1. **Einleitung**

In den letzten Jahren haben Entwicklungen in den verschiedensten Bereichen zu Fortschritten in der Sprach-, aber auch Bilderkennung geführt. Die Fähigkeit, Gebärdensprache aus Handzeichen zu erkennen, ist ein bedeutender technologischer Fortschritt, der die Kommunikation für gehörlose oder schwerhörige Menschen revolutionieren kann. In der Gebärdensprache gibt es viele verschiedene Handzeichen, die den Menschen in der Kommunikation untereinander helfen können. Unter anderem steht für jeden Buchstaben des Alphabets ein bestimmtes Handzeichen zur Verfügung. In der vorliegenden Arbeit sollen genau diese Buchstaben verwendet werden, um ein Modell mit Bildern zu den jeweiligen Handzeichen zu trainieren, damit später die Möglichkeit besteht, das Modell mit dem Zeigen von verschiedenen Handzeichen und der Erkennung dieser Gebärdenzeichen zu nutzen.

In dem vorliegenden Projekt wird mit Hilfe eines vortrainierten CNNs, einem Convolutional Neural Network, ein Modell trainiert und entwickelt, welches die Zeichensprache in Buchstaben übersetzen kann. Eine dafür genutzte Technologie ist YOLO, You Only Look Once, die sich durch ihre Fähigkeit auszeichnet, Objekte in Echtzeit zu erkennen und zu klassifizieren. Für diese Arbeit wird die spezielle Variante YOLOv8 genutzt, welches durch verbesserte Eigenschaften als die Vorgängermodelle in vielen Bereichen bevorzugt wird.

Der Fokus dieses Projekts liegt auf der Implementierung und Einschätzung von YOLOv8 für die Bildklassifizierung. Die Arbeit ist in mehrere Abschnitte unterteilt, die sich mit den verschiedenen Aspekten der Modellierung, dem Training und der Analyse von CNNs befassen. Zunächst werden der verwendete Datensatz und die erforderlichen Schritte zur Datenvorverarbeitung betrachtet. Anschließend erfolgt eine detaillierte Untersuchung der YOLOv8-Architektur sowie der zugrunde liegenden mathematischen Konzepte und Mechanismen.

Ein weiterer wesentlicher Bestandteil dieser Arbeit ist das Training des Modells. Hierbei werden verschiedene Strategien zur Hyperparameteranpassung und Optimierung untersucht. Schließlich werden die trainierten Modelle getestet und ihre Leistungen verglichen, um die besten Ansätze zu identifizieren und zu bewerten.

Die Ergebnisse dieser Analyse bieten gute Einblicke in die Leistungsfähigkeit und die Anwendungsmöglichkeiten von YOLOv8 in der Bildklassifizierung, aber auch allgemein in die Entwicklung eines Modells mit einem vortrainierten CNN.

Mit Hilfe eines verfügbaren Datensatzes für Gebärdenzeichen sowie eigener Bilder wird in der folgenden Arbeit ein Modell trainiert und getestet, welches die oben genannten Technologien umsetzt und ein anwendbares Modell erstellt.

1. **Der Datensatz**

In diesem Abschnitt wird nun die Datenvorverarbeitung für den ausgewählten Datensatz beschrieben sowie der Datensatz näher erläutert.

Der für diese Arbeit ausgewählte Datensatz „ASL Alphabet“ wurde 2014 entwickelt, um das Erkennen von Buchstaben des amerikanischen Gebärdensprachenalphabets zu erleichtern. „ASL Alphabet“ kann übersetzt werden in „American Sign Language“, kurz „ASL“. Der Datensatz besteht aus einer großen Anzahl von Bildern, die von Handgesten der einzelnen Buchstaben repräsentiert werden [Datensatz Seite].

In der folgenden Abbildung ? ist zunächst eine Übersicht des „ASL-Alphabets“ zu sehen, in der allen Buchstaben die passenden Handzeichen zugeordnet worden sind.

Abbildung ASL Alphabet

Auch inhaltlich und strukturell gibt es einige Informationen, die die Webseite des Datensatzes liefert. Der Datensatz enthält 87000 Bilder mit einer Auflösung von 200×200 Pixeln. Diese Bilder sind in 29 Klassen unterteilt, die die Buchstaben von A bis Z und die zusätzlichen Klassen SPACE, DELETE und NOTHING repräsentiert. Die drei zusätzlichen Klassen sind für die Echtzeit-Anwendung und -Klassifizierung sehr nützlich, da sie bestimmte Aktionen oder einen „leeren“ Schritt darstellen. Alle 87000 Bilder sind jeweils in dem Ordner abgelegt, welcher der jeweiligen Klasse zugeordnet ist. Dies erleichtert das Laden der Daten während des Trainingsprozesses [Datensatz Seite].

Der Testdatensatz erhält 29 Bilder, wobei jedes Bild eine der 29 Klassen repräsentiert. Damit die Leistungsfähigkeit des Modells unter realistischen Bedingungen bewertet werden kann, wird nur eine geringe Anzahl an Testbildern genutzt [Datensatz Seite].

Neben den aus dem Datensatz „ASL Alphabet“ genutzten Bildern wurden von allen vier Gruppenmitgliedern zudem eigene Bilder der Handzeichen aufgenommen, um die Datenvielfalt zu vergrößern und ein umfangreicheres Ergebnis zu erzielen. Dafür hat jedes Gruppenmitglied für jeden Buchstaben, ausgenommen der Buchstaben J und Z, da dort Bewegungen nötig sind, die nicht in einem reinen Bild aufgezeichnet werden können, zwei oder mehr Bilder aufgenommen und wieder in eine passende Orderstruktur zu den jeweiligen Buchstaben zugeordnet. Dies erweitert den bisherigen Datensatz um weitere Bilder mit unterschiedlichen Hintergründen sowie anderen Ausführungen der Handzeichen.

* 1. **Datenvorverarbeitung**

Um die Daten aus dem Datensatz für den Algorithmus und damit später auch in der realen Anwendung nutzen zu können, muss mit der Datenvorverarbeitung zunächst ein wichtiger Schritt durchgeführt werden. Insbesondere bei der Sprach- und Bilderkennung muss sichergestellt sein, dass die Daten in einer Form vorliegen, die für das Training des Modells optimal sind und damit auch das Testen mit anderen Daten ermöglichen. Wichtig ist eine korrekte Speicherung der einzelnen Bilder in den passenden, den dargestellten Gebärdenzeichen zugewiesen Ordnern, um dem erstellten Algorithmus die richtigen Daten für das Trainieren des Modells zu übermitteln. Zudem wird damit das Laden der Daten erleichtert sowie die Verwaltung der Klassifikationen während des Trainingsprozesses. Falls sich irrelevante Bilder zwischen den über 87000 Bildern befinden, die keine relevanten Informationen für das Training enthalten, da beispielsweise falsche Handzeichen gezeigt wurden oder unscharfe, nicht aussagekräftige Bilder zu sehen sind, sollen diese aus dem Datensatz entfernt werden. Das gleiche gilt für Duplikate, die keine zusätzlichen Informationen für das Modell liefern würden. In dem genutzten Datensatz befinden sich insgesamt viele Hintergrundbilder in den Daten, welche innerhalb der Datenvorverarbeitung bereinigt worden sind. Neben der Entfernung von irrelevanten Bildern muss zudem sichergestellt werden, dass alle Bilder auf eine einheitliche Größe skaliert werden. Da die Bilder des „ASL Alphabet“-Datensatzes bereits alle in einer Bildgröße von 200×200 Pixeln vorliegen, muss die Anpassung höchstens bei den selbst erstellten Bildern erfolgen, da diese von vier verschiedenen Mitgliedern und demnach von unterschiedlichen Kameras erstellt worden sind. Die Größenanpassung hilft, die Berechnungen während des Modelltrainings zu vereinfachen und die Konsistenz der Eingabedaten sicherzustellen.

Ein weiterer Datenvorverarbeitungsschritt wurde durch die Reduzierung der Klassenanzahl erreicht. In den 29 vorhandenen Klassen befinden sich drei leere Klassen SPACE, DELETE und NOTHING, welche für die Echtzeit-Anwendung zwar nützlich sind, in der reinen Gebärdenzeichenerkennung jedoch keine Rolle spielen und daher entfernt werden können. Somit sind in den Daten nur noch Buchstabenklassen enthalten, die das Trainieren des Modells erleichtern können und der Fokus nur noch auf den 26 Buchstaben liegt.

Da in der vorliegenden Arbeit mit YOLO gearbeitet wurde, was im folgenden Kapitel genauer erläutert wird, müssen vorab hingehend zu dem Thema einige Erweiterungseinstellungen getroffen werden, die die Arbeit mit YOLO erleichtern sollen. Dafür stehen verschiedene „Data-Augmentation“-Erweiterungsargumente zur Verfügung. Einige davon wurden in der vorliegenden Arbeit genutzt, um die Robustheit und Leistung von YOLO-Modellen zu verbessern, indem Variabilität in die Trainingsdaten eingebracht wird und das Modell unterstützt wird, um besser auf ungesehen Daten zu reagieren. Dabei wählt das Modell selbst aus, welche Argumente genutzt werden müssen. Ein Argument aus den Erweiterungseinstellungen, das in diesem Fall eine Rolle für die Datenvorverarbeitung spielt, ist die Verwendung von „hsv\_h“, welches den Farbton des Bildes um einen Bruchteil des Farbkreises anpasst und dafür sorgt, dass sich das Modell durch die Farbvariabilität an unterschiedliche Lichtverhältnisse anpassen kann. Mit der Methode „hsv\_s“ kann die Sättigung der Bilder geändert werden, auch hier ebenso wie bei der vorhergegangenen Methode durch die Änderung des Bruchteils des Bildes. Dadurch wird die Intensität der Farben beeinflusst, um unterschiedliche Umgebungsbedingungen zu realisieren, die mit der Nutzung des Modells vorkommen können. Durch die Verwendung des Arguments „translate“ können die Bilder horizontal und vertikal um einen Bruchteil der jeweiligen Bildgröße verschoben werden. Dies soll dabei dem Modell beim Lernen helfen, um teilweise sichtbare Objekte zu erkennen. Neben der Bildverschiebung kann auch die Bildskalierung, dargestellt durch das Argument „scale“, dabei helfen, das Modell zu verbessern. Hierfür wird ein Verstärkungsfaktor verwendet, der Objekte in unterschiedlichen Entfernungen zur Kamera simuliert. Dies ist ein notwendiger Vorverarbeitungsschritt, um die Realität bei der Nutzung des Modells darzustellen. Eine nützliche Erweiterungseinstellung von YOLO bietet das Argument „fliplr“, welches das jeweilige Bild mit der angegebenen Wahrscheinlichkeit von links nach rechts spiegelt, um symmetrische Objekte zu lernen und die Vielfalt der Datensätze zu erhöhen. Dies ist besonders hilfreich, da der Test des Modells und das Zeigen der Gebärdenzeichen häufig über die Frontkamera eines Computers geschieht und hier die Spiegelung essentiell ist. Weitere Argumente, wie beispielsweise „mosaic“ zur Kombination von vier Trainingsbildern zu einem einzigen Bild, wurden genutzt, um die Arbeit mit den Trainingsdaten zu vereinfachen und das Modell stärker zu machen [YOLO Seite].

Insgesamt können die Erweiterungseinstellungen für YOLO angepasst werden, um den spezifischen Anforderungen des Datensatzes gerecht zu werden. Dabei kann mit verschiedenen Werten experimentiert werden, um die beste Modelleistung zu erhalten. Warum genau diese Argumente in der vorliegenden Arbeit genutzt wurden, wird in Abschnitt 4.1 bei der Hyperparameteranpassung erläutert. Neben den genannten Argumenten stehen ebenfalls Argumente für das Drehen des Bildes, eine zufällige perspektivische Transformation oder dem Beschneiden eines Klassifizierungsbildes zur Verfügung, um eine optimale Erweiterungsstrategie für die beste Modellleistung zu finden [YOLO Seite].

Originalbild

In Abbildung ? ist eine Matrix an Originalbildern aus dem Datensatz zu sehen, welche von verschiedenen Kameras aufgenommen wurden. Nachfolgend ist in Abbildung ? eine Reihe von Trainingsbildern zu sehen, welche die verschiedenen Anpassungen durch die „Data-Augmentation“-Erweiterungsargumente zeigt. Dabei ist erkennbar, wie sich die jeweiligen Bilder verändern und anschließend weiterverarbeitet werden.

Abbildung Trainingsbatches

Um nun mit dem Trainieren und Testen der Bilder beginnen zu können, muss sichergestellt sein, dass der Datensatz in Trainings- und Validierungsbilder unterteilt wurde, damit das Modell nicht überangepasst wird. Wie im vorherigen Abschnitt schon erläutert, wird das Modell insgesamt mit nur wenigen Bildern trainiert, um ein gutes Modell zu erzielen, welches später die unterschiedlichsten Ausführungen der Gebärdenzeichen erkennen kann.

Die Datenvorverarbeitung für den vorliegenden Datensatz „ALS Alphabet“, aber auch für die Nutzung ähnlicher Datensätze, umfasst eine Vielzahl von Schritten, um sicherzustellen, dass die Bilder in einer geeigneten Form vorliegen, um Modelle effektiv zu trainieren. Dazu gehören die Organisation und Bereinigung der Daten, die Skalierung und Normalisierung der Bilder, aber auch die Datenerweiterung und die sinnvolle Aufteilung der Daten in Trainings- und Testdaten. Diese sorgfältigen Vorbereitungen sind entscheidend, um genaue Gebärdenspracherkennungsmodelle zu entwickeln, die in realen Anwendungen eingesetzt werden können. Mit diesen Vorverarbeitungsschritten kann nun das Modell trainiert und getestet werden.