Pulse → Geschwindigkeit

$$\left\{ \begin{array}{l} \omega_{\text{rad}}(t) = \frac{(n(t) - n(t-1))}{\Delta t} \cdot \frac{\pi}{180} \\ v_{\text{rad}}(t) = \omega_{\text{rad}}(t) \cdot R \end{array} \right.$$

Aktuelle Anzahl von  
 $n(t)$  = Messrad-Encoder Pulse  
 $n(t-1)$  = vorherige Anzahl von  
 Messrad-Encoder Pulse  
 $\Delta t$  - Abtastzeit  
 $R$  - Radius des Rades  
 $\omega_{\text{rad}}$  - Winkelgeschwindigkeit  
 $v_{\text{rad}}$  - Geschwindigkeit

Position berechnung (Odometrie)



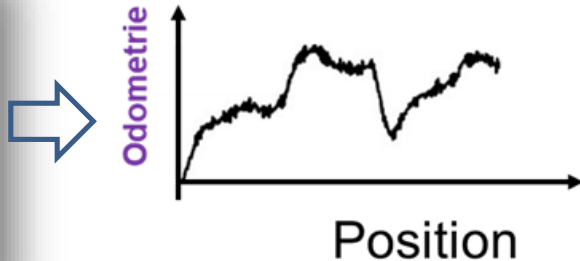
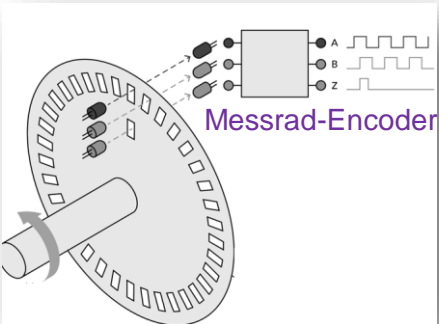
Roboter  
 Kinematik - Geschwindigkeit

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3}-\theta) & -\cos(\frac{\pi}{3}-\theta) & L \\ \sin(\frac{\pi}{3}+\theta) & -\cos(\frac{\pi}{3}+\theta) & L \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix}$$

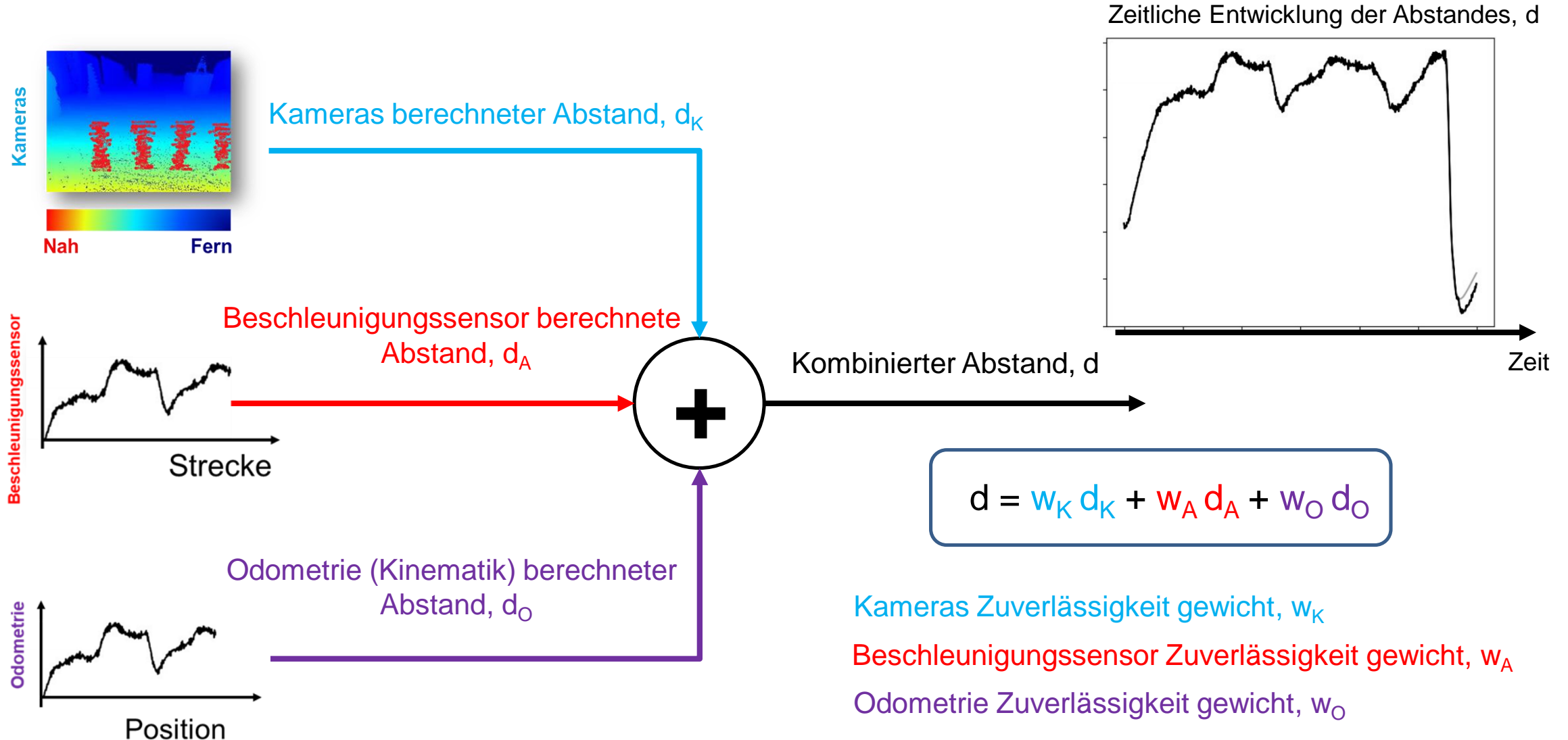
Kinematik-Position

$$\left\{ \begin{array}{l} x(k) = x(k-1) + v_x \cdot T \\ y(k) = y(k-1) + v_y \cdot T \end{array} \right.$$

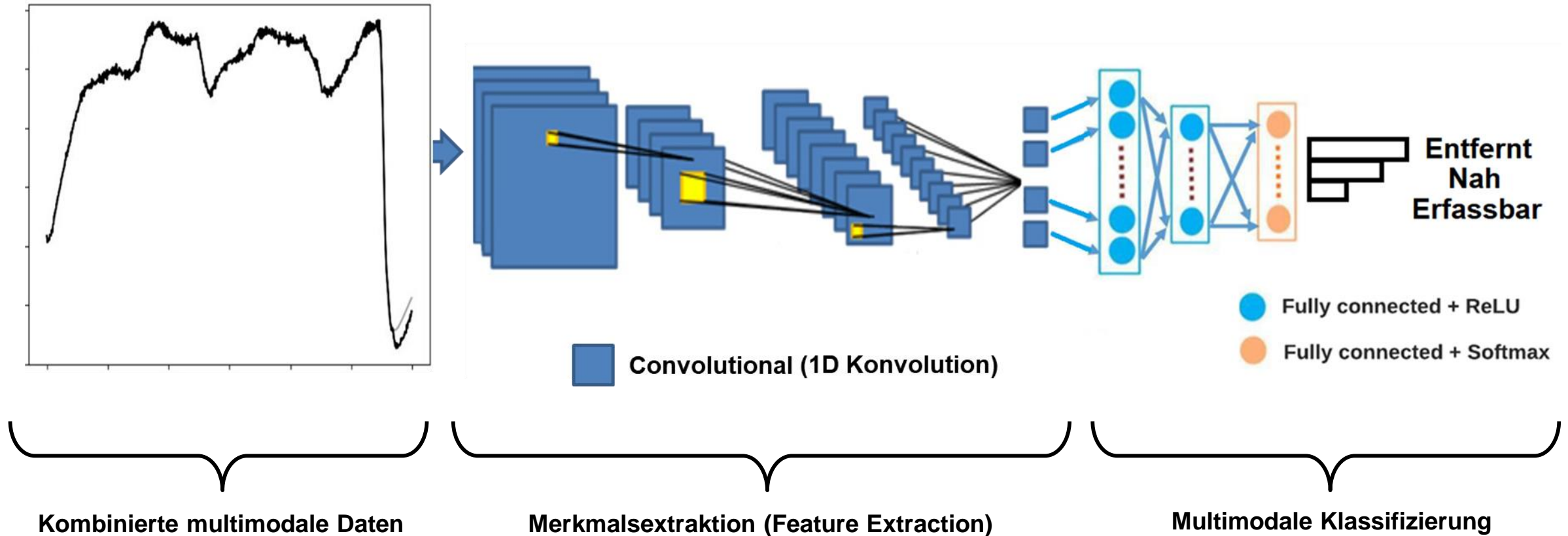
$T$  - Abtastzeit



# Deep Learning für Early Sensorfusion



# Deep Learning für Early Sensorfusion

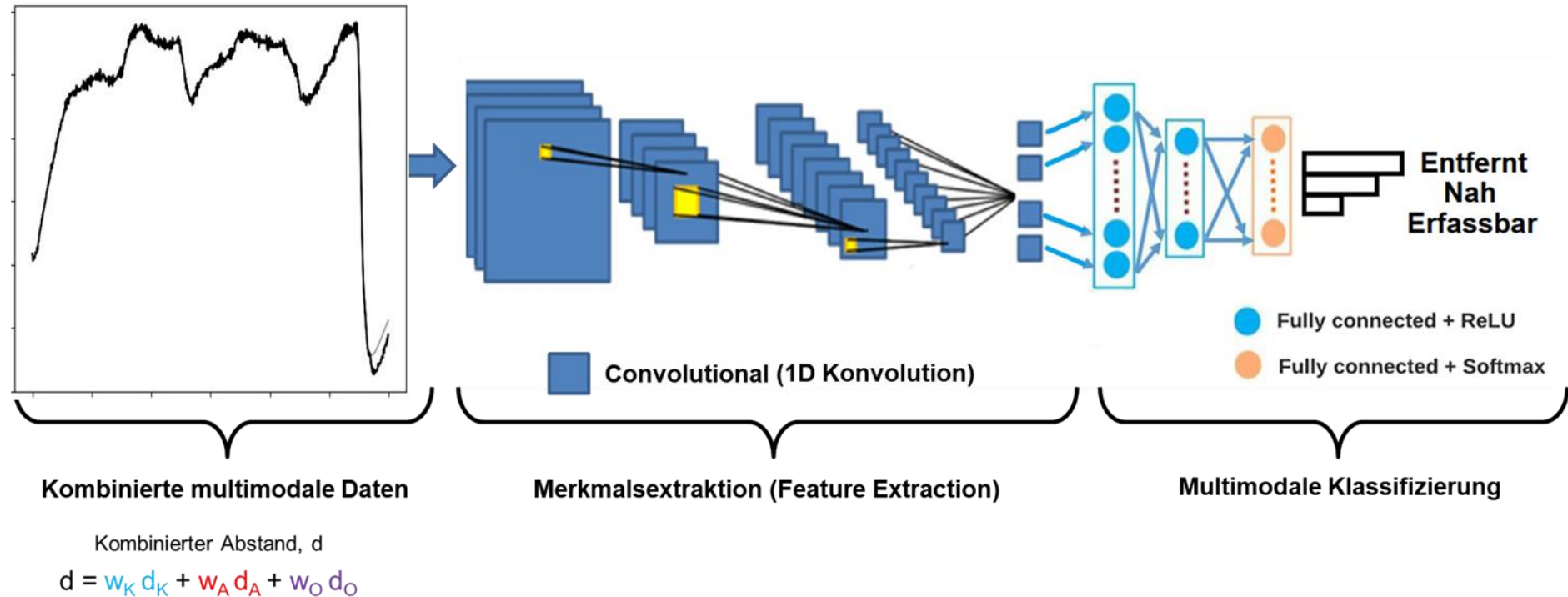


Kombinierter Abstand,  $d$

$$d = w_K d_K + w_A d_A + w_O d_O$$



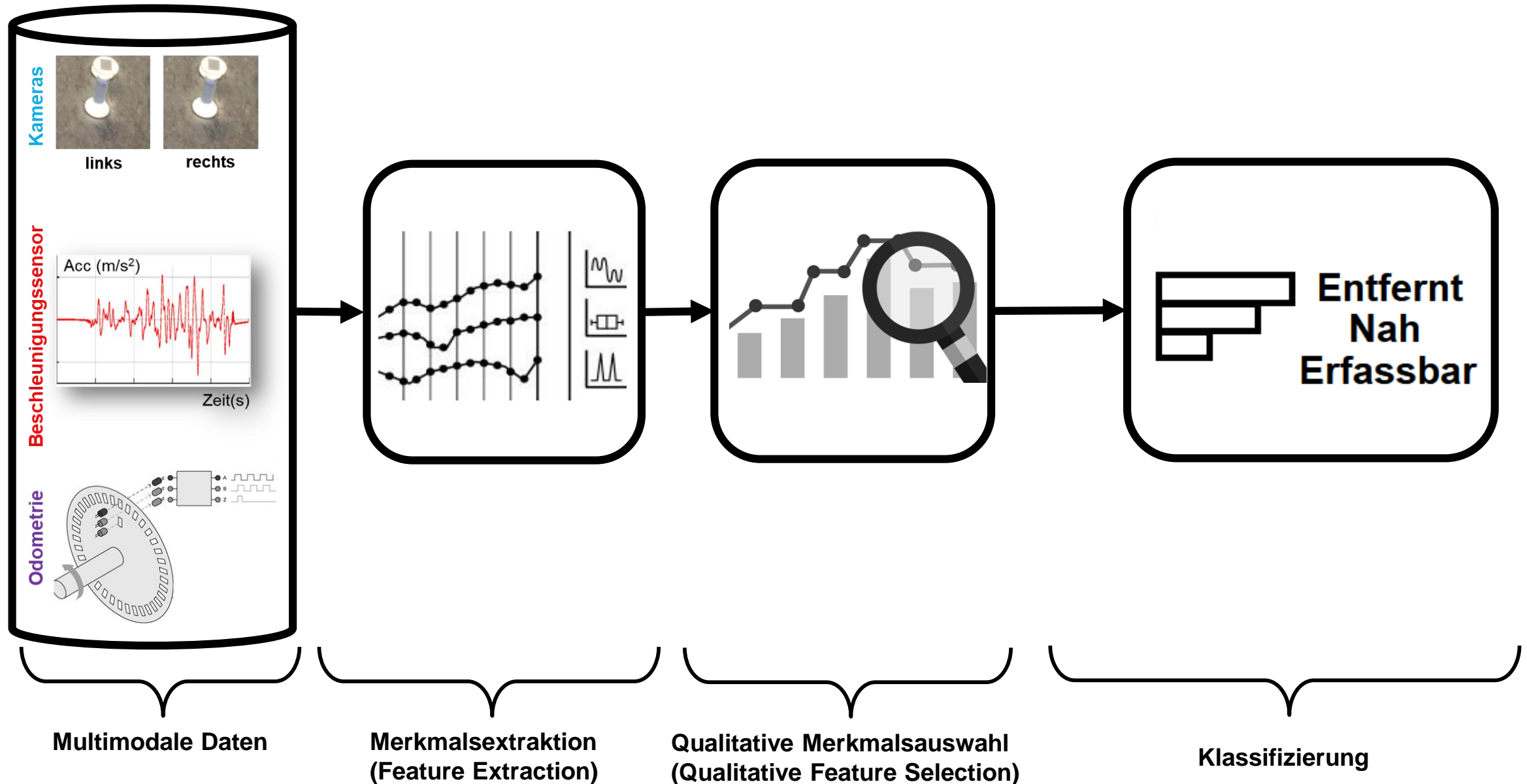
# Deep Learning für Early Sensorfusion



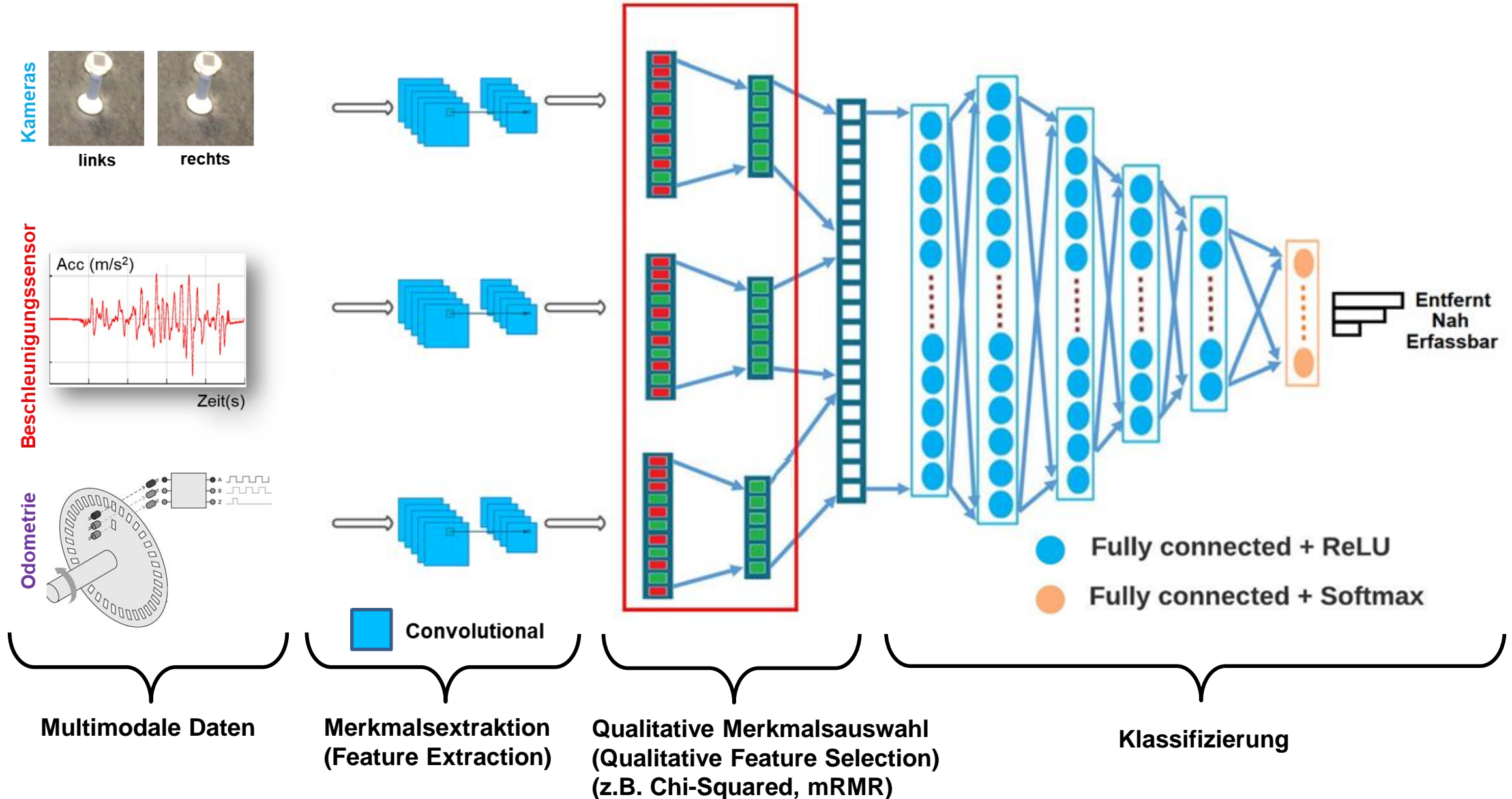
## Allgemeine Eigenschaften

- erfordert ein gutes Verständnis der **Sensorphysik**
- **manuelle Kombination** von Sensorenmodalitäten
- die **Merkmalsextraktion** ist in hohem Maße von der einheitlichen **Darstellung abhängig**
- **Entscheidung/Klassifizierung** wird **vereinfacht**

# Deep Learning für Intermediate Sensorfusion



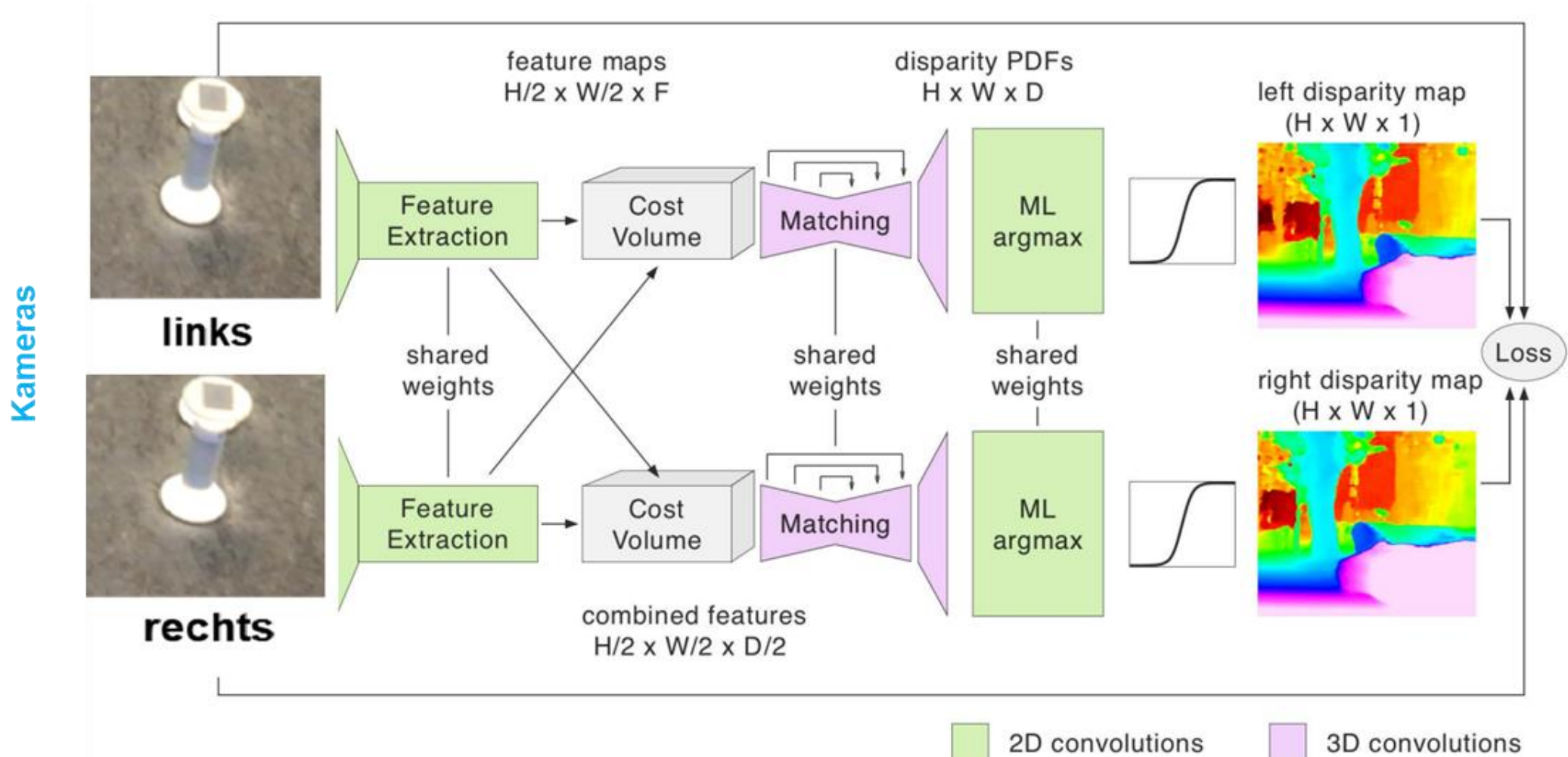
# Deep Learning für Intermediate Sensorfusion





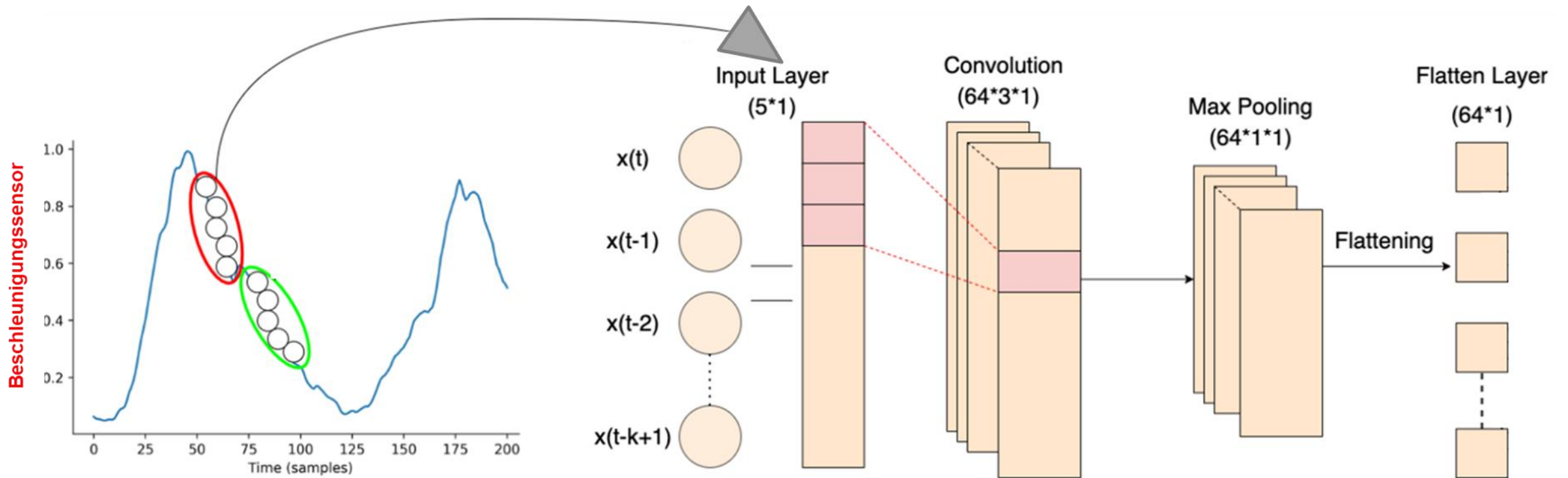
# Deep Learning für Intermediate Sensorfusion

Faltungsnetzwerk (Conv Net) zur Fusion der Daten von zwei Kameras zur Berechnung der Tiefe (siehe [8])

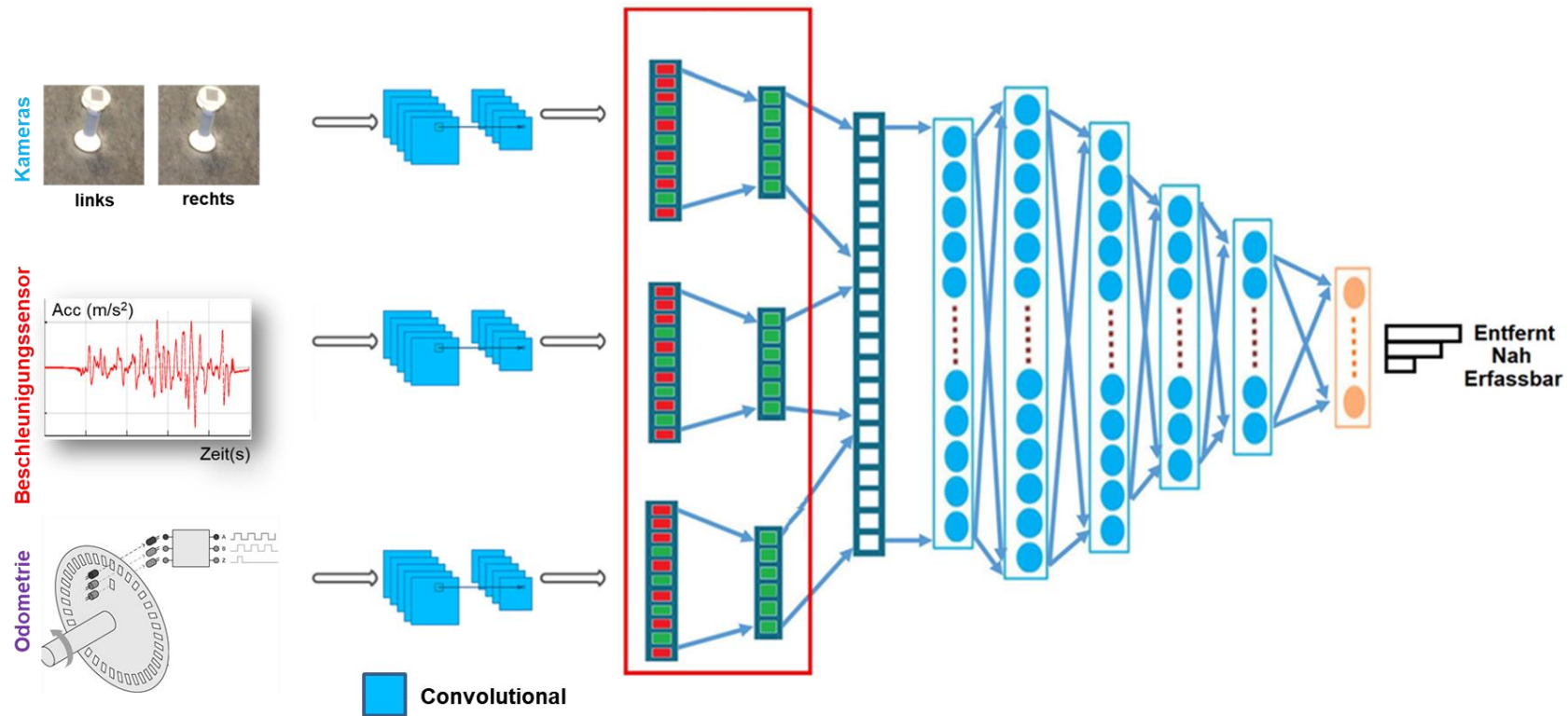


# Deep Learning für Intermediate Sensorfusion

Faltungsnetzwerk (Conv Net) zur Extraktion von Merkmalen aus sensorischen Zeitreihen (z. B. Beschleunigungsmesser, Rad-Encoder) (siehe [7])



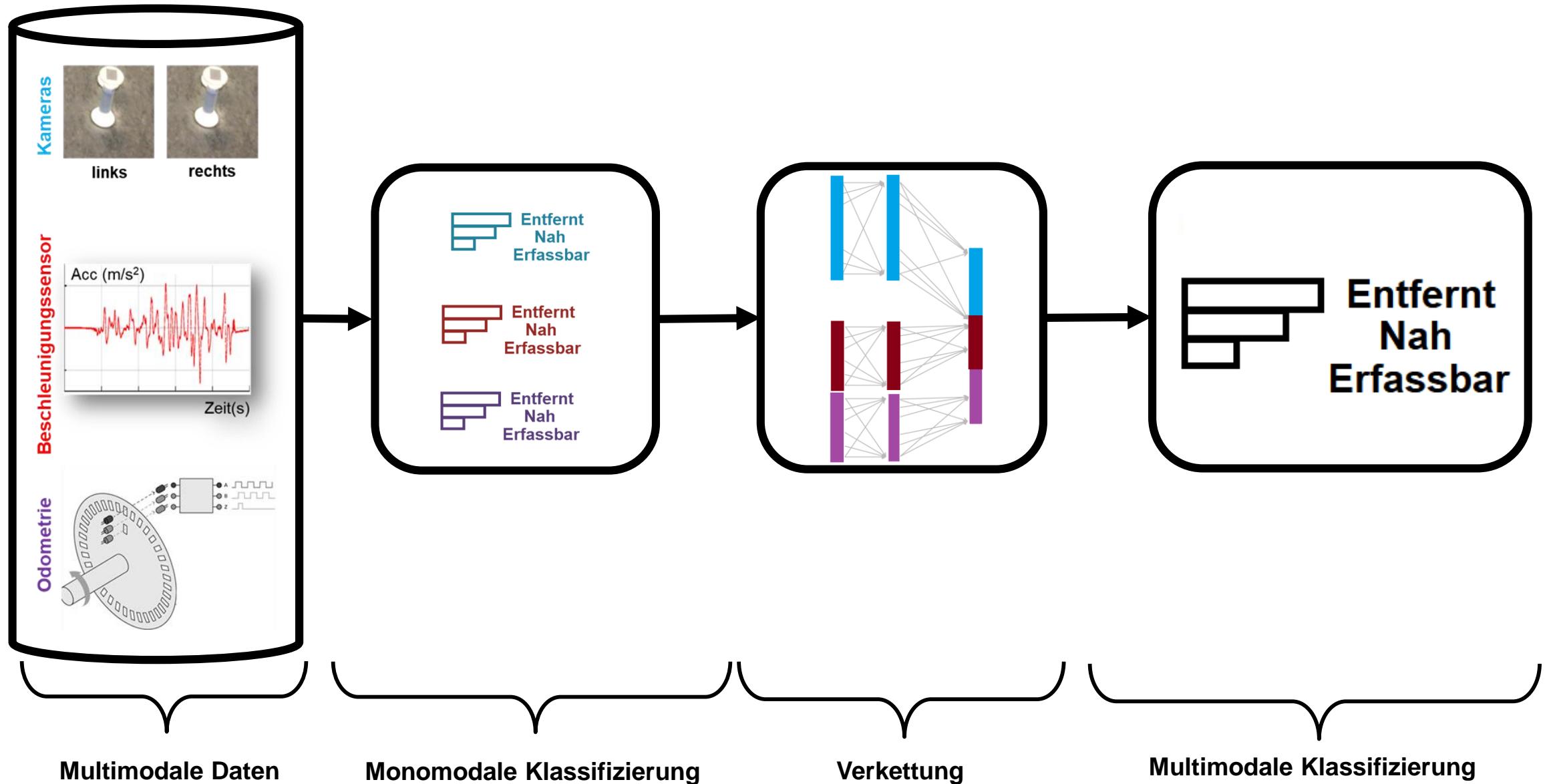
# Deep Learning für Intermediate Sensorfusion



## Allgemeine Eigenschaften

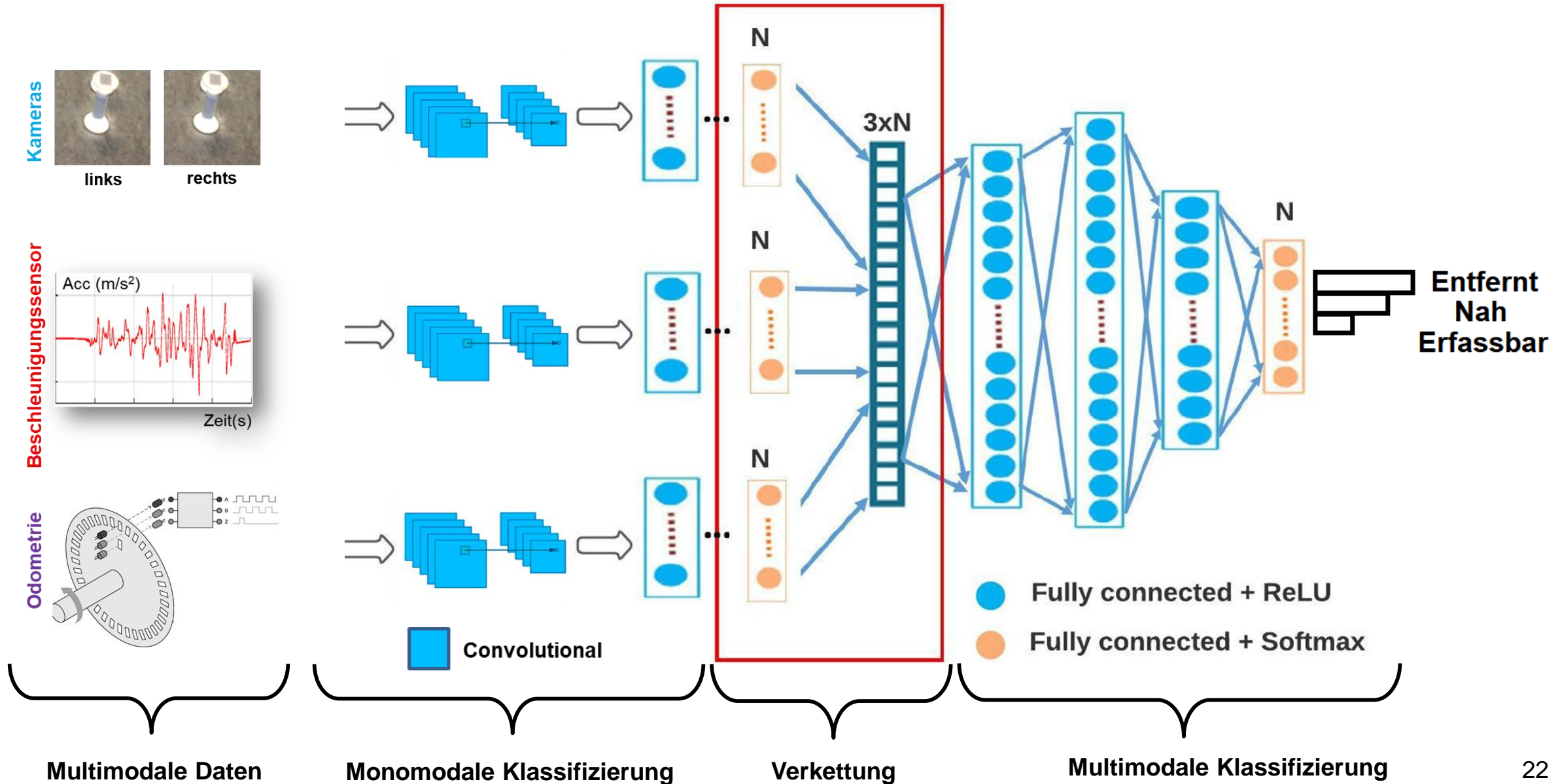
- Merkmalsextraktion erhöht die **Aussagekraft** der **Fusion**
- **Auswahl/Rangfolge** der Merkmale ist erforderlich, um **Redundanz** zu vermeiden
- erfordert **Selektionsregeln** für jede **Modalität**, die in den **Netzentwurf eingebettet** sind
- Komplizierter Entwurf des Teils der Merkmalsextraktion (typischerweise benutzerdefinierte Conv Nets)

# Deep Learning für Late Sensorfusion

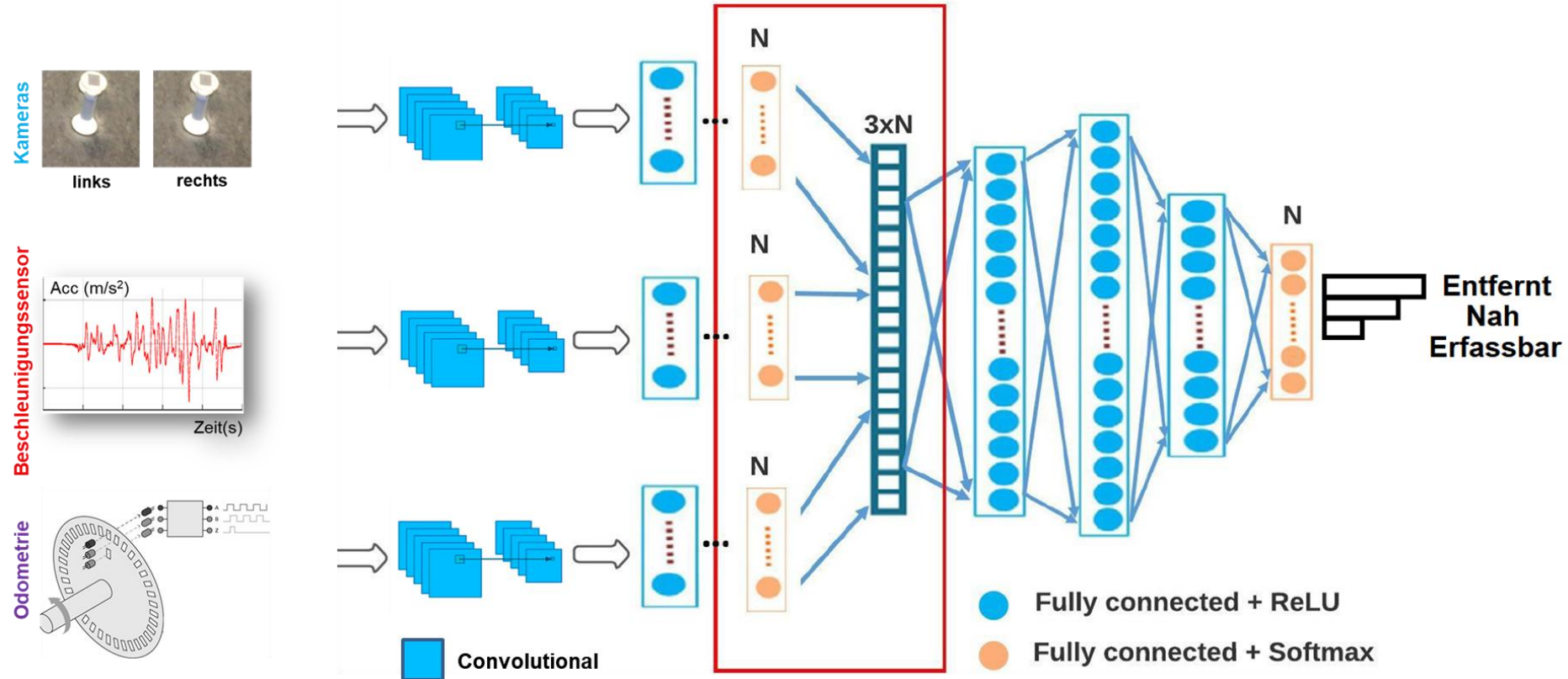




# Deep Learning für Late Sensorfusion



# Deep Learning für Late Sensorfusion



## Allgemeine Eigenschaften

- es handelt sich im Grunde um eine **Kombinationsstrategie** für die **Klassifizierung** der **einzelnen Modalitäten**
- verwendet **manuelle Regeln** zur **Verkettung** der **Einzelentscheidungen** auf der Grundlage der **Zuverlässigkeit**

# Tipps und Tricks in der Praxis

*Welche Art von Modell sollte ich verwenden?\**

Art der Fusion	Klassifizierung	Feature-Extraction	Feature-Selection
Early	DenseNet	VGG16, ResNet RBM	-
Intermediate	DenseNet	VGG16, ResNet, ST-GCN	RFE, ANOVA
Late	MLP, DenseNet	VGG16	DFC

**VGG16** - Convolutional Neural Network, Merkmalsextraktor, Einzelbild

**DenseNet** - vollständig verbundenes Netz mit Merkmalskarten, Einzelbild

**ResNet** - gruppenbezogenes neuronales Faltungsnetz, Sequenz von Einzelbildern

**ST-GCN** - räumlich-zeitliches Graph-Faltungsnetz

**RFE** - rekursive Merkmalseliminierung

**ANOVA** - Varianzanalyse

**DFC** - Tiefe Merkmalsverkettung

**RBM** – Restricted Boltzmann Machines

\* Analyse aus der Paper Review [2], [3], [4], [5], [6], [7], [8]

# Fazit

- **Sensorfusion** ist eine sehr wichtige **Komponente** für die **Autonomie** (sowohl in biologischen als auch in technischen Systemen)
- **tiefe neuronale Netze** sind ein sehr leistungsfähiges **Lernwerkzeug** für die **Sensorfusion**
- die Wahl der **Fusionsmethode** (früh, mittel, spät) ist **problemabhängig**
- **frühe Fusion** ist für **niedrigdimensionale** Daten geeignet und erfordert eine **physikalische Modellierung**
- **intermediäre Fusion** eignet sich sowohl für **niedrig- als auch für hochdimensionale Daten**, erfordert jedoch die Entwicklung einer **Merkmalsauswahl** für die **kombinierte Entscheidung**
- **späte Fusion** eignet sich sowohl für **niedrig- als auch für hochdimensionale Daten**, wirkt aber nur als **Kombination einzelner Modalitätsentscheidungen**



# Literaturverzeichnis

- [1] Mirus, F., **Axenie**, C., Stewart, T. C., & Conradt, J. (2018). Neuromorphic sensorimotor adaptation for robotic mobile manipulation: From sensing to behaviour. *Cognitive Systems Research*, 50, 52-66.
- [2] Boulahia, S. Y., Amamra, A., Madi, M. R., & Daikh, S. (2021). Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Machine Vision and Applications*, 32(6), 1-18.
- [3] Assa, A., & Janabi-Sharifi, F. (2015). A Kalman filter-based framework for enhanced sensor fusion. *IEEE Sensors Journal*, 15(6), 3281-3292.
- [4] Gao, J., Li, P., Chen, Z., & Zhang, J. (2020). A survey on deep learning for multimodal data fusion. *Neural Computation*, 32(5), 829-864.
- [5] Zhang, H., Xu, H., Tian, X., Jiang, J., & Ma, J. (2021). Image fusion meets deep learning: A survey and perspective. *Information Fusion*, 76, 323-336.
- [6] Kam, M., Zhu, X., & Kalata, P. (1997). Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 85(1), 108-119.
- [7] R. Chandra, S. Goyal and R. Gupta, "Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction," in *IEEE Access*, vol. 9, pp. 83105-83123, 2021, doi: 10.1109/ACCESS.2021.3085085.
- [8] Smolyanskiy, N., Kamenev, A., & Birchfield, S. (2018). On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1007-1015).

# Video Vortrag



**Youtube Vortrag**

in Produktion, verfügbar vor Montag 9.05.2022

# Maschinelles Lernen

<b>SWS</b>	4
<b>ECTS</b>	5
<b>Sprache(n)</b>	Deutsch (Standard) Englisch
<b>Lehrform</b>	SU mit Praktikum
<b>Angebot</b>	im Wechsel mit anderen Fächern der gleichen Fachgruppe
<b>Aufwand</b>	30 Präsenzstunden Vorlesung, 30 Präsenzstunden Praktikum, 45 Stunden Vor-/Nachbereitung des Praktikums, 45 Stunden Nachbereitung der Vorlesung und Prüfungsvorbereitung
<b>Voraussetzungen</b>	Kenntnisse in linearer Algebra und Analysis, Grundlegende Programmierkenntnisse. BA Inf: Mathematik I, Mathematik II, Prozedurale Programmierung, Objektorientierte Programmierung
<b>Ziele</b>	<p>Lernziele:</p> <p>Die Studierenden lernen verschiedene Modelltypen und passende Lernverfahren aus dem Bereich des maschinellen Lernen kennen und anwenden, um sie in ihrer späteren beruflichen Tätigkeit bei der Analyse von Daten verschiedenster Modalitäten hinsichtlich Erkenntnisgewinn und Vorhersage sinnvoll einsetzen zu können.</p> <p>Fach- &amp; Methodenkompetenz:</p> <p>Die Studierenden sind in der Lage</p> <ul style="list-style-type: none"> <li>• grundlegende und komplexere Konzepte hinter maschinellen Lernverfahren zu erläutern,</li> <li>• einfachere maschinelle Lernverfahren selbst zu implementieren,</li> <li>• grundlegende und komplexere Machine-Learning-Modelle in verschiedenen Problemstellungen mit Hilfe moderner Frameworks anzuwenden und zu evaluieren</li> <li>• sich anhand dieser Grundlagen selbstständig in weiterführende und komplexere Themengebiete einzuarbeiten</li> </ul> <p>Überfachliche Kompetenz:</p> <ul style="list-style-type: none"> <li>• Teamarbeit: Die Studierenden bearbeiten Problemstellungen in Kleingruppen</li> </ul>
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Wiederholung der mathematischen Grundlagen des maschinellen Lernens (Lineare Algebra und Multivariate Analysis)</li> <li>• Überblick Grundbegriffe des maschinellen Lernens</li> <li>• Lineare Regression und erweiterte Lineare Regression mit Basiswechsel, nichtlinearen Basisfunktionen und Norm-Penalties</li> <li>• Logistische Regression mit Maximum Likelihood Parameterschätzung</li> <li>• K-Nearest Neighbors</li> <li>• Entscheidungsbäume</li> <li>• Unsupervised Methoden: PCA und Clustering</li> <li>• Support Vector Machines für Klassifikation und Regression</li> <li>• Einführung in die Neuronale Netze mit Perceptron und Adaline</li> <li>• Ausblick Multilayer Perceptron Netze und Deep Learning</li> </ul>
<b>Medien und Methoden</b>	Beamer, Tafel, Jupyter Notebooks. Praktische Programmierungsaufgaben: Moodle CodeRunner, DOMJudge
<b>Literatur</b>	<p>Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.</p> <p>Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.</p> <p>Friedman, J., Hastie, T., &amp; Tibshirani, R. (2001). The elements of statistical learning. Springer.</p> <p>A. Aldo Faisal, Cheng Soon Ong, and Marc Peter Deisenroth (2020). Mathematics for Machine Learning. MIT press.</p>

# Deep Learning

<b>SWS</b>	4
<b>ECTS</b>	5
<b>Sprache(n)</b>	Deutsch (Standard) Englisch
<b>Lehrform</b>	SU mit Praktikum
<b>Angebot</b>	nach Ankündigung
<b>Aufwand</b>	30 Präsenzstunden Vorlesung, 30 Präsenzstunden Praktikum, 45 Stunden Vor-/Nachbereitung des Praktikums, 45 Stunden Nachbereitung der Vorlesung und Prüfungsvorbereitung
<b>Voraussetzungen</b>	Kenntnisse in linearer Algebra und Analysis, Grundlegende Programmierkenntnisse. BA Inf: Mathematik I, Mathematik II, Prozedurale Programmierung, Objektorientierte Programmierung
<b>Ziele</b>	Kennenlernen und Verstehen grundlegender theoretischer Prinzipien des Deep Learning sowie die praktische Anwendung der Algorithmen auf unterschiedlichste Probleme. Erlernen von Fähigkeiten, um die Algorithmen in Python (eine der führenden Programmiersprachen im Bereich des maschinellen Lernens) zu implementieren und anzuwenden. Fähigkeiten, verschiedene Deep Learning Architekturen zu verstehen, zu implementieren und anzuwenden.
<b>Inhalt</b>	Deep Learning hat sich zu einem rasant wachsenden Gebiet im Bereich des maschinellen Lernens entwickelt und wird in einer Vielzahl unterschiedlichster Technologien erfolgreich eingesetzt, z. B. in der natürlichen Spracherkennung, Bild- und Objekterkennung oder autonomen Systemen und Robotern. In diesem Kurs beschäftigen wir uns zuerst mit den Grundlagen neuronaler Netze und lernen dann komplexere Systeme und Architekturen kennen. * Einführung in neuronale Netze (Perceptron) * Adaptive Linear Neurons, Gradient Descent, Stochastic Gradient Descent, Mini-batch Gradient Descent * Multilayer Neural Networks und Trainieren mittels Backpropagation * Aktivierungsfunktionen und Lossfunktionen * Normalisierung und Regularisierung * Moderne Verfahren der Hyperparameteroptimierung * Komplexere Optimierungsverfahren (AdaGrad, RMSProp, Adam) * Convolutional Neural Networks * Unterschiedliche Netzwerkarchitekturen * Transfer Learning und Neural Style Transfer * Object Recognition, Object Detection * Recurrent Neural Networks * Restricted Boltzmann Machines, AutoEncoders
<b>Medien und Methoden</b>	Beamer, Tafel, Jupyter Notebooks. Praktische Programmierungsaufgaben: Moodle CodeRunner, DOMJudge
<b>Literatur</b>	Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer. A. Aldo Faisal, Cheng Soon Ong, and Marc Peter Deisenroth (2020). Mathematics for Machine Learning, MIT press. C. Aggarwal (2018), Neural Networks and Deep Learning: A Textbook. Springer