Projektarbeit 1 + 2

Miro Göttler, Pol Zeimet

M

Gliederung

Ziel
Was soll in dieser Arbeit erforscht werden?

Daten
Datensatz, Auswahl und Format

Vorverarbeitung
Teilung, Normalisierung und Sequenzierung der Daten

Modelle
Beschreibung und Ergebnisse der Netze

Vergleich
Genauigkeit, Konfusionsmatrix, Cross View/Subject

Fazit
Ergebnisse und nächste Schritte

Ziel der Arbeit

„Erkennung von Bewegungsvorgängen von Personen anhand 3D-Keypoint-Daten mit Rekurrenten und Convolutional Neuronalen Netzen.“

Datenauswahl

Klasse 0
A028 – phone call

Klasse 1
A028 – play with phone

Klasse 2
A028 – taking a selfie

A001	drink water
A003	brush teeth
A010	clapping
A014	put on jacket
A019	take off glasses
A021	take off hat/cap

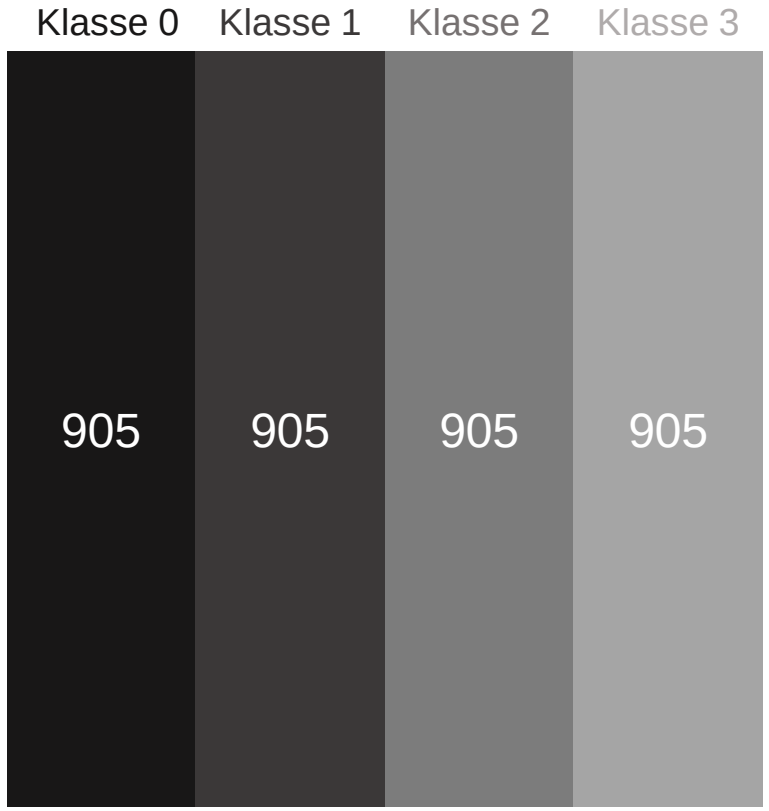
Klasse 3
Restklasse

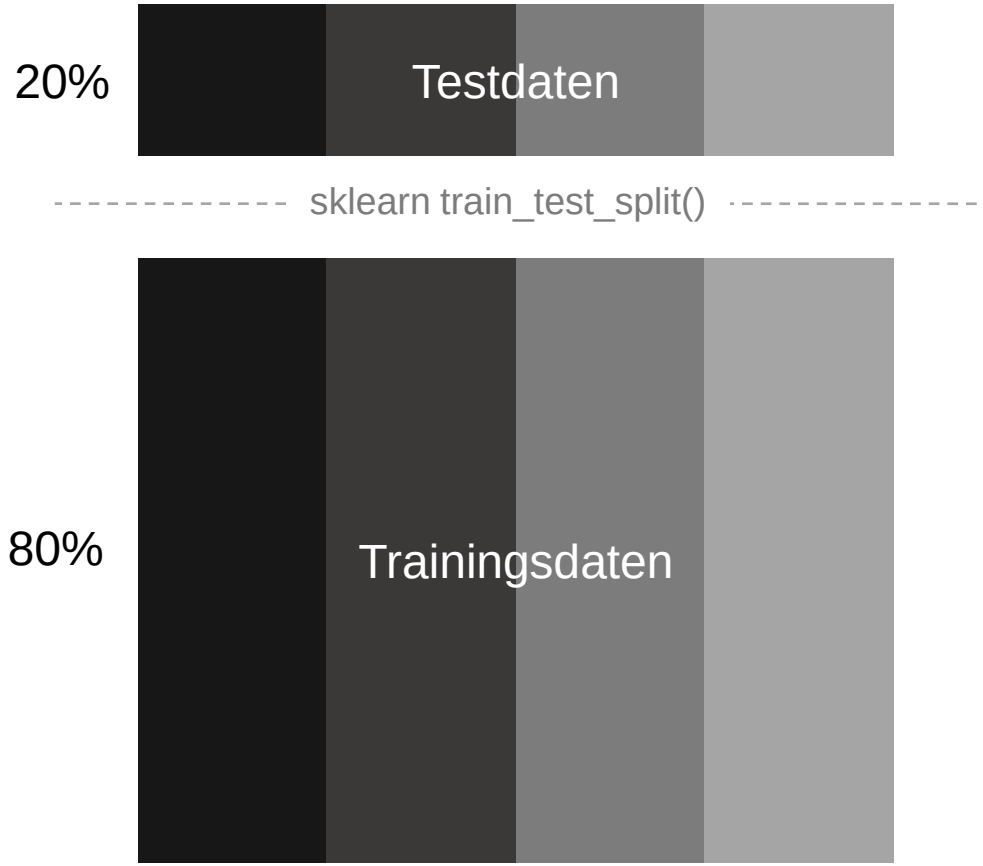
M

Datenvorverarbeitung

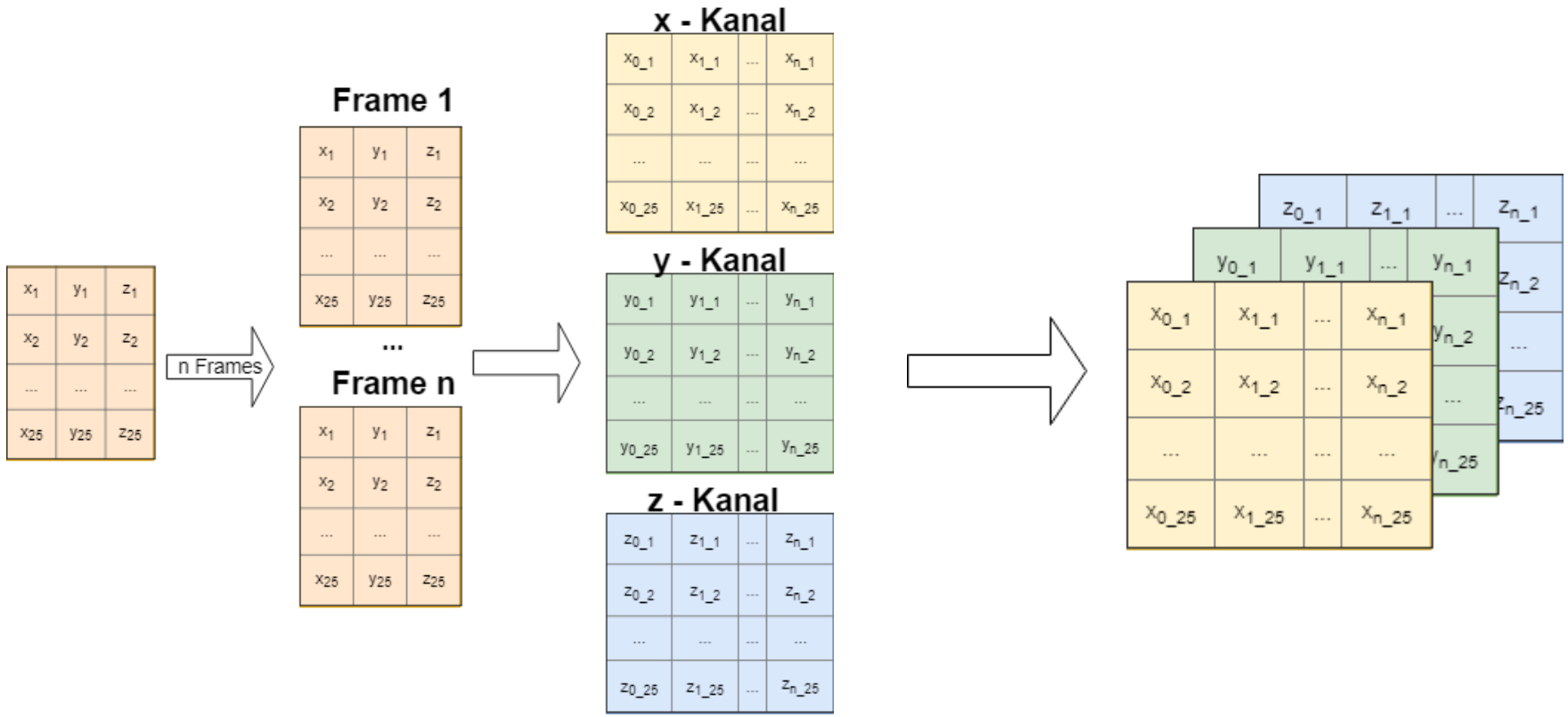
Daten teilen, Datenformat, Normalisierung

Test- und Trainingsdaten





Datenformat für CNN

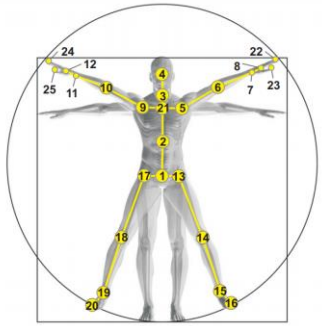


Datenformat für RNN

2-dimensionales Frame

X-Achse Y-Achse Z-Achse

```
array([[ 0.144105  ,  0.1797895 ,  3.770897  ],
       [ 0.1298895 ,  0.4353971 ,  3.716636  ],
       [ 0.1148097 ,  0.6858059 ,  3.65103   ],
       [ 0.1407793 ,  0.7899354 ,  3.626976  ],
       [ 0.00852378,  0.5821182 ,  3.602603  ],
       [-0.06197929,  0.3990757 ,  3.630781  ],
       [ 0.1899347 ,  0.4185481 ,  3.537421  ],
       [ 0.2423392 ,  0.416668  ,  3.551047  ],
       [ 0.2229546 ,  0.5959279 ,  3.709957  ],
       [ 0.2622133 ,  0.4049372 ,  3.580323  ],
       [ 0.2006394 ,  0.4397647 ,  3.380641  ],
       [ 0.215305  ,  0.4403287 ,  3.326553  ],
       [ 0.09136374,  0.174307  ,  3.716386  ],
       [ 0.05568824, -0.1528607 ,  3.809743  ],
       [ 0.04508911, -0.4464301 ,  3.96866   ],
       [ 0.05371094, -0.5288086 ,  3.935547  ],
       [ 0.1945098 ,  0.1821429 ,  3.762648  ],
       [ 0.206101  , -0.08332169,  3.867309  ],
       [ 0.1672105 , -0.4358586 ,  4.050102  ],
       [ 0.2079305 , -0.4915422 ,  3.953314  ],
       [ 0.118738  ,  0.624014  ,  3.669435  ],
       [ 0.2684598 ,  0.4491149 ,  3.511355  ],
       [ 0.2647159 ,  0.3881126 ,  3.570221  ],
       [ 0.1888318 ,  0.4359006 ,  3.304096  ],
       [ 0.1816407 ,  0.4335808 ,  3.306019  ]])
```



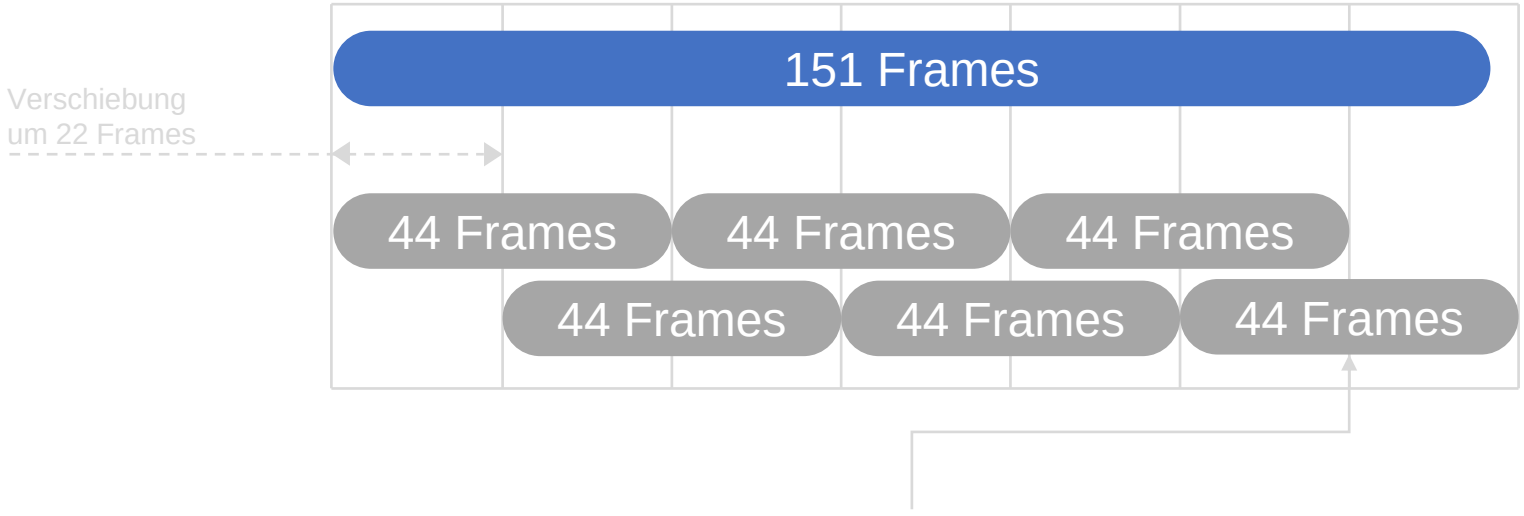
25 Gelenkpunkte

1-dimensionales Frame

$$\underbrace{[X_1, Y_1, Z_1, X_2, Y_2, Z_2, \ldots X_{25}, Y_{25}, Z_{25}]}$$

75 Gelenkpunkte in Reihe

Sliding Window



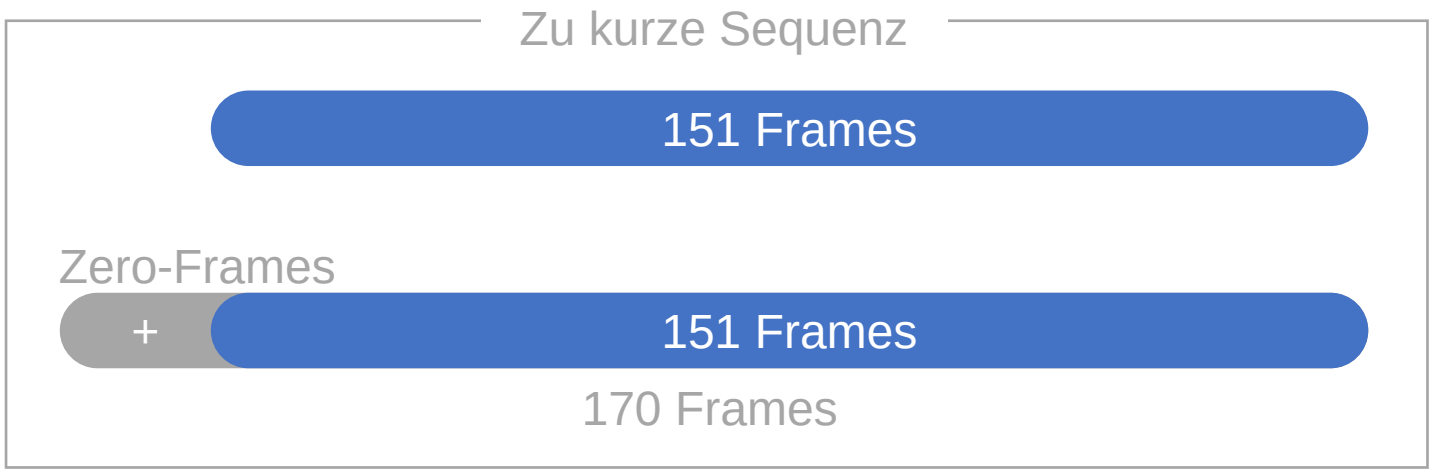
vier oder weniger Frames zu kurz:

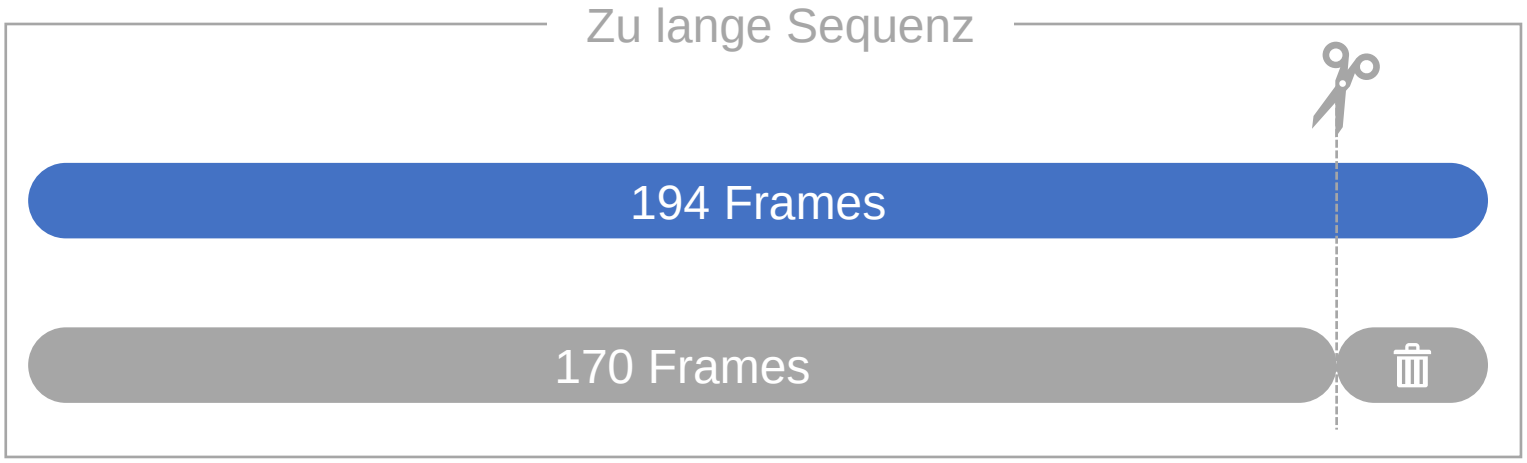
41 Frames + Standbilder anfügen

mehr als vier Frames zu kurz:

37 Frames Sequenz Löschen

Fillup mit Pre-Zero-Padding



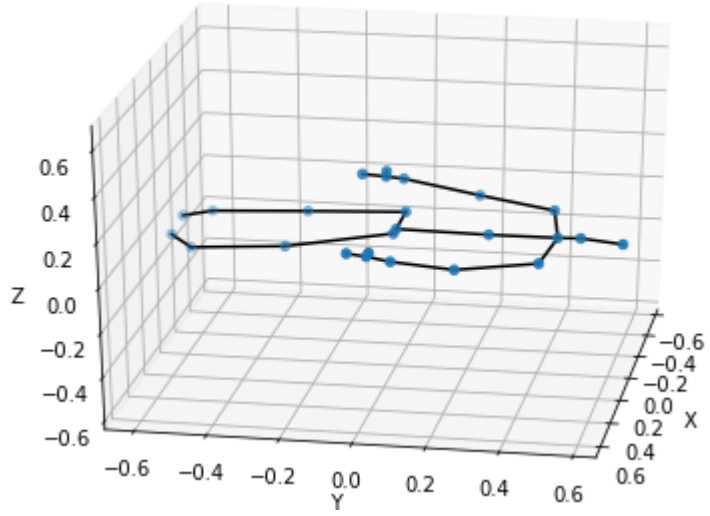


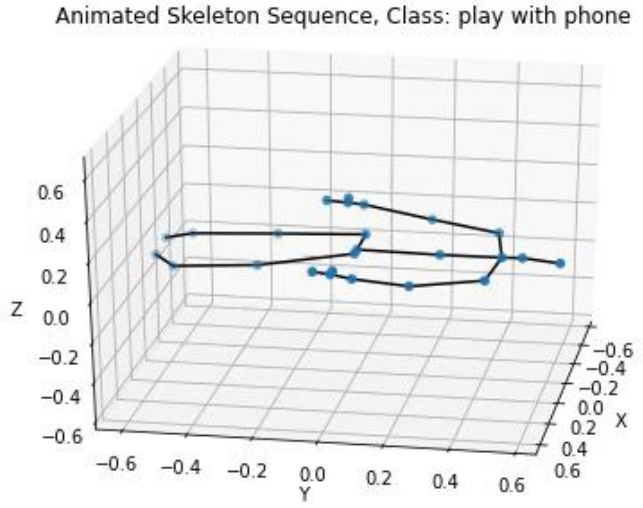
Normalisierung

Normalisierung der Daten

Als Versuch zum Vergleich zur Rotationsnormalisierung

- Übertragung der Gelenkpunkte in ein Körperkoordinatensystem.
- Als Basisvektoren dienen Hüfte und Wirbelsäule.
- Neuberechnung des Koordinatensystems in jedem Frame

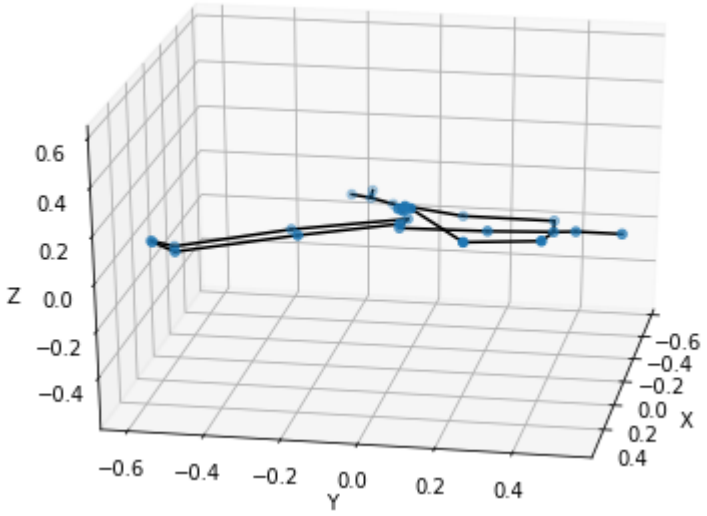


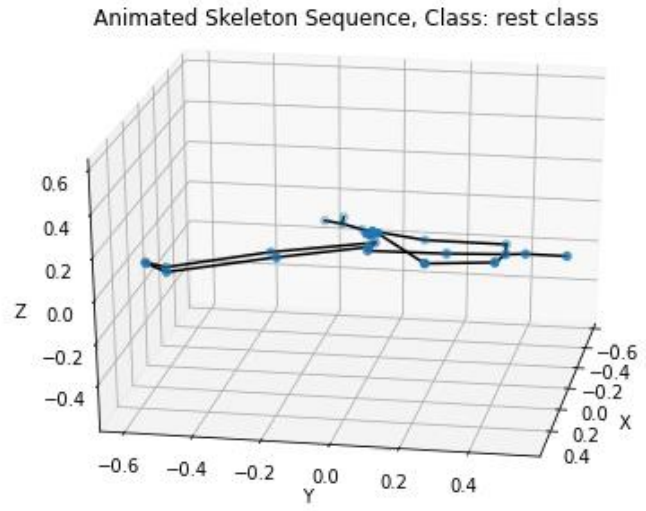


Normalisierung der Daten

Erweiterung der Normalisierung auf Körperkoordinatensystem:

- Körperkoordinatensystem wird nur im ersten Frame aufgestellt
- Folgeframes werden in Koordinatensystem des ersten Frame umgewandelt
- Erhalten der Räumlichen Bewegung





M

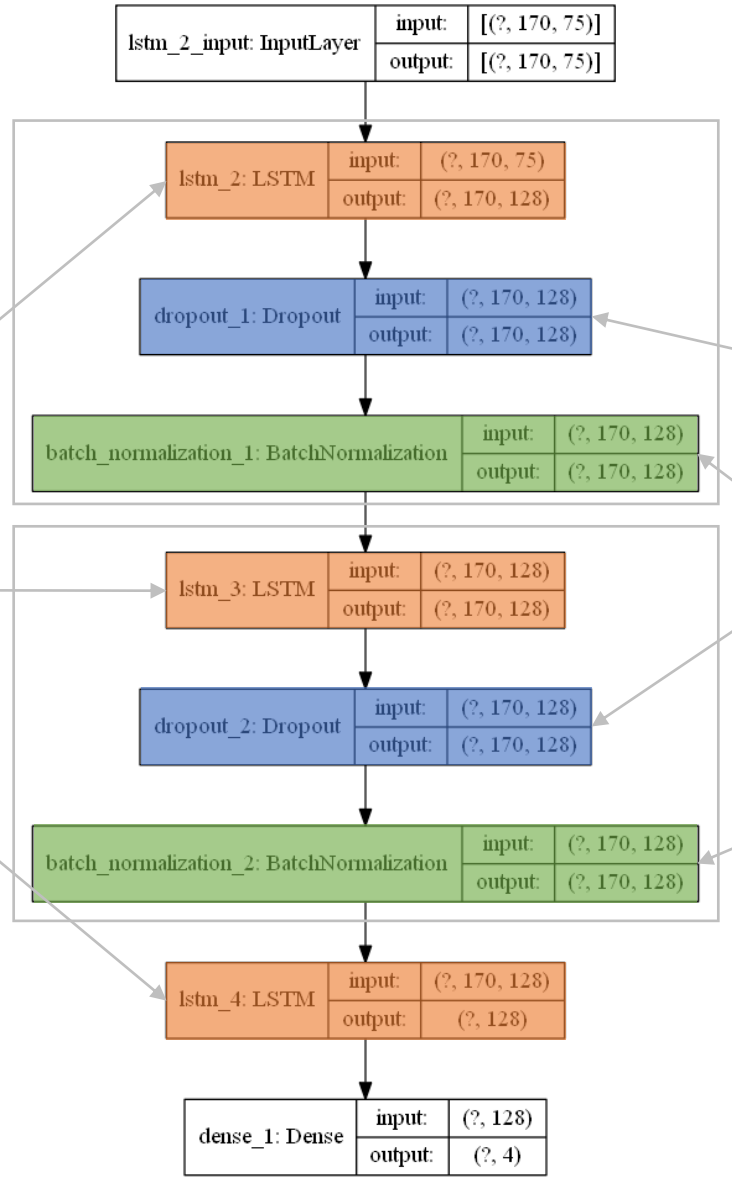
Rekurrente Neuronale Netze

Netzarchitektur und Ergebnisse

RNN Netzstruktur

128 Zellen pro Schicht

Parameter Anzahl: 237.060



LSTM Schichten

Dropout:
gegen Overfitting

Batch Normalization:
beschleunigt Lernprozess

Optimzer: Adam

Loss:
sparse_categorical_crossentropy

Methoden zur Evaluierung

Standard Accuracy

Konfusions Matrix

Precision, Recall und F1-Score

ROC - Area under Curve

Vorverarbeitungs-Ergebnisse

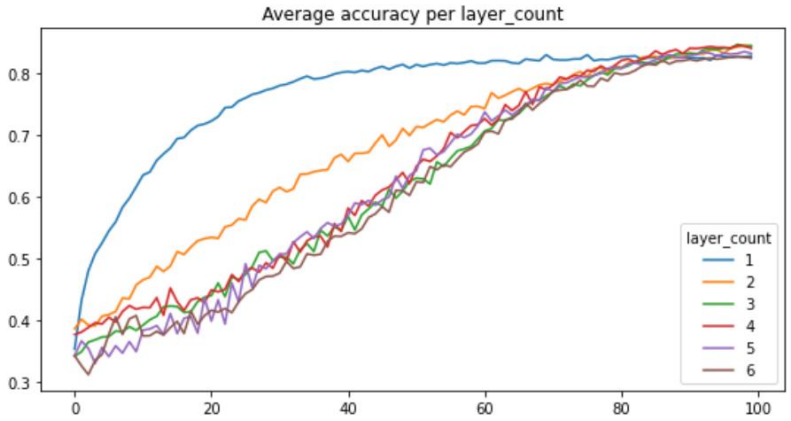
	Ausprägung	Genauigkeit	Differenz
Normalisierung	True	87,15%	4,42%
	False	82,73%	
Dimensionen	3D	87,15%	7,04%
	2D	80,11%	
Methode	Fillup	87,15%	10,44%
	Sliding	76,71%	

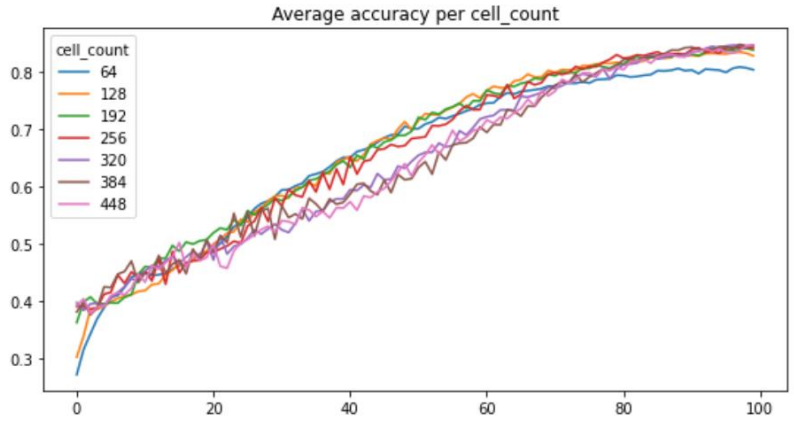
Netzarchitekturen

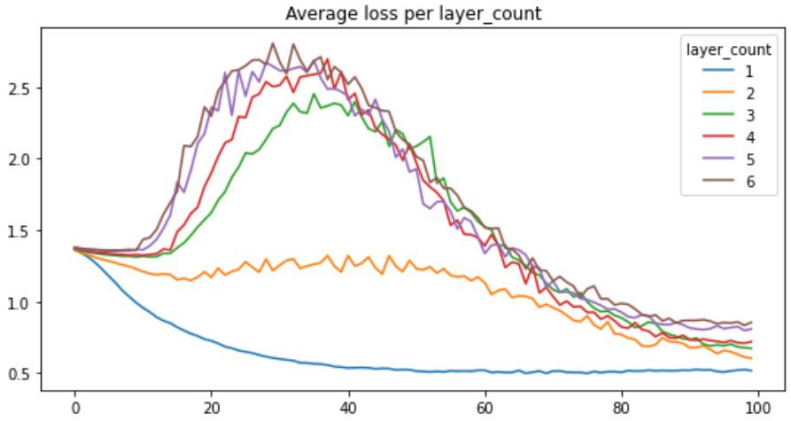
		Zellen pro Schicht						
		64	128	192	256	320	384	448
Schichten	1	78,87%	82,73%	83,15%	85,22%	85,50%	86,05%	84,25%
	2	82,04%	87,15%	84,25%	86,33%	84,67%	85,50%	85,36%
	3	82,04%	83,98%	85,50%	84,39%	86,46%	85,64%	85,64%
	4	82,46%	84,12%	86,19%	85,22%	84,94%	86,05%	86,19%
	5	81,63%	83,29%	84,81%	84,81%	85,22%	84,67%	83,84%
	6	80,80%	83,43%	82,87%	84,81%	84,25%	83,98%	84,67%

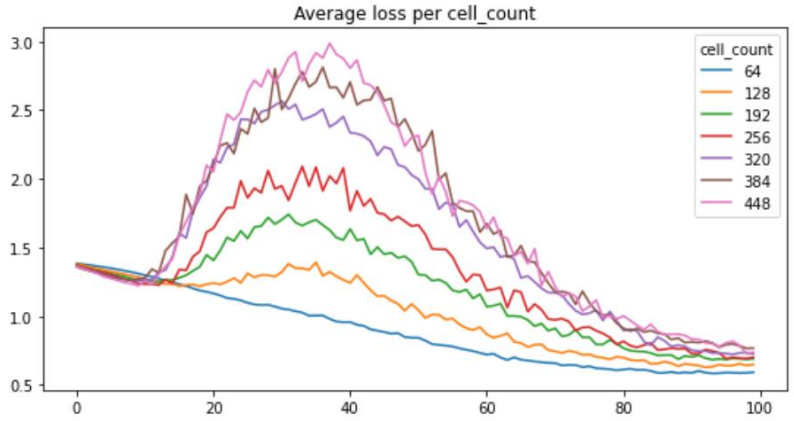
→ 64 Zellen reichen nicht aus
→ Weniger ist besser
→ Große Netze brauchen deutlich länger zum Lernen

Auswirkungen der Netzstruktur auf Geschwindigkeit









→ Je weniger Schichten desto schnelleres Lernen
→ Anzahl der Zellen hat keinen Einfluss auf Geschwindigkeit
→ Zu hohe Komplexität führt zu Overfitting

RNN Metriken

„Play with phone“ und „taking a selfie“ klar unterscheidbar.

Werden oft nur mit „phone call“ verwechselt.

		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	154	15	2	10
	play with phone	9	169	2	1
	taking a selfie	9	3	163	6
	rest class	15	14	7	145

„phone call“ und „play with phone“ oft falsch zugeordnet

	precision	recall	f1-score	support
phone call	82%	85%	84%	181
play with phone	84%	93%	88%	181
taking a selfie	94%	90%	92%	181
rest_class	90%	80%	85%	181

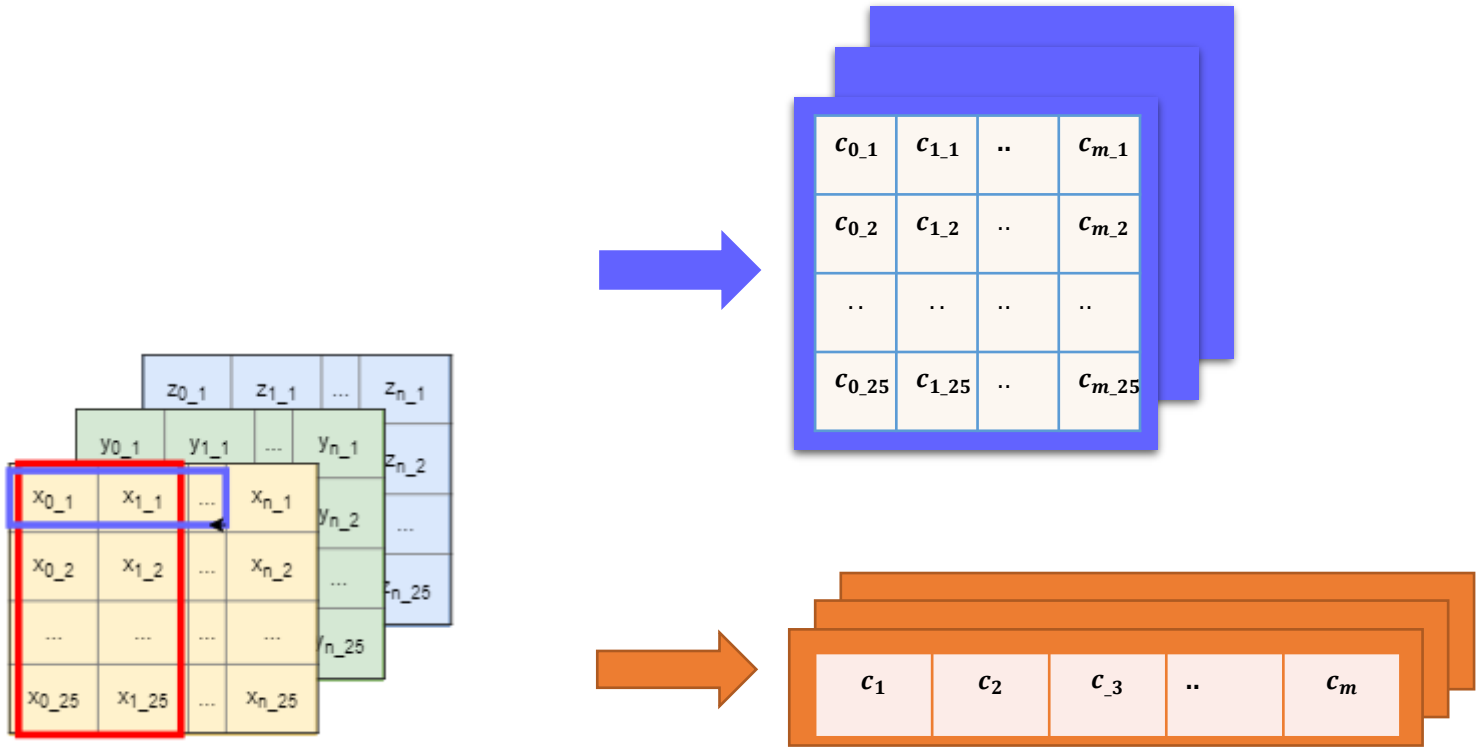
Nur geringer Anteil „Rest-Klasse“ richtig vorhergesagt

P

Convolutional Neuronale Netze

Netzarchitektur und Ergebnisse

Verschiedene Faltungskerne



In Blau

- Wir Falten jedes Gelenk individuell über die Zeit
  => Narrow Convolution

In Rot

Wir Falten alle Gelenke zusammen über die Zeit

=> Wide Convolution

Modellstrukturen

Layer (type)	Output Shape	Param #
conv2d_20 (Conv2D)	(None, 166, 25, 32)	512
max_pooling2d_10 (MaxPooling)	(None, 162, 25, 32)	0
conv2d_21 (Conv2D)	(None, 158, 25, 32)	5152
max_pooling2d_11 (MaxPooling)	(None, 154, 25, 32)	0
flatten_8 (Flatten)	(None, 123200)	0
dense_36 (Dense)	(None, 100)	12320100
dropout_12 (Dropout)	(None, 100)	0
dense_37 (Dense)	(None, 50)	5050
dropout_13 (Dropout)	multiple	0
dense_38 (Dense)	(None, 20)	1020
dense_39 (Dense)	(None, 20)	420
dense_40 (Dense)	(None, 4)	84

Total params: 12,332,338
Trainable params: 12,332,338
Non-trainable params: 0

Modell 1

- 2 narrow Covolution Layer
- 2 narrow Pooling Layer
- Faltung und Pooling jeweils über 5 Frames hinweg
- Stride bei Faltung: 1
- Stride bei Pooling: 1
- .35 Dropout nach erstem Fully Connected Layer
- .25 Dropout in den restlichen Layern

Modellstrukturen

Layer (type)	Output Shape	Param #
conv2d_22 (Conv2D)	(None, 166, 25, 32)	512
max_pooling2d_12 (MaxPooling)	(None, 165, 25, 32)	0
conv2d_23 (Conv2D)	(None, 163, 25, 32)	3104
max_pooling2d_13 (MaxPooling	(None, 162, 25, 32)	0
conv2d_24 (Conv2D)	(None, 160, 25, 32)	3104
max_pooling2d_14 (MaxPooling)	(None, 159, 25, 32)	0
flatten_9 (Flatten)	(None, 127200)	0
dense_41 (Dense)	(None, 100)	12720100
dropout_14 (Dropout)	(None, 100)	0
dense_42 (Dense)	(None, 50)	5050
dropout_15 (Dropout)	multiple	0
dense_43 (Dense)	(None, 20)	1020
dense_44 (Dense)	(None, 4)	84

Total params: 12,732,974
Trainable params: 12,732,974
Non-trainable params: 0

Modell 2

- 3 narrow Covolution Layer
- 3 narrow Pooling Layer
- Faltung einmal über 5, zweimal über 3 Frames
- Pooling über 2 Frames
- Stride bei Faltung: 1
- Stride bei Pooling: 1
- .35 Dropout nach erstem Fully Connected Layer
- .25 Dropout in den restlichen Layern

Modellstrukturen

```
_________________________________________________________________
Layer (type) Output Shape Param #
=================================================================
conv2d_25 (Conv2D) (None, 81, 25, 32) 992
_________________________________________________________________
conv2d_26 (Conv2D) (None, 39, 1, 32) 128032
_________________________________________________________________
flatten_10 (Flatten) (None, 1248) 0
_________________________________________________________________
dense_45 (Dense) (None, 100) 124900
_________________________________________________________________
dropout_16 (Dropout) multiple 0
_________________________________________________________________
dense_46 (Dense) (None, 50) 5050
_________________________________________________________________
dense_47 (Dense) (None, 20) 1020
_________________________________________________________________
dense_48 (Dense) (None, 20) 420
_________________________________________________________________
dense_49 (Dense) (None, 4) 84
=================================================================
Total params: 260,498
Trainable params: 260,498
Non-trainable params: 0
```

Idee:

- Erste Faltung extrahieren für jedes Gelenk Features
- Zweite Schicht faltet über extrahierte Features für jedes Gelenk hinweg
  => Geringere Parameteranzahl

Modell 3

- 1 narrow Covolution Layer.
  - Breite von 10
  - Stride von (2,1)
- 1 wide Covolution Layer
  - Breite von 5
  - Stride von (2,1)
- Keine Pooling Layer
- .2 Dropout in den Fully Connected Layern

Modellstrukturen

```
_________________________________________________________________
Layer (type) Output Shape Param #
=================================================================
conv2d_37 (Conv2D) (None, 81, 25, 32) 992
_________________________________________________________________
conv2d_38 (Conv2D) (None, 39, 25, 16) 2576
_________________________________________________________________
conv2d_39 (Conv2D) (None, 18, 1, 32) 64032
_________________________________________________________________
flatten_15 (Flatten) (None, 576) 0
_________________________________________________________________
dense_68 (Dense) (None, 100) 57700
_________________________________________________________________
dropout_23 (Dropout) multiple 0
_________________________________________________________________
dense_69 (Dense) (None, 50) 5050
_________________________________________________________________
dense_70 (Dense) (None, 20) 1020
_________________________________________________________________
dense_71 (Dense) (None, 4) 84
=================================================================
Total params: 131,454
Trainable params: 131,454
Non-trainable params: 0
```

Idee:

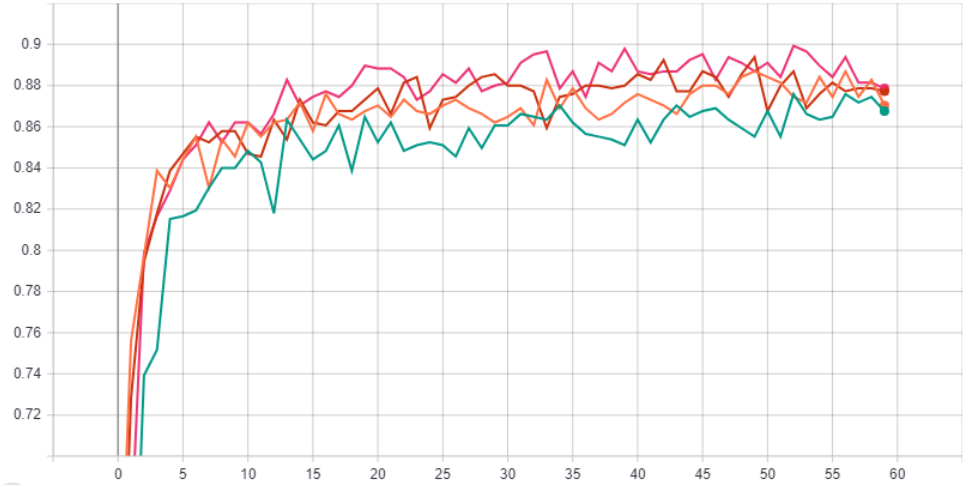
- Bessere Feature Extrahierung
- Geringere Parameteranzahl

Modell 4

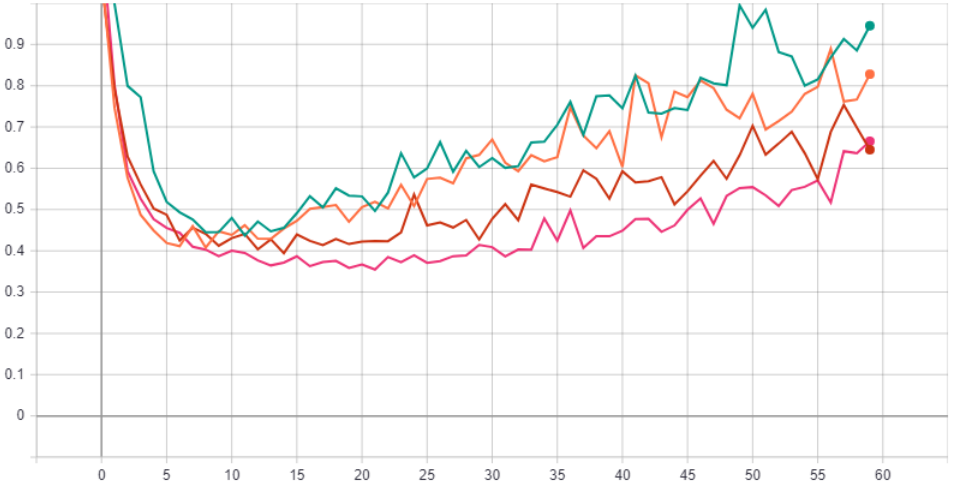
- 2 narrow Covolution Layer.
  - Einmal Breite von 10
  - Einmal Breite von 5
  - Stride von (2,1)
- 1 wide Covolution Layer
  - Breite von 5
  - Stride von (2,1)
- Keine Pooling Layer
- Ein Fully Connected Layer weniger
- .2 Dropout in den Fully Connected Layern

Vergleich der Netzstrukturen

Validierungs-Genauigkeit



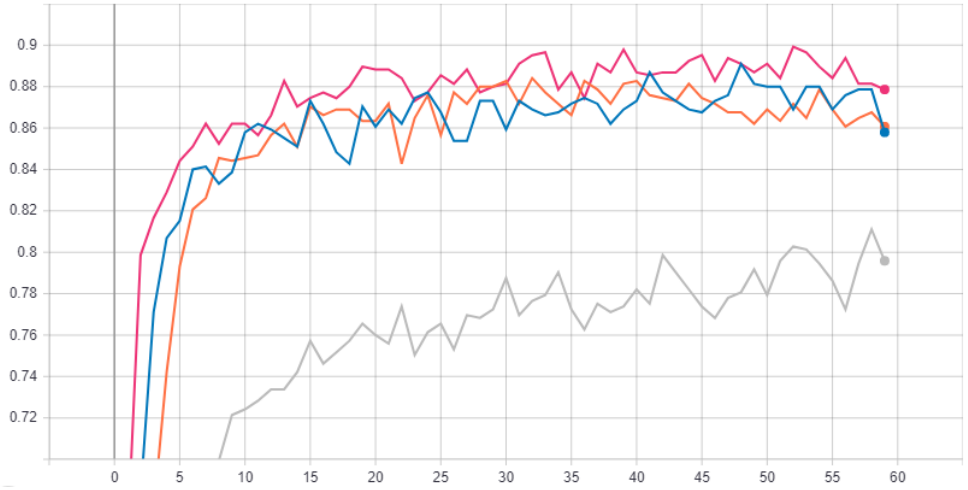
Validierungs-Loss



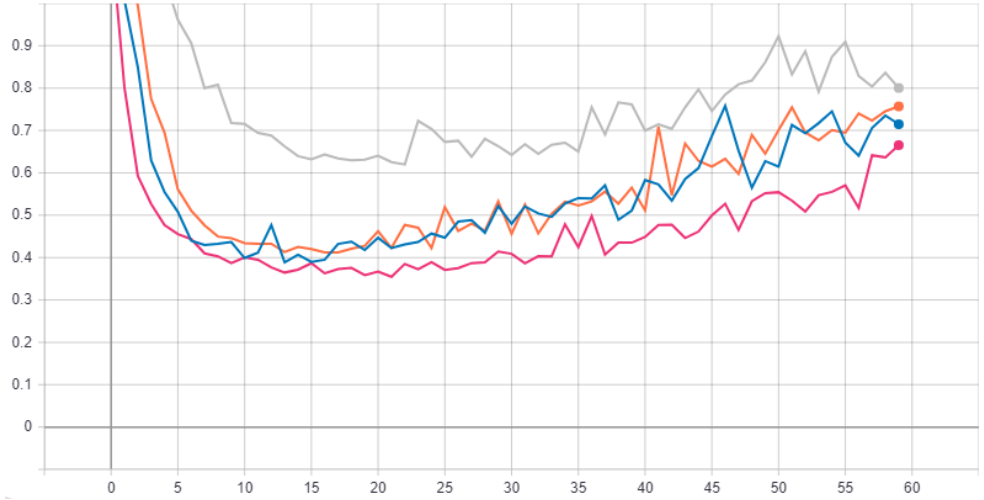
- rotation_normalized_model_regularized_1/validation
- rotation_normalized_model_regularized_2/validation
- rotation_normalized_model_regularized_3/validation
- rotation_normalized_model_regularized_4/validation

Vergleich der genutzten Daten

Validierungs-Genauigkeit



Validierungs-Loss



- advanced_normalized_model_regularized/validation
- normalized_model_regularized_4/validation
- rotation_normalized_model_regularized_4/validation
- unnormalized_model_regularized_4/validation

Auf Trainingsdaten

		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	721	1	0	3
	play with phone	0	725	0	0
	taking a selfie	2	1	717	5
	rest class	2	1	0	722

- **Accuracy:** 0.9948
- **Precision, Recall und F1-score für die Klassen:**

class	precision	recall	f1-score	support
phone call	0.99	0.99	0.99	725
play with phone	1.00	1.00	1.00	725
taking a selfie	1.00	0.99	0.99	725
rest class	0.99	1.00	0.99	725

- **Precision Recall und F1-score im Schnitt:**

Average Type	precision	recall	f1-score	support
micro avg	0.99	0.99	0.99	2900
macro avg	0.99	0.99	0.99	2900
weighted avg	0.99	0.99	0.99	2900
samples avg	0.99	0.99	0.99	2900

- **ROC AUC Score**: 0.9966

Auf Testdaten

		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	159	6	3	14
	play with phone	8	168	1	4
	taking a selfie	6	1	160	14
	rest class	4	1	6	165

- **Accuracy:** 0.8993
- **Precision, Recall und F1-score für die Klassen:**

class	precision	recall	f1-score	support
phone call	0.90	0.87	0.89	182
play with phone	0.93	0.93	0.93	181
taking a selfie	0.94	0.88	0.91	181
rest class	0.84	0.91	0.87	181

- **Precision Recall und F1-score im Schnitt:**

Average Type	precision	recall	f1-score	support
micro avg	0.90	0.90	0.90	725
macro avg	0.90	0.90	0.90	725
weighted avg	0.90	0.90	0.90	725
samples avg	0.90	0.90	0.90	725

- **ROC AUC Score**: 0.9329

Cross View Test

		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	280	12	4	20
	play with phone	5	297	6	8
	taking a selfie	1	3	306	6
	rest class	21	6	10	239

- **Accuracy:** 0.9167
- **Precision, Recall und F1-score für die Klassen:**

class	precision	recall	f1-score	support
phone call	0.91	0.89	0.90	316
play with phone	0.93	0.94	0.94	316
taking a selfie	0.94	0.97	0.95	316
rest class	0.88	0.87	0.87	276

- **Precision Recall und F1-score im Schnitt:**

Average Type	precision	recall	f1-score	support
micro avg	0.92	0.92	0.92	1224
macro avg	0.92	0.92	0.91	1224
weighted avg	0.92	0.92	0.92	1224
samples avg	0.92	0.92	0.92	1224

- **ROC AUC Score**: 0.9437

Cross Subject Test

		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	230	15	8	22
	play with phone	7	256	5	7
	taking a selfie	5	4	257	10
	rest class	25	13	10	228

- **Accuracy:** 0.8811
- **Precision, Recall und F1-score für die Klassen:**

class	precision	recall	f1-score	support
phone call	0.86	0.84	0.85	275
play with phone	0.89	0.93	0.91	275
taking a selfie	0.92	0.93	0.92	276
rest class	0.85	0.83	0.84	276

- **Precision Recall und F1-score im Schnitt:**

Average Type	precision	recall	f1-score	support
micro avg	0.88	0.88	0.88	1102
macro avg	0.88	0.88	0.88	1102
weighted avg	0.88	0.88	0.88	1102
samples avg	0.88	0.88	0.88	1102

- **ROC AUC Score**: 0.9208

P

Vergleich der Modelle

RNN vs. CNN

RNN vs. CNN

RNN

- **Parameteranzahl:** 237.060
- **Accuracy:** 0.8715
- **Precision, Recall und F1-score für die Klassen:**

	precision	recall	f1-score	support
phone call	82%	85%	84%	181
play with phone	84%	93%	88%	181
taking a selfie	94%	90%	92%	181
rest_class	90%	80%	85%	181

- **Precision Recall und F1-score im Schnitt:**

Average Type	precision	recall	f1-score	support
macro avg	87%	87%	87%	181
weighted avg	87%	87%	87%	181

- **ROC AUC Score**: 0.9723

CNN

- **Parameteranzahl:** 260.498
- **Accuracy:** 0.8993
- **Precision, Recall und F1-score für die Klassen:**

	precision	recall	f1-score	support
phone call	90%	87%	89%	182
play with phone	93%	93%	93%	181
taking a selfie	94%	88%	91%	181
rest_class	84%	91%	87%	181

- **Precision Recall und F1-score im Schnitt:**

Average Type	precision	recall	f1-score	support
micro avg	90%	90%	90%	725
macro avg	90%	90%	90%	725
weighted avg	90%	90%	90%	725
samples avg	90%	90%	90%	725

- **ROC AUC Score**: 0.9329

Konfusions Matrix

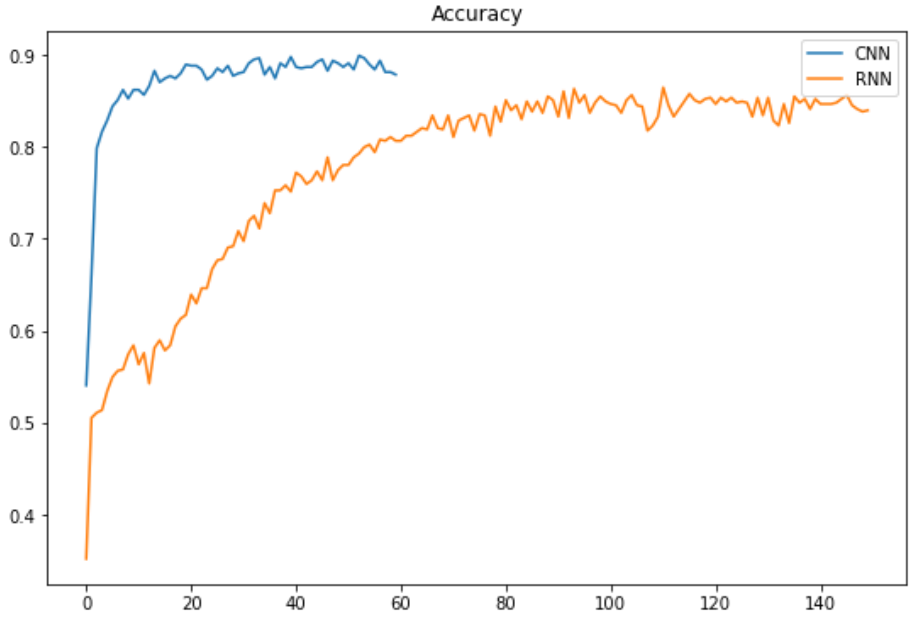
RNN

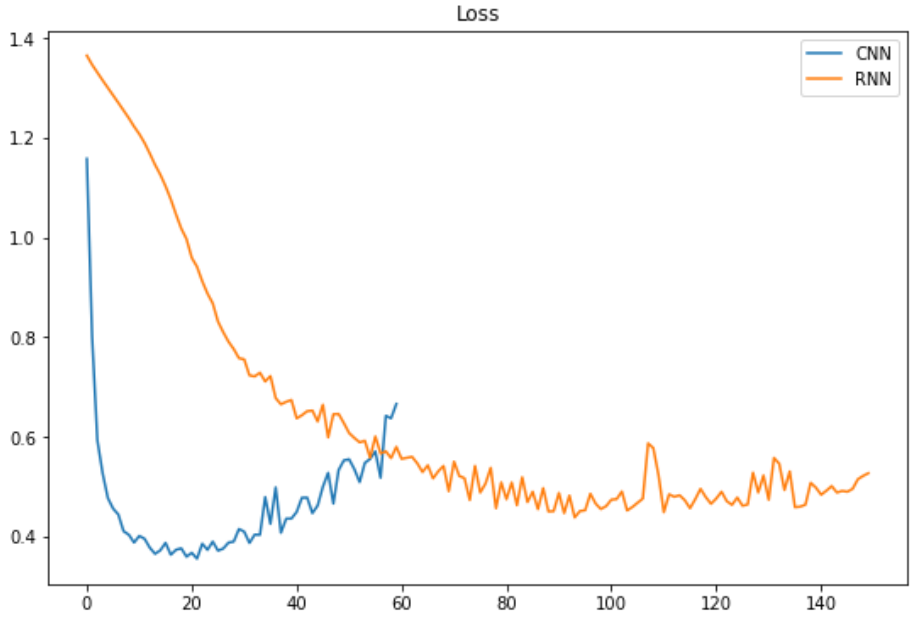
		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	154	15	2	10
	play with phone	9	169	2	1
	taking a selfie	9	3	163	6
	rest class	15	14	7	145

CNN

		Vorhersagen			
		phone call	play with phone	taking a selfie	rest class
Eigentliche Klasse	phone call	159	6	3	14
	play with phone	8	168	1	4
	taking a selfie	6	1	160	14
	rest class	4	1	6	165

Vergleich zwischen CNN und RNN



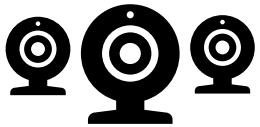


Spezifische Anwendungsfälle der Bewegungsklassifizierung

Cross Subject

Aktionen in Trainings- und Validierungsdaten von jeweils unterschiedlichen Darstellern.

Generalisiert das Modell gut oder trainiert es Merkmale der Darsteller?

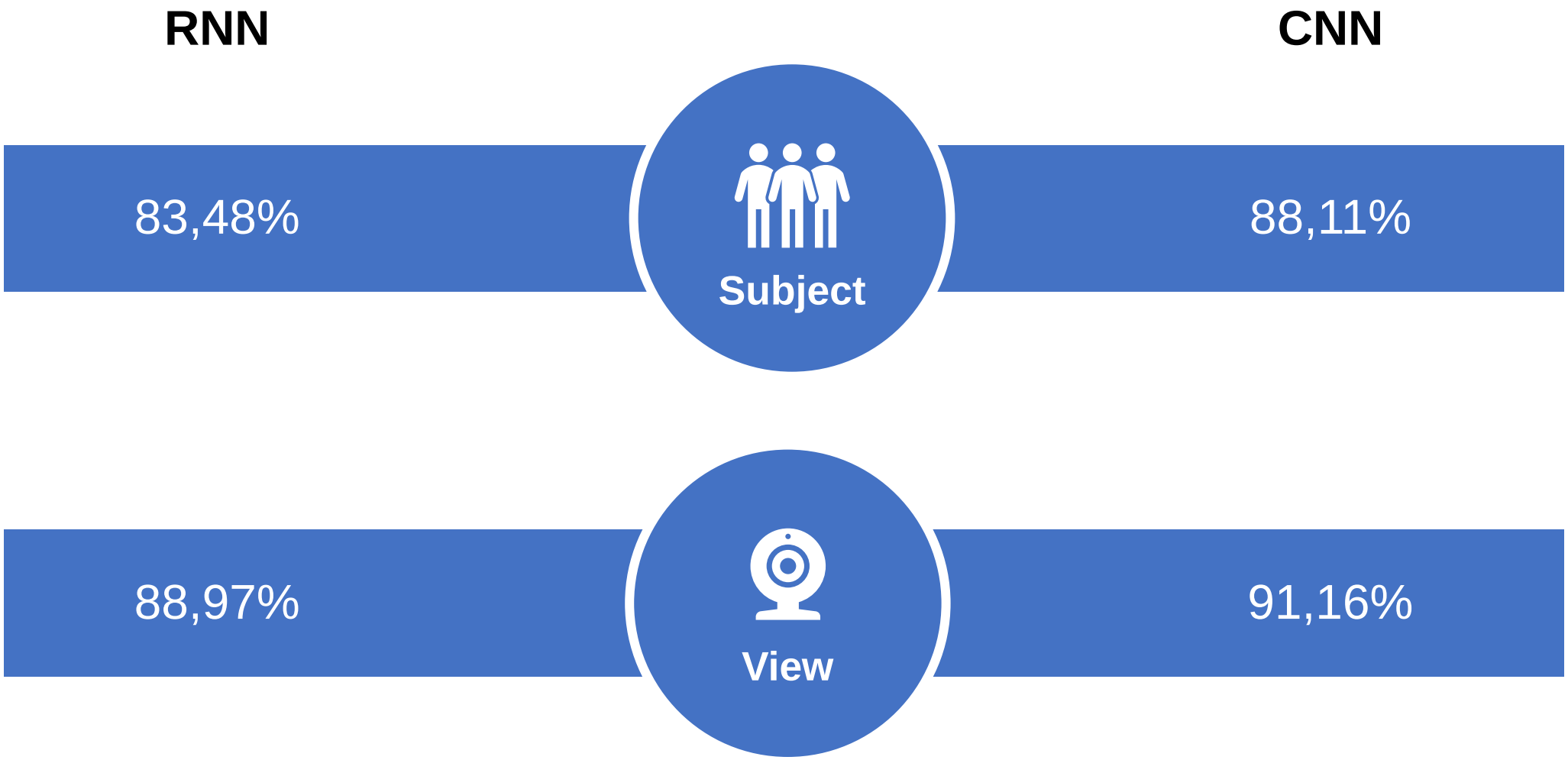


Cross View

Trainingsdaten aus verschiedenen Winkeln, Testdaten entstammen einem unbekannten Blickwinkel.

Wie gut verallgemeinert Modell unabhängig vom Blickwinkel?

Cross View & Cross Subject



Fazit

- Richtige Vorverarbeitung wichtig
- CCN > RNN
- Restklasse erschwert Klassifizierungsaufgabe deutlich

Nächste Schritte

- Cross Validation um Algorithmen noch besser zu vergleichen
- Falsch klassifizierte Fälle genauer untersuchen
- Algorithmen auf eigenen Daten testen (Kinect)
- Transformer