

## Bericht zum zweiten Meilenstein

### Arbeitsfortschritt

Ziel des zweiten Meilensteins seitens des Deployments ist es, die Integration von Mediapipe und OpenCV in das Kivy-Framework zu untersuchen und die App erfolgreich als apk-Datei bereitzustellen. Dabei wurden auch Herausforderungen im Zusammenhang mit dem Kamerazugriff unter Windows adressiert und alternative Lösungsansätze evaluiert.

Der aktuelle Stand ist wie folgt:

1. Integration von Kivy und Dlib: Es wurde erfolgreich Dlib in Kivy integriert, wodurch es möglich ist, den Code auszuführen und die GUI anzuzeigen. Allerdings besteht noch das Problem, dass das Bereitstellen der App als apk-Datei aufgrund von Inkompatibilitäten nicht reibungslos funktioniert.
2. Integration von Mediapipe und OpenCV: Zur Lösung der Bereitstellungsprobleme wurde die Integration von Mediapipe und OpenCV in das Kivy-Framework untersucht. Dabei wurde das Framework "Mediapipe for Android" verwendet, um eine apk-Datei zu erstellen. Es besteht jedoch noch die Herausforderung, Mediapipe und OpenCV erfolgreich in Kivy zu integrieren, um die gewünschte Funktionalität zu erreichen.
3. Nutzung eines Docker-Containers: Um die Erstellung von apk-Dateien mit Mediapipe for Android zu ermöglichen, wurde ein Docker-Container eingerichtet, der die erforderlichen Abhängigkeiten enthält.

Für die Weiterentwicklung der Algorithmen zur Müdigkeitserkennung wurde zuletzt daran gearbeitet, das Framework Mediapipe einzubinden und entsprechend zu nutzen. Prinzipiell ist die Funktionsweise von Mediapipe bereits gegeben, jedoch noch nicht in gemeinsamer Bedingung mit Kivy.

Weiterhin wurde Zeit in das Verbessern des Algorithmus zur Müdigkeitserkennung investiert, in dem bereits begonnen wurde, ein neues Feature zu implementieren. Dabei handelt es sich um PERCLOS, was den prozentualen Anteil der Frames mit geschlossenen Augen gemessen an den gesamt aufgenommenen Frames widerspiegelt. Das Testen und Interpretieren dieser Werte sind hierbei noch zu bearbeitende Schritte.

### Erkenntnisse

Bei der Integration von Mediapipe und OpenCV in Kivy sind noch, wie uns erkenntlich wurde, einige herausfordernde Schritte zu bewältigen. Es wird intensiv nach Lösungsansätzen gesucht, um die App erfolgreich zum Laufen zu bringen. Unter Windows besteht unter anderem die Schwierigkeit, dass der Kamerazugriff zwischen Mediapipe und Kivy nicht reibungslos funktioniert. Aus diesem Grund wurde beschlossen, ein natives Linux-System auf

einem Laptop zu verwenden, um die App unter optimalen Bedingungen zu testen. Dadurch soll eine umfassende Testumgebung geschaffen werden, um die Landmarkenerkennung und die gesamte App-Funktionalität erfolgreich zu überprüfen.

Es wird weiterhin daran gearbeitet, die Integrationsprobleme zwischen Mediapipe, OpenCV und Kivy zu lösen. Dazu werden verschiedene Ansätze und Lösungen untersucht, um die Funktionalität der App sicherzustellen. Sobald die Integration von Mediapipe und OpenCV in Kivy erfolgreich abgeschlossen ist, wird die weitere Entwicklung der App vorangetrieben. Das langfristige Ziel besteht darin, die App erfolgreich als apk-Datei bereitzustellen, um sie auf Android-Geräten nutzen zu können, worauf unsererseits noch hingearbeitet werden muss.

Noch ausstehende Schritte in der Entwicklung des Algorithmus sind die Evaluation beider Frameworks (Dlib und Mediapipe), um Vor- und Nachteile jeweils genauer herauszufinden und dokumentieren zu können. Weiterhin ist deutlich geworden, dass basierend auf dem EAR (Eye Aspect Ratio) noch viele Möglichkeiten zur Einschätzung der Müdigkeitsveränderung einer Zeitreihe gegeben sind. So werden wir beispielsweise dynamische Schwellenwerte anstelle eines Festwertes einführen, um personenunabhängiger das Blinzeln detektieren zu können und somit Individualitäten ansatzweise auszugleichen. Außerdem soll eine EAR-Trendanalyse folgen, in welcher nicht nur wie bisher der aktuelle EAR-Wert betrachtet wird, sondern dessen Veränderung in einem bestimmten Zeitrahmen. Ergeben diese Analysen einen Trend zur Abnahme des EAR-Wertes, so weist dies auf zunehmende Müdigkeit hin und kann potenzielle Ausfälle rechtzeitig vermeiden.

Weiterhin ausstehend sind jedoch Tests zur genaueren Evaluierung unserer Ideen sowie das genauere Untersuchen des Kombinierens unserer Algorithmen mit physiologischen Parametern. So könnte, sollte die Genauigkeit nicht den Vorstellungen entsprechen, über das Einbinden einer Pulsanalyse, noch eine weitere Informationsquelle einbezogen werden.