

Mask RCNN für Multi-Object Pose Estimation

Mert Ali Özmeral¹ and Jerome-Pascal Habanz¹ *

¹Fachbereich MNI, University of Applied Sciences Mittelhessen

Zusammenfassung

Diese Arbeit konzentriert sich auf die Multi-Object Keypoint Detection unter Verwendung der Mask Region-based CNN (Mask RCNN)-Architektur. In einer Ära zunehmender Bild- und Videodatenkomplexität liegt der Fokus auf der präzisen Lokalisierung spezifischer Punkte auf Objekten. Unsere Motivation besteht darin, die Leistungsfähigkeit von Mask RCNN in diesem Kontext zu vertiefen. Die Untersuchung beinhaltet eine eingehende Analyse der Mask RCNN-Architektur, das Training eines spezifischen Keypoint-Detection-Modells und einen Vergleich mit anderen Modellen. Die strukturellen Elemente von Mask RCNN, insbesondere das Region Proposal Network (RPN) und das Region-based CNN, stehen dabei im Fokus. Wir evaluieren die Skalierbarkeit und Anpassungsfähigkeit des Modells an verschiedene Szenarien sowie seine Genauigkeit und Effizienz im Vergleich zu bestehenden Modellen.

Einleitung

Die zunehmende Komplexität von Bild- und Videodaten erfordert fortschrittliche Methoden für die präzise Erfassung und Analyse von visuellen Informationen. Die Keypoint-Detektion spielt dabei eine entscheidende Rolle, indem sie die genaue Lokalisierung spezifischer Punkte auf Objekten ermöglicht. In diesem Kontext hat sich der Einsatz von Convolutional Neural Networks (CNNs) als äußerst effektiv erwiesen. Eine spezifische Architektur, die in diesem Zusammenhang große Beachtung findet, ist das Mask RCNN. Die Motivation hinter dieser Untersuchung liegt in der Notwendigkeit, die Leistung und Effizienz von Mask RCNN im Bereich der Multi-Object Pose Estimation zu vertiefen. Die erfolgreiche Entwicklung und Anwendung solcher Modelle sind entscheidend für Anwendungen in der Computer Vision, Ro-

botik und autonomen Systemen. Durch die präzise Identifizierung von Keypoints können komplexe Szenarien besser verstanden und interpretiert werden, was wiederum zu verbesserten Entscheidungsprozessen führt.

Verwandte Arbeiten

Unter den verwandten Arbeiten sind frühere Architekturen wie die RCNN-Architektur zu erwähnen, die einen bedeutenden Beitrag zu diesem Forschungsfeld geleistet haben. Die RCNN-Architektur, wie von Girshick et al. vorgeschlagen [1], markiert einen Meilenstein in der Entwicklung von Objekterkennungssystemen. Ihre grundlegende Struktur hat den Weg für weitere Innovationen gegeben.

Darüber hinaus sind auch die Architekturen der Fast und Faster RCNN zu nennen, die zu signifikanten Verbesserungen geführt haben. Die Arbeit von Girshick [2] und die von Ren et al. [3] haben durch die Einführung schnellerer und effizienterer Mechanismen

*Autoren:

mert.ali.ozmeral@mni.thm.de

jerome-pascal.habanz@mni.thm.de

für die Regionproposals die Leistungsfähigkeit der Objekterkennung erheblich gesteigert.

Schließlich sei die Mask RCNN-Architektur erwähnt, die bedeutende Fortschritte im Bereich der Instanzsegmentierung erzielt hat, wie von He et al. beschrieben [4]. Mit der Fähigkeit, nicht nur Objekte zu erkennen, sondern auch ihre Pixel genau zu lokalisieren, hat die Mask RCNN-Architektur neue Möglichkeiten für die präzise Segmentierung von Bildern eröffnet.



Abbildung 1: Ausschnitt aus dem COCO-Datensatz mit Keypoints und Masken. Quelle: COCO, <https://cocodataset.org/images/keypoints-splash-big.png>.

Datensatz

Der COCO-Datensatz ist ideal für die Pose Estimation, da er über 100.000 annotierte Bilder enthält und darunter etwa 250.000 Personen mit Keypoints. Der Datensatz bietet eine große Variation an Szenarien und Hintergründen, was einen klaren Vorteil für ein vielseitiges Modell darstellt. Die relevanten Annotationen für das Modell umfassen Bounding-Boxes¹ mit zugehöriger Klassifizierung sowie 17 Keypoints, die die Knochenstruktur eines Menschen darstellen (siehe Abbildung 1)[5]. COCO bietet außerdem eine einfache Methode zum Laden der Daten und eine Evaluations-API, die die Validierung der Ergebnisse erleichtert und den Vergleich mit anderen Modellen vereinfacht.[6]

das Region Proposal Network (RPN) weitergegeben, das Region of Interests (ROIs) identifiziert. Die ROIs werden dann an Box-Regressor, Softmax Classifier und Keypoint Predictor weitergeleitet.[4]

Entwicklung des Modells

Architektur

Das Keypoint RCNN nutzt ein ResNet Feature Pyramid Network (FPN) als Backbone, um Merkmale aus dem Bild zu extrahieren und eine hierarchische Merkmalspyramide zu erzeugen. Diese Pyramide wird an

Backbone ResNet50, ein tiefes neuronales Netzwerk, extrahiert komplexe visuelle Merkmale aus Bildern, jedoch sind diese Merkmale nicht immer optimal für Aufgaben wie die Objekterkennung. Um dies zu verbessern, wird ResNet mit FPN kombiniert (siehe Abbildung 2), einem Feature Pyramid Network, das die effiziente Nutzung von Merkmalen auf verschiedenen Skalenebenen ermöglicht. FPN wird an den Ausgängen verschiedener ResNet-Blöcke angebracht, um Merkmale auf mehreren Skalenebenen zu extrahieren. Diese Kombination ermöglicht robuste und skalierbare Merkmalsextraktion für eine verbesserte Leistung bei komplexen visuellen Aufgaben in verschiedenen Anwendungen der Computer Vision.[4]

¹ Bounding-Boxes sind rechteckige Rahmen, die die Abmessungen von Objekten festlegen und begrenzen.

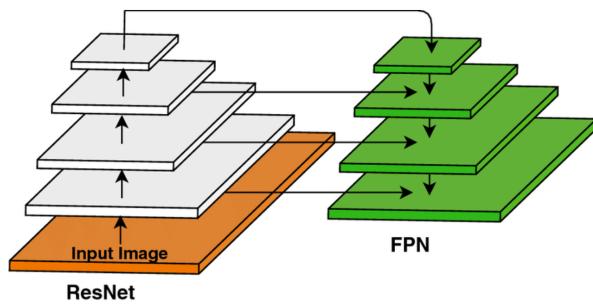


Abbildung 2: Kombination von ResNet und FPN zur verbesserten Merkmalsextraktion

Region Proposal Network (RPN) Das RPN nutzt die Feature Map vom FPN. Über diese Feature Map wird ein kleines 3×3 Fenster geschoben. Jedes dieser Fenster wird auf eine 256-dimensionale Feature Map abgebildet. Diese Feature Map wird dann in zwei gleichwertige, vollständig verbundene Schichten eingespeist: eine für die Box-Regression (*reg*) und eine für die Klassifizierung (*cls*) von Objekten (siehe Abbildung 3). An jeder Position des Fensters werden Anchor² in verschiedenen Formaten platziert, wie 1:2, 2:1 und 1:1. Die *reg*-Schicht erhält die Position und Größe des Anchors, während die *cls*-Schicht die Klassifizierung des Anchors erhält. Am Ende werden sich überlappende Anchors, die als Objekte identifiziert wurden, durch Non-Maximum-Suppression zu einer einzigen Box vereinigt.[3]

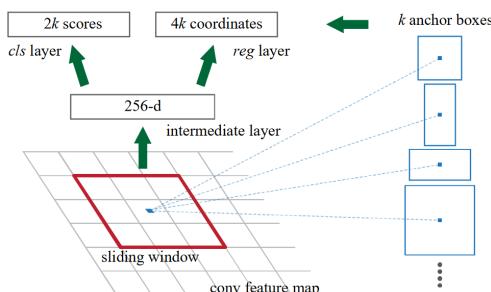


Abbildung 3: Region Proposal Network (RPN)[3]

Region of Interest Align (RoIAlign) Das RoIAlign-Verfahren wird eingesetzt, um Regionen auf Basis von Bounding-Boxes innerhalb der Feature Map des Backbones zu ex-

² Anchor sind vordefinierte Bounding-Boxes

trahieren[4]. Im Gegensatz zum RoIPooling, das Quantisierung verwendet, um die Regionen zu extrahieren[3], umgeht das RoIAlign-Verfahren dies, indem es die Bounding-Boxes mithilfe bilinearer Interpolation extrahiert (siehe Abbildung 4), was zu einer deutlichen Verbesserung der extrahierten Regionen führt[4].

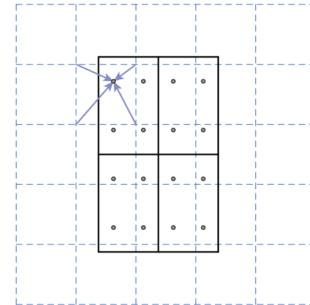


Abbildung 4: Das gestrichelte Gitter repräsentiert eine Feature Map, die durchgezogenen Linien stellen eine Region of Interest (RoI) dar (mit 2×2 Bins in diesem Beispiel), und die Punkte repräsentieren die 4 Abtastpunkte in jedem Bin. RoIAlign berechnet den Wert jedes Abtastpunkts durch bilineare Interpolation von den nahegelegenen Gitterpunkten auf der Feature Map.[4]

RoI-Heads RoI-Heads sind spezielle Module, die auf den Feature Maps von RoIs angewendet werden.

- **Box Regressor:** Der Box Regressor ist eine Komponente, die darauf abzielt, die Genauigkeit der Bounding-Boxes zu verbessern[1], die von dem RPN vorgeschlagen wurden[4]. Es verwendet eine Regressionstechnik, um die vorgeschlagenen Bounding-Boxes an die genauen Positionen und Größen der erkannten Objekte im Bild anzupassen. Dies geschieht durch das Lernen von Verschiebungen und Skalierungen, die die Abweichungen zwischen den vorgeschlagenen und den tatsächlichen Objektgrenzen minimieren[1].
- **Softmax Classifier:** Der Softmax Classifier nimmt die extrahierten Merkmale jeder vorgeschlagenen

Region als Eingabe und gibt eine Wahrscheinlichkeitsverteilung über verschiedene Objektklassen aus. Diese Wahrscheinlichkeiten werden mit Hilfe der Softmax-Funktion berechnet, die sicherstellt, dass die Summe der Wahrscheinlichkeiten für alle Klassen eins ergibt.

- **Keypoint Predictor:** Der Mask-Head des Mask-RCNN wird modifiziert, um die Identifizierung von Keypoints zu ermöglichen. In dieser Anpassung wird die Anzahl der Kanäle in der Ausgabeschicht von 80 auf 17 reduziert. Jeder dieser 17 Kanäle repräsentiert einen Keypoint, der durch eine binäre Maske erfasst wird, um seine Position im Bild zu lokalisieren.[4]

Hyperparameter

Die Hyperparameter wurden anhand früherer Experimente mit dem Mask RCNN-Modell ausgewählt[4]. Lediglich die Batchgröße wurde unter Berücksichtigung der verfügbaren Ressourcen gewählt.

- Epochen: 46
- Batchgröße: 8
- SGD:
 - Lernrate: 0.02
 - Momentum: 0.9
 - Gewichtsabnahme: 1e-4
- MultiStepLR:
 - Milestones: (36, 43)
 - Gamma: 0.1

Lernstrategie

Die Lernstrategie für die Pose Estimation vereint eine Vielzahl von Maßnahmen, um das Training des Modells zu optimieren. Diese umfassen die Filterung des Datensatzes, Transformationen der Bilder und Annotationen sowie die Anpassung der Lernrate

durch Optimierungsverfahren. Diese Maßnahmen arbeiten zusammen, um die Leistungsfähigkeit des Modells zu verbessern und seine Fähigkeit zur präzisen Keypoint-Erkennung zu steigern (siehe Abbildung 5).

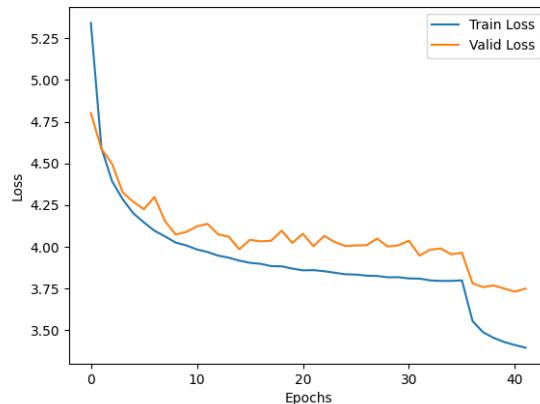


Abbildung 5: Trainings- und Validerungsverlust

Datensatz Der COCO-Datensatz wurde modifiziert, um nur Bilder zu verwenden, auf denen mindestens eine erkennbare Person zu sehen ist und die insgesamt mindestens 11 Keypoints aufweisen. Nach der Anpassung verblieben 46.529 Bilder im Datensatz. Diese Anpassungen zielen darauf ab, die Trainingsdaten für die Keypoint-Detection zu optimieren, indem sie sicherstellen, dass die Bilder relevante Informationen für diese Aufgabe enthalten.

Transformation Um die Vielfalt der Trainingsdaten zu steigern und die Robustheit des Modells zu verbessern, wurden die Bilder und zugehörigen Annotationen *horizontal gespiegelt*. Diese einfache Transformation verdoppelt effektiv die Trainingsdatenmenge und ermöglicht es dem Modell, eine breitere Palette von Situationen zu erlernen, was die Genauigkeit der Keypoint-Erkennung verbessert.

Aufwärmphase In der ersten Epoche wird eine Aufwärmphase eingeleitet, die auf dem LinearLR-Scheduler basiert. Dies hilft dabei, Instabilitäten zu Beginn des Trainings

zu vermeiden und einen gleichmäßigeren Verlauf des Trainingsprozesses zu gewährleisten. Durch die schrittweise Anpassung der Lernrate wird zudem eine *bessere Konvergenz* des Modells unterstützt.

MultiStepLR Der MultiStepLR-Ansatz ist optimal für das Training von Keypoint-Detektionsmodellen. Durch die Unterteilung des Trainings in aufeinanderfolgende Phasen ermöglicht er eine schrittweise Verbesserung der Modellleistung. Dies fördert eine *schnellere Konvergenz*, verhindert *Overfitting* und erlaubt die gezielte Anpassung verschiedener Aspekte des Modells während des Trainingsverlaufs.

Stochastischer Gradientenabstieg (SGD)
Der stochastische Gradientenabstieg (SGD) ist eine gute Wahl für die Keypoint-Erkennung aus mehreren Gründen. Er zeichnet sich durch seine Effizienz bei großen Datensätzen aus, ermöglicht eine schnellere Konvergenz des Modells und reguliert Overfitting durch den Einsatz von Mini-Batches. Zudem passt sich der SGD gut an nicht-konvexe Optimierungsräume an und bietet insgesamt eine effiziente und wirksame Methode zur Optimierung von Keypoint-Detektionsmodellen.

Ergebnisse

Metriken

Das Modell wurde anhand verschiedener Metriken wie IoU, Average Precision (AP) und Average Recall (AR) bewertet. Diese Metriken zeigen die Genauigkeit der Objekterkennung und -lokalisierung bei unterschiedlichen IoU-Schwellenwerten. Zum Beispiel zeigt ein AP von 86.0 bei einem IoU-Schwellenwert von 0.50, dass das Modell in etwa 86% der Fälle die Objekte richtig erkannt hat. Es gibt Unterschiede zwischen den Metriken, insbesondere bei verschiedenen IoU-Schwellenwerten. Zum Beispiel

liegt die AP bei einem IoU-Schwellenwert von 0.50:0.95 bei M bei 62.1, während sie bei L bei 72.0 liegt (siehe Tabelle 1). Ähnliche Unterschiede bestehen beim AR, was zeigt, wie die Leistung des Modells je nach Anwendung und Bewertungskriterien variieren kann.

Metrik	IoU	Fläche	%
AP	0.50:0.95	all	65.5
AP	0.50	all	86.0
AP	0.75	all	71.2
AP	0.50:0.95	M	62.1
AP	0.50:0.95	L	72.0
AR	0.50:0.95	all	72.3
AR	0.50	all	90.9
AR	0.75	all	77.5
AR	0.50:0.95	M	67.8
AR	0.50:0.95	L	78.7

Tabelle 1: Keypoint Average Precision & Average Recall mit 20 maximalen Erkennungen pro Bild

Metrik	IoU	Fläche	mD	%
AP	0.50:0.95	all	100	54.9
AP	0.50	all	100	82.5
AP	0.75	all	100	59.8
AP	0.50:0.95	S	100	37.7
AP	0.50:0.95	M	100	63.1
AP	0.50:0.95	L	100	70.8
AR	0.50:0.95	all	1	18.8
AR	0.50:0.95	all	10	55.8
AR	0.50:0.95	all	100	64.3
AR	0.50:0.95	S	100	49.4
AR	0.50:0.95	M	100	70.9
AR	0.50:0.95	L	100	78.8

Tabelle 2: Bounding-Box Average Precision & Average Recall. mD - maximale Erkennungen pro Bild

Vergleich mit anderen Modellen (State-of-the-art)

Das Keypoint RCNN-Modell zeichnet sich durch seine Flexibilität aus, da es im Gegensatz zu anderen Modellen Bilder mit ver-

schiedenen Auflösungen akzeptiert. Es bietet auch die Möglichkeit, Bounding-Boxes zu generieren, was in vielen Anwendungen der Computer Vision wichtig ist.

Ein Vergleich mit dem State-of-the-Art-Modell OmniPose zeigt, dass das Keypoint RCNN etwa 15% weniger Parameter hat. Obwohl es eine geringere Parameteranzahl aufweist und zusätzlich Bounding Box-Regression unterstützt, liegt die durchschnittliche Genauigkeit des Keypoint RCNN im Vergleich zum OmniPose-Modell um etwa 11% niedriger (siehe Tabelle 3)[7]. Dies war zu erwarten, da die Leistung eines Modells oft mit seiner Komplexität korreliert.

Trotzdem erbringt das Keypoint RCNN eine wertvolle Leistung, insbesondere in Anwendungen, die flexible Eingangsauflösungen erfordern und von der Bounding-Box-Generierung profitieren.

Fazit

Lessons learned

Es gibt einige wichtige Erkenntnisse, die aus der Optimierung des Keypoint RCNN-Modells gewonnen werden können.

- **RandomPhotometricDistort:** Die Integration von RandomPhotometricDistort in den Trainingsprozess des Keypoint RCNN könnte die Robustheit des Modells gegenüber Variationen in der Beleuchtung und den Farben der Bilder verbessern. Durch die Einführung von zufälligen Farbverzerrungen während des Trainings lernt das Modell, mit realistischen Szenarien umzugehen und könnte dadurch besser auf unterschiedliche Beleuchtungsbedingungen vorbereitet sein.
- **RandomRotation und RandomZoom:** Die Implementierung von RandomRotation und RandomZoom während des

Trainings könnte die Fähigkeit des Modells verbessern, mit Bildern umzugehen, die unterschiedliche Orientierungen und Skalierungen aufweisen. Diese Techniken könnten dazu beitragen, dass das Modell robuster gegenüber verschiedenen Ansichten von Objekten wird, was insbesondere in Anwendungen mit variablen Kameraperspektiven wichtig ist.

- **Optimierung der Hyperparameter:** Die Feinabstimmung der Hyperparameter des Keypoint RCNN-Modells könnte zu einer verbesserten Leistung führen. Dies umfasst die Optimierung von Lernraten, Batchgrößen, Regularisierungsparametern und anderen Modellparametern, um die Konvergenz zu beschleunigen und die Generalisierungsfähigkeit des Modells zu verbessern. Eine systematische Optimierung der Hyperparameter kann dazu beitragen, Overfitting zu reduzieren und die Leistung auf Testdaten zu verbessern.

Durch die Berücksichtigung und Implementierung dieser Optimierungen könnte das Keypoint RCNN-Modell seine Fähigkeiten verbessern und zuverlässigere Ergebnisse in verschiedenen Szenarien der Objekterkennung und -segmentierung liefern.

Ausblick

Ein Ausblick auf das Keypoint RCNN zeigt vielversprechende Perspektiven für die Weiterentwicklung der Objekterkennung und -segmentierung. Durch kontinuierliche Optimierung und Feinabstimmung der Architektur könnte die Genauigkeit des Modells weiter gesteigert werden. Besonders relevant ist die Möglichkeit, das Keypoint RCNN in Echtzeit-Anwendungen einzusetzen, was bedeutende Fortschritte in Bereichen wie Robotik, autonomem Fahren und Augmented Reality ermöglichen würde. Insgesamt bietet das Keypoint RCNN vielversprechende

	Params	<i>AP</i>	<i>AP₅₀</i>	<i>AP₇₅</i>	<i>AP_M</i>	<i>AP_L</i>
OmniPose	68.1M	76.4	92.6	83.7	72.6	82.6
HRNet	63.6M	75.5	92.5	83.3	71.9	81.5
Mask-RCNN		63.1	87.3	68.7	57.8	71.4
Keypoint RCNN	58.9M	65.5	86.0	71.2	62.1	72.3

Tabelle 3: Keypoint AP ausgewertet mit dem COCO-2017 Validerungs-Datensatz.[7]

Möglichkeiten für Fortschritte in der Computer Vision und hat das Potenzial, verschiedene Branchen zu transformieren.

Literatur

- [1] Ross Girshick u. a. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, S. 580–587.
- [2] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, S. 1440–1448.
- [3] Shaoqing Ren u. a. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).
- [4] Kaiming He u. a. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, S. 2961–2969.
- [5] Tsung-Yi Lin u. a. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Hrsg. von David Fleet u. a. Cham: Springer International Publishing, 2014, S. 740–755. ISBN: 978-3-319-10602-1.
- [6] cocoapi. *cocoapi*. <https://github.com/cocodataset/cocoapi>. 2019.
- [7] Bruno Artacho und Andreas Savakis. "Omnipose: A multi-scale framework for multi-person pose estimation". In: *arXiv preprint arXiv:2103.10180* (2021).