INTERNATIONALE HOCHSCHULE

Studiengang:	Angewandte Künstliche Intelligenz
Modul:	DLBAIPCV01_D - Projekt: Computer Vision
Tutor:	Ahmet Nasri
Semester:	4

Erkennung von Gebärdensprache für Kommunikationshilfe

Datum:	10.02.2026
Name:	Götz Möglinger
Adresse:	Werneckstr. 35
	80802 München
E-Mail:	goetz.moeglinger@iu-study.org
Matrikelnummer:	IU14106219

Inhalt

1 Einleitung

Die Deutsche Gebärdensprache (DGS) ist laut Becker & Jaeger eine natürliche Sprache, die im sozialen Miteinander gehörloser Menschen auf natürliche Weise entstanden ist. Sie ist auf sprachstruktureller Ebene wie die deutsche Sprache ebenfalls systematisch aufgebaut und ein zentrales Mittel sozialer Verständigung (2019, S. 22). Wichtige Bestandteile der sogenannten Gebärden sind Bewegungen nicht nur der Hände, sondern auch der Augenbrauen oder des Mundes sowie die Blickrichtung oder Neigung des Kopfes und des Oberkörpers (Becker & Jaeger, 2019, S. 22). Gerade diese multimodale Struktur macht die DGS im Alltag für Hörende ohne entsprechende Kompetenzen schwer zugänglich.

Daher könnte eine (simultane) Übersetzung dieser sehr komplexen und vielfältigen Sprache eine Brücke zwischen den Gehörlosen und den Hörenden bei der Zusammenarbeit im Alltag schaffen oder helfen Barrieren abzubauen. Unterstützt wird diese These von Pilling, die unterstreicht, dass Teilhabe und Zugehörigkeit durch den Abbau sozialer Kommunikationsbarrieren gestärkt werden sollen (2022, S. 45). Wenn Behinderung wesentlich durch gesellschaftliche Bedingungen entsteht, adressiert eine Übersetzungslösung genau diese Barriere – insbesondere in Situationen, in denen ohne sprachliche Vermittlung Verständigung erschwert oder verhindert wird (Pilling, 2022, S. 51).

Vor diesem Hintergrund stellt die Entwicklung datengetriebener, visueller Erkennungsverfahren für Gebärdensprache ein relevantes Anwendungsfeld moderner Computer-Vision- und Deep-Learning-Methoden dar.

1.1 Zielsetzung

Ziel des Projekts ist die prototypische Entwicklung eines Computer-Vision-Systems zur Erkennung von Gebärdensprache aus Videosequenzen auf Basis von Hand-Landmark-Features. Der Fokus liegt dabei nicht auf der vollständigen Übersetzung natürlicher Sprache, sondern auf der Klassifikation von Gebärden, sogenannten Glosses, als Zwischenschritt einer möglichen Übersetzungskette.

Ein zentrales Ziel der Arbeit ist die Untersuchung des Einflusses unterschiedlicher neuronaler Modellarchitekturen auf die Erkennungsleistung. Dazu werden zwei Modelle unter identischen Daten- und Vorverarbeitungsbedingungen verglichen:

(1) ein LSTM-basierter Window-Klassifikator, welcher feste, überlappende Zeitfenster einer Merkmalssequenz mit einem rekurrenten neuronalen Netz verarbeitet und jedem Fenster eine einzelne Klasse zuordnet, als Baseline, sowie
(2) ein hybrides CNN-LSTM-Modell, das lokale zeitliche Muster innerhalb eines festen Fensters mittels Convolutional Neural Networks extrahiert und diese anschließend mit einem LSTM zeitlich modelliert.

Beide Modelle nutzen das gleiche Inputformat und die gleiche Trainingspipeline. Die Arbeit verfolgt damit explizit einen methodischen Vergleichsansatz. Durch die bewusste Beschränkung auf Hand-

Landmark-Sequenzen und klar definierte Modellarchitekturen soll ein nachvollziehbarer methodischer Einblick in den Einsatz und die Wirkung zentraler Deep-Learning-Methoden im Kontext der Gebärdenspracherkennung gegeben werden.

1.2 Forschungsfragen

Die automatische Verarbeitung von Gebärdensprachdaten stellt hohe Anforderungen an die Modellierung zeitlicher Abfolgen visueller Merkmale. Unterschiedliche Deep-Learning-Architekturen adressieren diese Herausforderung auf verschiedene Weise. Während rekurrente neuronale Netze (RNNs) wie LSTM-Modelle primär zur Erfassung zeitlicher Abhängigkeiten eingesetzt werden, ermöglichen Convolutional Neural Networks (CNNs) die Extraktion lokaler Muster innerhalb begrenzter zeitlicher Abschnitte einer Sequenz. In hybriden CNN-LSTM-Architekturen werden beide Ansätze kombiniert, um sowohl lokale als auch sequenzielle Informationen zu nutzen.

Vor diesem Hintergrund stellt sich die Frage, ob ein solcher Hybridansatz bei einer reduzierten, abstrakten Datenrepräsentation – konkret Hand-Landmark-Sequenzen – einen messbaren Vorteil gegenüber einem reinen LSTM-Modell bietet. Da komplexe Teilprobleme wie eine explizite Sign-Segmentierung oder der Einsatz vollständiger Bilddaten den Umfang der Arbeit deutlich erhöhen würden, werden diese bewusst ausgeklammert.

Aus der beschriebenen Problemstellung ergibt sich folgende zentrale Forschungsfrage: Erzielt ein hybrides CNN-LSTM-Modell bei identischer Datenpipeline und identischer Hand-Landmark-Repräsentation eine höhere Erkennungsgenauigkeit als ein reines LSTM-Modell? Die Beantwortung dieser Frage erfolgt durch einen direkten Vergleich beider Modellarchitekturen unter gleichen Trainings- und Evaluationsbedingungen.

1.3 Geplantes Vorgehen und Aufbau des Berichts

Ausgehend von der definierten Zielsetzung und der Forschungsfrage wird zunächst der Datensatz *RWTH-PHOENIX-Weather 2014 T* ausgewählt und für die Modellierung vorbereitet. Dabei erfolgt eine Reduktion auf die für das Projekt relevanten Videosequenzen sowie die zugehörigen Gloss-Annotationen.

Im nächsten Schritt wird ein Subsystem zur Erkennung und Verfolgung der Hände in den Videosequenzen implementiert. Dieses Tracking basiert auf MediaPipe und liefert pro Frame normierte Hand-Landmark-Informationen, aus denen feste Merkmalsvektoren gebildet werden. Diese dienen als Eingabedaten für die neuronalen Modelle, während die eigentliche Merkmalsextraktion innerhalb der CNN- bzw. LSTM-Schichten erfolgt.

Darauf aufbauend werden zwei Modellarchitekturen implementiert und trainiert: ein LSTM-basiertes Basismodell sowie ein hybrides CNN-LSTM-Modell. Beide Modelle werden mit identischen Trainingsdaten, Hyperparametern und Evaluationsmetriken trainiert, um einen fairen Vergleich zu gewährleisten. Die Evaluation umfasst neben der Klassifikationsgenauigkeit auch qualitative

Analysen sowie Messungen der Inferenzlatenz im Rahmen einer prototypischen Anwendungspipeline.

2 Methodik und technische Umsetzung

2.1 Datensatz und methodische Abgrenzung

Für die prototypische Umsetzung wird der Datensatz *RWTH-PHOENIX-Weather 2014 T* verwendet. Der Datensatz enthält Aufzeichnungen von Wetter- und Nachrichtensendungen des Senders PHOENIX, die von professionellen Dolmetscherinnen und Dolmetschern in Deutsche Gebärdensprache übersetzt wurden. Ein Teilkorpus von 386 Ausgaben wurde mit Gloss-Annotationen versehen, wobei jedem Video eine vollständige Gloss-Sequenz zugeordnet ist.

Der Datensatz ist in der Forschung zur automatischen Gebärdensprachverarbeitung weit verbreitet und wird häufig als Benchmark für den Vergleich unterschiedlicher Modellarchitekturen eingesetzt. Überblicksarbeiten zeigen, dass auf diesem Datensatz zahlreiche rekurrente, konvolutionelle und hybride Deep-Learning-Ansätze untersucht und miteinander verglichen wurden (Koller, 2020, S. 1).

Aufgrund des Umfangs des Datensatzes sowie begrenzter Rechenressourcen wird in dieser Arbeit ein repräsentatives Teilkorpus verwendet. Ziel ist nicht die vollständige Ausschöpfung des Datensatzes, sondern die methodische Untersuchung unterschiedlicher Modellarchitekturen unter kontrollierten Bedingungen.

Weiterhin wird der Fokus bewusst auf manuelle Komponenten der Gebärdensprache gelegt. Nicht-manuelle Merkmale wie Mimik, Mundbild oder Oberkörperbewegungen werden nicht berücksichtigt. Diese Einschränkung stellt eine bewusste Abstraktion dar, um die Komplexität der Pipeline zu reduzieren und den Einfluss der Modellarchitektur isoliert untersuchen zu können.

2.2 Feature-Extraktion aus Videodaten

Zur Umwandlung der Videodaten in ein modellgeeignetes Zahlenformat wird pro Frame zunächst die Handpose erfasst und anschließend in einen festen Merkmalsvektor überführt. Für die Detektion und Verfolgung der Hände wird MediaPipe Hands verwendet, das pro Frame 21 dreidimensionale Hand-Landmarks sowie eine Zuordnung zu linker bzw. rechter Hand liefert (Lugaresi et al., 2019, S. 6f).

Um ein konsistentes Eingabeformat zu gewährleisten, werden die Landmark-Koordinaten beider Hände in einer definierten Reihenfolge zu einem Feature-Vektor fester Länge zusammengeführt. Zur Verringerung von Einflüssen durch Kameraposition und Handgröße werden die Koordinaten relativ zum Handgelenk zentriert und skaliert. Wird eine Hand in einem Frame nicht erkannt, werden die entsprechenden Koordinaten mit Nullen aufgefüllt; zusätzlich markiert ein binärer Indikator das Fehlen der jeweiligen Hand. Dadurch besitzt jedes Frame unabhängig von der Erkennungssituation dieselbe Feature-Dimension und kann als Sequenz in die nachfolgenden Zeitreihenmodelle eingegeben werden.

2.3 Sequenzbildung und Label-Zuordnung

2.3.1 Sequenzbildung und Windowing

Aus den pro Frame berechneten Merkmalen wird für jedes Video zunächst eine zeitlich sortierte Abfolge von Feature-Vektoren erstellt. Konkret wird das Video Bild für Bild (Frame für Frame) durchlaufen, wobei jedem Frame ein Merkmalsvektor mit fester Länge zugeordnet wird. Setzt man diese Vektoren der Reihe nach zusammen, entsteht eine Feature-Matrix der Größe TxF, Dabei steht T für die Anzahl der Frames im Video und F für die Anzahl der Merkmale pro Frame.

Da Videos unterschiedlich lang sind, können diese Feature-Sequenzen nicht direkt in gleicher Form verarbeitet werden. Deshalb werden sie in gleich große, sich überlappende Zeitfenster, sogenannte Sliding Windows, aufgeteilt. Jedes Fenster umfasst eine feste Anzahl an Frames und wird mit einem konstanten zeitlichen Versatz aus der Sequenz entnommen. Falls am Ende eines Videos ein Zeitfenster nicht vollständig gefüllt werden kann, werden die fehlenden Frames durch Padding ergänzt. So bleibt das Eingabeformat für alle Daten einheitlich.

Dieser Windowing-Ansatz hat mehrere Vorteile: Zum einen werden lange Videosequenzen in viele kleinere, vergleichbare Trainingseinheiten zerlegt, was die Rechenlast reduziert und das Training stabiler macht. Zum anderen ist sichergestellt, dass sowohl das reine LSTM-Modell als auch das hybride CNN-LSTM-Modell mit exakt derselben Eingabestruktur arbeiten, wodurch die Modelle fair miteinander verglichen werden können.

Insgesamt ergibt sich damit eine durchgängige Verarbeitungspipeline, in der die Videodaten zunächst in Frames zerlegt werden, anschließend Hand-Landmarks mittels MediaPipe extrahiert werden und daraus sequenzielle Featurevektoren entstehen. Diese werden in feste Zeitfenster segmentiert und anschließend einem sequenziellen neuronalen Modell zur Klassifikation einzelner Glossen zugeführt. Die Pipeline bildet damit alle zentralen Verarbeitungsschritte von der Rohvideoeingabe bis zur Modellvorhersage ab.

2.3.2 Label-Zuordnung (Weak Alignment auf Fensterebene)

Der verwendete Datensatz liefert keine exakten Start- und Endzeitpunkte einzelner Glossen auf Frame-Ebene. Das bedeutet, dass nicht eindeutig festgelegt werden kann, welches Frame oder welches Zeitfenster zu welcher Gloss gehört. Eine präzise Zuordnung ist daher mit den vorliegenden Daten nicht möglich.

Aus diesem Grund wird in dieser Arbeit ein vereinfachter Ansatz verwendet, der als sogenanntes *Weak Alignment* bezeichnet wird (d. h. eine nur näherungsweise Zuordnung zwischen Eingabesequenz und Labels ohne exakte zeitliche Synchronisation). Konkret wird die zu einem Video gehörende Gloss-Sequenz auf die zuvor extrahierten Sliding Windows abgebildet, indem die Zeitfenster gleichmäßig auf die vorhandenen Glossen verteilt werden. Jedes Fenster erhält dadurch genau ein Gloss-Label.

Dieses Vorgehen stellt keine echte zeitliche Segmentierung mit exakt erkannten Gebärdengrenzen dar. Es erlaubt jedoch ein konsistentes, nachvollziehbares und reproduzierbares Training der Modelle. Da sowohl das LSTM-Basismodell als auch das CNN-LSTM-Modell mit denselben Zeitfenstern und identischen Labels trainiert werden, eignet sich dieser Ansatz besonders gut für einen fairen Vergleich der beiden Modellarchitekturen.

2.4 Modellarchitekturen

2.4.1 LSTM-basierter Window-Klassifikator (Baseline)

Als Baseline wird ein einfacher, LSTM-basierter Klassifikator verwendet, der auf festen Zeitfenstern arbeitet. Das Modell erhält jeweils ein Zeitfenster mit einer Feature-Sequenz und ordnet diesem genau eine Gloss-Klasse zu. LSTM-Netzwerke (Long Short-Term Memory) sind speziell dafür entwickelt worden, zeitliche Abhängigkeiten zu lernen, und eignen sich daher gut, um Bewegungsabläufe über mehrere aufeinanderfolgende Frames hinweg zu erfassen.

Der Modellaufbau ist bewusst übersichtlich gehalten. Zunächst verarbeitet eine LSTM-Schicht die Feature-Sequenz innerhalb eines Zeitfensters und lernt dabei, wie sich die Merkmale über die Zeit verändern. Anschließend folgt eine Fully-Connected-Schicht, die auf Basis dieser zeitlichen Information die eigentliche Klassifikation der Gloss vornimmt. Da alle Zeitfenster die gleiche Länge haben, kann das Modell unabhängig von der Gesamtdauer eines Videos trainiert werden.

Der LSTM-basierte Ansatz dient in dieser Arbeit als Referenzmodell. Er stellt eine etablierte und gut verständliche Methode zur Verarbeitung zeitlicher Daten dar und bietet damit eine klare Ausgangsbasis, um die Leistung mit komplexeren Modellarchitekturen, wie etwa hybriden CNN-LSTM-Modellen, systematisch zu vergleichen.

2.4.2 Hybrides CNN-LSTM-Modell

Ergänzend zum Basismodell wird ein hybrides CNN-LSTM-Modell betrachtet. Ziel dieses Ansatzes ist es, zeitlich lokale Muster innerhalb eines festen Zeitfensters zunächst gezielt zu erfassen, bevor längerfristige zeitliche Zusammenhänge modelliert werden.

Dazu werden die Frame-Features eines Zeitfensters zuerst durch Faltungsschichten verarbeitet. Diese Schichten analysieren jeweils benachbarte Frames und können so kurze Bewegungsabfolgen sowie lokale Positionsänderungen erkennen. Die dadurch entstehenden, verdichteten Merkmalsrepräsentationen werden anschließend an eine LSTM-Schicht weitergegeben, die die zeitliche Entwicklung dieser Merkmale über das gesamte Fenster hinweg modelliert.

Solche hybriden CNN-LSTM-Architekturen wurden bereits in mehreren Arbeiten zur Gebärdensprachverarbeitung eingesetzt und gelten als vielversprechend für sequenzielle visuelle Daten. Der Grund dafür ist die Kombination aus lokaler Mustererkennung durch die CNN-Komponenten und der Modellierung längerfristiger Abhängigkeiten durch das LSTM. In dieser Arbeit wird der Hybridansatz unter denselben Daten- und Trainingsbedingungen wie das LSTM-

Basismodell evaluiert, um den Einfluss der zusätzlichen Verarbeitung übe ein CNN gezielt zu untersuchen.

2.5 Trainingssetup und Datenaufbereitung

Für das Training werden die vorbereiteten Zeitfenster zusammen mit den zugehörigen Gloss-Labels in Trainings-, Validierungs- und Testdaten aufgeteilt. Diese Aufteilung folgt den im RWTH-PHOENIX-Weather-Datensatz vorgegebenen Splits, sodass Trainings- und Evaluationsdaten klar voneinander getrennt sind und keine Überschneidungen entstehen.

Da die einzelnen Glossen im Datensatz sehr unterschiedlich häufig vorkommen, wird beim Training eine Klassengewichtung verwendet. Häufig auftretende Glossen werden dabei geringer gewichtet als seltene. Auf diese Weise wird verhindert, dass das Modell sich zu stark auf dominante Klassen konzentriert und seltene Glossen vernachlässigt.

Als Verlustfunktion wird die kategoriale Kreuzentropie verwendet, da jedes Zeitfenster genau einer Gloss-Klasse zugeordnet ist. Für das Training der Modelle kommt der Adam-Optimierer zum Einsatz, da er in vielen Deep-Learning-Anwendungen zuverlässig funktioniert. *Adam* passt die Lernrate automatisch an und nutzt dabei Informationen aus früheren Gradienten, was zu einem stabilen und effizienten Trainingsverlauf führt (Kingma & Ba, 2017, S. 2f). Die zentralen Trainingsparameter, insbesondere Lernrate, Batchgröße und Anzahl der Epochen, werden für beide Modellarchitekturen identisch gewählt, um einen fairen Vergleich sicherzustellen.

Zur Reduktion von Überanpassung werden Regularisierungstechniken wie Dropout eingesetzt. Zusätzlich wird die Modellleistung während des Trainings kontinuierlich auf dem Validierungsdatensatz überwacht. Dieser dient ausschließlich zur Bewertung des Trainingsfortschritts und wird nicht zur direkten Anpassung der Modellparameter verwendet.

2.6 Zwischentests und Verifikation der Pipeline

Vor dem eigentlichen Modelltraining wurden mehrere Zwischentests durchgeführt, um die korrekte Funktionsweise der gesamten Datenpipeline sicherzustellen. Zunächst wurde überprüft, ob die Datenstruktur des Datensatzes wie erwartet vorliegt und ob sich einzelne Video-Frame-Ordner korrekt in Feature-Sequenzen fester Dimension überführen lassen.

Anschließend wurden die Gloss-Annotationen geladen und stichprobenartig verifiziert, um sicherzustellen, dass die Video-IDs korrekt mit den zugehörigen Gloss-Sequenzen verknüpft sind. Darüber hinaus wurde geprüft, ob aus einzelnen Samples konsistente Zeitfenster und die zugehörigen Labels erzeugt werden können.

Abschließend wurden kurze Vorwärtsdurchläufe der Modelle mit einer kleinen Anzahl von Trainingsinstanzen durchgeführt, um die Kompatibilität der Datenformate mit den Modellarchitekturen zu validieren. Diese Zwischentests dienen dazu, Implementierungsfehler

frühzeitig zu erkennen und stellen sicher, dass die Pipeline vor dem eigentlichen Training end-to-end konsistent ist.

2.7 Fensterbasierte Klassifikation und Nachverarbeitung

Die Modelle erzeugen ihre Vorhersagen nicht auf Ebene einzelner Frames, sondern auf Basis fester, überlappender Zeitfenster. Jedes Zeitfenster wird unabhängig klassifiziert und genau einer Gloss-Klasse zugeordnet, sodass für ein Video eine zeitlich geordnete Folge von Fenster-Vorhersagen entsteht, die zunächst eine rohe Label-Sequenz bildet. Überlappende Fenster werden bewusst eingesetzt, da Gebärden nicht exakt an feste Zeitgrenzen gebunden sind. Durch die Überlappung wird die Robustheit der Zuordnung erhöht und sichergestellt, dass eine Gebärde in mindestens einem Fenster vollständig erfasst wird, auch wenn dadurch mehrere identische Vorhersagen für denselben Zeitabschnitt entstehen.

Zur Stabilisierung der Ausgabe wird anschließend ein einfaches Postprocessing durchgeführt. Dabei werden identische Vorhersagen in aufeinanderfolgenden Fenstern zusammengefasst (Collapse-Schritt). Optional kann zusätzlich eine Mehrheitsentscheidung über benachbarte Fenster angewendet werden, um instabile Übergänge weiter zu glätten. Aus dieser Verarbeitung entsteht eine konsistente Gloss-Sequenz, die die finale Systemausgabe darstellt und für Analyse sowie Evaluation verwendet wird.

Das trainierte Modell ist dabei in eine prototypische Verarbeitungspipeline eingebettet, in der Videoframes fortlaufend verarbeitet, in Zeitfenster segmentiert und anschließend klassifiziert werden. Die Pipeline bildet somit eine vereinfachte Echtzeitverarbeitung ab, da die Modellinferenz direkt auf kontinuierliche Videodaten angewendet werden kann. Ziel ist hierbei nicht die Entwicklung einer vollständig optimierten Produktionslösung, sondern der prototypische Nachweis der technischen Umsetzbarkeit einer automatischen Gebärdenspracherkennung.

3 Evaluation und Metriken

Die Evaluation dient dazu, die im Projekt entwickelten Modelle systematisch miteinander zu vergleichen. Im Mittelpunkt steht dabei der Einfluss der jeweiligen Modellarchitektur auf die Erkennungsleistung. Verglichen werden ein LSTM-basierter Window-Klassifikator und ein hybrides CNN-LSTM-Modell, die unter identischen Daten-, Vorverarbeitungs- und Trainingsbedingungen eingesetzt werden. Unterschiede in den Ergebnissen lassen sich dadurch direkt auf architekturelle Aspekte zurückführen.

Der Fokus der Evaluation liegt nicht auf der Entwicklung eines produktionsreifen Systems, sondern auf einer methodischen Einordnung der verwendeten Ansätze. Neben den quantitativen Ergebnissen werden auch qualitative Beobachtungen berücksichtigt, um das Modellverhalten im Rahmen einer prototypischen Gebärdenspracherkennung besser zu verstehen.

3.1 Evaluationskonzept

Die Evaluation erfolgt auf getrennten Validierungs- und Testdatensätzen, die während des Trainings nicht zur Anpassung der Modellparameter verwendet wurden. Beide Modellvarianten werden mit identischen Eingabedaten sowie gleichen Hyperparametern evaluiert, um einen fairen und nachvollziehbaren Vergleich zu gewährleisten.

Als zentrale Bewertungsmetrik wird die Klassifikationsgenauigkeit *Accuracy* verwendet, da jedes Zeitfenster genau einer Gloss-Klasse zugeordnet wird. Ergänzend werden qualitative Analysen durchgeführt, um typische Fehlermuster zu identifizieren, etwa instabile Vorhersagen an Übergängen zwischen Gebärden oder eine Bevorzugung häufiger Gloss-Klassen.

Zusätzlich wird die Inferenzlatenz betrachtet, um eine grobe Einschätzung der praktischen Einsetzbarkeit der Modelle innerhalb einer prototypischen Pipeline zu ermöglichen. Für erste Experimente wurde ein reduzierter Datenausschnitt verwendet, um Rechenzeit zu begrenzen und die Funktionsfähigkeit der Pipeline zu überprüfen.

3.2 Ergebnisse

3.2.1 Baseline LSTM

Das LSTM-Basismodell erreicht auf dem Validierungsdatensatz eine Klassifikationsgenauigkeit von etwa 4,68 % im letzten und 4,94 % im vorletzten Durchlauf. Damit liegt die Leistung deutlich über dem Zufallsniveau. Bei insgesamt 501 möglichen Gloss-Klassen würde ein zufälliges Raten lediglich eine Trefferquote von rund 0,2 % erwarten lassen. Die deutlich höhere Accuracy zeigt, dass das Modell systematische Zusammenhänge in den Daten nutzt und insbesondere wiederkehrende zeitliche Muster in den Hand-Landmark-Sequenzen erlernt.

Die Trainings- und Validierungskurven weisen über mehrere Epochen hinweg einen gleichmäßigen Anstieg ohne starke Divergenzen auf, was auf ein stabiles Lernverhalten hindeutet. Im Kontext der gewählten Vereinfachungen, insbesondere der Beschränkung auf Hand-Landmarks sowie der heuristischen Fenster-zu-Gloss-Zuordnung, ist die erzielte Genauigkeit als solides Baseline-Ergebnis einzuordnen und dient als Referenz für die weiteren Experimente.

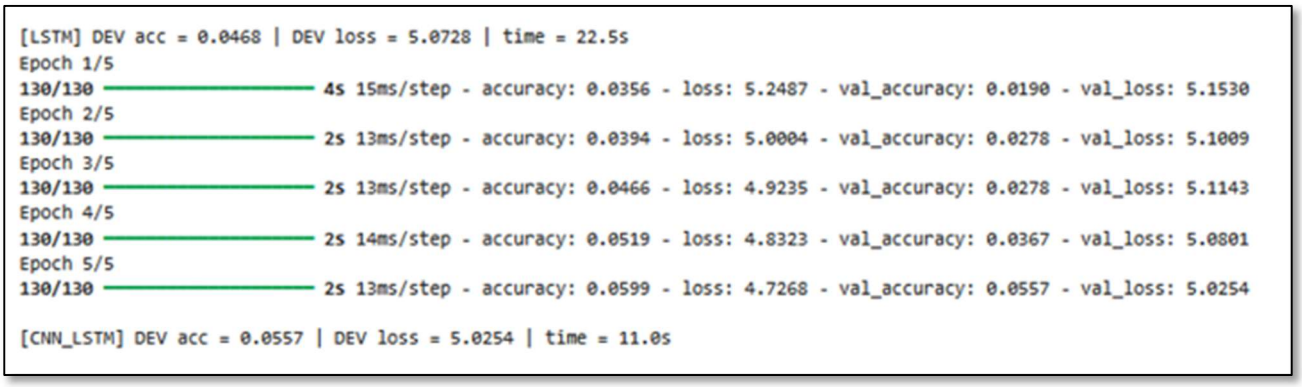
```
[Build] Landmark datasets gebaut (2993.4s)
X_train: (4159, 60, 128) y_train: (4159,) | classes in y_train: 351
X_dev  : (790, 60, 128) y_dev  : (790,) | classes in y_dev  : 201
Epoch 1/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 6s 33ms/step - accuracy: 0.0337 - loss: 5.2763 - val_accuracy: 0.0177 - val_loss: 5.1289
Epoch 2/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 4s 31ms/step - accuracy: 0.0457 - loss: 4.9775 - val_accuracy: 0.0418 - val_loss: 5.1127
Epoch 3/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 4s 32ms/step - accuracy: 0.0474 - loss: 4.9060 - val_accuracy: 0.0443 - val_loss: 5.1090
Epoch 4/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 4s 32ms/step - accuracy: 0.0476 - loss: 4.8164 - val_accuracy: 0.0494 - val_loss: 5.0564
Epoch 5/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 4s 31ms/step - accuracy: 0.0604 - loss: 4.7159 - val_accuracy: 0.0468 - val_loss: 5.0728

[LSTM] DEV acc = 0.0468 | DEV loss = 5.0728 | time = 22.5s
```

3.2.2 CNN-LSTM Hybridmodell

Das hybride CNN-LSTM-Modell wurde mit denselben Eingabedaten, Hyperparametern und Trainingsprotokollen trainiert. Es erreicht auf dem Validierungsdatensatz eine Accuracy von rund 5,57 % und liegt damit geringfügig über dem reinen LSTM-Basismodell.

Durch die vorgeschalteten Faltungsschichten kann das Modell lokale zeitliche Muster innerhalb eines Fensters erfassen, bevor diese durch das LSTM über längere Zeiträume hinweg verarbeitet werden. Der Leistungsgewinn fällt insgesamt moderat aus, zeigt jedoch, dass die Kombination aus lokaler Mustererkennung und sequenzieller Modellierung grundsätzlich sinnvoll ist. Gleichzeitig bleibt der Unterschied zum LSTM-Basismodell relativ gering, was auf die stark abstrahierte Eingaberepräsentation und die vereinfachte Labelzuordnung zurückzuführen ist.



3.2.3 Direkter Vergleich der Modellarchitekturen

Der direkte Vergleich der beiden zeitbasierten Modelle zeigt, dass sowohl das LSTM-Basismodell als auch das hybride CNN-LSTM-Modell sehr ähnliche Klassifikationsgenauigkeiten erreichen. Zwar erzielt das CNN-LSTM-Modell eine geringfügig höhere Accuracy, der Unterschied ist jedoch insgesamt so klein, dass daraus kein eindeutiger Vorteil der komplexeren Architektur abgeleitet werden kann.

Dieses Ergebnis deutet darauf hin, dass die erreichbare Erkennungsleistung in diesem Setup weniger durch die konkrete Modellarchitektur begrenzt wird, sondern vielmehr durch die gewählte Problemformulierung. Insbesondere die fehlenden framegenauen Annotationen sowie der eingesetzte Weak-Alignment-Ansatz führen zu unvermeidbaren Ungenauigkeiten in den Trainingslabels, wodurch die maximal erreichbare Modellleistung eingeschränkt wird.

Auffällig ist, dass das CNN-LSTM-Modell eine kürzere Trainingszeit aufweist als das reine LSTM-Basismodell. Dies lässt sich dadurch erklären, dass die vorgelagerten konvolutionellen Schichten die zeitliche Auflösung der Eingabesequenzen reduzieren, sodass das nachgeschaltete LSTM weniger Zeitschritte verarbeiten muss.

Obwohl das CNN-LSTM damit sowohl eine leicht höhere Accuracy als auch eine geringere Trainingszeit erzielt, fällt der Gesamtunterschied zwischen den Modellen moderat aus. Der Geschwindigkeitsvorteil allein stellt daher kein ausreichendes Argument für die komplexere

Architektur dar, insbesondere im Kontext einer prototypischen Pipeline mit begrenztem Datensatz und bewusst vereinfachter Problemformulierung.

Vor diesem Hintergrund erscheint der Einsatz des einfacheren LSTM-Basismodells für eine prototypische Pipeline als sinnvoll. Es erreicht bei vergleichbarer Erkennungsleistung eine geringere Modellkomplexität und einen niedrigeren Implementierungsaufwand, während es dennoch in der Lage ist, relevante zeitliche Muster in den Hand-Landmark-Sequenzen zuverlässig zu erfassen.

3.2.4 Bildbasiertes CNN auf Rohdaten

Als weiteres Vergleichsexperiment wurde zusätzlich ein CNN-LSTM-Modell auf Rohbilddaten untersucht. Im Unterschied zu den Landmark-basierten Ansätzen erfolgt hier zunächst eine konvolutionelle Merkmalsextraktion direkt aus den Video-Frames, bevor ein LSTM die zeitliche Abfolge dieser Merkmale modelliert. Ziel dieses Experiments war es zu prüfen, ob eine Kombination aus visueller Merkmalsextraktion und zeitlicher Modellierung auch ohne explizit vorverarbeitete Hand-Landmarks zu vergleichbaren Ergebnissen führen kann.

Das CNN-LSTM-Modell auf Rohdaten erreicht auf dem Validierungsdatensatz eine Klassifikationsgenauigkeit von etwa 3,9 % und bleibt damit unter den Ergebnissen der Landmark-basierten LSTM- und CNN-LSTM-Modelle. Gleichzeitig zeigt sich ein deutlich höherer Rechenaufwand, was sich in einer erheblich längeren Trainingszeit widerspiegelt.

Dieses Ergebnis ist plausibel, da die zeitliche Modellierung hier auf hochdimensionalen Rohbildmerkmalen basiert, die im Vergleich zu Hand-Landmarks stärker verrauscht und weniger direkt auf die eigentliche Gebärdenausführung fokussiert sind. Relevante Informationen wie Handform, Position und Bewegung müssen vom Modell zunächst implizit aus den Bilddaten gelernt werden. Dadurch wird die Lernaufgabe erheblich erschwert, selbst wenn eine zeitliche Modellierung durch ein LSTM vorhanden ist.

Das Experiment verdeutlicht somit, dass die Kombination aus CNN und LSTM allein nicht automatisch zu besseren Ergebnissen führt. Entscheidend ist vielmehr die Qualität und Aufgabenangemessenheit der verwendeten Eingaberepräsentation. Im vorliegenden Setup erweisen sich explizit extrahierte Hand-Landmarks als deutlich effizientere und besser geeignete Grundlage für zeitbasierte Modelle als Rohbilder.

```
[Raw-CNN-LSTM] instances=790 | skipped_no_gloss=0 | skipped_no_frames=0
Epoch 1/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 1241s 9s/step - accuracy: 0.0368 - loss: 5.2651 - val_accuracy: 0.0190 - val_loss: 5.1882
Epoch 2/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 1113s 9s/step - accuracy: 0.0409 - loss: 5.0419 - val_accuracy: 0.0291 - val_loss: 5.1870
Epoch 3/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 1123s 9s/step - accuracy: 0.0418 - loss: 5.0325 - val_accuracy: 0.0291 - val_loss: 5.2105
Epoch 4/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 1183s 9s/step - accuracy: 0.0366 - loss: 5.0290 - val_accuracy: 0.0190 - val_loss: 5.2145
Epoch 5/5
130/130 ━━━━━━━━━━━━━━━━━━━━ 1115s 9s/step - accuracy: 0.0416 - loss: 5.0218 - val_accuracy: 0.0392 - val_loss: 5.2110

[RAW_CNN_LSTM] DEV acc = 0.0392 | DEV loss = 5.2110 | time = 5834.2s
```

3.3 Diskussion und Einordnung der Ergebnisse

Die Ergebnisse der durchgeführten Experimente zeigen, dass die Wahl der Modellarchitektur einen erkennbaren Einfluss auf die Erkennungsleistung bei der Gebärdenspracherkennung hat. Gleichzeitig wird deutlich, dass sich alle erzielten Genauigkeiten auf einem insgesamt niedrigen absoluten Niveau bewegen. Dieses Ergebnis ist jedoch im Kontext der Aufgabenstellung und der bewusst gewählten Vereinfachungen zu betrachten.

Zunächst bestätigt das LSTM-Basismodell die zentrale Bedeutung zeitlicher Modellierung für die Gebärdenspracherkennung. Trotz der Beschränkung auf Hand-Landmark-Sequenzen und der nur näherungsweisen Zuordnung von Zeitfenstern zu Gloss-Labels erzielt das Modell eine Erkennungsleistung oberhalb des Zufallsniveaus. Dies zeigt, dass das LSTM in der Lage ist, grundlegende Bewegungsmuster und zeitliche Abfolgen aus den Daten zu lernen, selbst wenn keine framegenauen Annotationen vorliegen.

Das hybride CNN-LSTM-Modell erreicht im Vergleich zum reinen LSTM eine leicht höhere Erkennungsleistung. Durch die vorgelagerte konvolutionelle Verarbeitung können lokale Muster innerhalb eines Fensters zunächst separat erfasst und anschließend durch das LSTM über längere Zeiträume hinweg modelliert werden. Der beobachtete Leistungsgewinn fällt jedoch insgesamt moderat aus. Dies lässt sich unter anderem dadurch erklären, dass die verwendeten Hand-Landmark-Features bereits eine stark abstrahierte Darstellung der Bewegungen liefern, sodass der zusätzliche Nutzen der CNN-Schichten begrenzt bleibt. Dennoch verdeutlicht das Ergebnis, dass die Kombination aus lokaler Mustererkennung und sequenzieller Modellierung grundsätzlich sinnvoll ist.

Ein wichtiges Vergleichsexperiment stellt das CNN-LSTM-Modell auf Rohbilddaten dar. Obwohl hier, im Gegensatz zu einem reinen CNN, eine explizite zeitliche Modellierung durch ein LSTM erfolgt und das Modell insgesamt deutlich mehr Parameter besitzt, bleibt die Erkennungsleistung unterhalb der landmark-basierten LSTM- und CNN-LSTM-Modelle. Dieses Ergebnis zeigt, dass Rohbilder für sich genommen keine optimale Eingaberepräsentation für die Gebärdenspracherkennung darstellen. Beim Training auf Rohbildern muss das Modell relevante Informationen wie Handform, Position und Bewegung zunächst implizit aus hochdimensionalen Bilddaten extrahieren, bevor diese zeitlich modelliert werden können. Dadurch wird die Lernaufgabe deutlich komplexer, was zu unsicheren und weniger stabilen Vorhersagen führt – selbst dann, wenn eine zeitliche Modellierung durch ein LSTM vorhanden ist.

Über alle Modelle hinweg ist zu berücksichtigen, dass mehrere Faktoren die erreichbare Genauigkeit begrenzen. Dazu zählen insbesondere die Beschränkung auf Hand-Landmarks ohne non-manuelle Komponenten, der reduzierte Umfang der verwendeten Trainingsdaten sowie der eingesetzte Weak-Alignment-Ansatz zur Labelzuordnung. Da die tatsächlichen zeitlichen Grenzen der Gebärden nicht bekannt sind, enthalten die Trainingslabels zwangsläufig Ungenauigkeiten. Die Ergebnisse sind

daher nicht als absolute Leistungsbewertung der Modelle zu verstehen, sondern als vergleichende Einordnung unter kontrollierten und vereinfachten Bedingungen.

Zusammenfassend zeigen die Experimente, dass zeitliche Modellierung für die automatische Gebärdenspracherkennung unverzichtbar ist. Sowohl das LSTM-Basismodell als auch das hybride CNN-LSTM-Modell demonstrieren, dass selbst vereinfachte Feature-Repräsentationen verwertbare Informationen enthalten, sofern ihre zeitliche Struktur berücksichtigt wird. Die Ergebnisse liefern damit eine nachvollziehbare Begründung für den Einsatz sequenzieller oder hybrider Architekturen in weiterführenden, komplexeren Systemen.

3.4 Limitationen und Ausblick

Die in dieser Projektarbeit entwickelte Pipeline stellt einen prototypischen Ansatz dar, der bewusst auf Verständlichkeit, Nachvollziehbarkeit und einen sauberen methodischen Vergleich ausgelegt ist. Daraus ergeben sich mehrere Einschränkungen, die bei der Interpretation der Ergebnisse berücksichtigt werden müssen.

Eine zentrale Limitation liegt in der vereinfachten Datenrepräsentation. Das System verwendet ausschließlich Hand-Landmark-Features, während non-manuelle Komponenten der Gebärdensprache wie Gesichtsausdruck, Mundbild sowie Kopf- und Oberkörperbewegungen nicht einbezogen werden. Diese Entscheidung wurde aus Gründen des Projektumfangs und der verfügbaren Rechenressourcen getroffen, schränkt jedoch die Ausdruckskraft des Systems ein, da viele Gebärden erst durch das Zusammenspiel manueller und non-manueller Signale eindeutig interpretiert werden können.

Eine weitere Einschränkung ergibt sich aus der heuristischen Zuordnung von Zeitfenstern zu Gloss-Labels. Da der verwendete Datensatz keine verlässlichen framegenauen Start- und Endpunkte der Gebärden bereitstellt, kommt ein Weak-Alignment-Ansatz zum Einsatz. Dieser führt zwangsläufig zu ungenauen Trainingslabels und begrenzt damit die maximal erreichbare Klassifikationsgenauigkeit. Die erzielten Ergebnisse sind daher primär als vergleichende Referenzwerte zwischen den Modellarchitekturen zu verstehen.

Auch der fensterbasierte Klassifikationsansatz stellt eine Vereinfachung dar. Die Modelle ordnen jedem Zeitfenster genau eine Gloss-Klasse zu, anstatt vollständige Gloss-Sequenzen variabler Länge vorherzusagen. Dieser Ansatz ermöglicht zwar einen stabilen Vergleich und eine prototypische Echtzeitverarbeitung, ersetzt jedoch keine echte linguistische Sequenzmodellierung.

Darüber hinaus war der Umfang der Experimente durch begrenzte Rechenressourcen eingeschränkt. Aus diesem Grund wurden reduzierte Datensätze und kompakte Modellarchitekturen verwendet. Umfangreiche Hyperparameter-Optimierungen oder größere Modellvarianten konnten nicht vollständig untersucht werden.

Trotz dieser Einschränkungen zeigt die Arbeit, dass sich bereits mit Hand-Landmark-Sequenzen und sequenziellen neuronalen Netzen verwertbare Ergebnisse erzielen lassen. Für zukünftige Arbeiten

bieten sich mehrere Erweiterungen an, darunter die Integration von Gesichts- und Pose-Landmarks sowie der Einsatz alignment-freier Trainingsverfahren, also ohne exakte zeitliche Zuordnung zwischen Eingabe und Labels, wie Connectionist Temporal Classification (CTC). Solche Verfahren wurden in der Forschung zur automatischen Gebärdenspracherkennung bereits eingesetzt, da sie das Training auf Sequenzdaten ohne exakte framegenaue Annotationen ermöglichen (Camgoz et al., 2020, S. 1-3). Auch adaptive Fenstergrößen und verbesserte Nachverarbeitungsverfahren könnten die Stabilität weiter erhöhen.

4 Fazit

Ziel dieser Projektarbeit war die prototypische Entwicklung eines Computer-Vision-Systems zur Echtzeiterkennung und Übersetzung von Gebärdensprache in Text sowie die methodische Untersuchung geeigneter Deep-Learning-Ansätze. Auf Basis des Datensatzes *RWTH-PHOENIX-Weather 2014 T* wurde eine vollständige Verarbeitungspipeline umgesetzt, die von der Datenaufbereitung über die Merkmalsextraktion bis hin zur Modellierung, Evaluation und prototypischen Anwendung reicht.

Im Zentrum stand die Verarbeitung kontinuierlicher Videosequenzen mit zeitbasierten neuronalen Netzen. Mithilfe von MediaPipe wurden Hand-Landmark-Sequenzen extrahiert und als Grundlage für die Modellierung verwendet. Die Ergebnisse zeigen, dass selbst bei einer stark reduzierten Repräsentation grundlegende gebärdensprachliche Muster erlernt werden können.

Der Vergleich zwischen einem LSTM-basierten Klassifikator und einem hybriden CNN-LSTM-Modell verdeutlicht die Bedeutung zeitlicher Modellierung. Während das LSTM-Basismodell bereits stabile Ergebnisse liefert, kann der Hybridansatz leichte Verbesserungen erzielen. Ein zusätzliches Experiment mit einem reinen CNN auf Rohbildern bestätigt, dass statische Einzelbilder ohne zeitliche Modellierung für diese Aufgabe nur eingeschränkt geeignet sind.

Die erzielten Genauigkeiten sind insgesamt niedrig, lassen sich jedoch durch die getroffenen Vereinfachungen erklären. In diesem Kontext erfüllen die Ergebnisse ihre Rolle als Baseline und ermöglichen eine nachvollziehbare Einordnung der untersuchten Ansätze.

5 Abschlussreflexion

Die Bearbeitung dieses Projekts hat gezeigt, dass die Entwicklung eines Systems zur automatischen Gebärdenspracherkennung sowohl konzeptionell als auch technisch anspruchsvoll ist, insbesondere bei kontinuierlichen Videodaten ohne framegenaue Annotationen. Eine zentrale Herausforderung bestand darin, einen praktikablen Kompromiss zwischen Modellkomplexität, Datenstruktur und verfügbaren Ressourcen zu finden.

Der schrittweise Aufbau der Pipeline erwies sich als besonders lehrreich, da er ein besseres Verständnis für das Zusammenspiel von Daten, Feature-Repräsentation und Modellarchitektur ermöglichte. Der direkte Vergleich verschiedener Ansätze machte deutlich, dass höhere Modellkomplexität nicht automatisch zu besseren Ergebnissen führt.

Insgesamt hat das Projekt sowohl fachliche Kompetenzen im Bereich Computer Vision und Deep Learning vertieft als auch ein realistisches Verständnis für die Grenzen datengetriebener Systeme vermittelt. Die gewonnenen Erkenntnisse bilden eine solide Grundlage für weiterführende Arbeiten und zeigen, wie theoretische Konzepte in praktikable Prototypen überführt werden können.

Alle Unterlagen zu diesem Bericht finden sie auch in meinem GitHub Rep unter dem Link: https://github.com/goetzmoeglinger/IU-Projekt-Computer-Vision

Literaturverzeichnis

Becker, C., & Jaeger, H. (2019). *Deutsche Gebärdensprache: Mehrsprachigkeit mit Laut- und Gebärdensprache* (1st ed). Narr Francke Attempto Verlag.

Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). *Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation* (arXiv:2003.13830). arXiv. https://doi.org/10.48550/arXiv.2003.13830

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. https://doi.org/10.48550/arXiv.1412.6980

Koller, O. (2020). *Quantitative Survey of the State of the Art in Sign Language Recognition* (arXiv:2008.09918). arXiv. https://doi.org/10.48550/arXiv.2008.09918

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). *MediaPipe: A Framework for Building Perception Pipelines* (arXiv:1906.08172). arXiv. https://doi.org/10.48550/arXiv.1906.08172

Pilling, C.-S. (mit Wilke, T.). (2022). *Gehörlose und Hörende: Raummodellierung im Kontext von Behinderung und Interkulturalität*. transcript Verlag.