



# Blink Detector

## Seminararbeit

vorgelegt von

**Jan-Nicolas Weider, Felix Tischler und Mattis Dietrich Dietrich**

angefertigt am

**Lehrstuhl für Digitale Bildverarbeitung  
Fakultät für Mathematik und Informatik  
Friedrich-Schiller-Universität Jena**

Seminar: Seminar Rechnersehen

Betreuer: Niklas Penzel

Datum: 29. September 2023

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
1.1	Aufgabenstellung . . . . .	2
1.2	Motivation . . . . .	2
1.3	Ziele, Arbeitspakete und GANTT-Diagramm . . . . .	3
<b>2</b>	<b>Interface/Front-End</b>	<b>3</b>
<b>3</b>	<b>Theorie</b>	<b>3</b>
3.1	Gesichtserkennung . . . . .	4
3.2	Blinzeldetektion . . . . .	5
3.3	Features . . . . .	7
3.4	Kalibrierung . . . . .	8
3.5	Datensatz . . . . .	8
3.6	Klassifikation . . . . .	10
3.7	Evaluationsmetriken zur Beurteilung der Klassifikatoren . . . . .	11
<b>4</b>	<b>Evaluation</b>	<b>11</b>
4.1	Selbsttest zu Metriken und Frameworks . . . . .	11
4.2	Vergleich der Klassifikatoren . . . . .	13
4.3	Müdigkeitserkennung in der Anwendung . . . . .	15
<b>5</b>	<b>Lessons Learned</b>	<b>16</b>
5.1	Lessons learned . . . . .	16
5.2	Future Work . . . . .	16
<b>6</b>	<b>Zusammenfassung</b>	<b>16</b>
	<b>Literatur</b>	<b>17</b>
	<b>Abbildungsverzeichnis</b>	<b>19</b>

# 1 Einführung

Die Anwendung von digitaler Bildverarbeitung und maschinellem Lernen hat in den letzten Jahren eine transformative Bedeutung erlangt und spielt eine zunehmend entscheidende Rolle in einer breiten Palette von Anwendungsgebieten. Diese Technologien ermöglichen es, visuelle Informationen präzise zu analysieren, Muster zu erkennen und darauf basierend intelligente Entscheidungen zu treffen. Die Relevanz dieser Technologien erstreckt sich über eine Vielzahl von Anwendungsgebieten, von der Medizin über die Robotik bis hin zur Automobilindustrie und zur Sicherheitstechnik.

## 1.1 Aufgabenstellung

Im Rahmen der Bachelorstudiengänge Informatik und Angewandte Informatik an der Friedrich-Schiller-Universität Jena wird die Möglichkeit geboten, das Modul Projekt Intelligente Systeme zu belegen. Da in dem Bereich Rechnersehen und Mustererkennung bereits erste Kenntnisse vermittelt wurden und sich ein Interesse ausbilden konnte, ist es naheliegend, die erlangten theoretischen Fähigkeiten mit einem Projekt in die Praxis umsetzen. Infolgedessen entschied man sich für dieses Modul, um in Gruppenarbeit aktuelle Techniken und Methodiken der digitalen Bildverarbeitung und des maschinellen Lernens weiter zu erproben. Dabei sollten bestehende Verfahren nach Stand der Kunst genutzt und sinnvoll eingesetzt werden, um ein umfassenderes Bildverarbeitungsprogramm zu entwickeln. Der Umfang dieses Projektes reicht von der Projektplanung und Konzeptionierung über die Dokumentation bis zur Vorstellung der fertigen Anwendung und der Anfertigung eines Abschlussberichts. Dabei soll man sich zu Beginn mit der Formulierung der Projektziele nach der SMART-Methode auseinandersetzen. Anschließend sind Arbeitspakete und erste Zuständigkeiten auszuformulieren, welche in einem Gantt-Diagramm zur Planung wiederzufinden sind. Das Projekt wird in Gruppenarbeit angefertigt und bis zur Vorstellung dem Arbeitsplan gemäß entwickelt. Welche Thematik die Anwendung gezielt anspricht, ist den Gruppenmitgliedern freigestellt und unterliegt deren Interesse und Motivation.

## 1.2 Motivation

Durch die thematische Freistellung musste zu Beginn festgelegt werden, welcher Anwendungsbereich von Interesse ist. Schnell herrschte Einigkeit darüber, dass man etwas entwickeln möchte, was eine große Menge potenzieller Anwender anspricht. Das Projekt

sollte demzufolge ein Ergebnis liefern, welches im Alltag anwendbar ist und bestenfalls zusätzlich einen gesellschaftlichen Nutzen aufweist, in dem für bestimmte Vorgehen die Sicherheit erhöht wird. Eine Erhöhung der Sicherheit ist im stressgeprägten Alltag, wobei Vieles schnellstmöglich vonstattengehen soll, von großer Relevanz. Vor allem im Straßenverkehr herrscht durch die hohe Grundgeschwindigkeit ein erhöhtes Risiko, weshalb kurze Unaufmerksamkeiten, durch beispielsweise Sekundenschlaf, bereits schwerwiegende Folgen mit sich ziehen können. In diesem Bereich kann die Kontrolle des Aktivitätsstatus von großer Bedeutung sein, da hiermit die Sicherheit aller Verkehrsteilnehmer positiv beeinflusst werden kann. Eine weitere Anwendung eines solchen Müdigkeitsdetektors kann in der aktiven Kontrolle ausgewählter Mitarbeiter mit sicherheitsrelevanten Jobs sein, oder auch in Online-Klassen zur Aktivitätsüberwachung der Teilnehmer. All diese Anwendungsgebiete, welche einen Fußabdruck in der Gesellschaft hinterlassen können, motivieren uns dazu, einen Müdigkeitsdetektor im Rahmen dieses Projekts zu entwickeln. Das Programm soll über eine Webcam laufen, die Person wird detektiert und verschiedene signifikante Merkmale für Müdigkeitsanzeichen werden stetig überprüft. Weiterhin sollen potenzielle Ausfälle frühzeitig erkannt und durch eine Warnung verhindert werden, somit kann ein wesentlicher Teil zur Sicherheit beigetragen werden.

### **1.3 Ziele, Arbeitspakete und GANTT-Diagramm**

Optionaler Teil, eventuell auslassen, in Einführung hinten dran ballern Aus der gewählten Thematik leiten sich somit die Ziele ab, welche mit diesem Projekt verfolgt werden sollen. Diese gemeinsamen auszuarbeiten und zu besprechen war der Start des Moduls. Hierbei ERGEBNISSE PRÄSENTIEREN

## **2 Interface/Front-End**

## **3 Theorie**

Zur Müdigkeitserkennung im Zusammenhang mit Auto fahren gibt es verschiedene Methoden, wie sie von Arunasalam et al. [AYA<sup>+</sup>20] und Ramzan et al. [RKA<sup>+</sup>19] beschrieben werden. Diese lassen sich in drei Hauptkategorien unterteilen:

- Physiologische Methoden: Hierzu gehören Messungen wie Puls und EEG.
- Fahrverhaltensbasierte Methoden: Dies umfasst die Analyse des Fahrverhaltens mittels Sensoren am Auto.

- Verhaltensbasierte Methoden: Hierbei liegt der Fokus auf Beobachtungen von Auge, Gesicht und Kopfbewegungen.

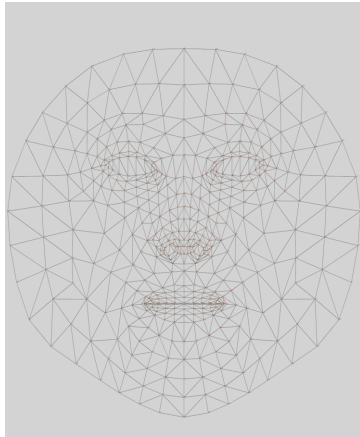
Die visuelle Analyse des Verhaltens bietet eine praktikable Lösung, da diese auch vielfältig einsetzbar ist, beispielsweise auch um Müdigkeit beim Lernen und Arbeiten zu erkennen.

Unser Hauptziel, wie bereits in der Aufgabenstellung erwähnt, besteht darin, eine frühzeitige Warnung auszugeben, wenn eine Person Anzeichen von Müdigkeit zeigt. Wir möchten damit eine Methode entwickeln, um Sekundenschlaf frühzeitig zu verhindern. Dies erfordert die Erkennung von Müdigkeit sowie die Identifizierung von Ablenkung (z. B. wenn eine Person zu lange nicht nach vorne schaut). Die Herausforderung besteht darin, den Zeitpunkt zu bestimmen, ab dem eine Person als müde gelten kann, und objektive Kriterien dafür zu entwickeln. Eine weitere wichtige Herausforderung besteht darin, dass Müdigkeit bei jeder Person unterschiedlich ist und sich visuelle Verhaltensweisen individuell manifestieren. Zum Beispiel variiert die Blinzelrate erheblich von Person zu Person, selbst bei Ruhe (Bentivoglio et al. [BBC<sup>+</sup>97] berichtet von 4 bis zu 48 Blinzeln pro Minute).

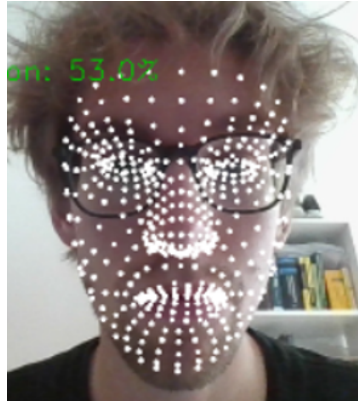
Unsere Vorgehensweise orientiert sich an Ghodoosian et al. [GGA] und verwendet deren Datensatz als Ausgangspunkt. Wir verarbeiten Videodaten einer Person und extrahieren Landmarks, um quantitative Merkmale zu berechnen. Unsere Schwerpunkte liegen auf den visuellen Eigenschaften des Auges, wobei die Erkennung von Blinzeln eine entscheidende Rolle spielt. Um die Ergebnisse auf individuelle Unterschiede abzustimmen, führen wir eine Kalibrierung durch, die absolute Daten in relative Werte umwandelt. Diese Merkmale werden dann für die Klassifikation von "müde/und" "nicht müde" verwendet, woraufhin bei "müde" immer wieder eine Warnung ertönt. Auch ertönt eine Warnung, falls eine Person in Sekundenschlaf verfällt und die Augen über einen längeren Zeitraum geschlossen sind.

### 3.1 Gesichtserkennung

Um verwertbare Daten zu generieren, ist die Echtzeit-Erkennung der Augenstruktur entscheidend, um Blinzler zu identifizieren, was die Grundlage für weitere Analysen bildet. Eine höhere Bildwiederholrate (auch Framerate; Bilder/Sekunde) ermöglicht präzisere Ergebnisse, obwohl die meisten Kameras und Videodaten üblicherweise eine Framerate von etwa 30 Bildern pro Sekunde haben. Es ist von Bedeutung, dass die Framerate durch die Berechnungen des Algorithmus nicht erheblich verlangsamt wird.



(a) Die Landmarks des Mediapipe Face Mesh mit Nummerierungen.



(b) Alle Landmarks des Face Mesh während der Anwendung.



(c) Die Augenlandmarks des Face Mesh zur Berechnung der EAR.

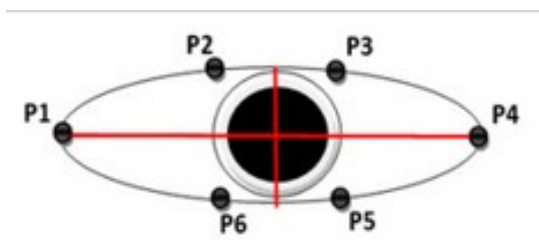
Abbildung 1: Prozess der Extraktion der entscheidenden Landmarks für unsere Müdigkeitsdetektion.

Ein leistungsstarkes und schnelles System dafür ist Mediapipe [LTN<sup>+</sup>]. Das Mediapipe Face Mesh erkennt in Echtzeit 468 3D-Gesichtslandmarks, selbst auf mobilen Geräten, und ermöglicht die Approximation einer 3D-Oberfläche des Gesichts durch maschinelles Lernen. Für unsere Anwendung verwendeten wir Version 0.9.0, um eine Integration mit der Kivy-Bibliothek zu ermöglichen, da eine neuere Version von Mediapipe auf unerklärliche Probleme stieß. In der Version 0.9.0 des Mediapipe Face Mesh bleibt die Grundeinstellung statisch und bewegt sich als Ganzes mit, was jedoch für unsere Untersuchung unzureichend ist. Daher muss die Einstellung "refine" auf "True" gesetzt werden, um 10 zusätzliche Landmarks und Bewegungen innerhalb des Gesichts zu erfassen. Darüber hinaus muss der "static image mode" auf "False" gesetzt werden, um das Face Mesh auf Videos anzuwenden.

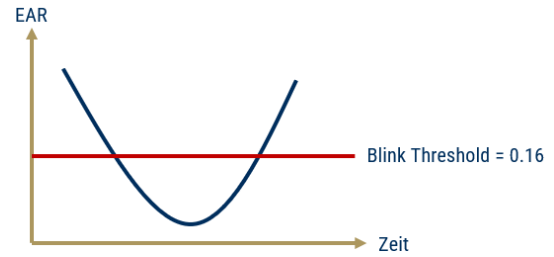
Wir haben auch andere Ansätze für ein Face Mesh evaluiert, darunter den Dlib Face Landmark Detector (Davis E. King [Kin09]), basierend auf Dalal et al. [DT05]. Im Vergleich zum Mediapipe Face Mesh erzielte dieser jedoch in Bezug auf Genauigkeit schlechtere Ergebnisse und zeigte gelegentlich Aussetzer. Weitere Details dazu finden sich in der Evaluation (siehe 4.1).

## 3.2 Blinzeldetektion

Durch die Berechnung von Landmarks können wir nun das Blinzeln einer Person erkennen. Eine gängige Methode hierfür ist die Berechnung des "Eye Aspect Ratio" (EAR), wie



(a) Die sechs Landmarks von P1 bis zu P6 in ihrer Anordnung am Auge [DCC<sup>+</sup>22]



(b) Vereinfachte Darstellung eines Blinzelvorgangs und der Auswirkung auf die EAR.

Abbildung 2: Prozess der Berechnung eines Blinzlers über die sechs Augenlandmarks (a) über die EAR Kurve und dem Blink Threshold (b)

von Soukupova [Sou16] präsentiert. Die EAR wird anhand des Auges und sechs Landmarks (P1 bis P6) ermittelt und entspricht dem Verhältnis der vertikalen Öffnung zur horizontalen Öffnung:

$$EAR = \frac{\|P2 - P6\| + \|P3 - P5\|}{2\|P1 - P4\|} \quad (1)$$

In dieser Formel repräsentieren P1 bis P6 die Koordinaten der Landmarks, wobei P1 und P4 die äußeren Augenwinkel sind, P2 und P6 die oberen Augenlider und P3 und P5 die unteren Augenlider (Abbildung 2a). Die  $\| \cdot \|$  Symbole stellen die euklidische Norm dar, die den Abstand zwischen den Punkten berechnet.

Die EAR ist stets größer, wenn das Auge geöffnet ist, im Vergleich dazu, wenn es geschlossen ist. Allerdings variieren diese Werte erheblich zwischen Personen, wie auch unser eigener Test zeigt, beispielsweise hat Jan-Nicolas Weider einen Wert von x und Mattis Dietrich einen Wert von y (Siehe blabla). Der Durchschnitts-EAR wird über beide Augen berechnet und ermöglicht die kontinuierliche Erfassung dieses Werts für jedes Einzelbild, was weitere Analysen ermöglicht:

$$\text{Durchschnittliche EAR} = \frac{EAR_{\text{linkes Auge}} + EAR_{\text{rechtes Auge}}}{2} \quad (2)$$

Bei einem Blinzeln zeigt die EAR-Kurve einen typischen Verlauf, der in Abbildung 2b dargestellt ist - sie sinkt zunächst parabelförmig ab und steigt dann wieder an. Ein wichtiger Aspekt ist nun der Schwellenwert, um zu erkennen, ab wann das Auge als "geschlossen oder öffnen" gilt, um ein Blinzeln zu detektieren und seine Dauer zu berechnen. In unseren Selbsttests (siehe 4.1) hat sich vorläufig ein fester Schwellenwert von 0.16 als am besten geeignet erwiesen. Sobald der EAR den Schwellenwert unterschreitet, wird

das Auge als geschlossen betrachtet, und wenn er ihn wieder überschreitet, wird der Vorgang als Blinzeln erkannt. Die Blinzzeit kann anhand der Anzahl der Frames ermittelt werden, in denen der EAR-Wert kleiner als 0.16 ist. Um verschiedene Bildwiederholungsraten auszugleichen, wird die Framerate des Videos verwendet, um einen zeitlichen Wert in Millisekunden zu erhalten. Dies ist entscheidend für die Trainingsphase des Klassifikators und die Anwendbarkeit über verschiedene Bildwiederholungsraten und Geräte hinweg. Allerdings sollten weitere Untersuchungen zur Verbesserung der Genauigkeit sowie zur Berücksichtigung individueller Unterschiede und möglicher Augenkrankheiten wie Grauer Star oder altersbedingter Verengung der Augen in zukünftigen Arbeiten durchgeführt werden. Eine agile Berechnung der EAR-Schwelle zu Beginn jeder Aufnahme für jede Person könnte eine Möglichkeit sein, wie es von Ghoddoosian et al. [GGA] in ihrem "Blink Retrieval Algorithm" beschrieben wird.

### 3.3 Features

Der nächste Schritt in unserem Verarbeitungsprozess besteht darin, Features zu sammeln. Insbesondere für das Auge stehen verschiedene mögliche Features zur Verfügung, und unterschiedliche Kombinationen können zu besseren Ergebnissen führen, wie von Dreißig et al. [Mar] untersucht wurde. Dreißig et al. hat auch Kopfbewegungen in seine Untersuchungen einbezogen. Aich Ebrahim [Par16] stellt 19 Features vor, die sich ausschließlich auf den Blinzelvorgang beziehen.

Für uns haben sich folgende Features bewährt: die einfache Betrachtung des "Eye Aspect Ratio" (EAR), die durchschnittliche Blinzdauer und der "Percentage of Eye Closure" (PERCLOS) Wert. Der EAR bildet die Grundlage für weitere Features und besitzt auch allein schon Aussagekraft. Wir betrachten die gemittelte EAR über eine Zeitspanne und vergleichen sie mit vorherigen Zeitspannen. Außerdem analysieren wir die gemittelte EAR über alle Werte kleiner als 0.16, um die Werte zu entfernen, wenn das Auge als geschlossen gilt.

Ein weiteres Feature ist die durchschnittliche Blinzdauer. Wenn eine Person müde wird, sollte die Blinzdauer länger sein. Daher speichern wir für jeden Blinzelvorgang die Dauer in Millisekunden und berechnen den Durchschnitt über eine Zeitspanne. Der PERCLOS-Wert, erstmals von Wierwille et al. [WWK<sup>+</sup>94] beschrieben, repräsentiert den Anteil der Zeit in einer Minute, in der die Augen zu mindestens 80 Prozent geschlossen sind. Bei müden Personen wird dieser Wert tendenziell größer. Für eine einfachere Berechnung verwenden wir den PERCLOS-Wert als den Anteil der Zeit in einer Minute, in der der EAR-Wert unterhalb des Schwellenwerts von 0.16 liegt, was darauf hinweist,



dass das Auge als geschlossen betrachtet wird.

Die Frage nach der Festlegung der Zeitspanne für diese Berechnungen stellt sich nun als nächstes. Unsere Recherchen ergaben, dass eine Dauer von 60 Sekunden als gängige Spanne angesehen wird, da Werte wie der PERCLOS-Wert pro Minute berechnet werden. Diese Berechnungen erfolgen überlappend, was bedeutet, dass bei jedem neuen Frame neue Werte für die Features generiert werden.

Ein Beispielhafter Feature-Vektor nach der Berechnung ist in Tabelle 1 zu erkennen.

	PERCLOS (%)	Blinzeldauer (ms)	EAR bei geöffnete Augen	EAR
Wach	0.0094	80.24	0.2639	0.2627
Fraglich	0.0879	174.77	0.2267	0.2188
Müde	0.1780	168.60	0.1985	0.1878

Tabelle 1: Beispielhafter Featurevektor

### 3.4 Kalibrierung

Eine spezifische Herausforderung, die wir in Anbetracht unserer Aufgabenstellung betrachten müssen, ist die Individualität in Bezug auf die Augenöffnung und das Verhalten bei Müdigkeit. Bei verschiedenen Personen zeigt sich Müdigkeit auf unterschiedliche Weisen: Bei einer Person kann die Blinzeldauer länger werden, während bei einer anderen Person eher die Blinzelhäufigkeit zunimmt. Zusätzlich variiert die EAR bei jeder Person erheblich, wie auch unser Selbsttest (siehe ??) bestätigt hat. Daher ist eine Relativierung der Werte unerlässlich.

Zu Beginn jeder Aufzeichnung gehen wir davon aus, dass die Person wach ist, und verwenden die ersten 60 Sekunden, um Referenzwerte für den Wachzustand zu erfassen. Diese Zeitspanne entspricht derjenigen, die für die weiteren Berechnungen verwendet wird. Die für jeden neuen Frame berechneten Werte werden dann in Beziehung zu den Werten des Wachzustands gesetzt. Auf diese Weise erhalten wir immer relative Werte für jedes einzelne Feature.

### 3.5 Datensatz

Bei dem ausgewählten Datensatz zum Trainieren des Klassifikators handelt es sich um den UTA Real-Life Drowsiness Dataset von Ghodoosian et al. [GGA]. Der Datensatz stand unglücklicherweise nur zu zwei Drittel zur Verfügung, da auf der offiziellen Seite der Veröfentlichenden technische Probleme zur Zeit der Ausarbeitung vorlagen. In dem

1- Extremely alert
2- Very alert
3- Alert
4- Rather alert
5- Neither alert nor sleepy
6- Some signs of sleepiness
7- Sleepy, no difficulty remaining awake
8- Sleepy, some effort to keep alert
9- Extremely sleepy, fighting sleep

Abbildung 3: Karolinska Sleepiness Scale (KSS)

verwendeten Datensatz befinden sich jeweils drei Videos von 48 Individuen. Die teilnehmenden Personen wurden so ausgewählt, dass verschiedene Ethnien, Altersgruppen und Geschlechter abgedeckt sind. Weiterhin trug ein Teil der Probanden eine Brille. Die Aufnahmen entstanden jeweils in natürlicher Umgebung mit verschiedenen Perspektiven, Hintergründen und Belichtungsverhältnissen. Pro Person ist ein Video für die Klasse „wach“, „fraglich“ und „müde“ gegeben.

Die drei Kategorien ergeben sich, wie in Abbildung 3 erkenntlich, durch eine Zusammenfassung des offiziellen Karolinska Sleepiness Scale (KSS). Die KSS wurde von Akerstedt et al. [AG90] zuerst vorgestellt. Als „wach“ werden die Stufen eins bis drei betitelt, wobei für „fraglich“ die Stufen 6 und 7 repräsentativ sind. Auf den Zustand „müde“ weisen die Einordnungen in die Stufen acht und neun hin. Die durchschnittliche Länge eines einzelnen Videos beträgt in etwa 10 Minuten, so dass die für die Klassifikation benötigten Featurewerte über einen längeren Zeitraum möglichst genau und fehlerunabhängig ermittelt werden können.

Zu beachten bei der vorliegenden Datengrundlage ist die Tatsache, dass die Videos über verschiedene Dateiformate, Frameraten und Längen verfügen. Einerseits sichert dies die Robustheit des Programms, um mit verschiedener Hardware umgehen zu können,

jedoch war andererseits für das automatisierte Einlesen und Featureextrahieren ein Mehraufwand benötigt.

Es wurde ein Skript erstellt, welches alle Videos mit den entsprechenden Datenformat aufruft und abspielt und dabei die Extraktion der Features über eine bestimmte Zeit laufen lässt, sodass aus jedem Video relevante Informationen gewonnen werden, alle Werte aus den Videos gleich stark gewichtet sind und zusätzlich kein Video vor einer vollständigen Extraktion aufgrund zu kurzer Aufnahmedauer eines Probanden abbricht.

### 3.6 Klassifikation

In Bezug auf die drei Videos pro Person setzen wir für jedes Feature den Zustand halbwach und müde in Relation zum Wachzustand und führen daraufhin Klassifikationen basierend auf den relativen Werten durch. Hierbei konzentrieren wir uns auf drei gängige Klassifikatoren und nutzen die Funktionen aus der Bibliothek scikit-learn von Pedregosa et al. [PVG<sup>+</sup>11]. Die Ergebnisse werden anhand der folgenden drei Klassifikatoren ausgewertet und verglichen:

- Logistische Regression: Für zwei Klassen wird die „One-Versus-Rest“ (ovr) Multi-Class-Strategie verwendet, während die Strategie für drei Klassen automatisch ausgewählt wird. Das Modell nutzt den 'lbfgs'-Solver und führt bis zu 1000 Iterationen durch, um das Modell zu trainieren.
- K-Nearest Neighbour mit K=3.
- Support Vector Machine.

Zudem untersuchen wir die Ergebnisse für drei Klassen (müde/fraglich/wach) sowie für zwei Klassen (müde/wach). Wir erwägen auch die Entwicklung eines Scores, haben diese Idee jedoch aufgrund der begrenzten Datenmenge verworfen, da uns lediglich Daten für die Zustände müde, fraglich und wach zur Verfügung standen, jedoch keine weiteren Zwischenwerte.

Die Details zu den Klassifikationsergebnissen werden in der Evaluation 4.2 genauer analysiert. Für unser Projekt hat sich jedoch der K-Nearest Neighbour als die beste Option erwiesen. Um das System weiter zu verbessern, wäre eine größere Datenmenge von entscheidender Bedeutung. Zusätzlich könnten auch weitere Klassifikatoren in Betracht gezogen werden.

### 3.7 Evaluationsmetriken zur Beurteilung der Klassifikatoren

Für die Evaluierung unserer Klassifikatoren haben wir verschiedene Metriken angewendet. Dabei haben wir die Funktionen aus der scikit-learn-Bibliothek von Pedregosa et al. [PVG<sup>+</sup>11] genutzt. Konkret haben wir den Precision Score, Recall Score, F1 Score, Accuracy Score und die Fehlermatrix (auch als Confusion Matrix bekannt) verwendet.

Der Precision Score gibt an, wie viele der vorhergesagten positiven Instanzen tatsächlich positiv sind. Sein Wert liegt zwischen 0 und 1, wobei 1 für eine perfekte Präzision steht.

Der Recall Score zeigt, wie viele der tatsächlich positiven Instanzen korrekt als positiv vorhergesagt wurden. Auch hier reicht der Wertebereich von 0 bis 1, wobei 1 für einen perfekten Recall steht.

Der F1 Score ist das harmonische Mittel zwischen Precision und Recall und bietet ein ausgewogenes Maß für die Modellleistung. Auch dieser Wert liegt zwischen 0 und 1, wobei 1 eine perfekte Balance zwischen Precision und Recall repräsentiert.

Der Accuracy Score gibt das Verhältnis der korrekt klassifizierten Instanzen zur Gesamtanzahl der Instanzen im Testdatensatz an, wobei höhere Werte auf eine bessere Klassifikationsgenauigkeit hinweisen.

Die Fehlermatrix, auch als Confusion Matrix bezeichnet, ist eine tabellarische Darstellung der Anzahl der wahren positiven, wahren negativen, falsch positiven und falsch negativen Vorhersagen eines Klassifikationsmodells. Sie bietet eine aufgeschlüsselte Übersicht über die Verteilung der Ergebnisse.

## 4 Evaluation

### 4.1 Selbsttest zu Metriken und Frameworks

Zur Evaluation verschiedener Schwellwerte zur Blinzeldetektion und Frameworks wurde einen Selbsttest im Wachzustand durchgeführt, wobei zu zweit in einer alltäglichen Arbeitsumgebung mit natürlicher Beleuchtung über die interne Webcam des eigenen Laptops das Programm circa eine Minute lang ausgeführt wurde. Von diesen Versuchen entnahmen wir die selbstgezählte reale und die detektierte Anzahl der Blinzelschläge in diesem Zeitraum für verschiedene Parametervoreinstellungen. Zusätzlich führten wir dies für zwei verschiedene Frameworks aus, um durch Selbsttests herauszufinden, welches für unseren Anwendungsbereich geeigneter ist. Für Proband 1 ist zu erkennen, dass MediaPipe im Durchschnitt eine signifikant bessere Detektionsrate aufweisen kann als Dlib. Hierbei ist es wichtig, dass weder zu viele noch zu wenig Blinzelschläge detektiert wer-

den. Da keine Toleranz für Fehldetektionen geplant war, ist dieser Wert zu optimieren. Für Probanden 2 sind ähnliche Ergebnisse zu verzeichnen, eine fehlerfreie Erkennung kann auch hier nur MediaPipe liefern. Jedoch ist zu bemerken, dass für diese Versuchsperson die optimale Detektion nur bei einem kleineren Schwellwert des EAR-Wertes gegeben ist. Zu begründen ist dies durch die generellen Unterschiede in Individuen sowie die persönliche Form und Anatomie des Auges [Verweis/Quelle]. Für jede Person ist der EAR-Wert im geöffneten Zustand verschieden, ein kleinerer Wert tendiert dazu, dass bei einem vergleichsweise hohen Schwellwert einige Blinzelschläge fälschlicherweise erkannt werden, da der Schwellwert bereits ohne ein Schließen des Auges unterschritten wird. Dies ist auch in dem Test aufgefallen, so wurden für einige Kombinationen zu viele Blinzelschläge detektiert, weshalb der Detektor nicht vertrauenswürdig wäre. Ein zu kleiner Schwellwert resultiert jedoch darin, dass aufgrund der möglicherweise geringen Frame-rate der Videos ein Blinzeln nicht erkannt wird, wenn nicht zufälligerweise ein Bild bei der vollständigen Schließung des Auges aufgenommen wird. Nach Beachtung dieser Zusammenhänge und Auswertung der Tabelle des Selbsttest ist der Entschluss zu ziehen, dass die Benutzung von MediaPipe als Framework das geeignetere Mittel ist. Weiterhin stellte sich der Schwellwert für die Blinzeldetektion von 0,16 als passend heraus, da dieser verschiedene Probleme minimiert. Für diesen Wert wurde die höchste Genauigkeit erreicht, da sowohl die Anzahl der falschpositiven als auch falschnegativen Werte für das ausgewählte Framework gleich Null ist. Somit ist zu sagen, dass die implementierte Detektion verlässlich und genau arbeitet.

Tabelle einfügen

Neben der Anzahl der Blinzelschläge wurden zusätzlich die durchschnittlichen EAR- und PERCLOS-Werte betrachtet, um eventuelle Zusammenhänge nachvollziehen zu können. Hierbei ist deutlich zu entnehmen, dass der EAR-Wert für Proband 2 deutlich kleiner und der PERCLOS-Wert deutlich größer als die des Probanden 1 sind. Dies spiegelt genau die Zusammenhänge der Metriken und der verschiedenen geeigneten Schwellwerts wider, da für Proband 2 ein kleinerer Schwellwert besser geeignet ist, was auf ein generell schwächer geöffnetes Auge hinweist. Dies würde in einem geringeren EAR-Wert resultieren, was die Daten verifizieren. Weiterhin sollte der PERCLOS-Wert höher sein, je größer die Anzahl an Blinzelschlägen oder je länger die Blinzeldauer pro Zeitspanne ist, da der Anteil, in dem die Augen geschlossen sind, größer ist. Unsere Beobachtungen stimmen mit diesem Verhalten überein, für Proband 1 wurde ein kleinerer Wert als für Proband 2 ermittelt.

Klassifikator	F1-Score	fRecall	Precision	Accuracy
KNN Klassifikator mit $k = 3$	0.743962	0.758621	0.801724	0.680542
Logistische Regression	0.663643	0.689655	0.671121	0.602709
Support Vector Machine	0.552622	0.62069	0.79454	0.489655

Tabelle 2: Ergebnisse der drei verglichenen Klassifikatoren über die vorgestellten Metriken für die drei Klassen wach, fraglich und müde.

## 4.2 Vergleich der Klassifikatoren

Wie in der Theorie im Abschnitt zur Klassifikation beschrieben, wurden verschiedene Klassifikatoren getestet, um den am besten geeigneten für unsere Anwendung zu ermitteln. Dabei haben wir sowohl den Fall mit zwei Klassen (müde und wach) als auch den mit drei Klassen (müde, fraglich und wach) getestet. Die Metriken, die wir verwendet haben, wurden im Abschnitt zur Klassifikationsbewertung vorgestellt.

Die Trainingsdaten für die Klassifikatoren wurden aus den berechneten Feature-Vektoren abgeleitet, die aus den Videos des verwendeten Datensatzes gewonnen wurden. Insgesamt hatten wir 144 Feature-Vektoren, davon jeweils 48 für die Zustände müde, fraglich und wach, jeweils einmal für jeden Probanden. Um die Feature-Vektoren zu erstellen, haben wir die ersten sechs Minuten jedes Videos für die Kalibrierung genutzt. Die Aufteilung in Test- und Trainingsdatensätze erfolgte zufällig über die 144 Feature-Vektoren, unabhängig von der Person. Die Werte für halbwach und müde wurden im Verhältnis zu den Wachwerten gesetzt, und danach wurden die Klassifikatoren trainiert. Dies führte dazu, dass für jedes Feature ein Wach-Wert von eins vorlag, was die Ergebnisse der Klassifikatoren beeinflusste und bessere Ergebnisse für die verwendeten Metriken lieferte. Trotzdem ergibt die Klassifikation Sinn, da sie eine sinnvolle Trennung zwischen den Klassen ermöglicht und eine signifikante Abweichung von Wach-Werten für die Klassifikation von fraglich und müde aufzeigt.

Die Ergebnisse zeigen im direkten Vergleich der Klassifikatoren für drei Klassen:

- Alle Metriken sind bei dem KNN-Klassifikator am besten, wobei die Accuracy und die Precision die höchsten Werte aufweisen.
- Bei der Logistic Regression ist die Accuracy höher (0.60) als bei der SVM (0.49), jedoch ist die Precision niedriger (0.67 gegenüber 0.79).
- Der F1 Score und der Recall sind ebenfalls beim KNN-Klassifikator am höchsten.

Ein Blick auf die Confusion Matrizen eines Durchlaufs der Crossvalidierung lohnt sich, insbesondere wenn man bedenkt, dass eine Fehlklassifikation von wach/fraglich als

KNN Klassifikator mit k=3			
	vorhergesagt wach	vorhergesagt fraglich	vorhergesagt müde
Wahr wach	13	0	0
Wahr fraglich	0	3	6
Wahr müde	0	1	6

Support Vector Machine			
	vorhergesagt wach	vorhergesagt fraglich	vorhergesagt müde
Wahr wach	13	0	0
Wahr fraglich	8	1	0
Wahr müde	3	0	4

Logistische Regression			
	vorhergesagt wach	vorhergesagt fraglich	vorhergesagt müde
Wahr wach	13	0	0
Wahr fraglich	2	3	4
Wahr müde	1	2	4

Tabelle 3: Confusion Matritzen für die drei Klassifikatoren für die drei Klassen wach, fraglich und müde.

Klassifikator	F1-Score	Recall	Precision	Accuracy
KNN Klassifikator mit k = 3	1	1	1	0.947368
Logistische Regression	1	1	1	0.904094
Support Vector Machine	0.88797	0.894737	0.908772	0.712865

Tabelle 4: Ergebnisse der drei verglichenen Klassifikatoren über die vorgestellten Metriken für die zwei Klassen wach und müde.

müde in unserem Fall weniger problematisch ist als die Fehlklassifikation von müde als wach/fraglich. Da wir vor Sekundenschlaf warnen möchten, ist eine frühzeitige fehlerhafte Warnung weniger problematisch. Natürlich unter der Berücksichtigung, dass zu viele Warnungen als "nervig empfunden werden könnten. Die Confusion Matrix für den KNN-Klassifikator zeigt beispielsweise, dass 6 fragliche Videos als müde klassifiziert wurden, während bei der SVM und der Logistischen Regression eine größere Streuung zu sehen ist: Bei der SVM wurden 8 Videos als wach statt fraglich klassifiziert, und bei der logistischen Regression 2 als wach statt fraglich. Dies ist in unserem Fall ungünstig. Der KNN-Klassifikator hat keine Videos fehlerhaft als wach klassifiziert, sondern lediglich ein Video als fraglich statt müde. Daher zeigt sich der KNN-Klassifikator als die beste Wahl zur Detektion von Müdigkeit.

Der Vergleich von zwei Klassen führte aufgrund der in Relation gesetzten Werte zu

KNN Klassifikator mit k=3		
	vorhergesagt wach	vorhergesagt müde
Wahr wach	13	0
Wahr müde	0	6

Support Vector Machine		
	vorhergesagt wach	vorhergesagt müde
Wahr wach	13	0
Wahr müde	0	6

Logistische Regression		
	vorhergesagt wach	vorhergesagt müde
Wahr wach	13	0
Wahr müde	2	4

Tabelle 5: Confusion Matritzen für die drei Klassifikatoren für die zwei Klassen wach und müde.

einem noch besseren Ergebnis, und dennoch konnte auch hier eine klare Trennung erreicht werden. Die logistische Regression und der KNN-Klassifikator haben beide sehr gut abgeschnitten, wobei die Accuracy beim KNN-Klassifikator höher war als bei der logistischen Regression, was bedeutet, dass der KNN-Klassifikator in allen Durchläufen eine bessere Klassifikation durchgeführt hat. Ein Beispiel aus einer Crossvalidierungsrunde zeigt, dass der KNN-Klassifikator und die logistische Regression in diesem Fall alles richtig klassifiziert haben, wie durch die Werte von eins bei den F1-, Recall- und Precision-Werten gezeigt wird. Die SVM hat in diesem Fall jedoch 2 Werte falsch klassifiziert. Hier ist der Accuracy Score ausschlaggebend.

### 4.3 Müdigkeitserkennung in der Anwendung

In der Anwendung erfolgt die Kalibrierung des Wachzustands in den ersten 60 Sekunden, in denen die Feature-Vektoren für den Wachzustand berechnet werden. Es sei denn, die Person dreht sich zur Seite oder bewegt sich aus dem Blickfeld, was die Kalibrierung anhält. Wenn die Person für mehr als 3 Sekunden nicht nach vorne schaut, ertönt ein Warnton, um den Fahrer zur Konzentration auf die Straße und erhöhte Aufmerksamkeit aufzufordern, da Ablenkung ebenfalls zu Unfällen führen kann.

Nach der Kalibrierung wird für jedes Einzelbild der aktuelle Feature-Vektor neu berechnet und im Verhältnis zu den Wachwerten gesetzt. Wenn der KNN-Klassifikator eine signifikante Abweichung von den Wachwerten feststellt, ändert sich die Farbe um das



Video herum, zuerst zu Orange (halbwach) und anschließend zu Rot (müde). Um abrupte Änderungen, wie sie in unseren Tests auftraten, auszugleichen, wird der Medianwert (wach/fraglich/müde) der letzten 60 Sekunden als maßgebend betrachtet.

Wenn eine Person über einen längeren Zeitraum im müden Zustand verbleibt, erfolgt eine wiederholte Warnung zur Müdigkeit. Dies geschieht alle zehn Sekunden, um eine angemessene Häufigkeit sicherzustellen.

Zusätzlich wird eine Warnung ausgegeben, wenn eine Person unabhängig von ihrem Status die Augen zu lange geschlossen hält (bei uns zwei Sekunden).

Unsere Tests haben gezeigt, dass unser Müdigkeitsdetektionskonzept eine solide Grundlage bietet, jedoch die verschiedenen Parameter nicht für jede Person optimal sind. Es gibt auch feststellbare Abweichungen in Bezug auf den Zeitpunkt, zu dem Warnungen ausgegeben werden, was vermutlich auf gelegentliche Berechnungsfehler zurückzuführen ist. Dies sind normale Schwankungen, die in Zukunft genauer betrachtet werden sollten. Trotzdem funktioniert der Übergang zwischen den Zuständen wach/fraglich/müde, auch wenn der fragliche Zustand manchmal sehr kurz ausfällt.

## **5 Lessons Learned**

### **5.1 Lessons learned**

### **5.2 Future Work**

## **6 Zusammenfassung**

## Literatur

- [AG90] AKERSTEDT, T. ; GILLBERG, M.: Subjective and objective sleepiness in the active individual. In: *The International journal of neuroscience* 52 (1990), Nr. 1-2, S. 29–37. <http://dx.doi.org/10.3109/00207459008994241>. – DOI 10.3109/00207459008994241. – ISSN 0020–7454
- [AYA<sup>+</sup>20] ARUNASALAM, M. ; YAAKOB, N. ; AMIR, A. ; ELSHAIKH, M. ; AZAHAR, N. F.: Real-Time Drowsiness Detection System for Driver Monitoring. In: *IOP Conference Series: Materials Science and Engineering* 767 (2020), Nr. 1. <http://dx.doi.org/10.1088/1757-899X/767/1/012066>. – DOI 10.1088/1757-899X/767/1/012066. – ISSN 1757–8981
- [BBC<sup>+</sup>97] BENTIVOGLIO, A. R. ; BRESSMAN, S. B. ; CASSETTA, E. ; CARRETTA, D. ; TONALI, P. ; ALBANESE, A.: Analysis of blink rate patterns in normal subjects. In: *Movement disorders : official journal of the Movement Disorder Society* 12 (1997), Nr. 6, S. 1028–1034. <http://dx.doi.org/10.1002/mds.870120629>. – DOI 10.1002/mds.870120629. – ISSN 0885–3185
- [DCC<sup>+</sup>22] DEWI, Christine ; CHEN, Rung-Ching ; CHANG, Chun-Wei ; WU, Shih-Hung ; JIANG, Xiaoyi ; YU, Hui: Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety. In: *Electronics* 11 (2022), Nr. 19, S. 3183. <http://dx.doi.org/10.3390/electronics11193183>. – DOI 10.3390/electronics11193183
- [DT05] DALAL, N. ; TRIGGS, B.: Histograms of Oriented Gradients for Human Detection. In: SCHMID, Cordelia (Hrsg.): *Proceedings, IEEE Computer Society*, 2005. – ISBN 0–7695–2372–2, S. 886–893
- [GGA] GHODDOOSIAN, Reza ; GALIB, Marnim ; ATHITSOS, Vassilis: *A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection*. <http://arxiv.org/pdf/1904.07312v1>
- [Kin09] KING, Davis E.: Dlib-ml: A Machine Learning Toolkit. In: *Journal of Machine Learning Research* 10 (2009), 1755–1758. [http://dlib.net/face\\_landmark\\_detection.py.html](http://dlib.net/face_landmark_detection.py.html)
- [LTN<sup>+</sup>] LUGARESI, Camillo ; TANG, Jiuqiang ; NASH, Hadon ; MCCLANAHAN, Chris ; UBOWEJA, Esha ; HAYS, Michael ; ZHANG, Fan ; CHANG, Chuo-Ling ; YONG, Ming G. ; LEE, Juhyun ; CHANG, Wan-Teh ; HUA, Wei ; GEORG, Manfred ; GRUNDMANN, Matthias: *MediaPipe: A Framework for Building Perception Pipelines*. <https://arxiv.org/pdf/1906.08172>
- [Mar] MARIELLA DREISSIG, MOHAMED HEDI BACCOUR, TIM SCHÄCK, ENKELEJDA KASNECI: Driver Drowsiness Classification Based on Eye Blink and Head Movement Features Using the k-NN Algorithm.

- [Par16] PARISA EBRAHIM: Driver drowsiness monitoring using eye movement features derived from electrooculography: Doktorarbeit. (2016)
- [PVG<sup>+</sup>11] PEDREGOSA, F. ; VAROQUAUX, G. ; GRAMFORT, A. ; MICHEL, V. ; THIRION, B. ; GRISEL, O. ; BLONDEL, M. ; PRETTENHOFER, P. ; WEISS, R. ; DUBOURG, V. ; VANDERPLAS, J. ; PASSOS, A. ; COURNAPEAU, D. ; BRUCHER, M. ; PERROT, M. ; DUCHESNAY, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830
- [RKA<sup>+</sup>19] RAMZAN, Muhammad ; KHAN, Hikmat U. ; AWAN, Shahid M. ; ISMAIL, Amina ; ILYAS, Mahwish ; MAHMOOD, Ahsan: A Survey on State-of-the-Art Drowsiness Detection Techniques. In: *IEEE Access* 7 (2019), S. 61904–61919. <http://dx.doi.org/10.1109/ACCESS.2019.2914373>. – DOI 10.1109/ACCESS.2019.2914373
- [Sou16] SOUKUPOVA, Tereza: Eye-Blink Detection Using Facial Landmarks. In: *Research Reports of CMP* 5 (2016)
- [WWK<sup>+</sup>94] WIERWILLE, Walter W. ; WREGGIT, Steven S. ; KIRN, Chris ; ELLSWORTH, Lynne A. ; FAIRBANKS, Rollin J.: Research on vehicle-based driver status/performance monitoring; Development, validation, and refinement of algorithms for detection of driver drowsiness. Final report, 1994

**Abbildungsverzeichnis**

1    Prozess der Extraktion der entscheidenden Landmarks für unsere Müdigkeitsdetektion.    5

2    Prozess der Berechnung eines Blinzlers über die sechs Augenlandmarks

     (a) über die EAR Kurve und dem Blink Threshold (b)    . . . . .    6

3    Karolinska Sleepiness Scale (KSS) . . . . .    9