Latenzarme Erkennung von Wassersprungbewegungen in Trainingsvideos

Pramod Murthy[1,2], Bertram Taetz[1,2] & Didier Stricker[1,2]

[1]DFKI, Kaiserslautern

[2]TU Kaiserslautern

E-Mail: pramod.murthy@dfki.de

Schlüsselwörter: Turmspringen, KI, Deep learning, Big Data, Computer Vision

Einleitung

Die Analyse der Körperhaltung und Bewegung von Sportlern ist ein aufstrebendes Forschungsgebiet im Bereich Computer Vision und Deep Learning (Le, V.H 2020, Zhang et. al 2017, Zhang 2019, Cust et. al 2019). Die Analyse von Sportvideos ist eine anspruchsvolle Aufgabe, da die Körperhaltungen und Manöver in Hochleistungssportarten wie Wasserspringen, Turnen und Schwebebalken von Natur aus komplex sind. Darüber hinaus leidet die Bildgebung mit monokularen RGB-Kameras aufgrund der hohen Geschwindigkeit von Sportaktionen häufig unter Selbstausschluss und Bewegungsunschärfe (Nabali 2017). Die Sportvideoanalyse zielt darauf ab, objektive Messwerte zum Vergleich der Leistung von Sportlern zu liefern. So kann die Videoanalyse beispielsweise Trainern helfen, in den frühen Phasen des Trainings sofortiges und genaues Feedback zu geben. Dies wiederum führt zu weniger Fehlern und kürzeren Trainingszyklen für die Athleten während der Vorbereitung auf den Wettkampf. So wird die Gesamteffizienz des Trainingsprozesses erhöht (Parmar et. al. 2019].

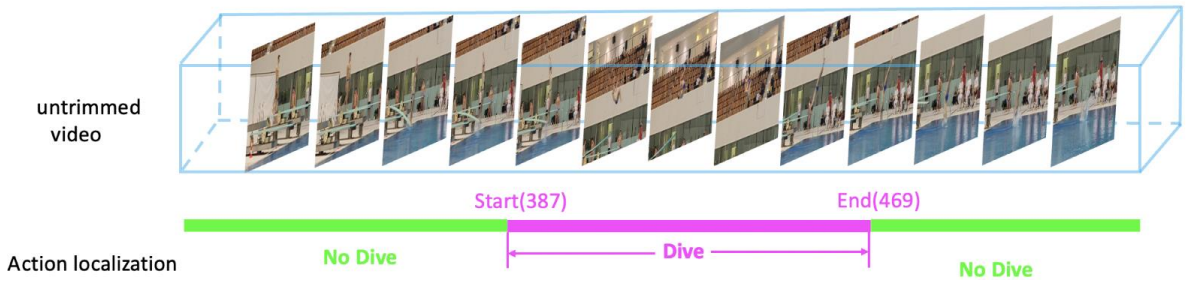


Abbildung 1. Die Abbildung zeigt die Videozeitleiste auf der horizontalen Achse. Der Beginn und das Ende eines Wassersprungs sind durch die jeweiligen genauen Bildnummern für eine Beispielvideoaufnahme gekennzeichnet.

In dieser Arbeit konzentrieren wir uns in erster Linie auf die latenzarme Erkennung von Anfang und Ende einzelner Wettkampfsprünge des Wasserspringens, aus einem Videostream. Die Klassifizierung von Sprünge und Unterabschnitte, wie beispielsweise in (Kanoji et.al 2019), können hierbei nachgelagert betrachtet werden. Ein häufig vorkommendes Szenario während der Trainingseinheiten ist das Trainieren einer Gruppe von Sportlern, indem diese kontinuierlich nacheinander Sprünge ausführen. Unser Ziel ist es, den Beginn und das Ende der Sprungbewegung für jeden Sportler in einem Videostream zu lokalisieren, wie in Abbildung 1 dargestellt. Die Latenzzeit soll hierbei weniger als 25 Millisekunden (ms) betragen.

Methode

Das Wasserspringen beinhaltet komplexe Körperbewegungen, die von einem Athleten in einer kurzen Zeitspanne (1-3 Sekunden) ausgeführt werden. Wir wenden unsere Methode auf Videostreams an, die mit einer Bildfrequenz von 60 Hz laufen. Die Aufgabe ist, die Sprungbewegung zu erkennen, indem wir das Start- und Endbild, dieser, in den aufgezeichneten Videos abgrenzen (siehe Abbildung 2).
Wir definieren das *Startbild*, als jenes, bei dem der Sportler keine Kraft mehr auf das Sprungbrett ausübt. Dies wird visuell markiert. Das *Endbild* ist dadurch definiert, dass der Sportler das Wasser berührt. Als Nächstes verwenden wir eine Drei-Frame-Schrittfunktion, um die Grenzklassen für jede Sprungbewegung darzustellen.

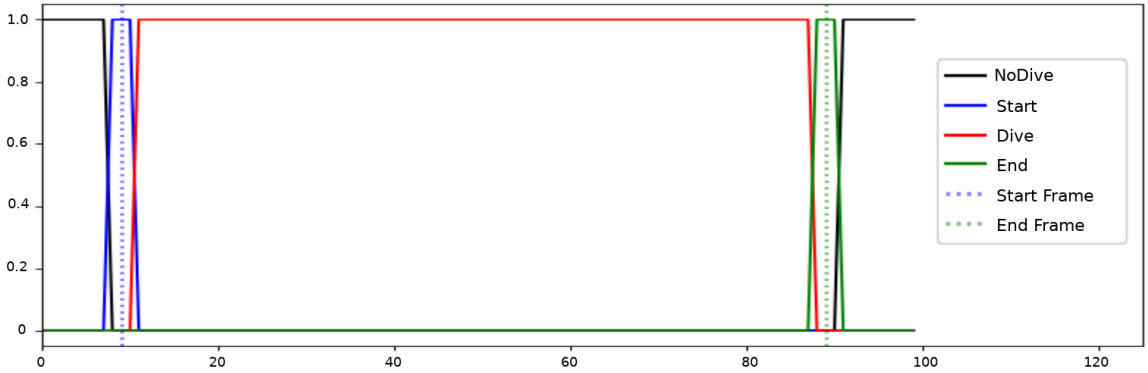
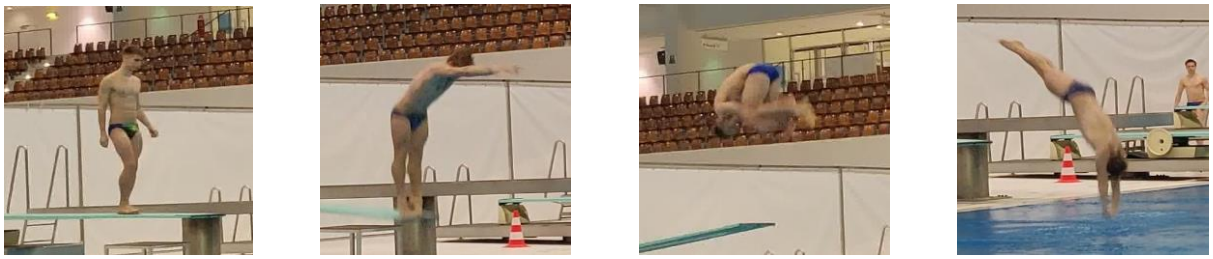


Abbildung 2. Die Abbildung veranschaulicht die kontinuierliche Markierung von Frames in 4 verschiedenen Klassenlabels für Wassersprungvideos. Die gepunktete Linie zeigt die genauen Frames der Grenzklassen für die Sprungaktion.

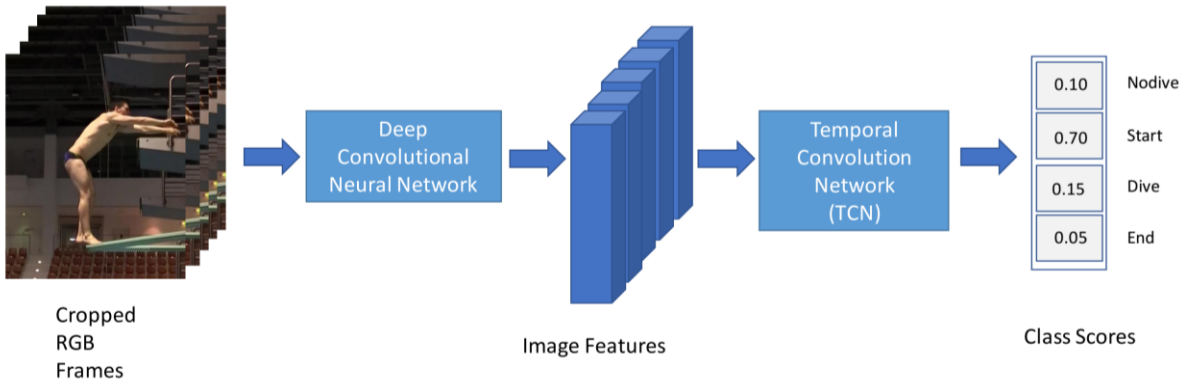
Wir fügen außerdem zwei weitere Klassen hinzu, nämlich "NoDive" und "Dive". Die Klasse "Dive" repräsentiert alle Frames, bei denen sich der Sportler in der Luft befindet, und die Klasse "No Dive" repräsentiert Frames, bei denen der Sportler entweder auf dem Brett steht oder ins Wasser eingetaucht ist. Die verschiedenen Klassen, sind in Abbildung 3 dargestellt.



(a) NoDive **(b) Start** **(c) Dive** **(d) End**

Abbildung 3. Illustration der Videobilder, die jede Klasse bei der Klassifizierung von Wassersprungbewegungen darstellen.

Wir haben einen Videodatensatz mit 450 gelabelten Videos, der 15 Stunden Videodaten umfasst, wobei jeder Bewegungsclip etwa 2 Minuten dauert. Jedes Video ist mit 4 Klassen gekennzeichnet. Der Datensatz umfasst 5 verschiedene Sprungbrett-Höhen: 1 Meter, 3 Meter, 5 Meter, 7,5 Meter und 10 Meter. Die Videos des Datensatzes wurden aus verschiedenen Blickwinkeln aufgenommen, um die Sprungbewegung zu erfassen. Bei einer Videosequenz werden die zugeschnittenen Bilder durch ein künstliches neuronales Netzwerk mit unterschiedlichen internen Strukturen geleitet.



Die Abbildung 4. veranschaulicht die Gesamtarchitektur für die Wassersprungerkennung. Zunächst lokalisieren wir die Person mit einem quadratischen Begrenzungsrahmen. Wir schneiden die im Bild dargestellte Person aus und nutzen diese als Eingabe für das Deep Convolutional Neural Network. Die extrahierten Merkmale für jedes Bild werden verkettet und an das Temporal Convolutional Neural Network (TCN) weitergeleitet, um die endgültigen Klassenwerte zu erhalten.

Die wesentlichen Blöcke sind in Abbildung 4 dargestellt. Bei einer Videosequenz, extrahiert das Merkmalsextraktionsnetzwerk (Kanazawa 2018) Merkmale für jedes Bild (Enkodierung). Die Merkmale werden in zeitliche Fenster mit 13 Frames aufgeteilt und als Eingabe für den zeitlichen Encoder (Koh 2021) verwendet, der aus mehreren TCN-Blöcken besteht. Je höher die Anzahl der TCN-Blöcke ist, desto größer ist das rezeptive Feld, d. h. das Netzwerk kann mehr zeitliche Merkmale in den Bildern berücksichtigen. Die Ausgabe des letzten TCN-Blocks sind T Merkmale für jedes Element der Sequenz, die verkettet und durch mehrere 1D-Faltungsschichten transformiert werden, um die Anzahl der Merkmale auf 512 zu verkleinern, diese wird als Eingabe für einen Linearen Layer genutzt. Schließlich wird die Softmax-Funktion auf die Ausgaben der linearen Schichten angewendet, welches das Ableiten von Klassifikationswahrscheinlichk

Tab. 1. *Ergebnisse der Studie für verschiedene Arten von Merkmalsextraktoren mit unterschiedlicher Eingabelänge. TCN-5 zum Beispiel ist ein Merkmalsextraktionsmodell mit fünf Bildern als Eingabe. Die Metrik ist die Klassifizierungsgenauigkeit pro Bild, je höher, desto besser. Die besten Ergebnisse sind hervorgehoben.*

	Detection Class				**Mean Accuracy**
	NoDive	**Start**	**Dive**	**End**	
TCN-5	76.9	93.7	92.9	90.9	88.6
TCN -9	92.1	92	95.1	89.2	92.1
TCN -13	95.2	93.4	95.2	89.7	**93.37**
TCN -17	91.5	94.5	**96**	90.6	93.15
TCN -21	**88.1**	**95.7**	93.5	**90.9**	92.05

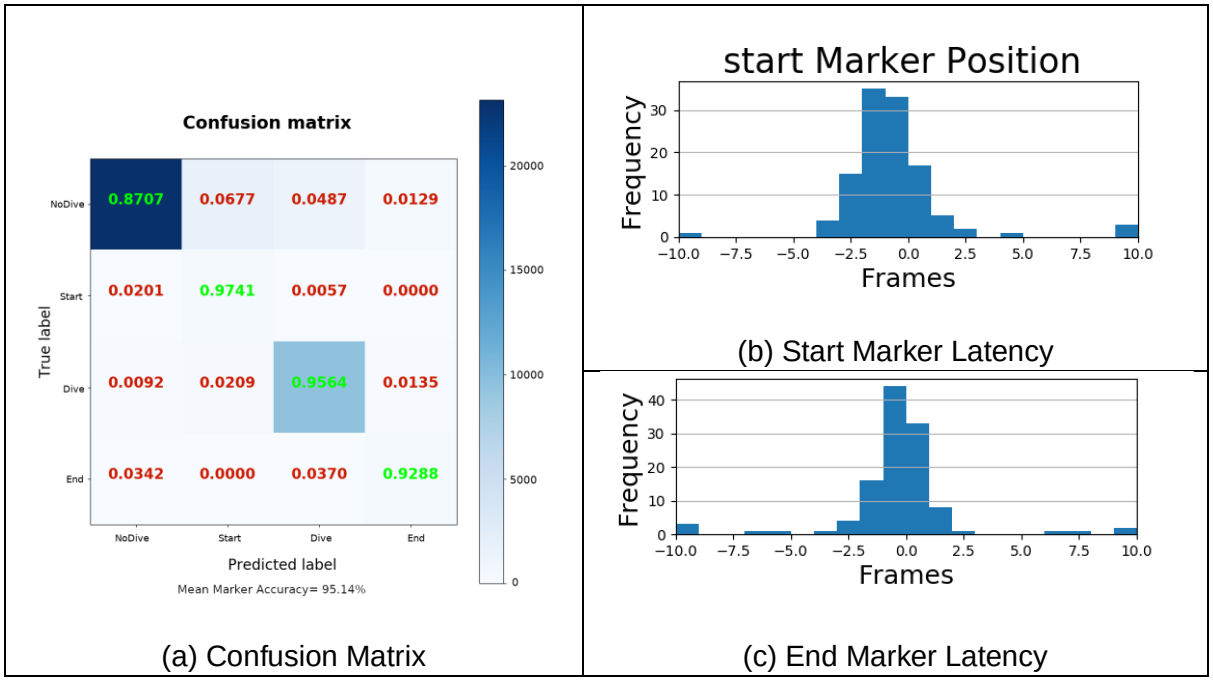
Wir verwenden den zeitlichen TCN-Codierer, um ein zeitliches rezeptives Feld zu erhalten. Hierbei hat sich über Kreuzvalidierung die Länge von 13 als beste herausgestellt. Die 1D-Faltungsschichten haben eine Kernelgröße von 3 und eine Padding-Größe von 2. Die Dilatationsgröße wird auf 1 gesetzt. Wir trainieren alle Netzwerke für 50 Epochen mit einer Batch-Größe von 16. Wir nutzen den Adam Optimierer (Kingma 2015) für die Optimierung der

Netze. Die Dropout-Rate wird auf 0,5 gesetzt. Die Lernrate und die L2 Regularisierung werden auf 1.e-5 bzw. 1.e-3 gesetzt. Wir nutzen eine gewichtete Kreuzentropieverlustfunktion mit den folgenden Gewichten für die verschiedenen Klassen [.15, .35, .15, .35].

Ergebnisse

Die folgenden Diagramme zeigen die Ergebnisse von Experimenten mit verschiedenen Sequenzlängen als Eingabe für das TCN Netz mit zeitlicher Faltung. Wir trainieren den mit der folgenden randomisierten Aufteilung: 300 Sequenzen für das Training, 50 Sequenzen für die Validierung und 100 Sequenzen für den Test. Tabelle 1 zeigt die Ergebnisse der Experimente mit einer unterschiedlichen Anzahl von Einzelbildern als Eingabe. Die Größe des zeitlichen Fensters, das dem zeitlichen Netzwerk als Eingabe zugeführt wird, beeinflusst die Leistung des Modells. Wir haben verschiedene Modelle mit N= 5, 9, 13, 17, 21 trainiert. Die Modelle mit der Eingabefenstergröße von 13 erlangt die höchste Durchschnittsgenauigkeit.

Diskussion



(a) Confusion Matrix

(b) Start Marker Latency

(c) End Marker Latency

Abbildung 5. Die Grafik (a) zeigt die Konfusionsmatrix der verschiedenen Erkennungsklassen. Die Abbildungen (b) und (c) zeigen Histogramme des Latenzmaßes der geschätzten Start- (oben) bzw. End-Grenzmarker (unten). Die y-Achse steht für die Anzahl der Sequenzen und die x-Achse für die Verzögerung in Frames. Die negativen Werte auf der x-Achse stehen für die Vorhersage vor dem Referenzframe und die positiven Werte für die Vorhersage nach dem Referenzframe.

Wir bewerten unser Modell zur Lokalisierung von Wassersprungaktionen anhand von zwei Metriken: (i) Klassifizierungsgenauigkeit des Bildes und (ii) Latenzzeit bei der Positionierung von Grenzmarkierungen.

Klassifizierungsgenauigkeit: Die mittlere Klassifizierungsgenauigkeit ist das Verhältnis zwischen der Anzahl der korrekt klassifizierten Frames und der Gesamtzahl der Eingabeframes, die Grenzklassen darstellen. Aufgrund des Vorhandenseins von stark unausgewogenen Klassen (Frames der Start- und Endklasse mit sehr geringer Repräsentation) kann die mittlere Genauigkeit irreführend und eine falsche Metrik sein. Zu

diesem Zweck analysieren wir die Leistung eines Modells mit einer Konfusionsmatrix wie in Abb. 5(a) dargestellt. Die Konfusionsmatrix zeigt deutlich eine geringe Anzahl von Fehlklassifizierungen für alle Klassen, was für eine genaue Lokalisierung der Sprungbewegung unerlässlich ist. Welche Klassen richtig und welche falsch vorhergesagt werden und welche Art von Fehlern gemacht werden.

Erkennungslatenz: Wir bewerten auch die Latenz der vorhergesagten Grenzmarkierungen des trainierten Modells. Die Latenz gibt zusätzliche Informationen darüber, wie weit die Start- und Endmarkierungen zeitlich von der Referenz entfernt sind. Bei einer Sequenz, bei welcher der Startmarker das Bild des Videos ist und das Netzwerk das Bild als Startmarker vorhersagt, wird die Sequenz als korrekt erkannt, wenn die Latenz innerhalb eines Schwellenwerts von einem Frame liegt. Wir messen den Prozentsatz der Testsequenzen, die innerhalb des Schwellenwerts für die geringe Latenz liegen.

Danksagungen

Diese Forschung wurde freundlicherweise vom DSV unterstützt. Wir möchten uns auch bei Dr. Thomas Köthe und seinen Kollegen am Institut für Angewandte Trainingswissenschaft bedanken. Herr Dr. Köthe hat uns sehr mit seinem immensen Wissen und seiner Erfahrung bei der Recherche geholfen. Wir möchten ihm auch für seine Vorschläge zur Verwendung korrekter Fachbegriffe danken, die das Manuskript erheblich verbessert haben.

Literatur

Kolotouros, N.; Pavlakos, G.; Black, M.J.; Daniilidis, K. (2019). Learning to reconstruct 3D human pose and shape via model-fitting in the loop. Proceedings of the IEEE International Conference on Computer Vision, 2252–2261.

Zhang, J.Y.; Felsen, P.; Kanazawa, A.; Malik, J. (2019). Predicting 3d human dynamics from video. Proceedings of the IEEE International Conference on Computer Vision, 7114–7123

Zhang, W.; Liu, Z.; Zhou, L.; Leung, H.; Chan, A.B. (2017). Martial arts, dancing and sports dataset: A challenging stereo and multi-view dataset for 3d human pose estimation. Image and Vision Computing (61), 22–39.

Le, V.H. 3-D Human Pose Estimation in Traditional Martial Art Videos. (2020) . International Journal of Machine Learning and Computing,10.

Nibali, A.; He, Z.; Morgan, S.; Greenwood, D. (2017) Extraction and classification of diving clips from continuous video footage. Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 38–48.

Cust, E.E.; Sweeting, A.J.; Ball, K.; Robertson, S. (2019). Machine and deep learning for sport-specific movement recognition: A systematic review of model development and performance. Journal of sports sciences, 37, 568–600.

Parmar, P.; Morris, B.T. (2019). What and How Well You Performed? A Multitask Learning Approach to Action Quality Assessment. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Kanazawa, A., Black, M. J., Jacobs, D. W., & Malik, J. (2018). End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (7122-7131).

Koh, B. H. D., Lim, C. L. P., Rahimi, H., Woo, W. L., & Gao, B. (2021). Deep temporal convolution network for time series classification. *Sensors*, *21*(2), 603.

Kanojia, G.; Kumawat, S.; Raman, S. (2019) Attentive spatio-temporal representation

learning for diving classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.

Kingma, D.P.; Ba, J. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations.