Verwendung von künstlicher Intelligenz und Computer Vision bei der Bewegungsanalyse von Hochleistungskanuten/innen - die nächste Ausbaustufe

Marc Schuh[1], Jonas Mayer[1], Thomas Endres[1]

[1]TNG Technology Consulting, Unterföhring, Germany

Email: marc.schuh@tngtech.com

Kanu, KI, Computer Vision, neuronale Netze

Einführung

Die visuelle Technik-Analyse im Hochleistungs-Kanusport ist bislang ein manueller und zeitaufwändiger Prozess. Hierbei werden zunächst die Athleten von einem Beiboot aus im Kanu gefilmt, das Videomaterial wird im Nachgang durch Sportwissenschaftler*innen ausgewertet. Untersucht wird der Winkel des Paddels zur Wasseroberfläche bei der Paddelbewegung, genauer gesagt in vier diskreten Positionen (siehe Abbildung 1), im nachfolgenden auch vereinfachend "Paddelposen" genannt. Die erzielten Paddelwinkel werden mit jeweiligen Sollwinkeln des Technik-Leitbilds verglichen, um Rückmeldung zur Verbesserung der Paddeltechnik an die Kanutin / den Kanuten zu gegeben.

Hierfür muss aktuell das gesamte Video Einzelbild für Einzelbild nach den Paddelposen durchsucht werden, um anschließend händisch Wasserlinie und Paddellinie einzuzeichnen. Dies kommt einem menschlichen Aufwand von 15 Minuten pro analysiertem Aufwand gleich. Unser Ziel ist es, diesen manuellen Aufwand zu automatisieren.

Abbildung 1: Die vier Paddelposen nach Technikleitbild zeigen das Einstechen, vollständige Eintauchen, Auftauchen, Ausstechen.

Bei der Konferenz Spinfortec 2020 konnten wir einen ersten Prototypen zeigen, der die Paddelposen findet, sowie den Paddelwinkel via Wasser- und Paddellinie erfasst. Bei der Vorstellung 2020 berichteten wir, dass die Paddelwinkel-Erkennung robust funktionierte, aber

dennoch in 20 - 30% Fälle fehlerhaft war. Die Paddelposen-Erkennung entsprach hingegen nur in 37% der Fälle auf +/- 1 Bild der menschlichen Auswertung.

In diesem Vortrag stellen wir die Verbesserungen vor, die dazu geführt haben, dass die Paddelwinkel-Erkennung in >99% der Fälle funktioniert und die Paddelposen-Erkennung im angegebenen Intervall zu 60% korrekt ist.

Methode

Wie auch in unserem ersten Prototypen haben wir die Analyse-Aufgabe in 3 unabhängige Unteraufgaben zerteilt:

1. Bestimmung der Wasserlinie
2. Bestimmung der Paddellinie
3. Finden der Paddelposen

Im Folgenden gehen wir auf den Ansatz, die technische Umsetzung, sowie die Verbesserungen der einzelnen Teilschritte ein.

Bestimmung der Wasserlinie

Wir orientieren uns hier an der Lösung von von Braun et al [1]: Zunächst wird das Kanu mithilfe eines eigens trainierten neuronalen Netzes (MaskRCNN [2]) vom Hintergrund segmentiert, was eine Bestimmung der Kanu-Umrisse erlaubt.

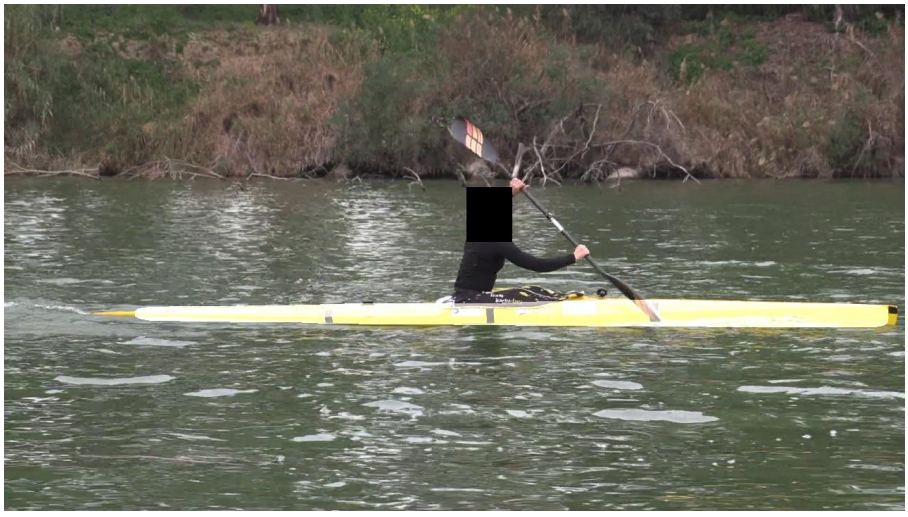

Abbildung 2: Es wird eine Kanu-Athletin auf dem Wasser gezeigt. Der durch das MaskRCNN als Vordergrund erkannte Kanuteil ist hell hervorgehoben.

Mithilfe der gefunden Umrisse, kann die Wasserlinie durch eine Gerade am unteren Rand des Kanus approximiert werden.

Bestimmung der Paddellinie

Zunächst muss die Position der Handmittelpunkte schrittweise bestimmt werden: Im ersten Schritt wird die Athletin / der Athlet grob mithilfe eines Personendetektors (YOLOv3 [3]) geortet. Auf diesen Bildausschnitt wird eine Körperposen-Erkennung (TransPose [4]) angewandt, die insbesondere die genaue Position der Handgelenke bestimmt. In einem letzten Schritt wird mittels einer Handposen-Erkennung (DeNa [5]) die Position der Handmittelpunkte erkannt.

Abbildung 3: Die Heatmeap des Personendetektors TransPose zeigt auf der Athletin die verschiedenen Gelenke. Für uns besonders interessant sind die Positionen der Handgelenke.

Um eine robuste Schätzung der Handposition zu gewährleisten, auch wenn die Hand kurzzeitig nicht sichtbar ist, werden die Handpositionen über die zeitliche Historie gefiltert. Dazu nähern wir ein Polynom mindestens dritten und maximal achten Grades auf diskrete Zeitfenster der Handpositionen an. Durch Gewichtung der einzelnen Beobachtung anhand der Sicherheit des Posen-Erkenners, können unsichere und fehlende Detektionen herausgefiltert werden.

Unsere Erfahrungen aus dem ersten Prototypen haben gezeigt, dass eine Approximation des Paddels nur über die Handposition in einzelnen Randfällen erhebliche Ungenauigkeiten verursachen kann. Um diese zu minimieren setzen wir jetzt auch eine weitere, direkte Erkennung des Paddels. Unsere Paddelerkennung basiert auf der Annahme, dass ein Paddel in der Regel gerade und ungefähr parallel zur Linie zwischen den beiden Händen ist. Wir nutzen diese beiden Annahmen, um mit einer Linienerkennung zwischen den Händen direkt die Umrisse des Paddels im Bild zu finden. Das Paddel wird als Durchschnitt der gefundenen Linien festgelegt.

Auch die Position und Rotation des Paddels wird wieder über ein zeitliches Fenster interpoliert. Die Gewichtungen der einzelnen Beobachtungen ergeben sich aus der Anzahl der gefundenen Linien. Dies ermöglicht uns eine robuste Approximation der Paddelposition, auch wenn das Paddel gerade verdeckt, oder die Hände nicht zu sehen sind (siehe Abbildung 4).

Abbildung 4: Durch die Paddelmitte verläuft eine parabelförmige braune Linie mit blauen Punkten. Dies ist die interpolierte Bewegung der Paddelmitte.

Aus der Paddellinie und der Wasserlinie kann nun der Einstichwinkel berechnet werden.

Finden der Paddelpose

Das Problem, die Paddelpose zu finden, reduzieren wir wie im ersten Prototypen wieder auf das Erkennen von einzelnen Paddelzuständen. Wir unterscheiden hier zwischen drei möglichen Paddelzuständen:

1. Paddel komplett über Wasser (a)
2. Paddelblatt teilweise eingetaucht (p)
3. Paddelblatt komplett eingetaucht (u)

Die vier möglichen Übergänge zwischen diesen Zuständen korrespondieren zu den vier Paddelposen des Technikleitbilds. Zur Erkennung des Paddelzustands haben wir einen temporär konsistenten Zustands-Erkenner basierend auf MobilenetV2 [6]. In unserem Training kommt unser Zustandserkenner auf 95% Präzision auf zuvor ungesehenen Daten (siehe Abbildung 5).

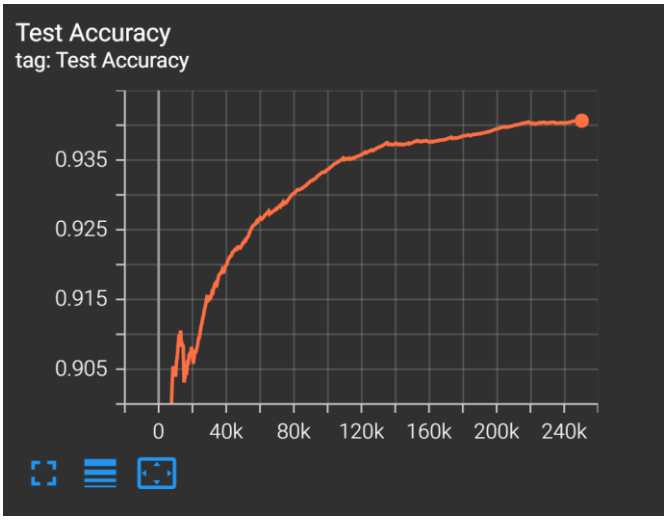

Abbildung 5: Es zeigt die Genauigkeit des neuronalen Netzes in der Erkennung der Paddelzustände in einem Trainingsvideo im Verlauf über verschiedene Trainingsiterationen.

Mithilfe der bereits erkannten Paddellinie und der Position der Hände kann als erstes ein grober Ausschnitt des Paddels vorgenommen werden. Um zeitliche Zusammenhänge abbilden zu können, werden die Paddelausschnitte von drei aufeinander folgenden Einzelbildern als Input in den Paddelzustands-Erkenner gegeben.

Um vereinzelte Fehldetektionen auszugleichen und eine robuste Erkennung des Zustandsübergangs zu ermöglichen, filtern wir auch über die zeitliche Historie der Paddelzustände. Unter der Annahme, dass nur bestimmte Zustandsübergänge physikalisch möglich sind (z.B. p → u) und ein Zustand auch für eine gewisse Mindestzeit gehalten werden muss, können wir die Zustandsverteilungen über das zeitliche Fenster mit Sigmoiden annähern. Dadurch werden einzelne Falschdetektionen (a→ p → a) und unmögliche Zustandsübergänge (a→u) effektiv herausgefiltert. Auch wird für alle Einzelbilder mit einem nach oben zeigenden Paddel angenommen, dass das Paddel über Wasser ist.

Die vier Paddelposen werden von unserem System im Einzelbild vor einem Zustandsübergang erkannt.

Technische Umsetzung

Unsere Software ist in Python implementiert und verwendet einige Open Source Bibliothek wie OpenCV und TensorFlow. Im Gegensatz zum ersten Prototypen arbeitet unser Analysator nicht

Bild für Bild, sondern in mehreren Analyse-Schritten über das gesamte Video. Dies ermöglicht das Berücksichtigen von zeitlichen Abhängigkeiten.

Die einzelnen Analyse-Schritte sind als Module implementiert, die nacheinander über das zu analysierende Video laufen und teilweise mit vorherigen Zwischenergebnissen arbeiten. Die Zwischenergebnisse mit sämtlichen Analysedaten werden in einem persistenten Speicherstand gehalten.

Am Ende der Analyse wird ein Zusammenfassung mit Paddelwinkeln, Paddelzuständen und Zeitstempeln im csv-Format ausgegeben. Zusätzlich werden die ausgewählten Einzelbilder der Paddelposen mit Wasserlinie, Paddellinie, Soll- und Ist-Winkel annotiert.

Ergebnisse

Das neuronale Netz wurde auf zehn Videos, die nicht im Trainingssatz enthalten waren, angewendet und verglichen, um wie viele Bilder der detektierte Paddelzustandwechsel von einer menschlichen Detektion abweicht (siehe Tabelle 1.).

	Exakt	**+/-1**	**+/-2**	**Mehr als 2**	**Verpasste Detektion**	**TOTAL**
Summe	54	87	60	35	16	236
Prozent	22.88%	36.86%	25.42%	14.83%	6.78%	

Tabelle 1: Vergleich der automatischen Paddelzustandserkennung mit einer manuellen Auswertung. Der Fehler in der Winkelbestimmung liegt bei etwa 3° pro Bild.

Diskussion und Ausblick

Gegenüber der im September 2020 vorgestellten Software konnten wir wesentliche Verbesserungen erreichen. Die Paddelschafterkennung ist robust und fällt nur noch in Extremsituationen, wenn zum Beispiel mehrere Kanuten im Bild sind aus. Die Paddelzustandserkennung hat sich von 37% auf 60% verbessert, wenn weiterhin angenommen werden kann, dass auch der Mensch bei einer manuellen Annotation gelegentlich ein Bild falsch liegt. Besonders der Wechsel in den vollständig eingetauchten Zustand bzw. gerade auftauchend ist selbst bei menschlichen Analysen fehleranfällig.

Insgesamt ist die Paddelzustandserkennung jedoch noch nicht vollständig zufriedenstellend. Auch wenn das neuronale Netz im Training 95% der Bilder richtig erkannt, bezieht sich dies aber auf den kompletten Zyklus der Paddelbewegung. Die für uns interessanten und scheinbar auch schwieriger zu analysierenden Einzelbilder um die vier Paddelposen herum sind in unserem bisherigen Datensatz nur zu 5-10% repräsentiert. Unsere Beobachtungen zeigen, dass sich aber gerade hier Fehler in der Detektion häufen. Weitere Faktoren wie unterschiedliche Bewegungsunschärfe, Kanu- und Wasserfarbe sowie eine reflektierende Wasseroberfläche bringen weitere Veränderungen in die Bilder, mit denen das neuronale Netz umgehen muss.

Um die Paddelposen-Findung zu verbessern, schlagen wir eine direkte neuronale Detektion der Paddelposen vor, ohne über einen Paddelzustand zu gehen. Außerdem könnte das Training stark von mehr Trainingsdaten und konsistenteren Aufnahmebedingungen profitieren.

References

1. von Braun, M., Frenzel, P., Käding, C., Fuchs, M., 2020, https://arxiv.org/abs/2004.09573
2. He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017, https://arxiv.org/abs/1703.06870
3. Redmon, J., Farhadi, A, 2018, https://arxiv.org/abs/1804.02767
4. Yang, S., Quan, Z., Nie, M., Yang, W., 2020, https://arxiv.org/abs/2012.14214
5. Cao, Z., Simon, T., Wei, S., Sheikh, Y., 2016, https://arxiv.org/abs/1611.08050

6. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L., 2018, https://arxiv.org/abs/1801.04381
7. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2017, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, 2018
8. Shi, J., Tomasi, C. 1994 https://ieeexplore.ieee.org/document/323794
9. Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y., 2018, https://arxiv.org/abs/1812.08008
10. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018, https://arxiv.org/abs/1801.04381