

Bericht zum vierten Meilenstein

Arbeitsfortschritt

Für den letzten Meilenstein würden im Deployment noch weitere Anpassungen und Optimierungen vorgenommen:

Darkmode-Implementierung: Es wurde erfolgreich ein Darkmode in die App integriert. Benutzer haben nun die Möglichkeit, das Erscheinungsbild der App anzupassen und zwischen einem hellen und einem dunklen Modus zu wählen.

Layout-Überarbeitung: Das Layout der App wurde umfassend überarbeitet. Es wurden verschiedene visuelle Verbesserungen vorgenommen, einschließlich einer neuen Schriftart, eines überarbeiteten Logos und neuer Designelemente. Diese Änderungen tragen dazu bei, die Benutzererfahrung zu verbessern und das Erscheinungsbild der App ansprechender zu gestalten.

Helpscreen-Implementierung: Ein Helpscreen wurde hinzugefügt, in dem Benutzer eine Erklärung zur Verwendung der App finden können. Zudem wurde ein Verweis auf das Repository hinzugefügt, sodass Benutzer bei Bedarf weitere Informationen und Ressourcen finden können.

Optimierung der Übergänge zwischen den Screens: Die Übergänge zwischen den verschiedenen Screens der App wurden optimiert. Dies umfasst Animationen und Effekte, die den Wechsel zwischen den Screens flüssiger und ansprechender gestalten.

Anpassung des Layouts: Es wurde eine neue Funktionalität hinzugefügt, mit der in der "MainMediapipe.py"-Datei dynamisch festgelegt werden kann, welche Farben, Schriftarten, Buttons und andere Elemente verwendet werden sollen. Dadurch wird es möglich, das Layout der App flexibel anzupassen und weitere Anpassungen vorzunehmen.

Seitens der Entwicklung gab es weitere Fortschritte.

In der Anwendung sind nun weitere Funktionalitäten implementiert. Wenn die Landmarks im Gesicht nicht erkannt werden können, so wird dies detektiert und es wird ein Warnsound abgespielt, um auf dieses Problem hinzuweisen.

Ein Trainingskript wird nun verwendet, um Videos als Eingabe nehmen zu können und eine Liste von Featurevektoren zu generieren. Diese Featurevektoren werden für das Training des Klassifikators verwendet.

Der Klassifikator kann einer der drei Klassen zurückgeben: "müde", "fraglich" oder "wach". Die Zuordnung basiert auf den berechneten Featurevektoren und wird in der Anwendung verwendet. Bei der Zuweisung zur Klasse "müde" wird ein Warnsound ausgegeben, um auf die Müdigkeit hinzuweisen.

Ein spezielles Skript wird für den Klassifikator verwendet, um Daten zu sammeln und Verhältnisse der Features zu berechnen. Diese Verhältnisse entsprechen einer Art Trendanalyse, bei der auf die Startwerte durch die Berechnung von Verhältnissen normalisiert wird, anstatt nur Differenzen zu verwenden.

Der Featurevektor wurde angepasst und enthält bisher nur Informationen zu PERCLOS und open EAR.

Es gibt eine Funktion zur Klassifikation, die den besten Klassifikator auswählt. Dabei werden verschiedene Klassifikatoren wie Logistische Regression, K-Nearest Neighbours mit $k=1$ und $k=3$ sowie SVM (Support Vector Machine) verwendet. Die Genauigkeit der Features und Klassifikatoren kann angezeigt werden.

Es wurde auch eine Regression getestet, um kontinuierliche Werte (Müdigkeits-Score) anstelle von Klassen zu erreichen. Aufgrund der Unsicherheit, insbesondere aufgrund des kleinen Testdatensatzes, wird vorerst jedoch nur die Klassifikation in Klassen verwendet.

Erkenntnisse

Um eine höhere Genauigkeit zu erreichen, ist es wichtig, das Training auf einem großen Datensatz durchzuführen. Es besteht die Möglichkeit, dass sich die Genauigkeiten der Klassifikatoren ändern und andere Klassifikatoren möglicherweise besser geeignet sind, sobald mit einem großen Datensatz trainiert wird.

Es könnte auch sinnvoll sein, weitere Features zu implementieren und zu testen. Durch die Kombination verschiedener Features könnte die Genauigkeit verbessert werden, dies ist bereits in Bearbeitung.

Bei Verwendung eines kleinen Datensatzes zeigte sich, dass der 3NN (3-Nearest Neighbours) Klassifikator die beste Leistung erbrachte. Zusätzlich ergab sich, dass die Information des open EAR Features im Vergleich zu PERCLOS keinen zusätzlichen Nutzen brachte.

In Bezug auf die Genauigkeit der verwendeten Features (vermutlich) gilt folgendes:

$$\text{PERCLOS} > \text{PERCLOS} + \text{open_EAR} > \text{open_EAR}$$