# Convolutional Pose Machines

Clemens Pollak

## I. Einleitung

CONVOLUTIONAL POSE MACHINES sind neuronale Netze, welche die zweidimensionale Pose von Menschen aus einem Bild extrahieren können. Mithilfe eines mehrstufigen Modells werden Schlüsselpunkte des Körpers detektiert. In dieser Seminararbeit wird beleuchtet, woher die Architektur und der Name kommen. Außerdem wird die Effektivität des Netzes untersucht und begründet.

Diese Ausarbeitung beschäftigt sich mit der Fragestellung der Pose Estimation. Als erster Lösungsansatz werden Pose Machines von Ramakrishna et al. [1] betrachtet. Der Kern der Arbeit ist die Untersuchung der Architektur von Convolutional Pose Machines [2] und den damit verbundenen Designentscheidungen.

## II. Pose estimation

Eine Pose ist nach der DIN EN ISO 8373 (Industrieroboter Wörterbuch) eine Kombination von Position und Orientierung im dreidimensionalen Raum. In diesem Fall wird das Problem der Posen-Schätzung von Menschen in unterschiedlichen Bildern betrachtet. Das bedeutet: Die erkannten Posen werden zunächst nur zweidimensional erfasst. Im Bereich der Computer Vision werden Posen oft durch sogenannte Schlüsselpunkte dargestellt. Diese Punkte liegen unter anderem an Gelenken, sodass durch eine Pose die komplette Haltung des Köpers beschrieben werden kann. Für das Benchmark des MPII Datensatzes [3] werden zum Beispiel folgende Schlüsselpunkte verwendet:

- Kopf
- Nacken
- Rechte & Linke Schulter
- Rechter & Linker Ellenbogen
- Rechtes & Linkes Handgelenk
- Brust
- Rechte & Linke Hüfte
- Rechtes & Linkes Knie
- Rechter & Linker Knöchel

Zusammen mit der Markierung für den Hintergrund gibt es in diesem Datensatz demzufolge 15 Markierungen. In anderen Fällen können auch mehr Punkte nötig sein [4]. In Abbildung 1 sieht man Beispiele, wo das Erkennen von Schlüsselpunkten sehr schwierig ist. Einige Schlüsselpunkte, wie der Ellenbogen, sind überhaupt nicht sichtbar und können nur durch holistisches, logisches Denken erkannt werden. [5] Außerdem sind die Winkel, aus denen Personen abgebildet sein können, sehr unterschiedlich und die Gliedmaßen folgen keinen simplen Anordnungsmustern (siehe Abbildung 1).

Betreuer: Bastian Wandt, Institut für Informationsverarbeitung, Leibniz Universität, WS 2018/2019.

Abbildung 1. Beispiele für schwierigen Posen aus [5]. Die Schätzung der Posen ist nicht erfolgreich, weil der Winkel der Aufnahme und die Haltung des Menschen entscheidende Merkmale verdecken (z.B. den linken Arm im rechten Bild). Für Menschen ist die Haltung jedoch leicht zu erkennen. Dies lässt vermuten, dass ein holistisches Verständnis des Körpers notwendig ist, um schwierige Posen akkurat zu schätzen.

## III. Pose Machines und andere Algorithmen zur Posen-Schätzung

Die Idee der Convolutional Pose Machines [2] geht auf die Pose Machines [1] zurück. Es handelt sich um eine Reihe von Klassifikatoren, die von vorhergegangenen Fehlern lernen, um immer bessere Schätzungen für Schlüsselpunkte zu erzeugen.

In der ersten Stufe produziert ein Klassifikator[1] initiale Schätzungen für die Positionen aller Schlüsselpunkte. Die Ausgabe sind sogenannte Belief Maps (siehe Abbildung 2). Die Klassifikatoren in den weiteren Stufen lernen die Belief Maps des jeweils vorigen Klassifikators zu korrigieren, indem sie von vorigen Fehlern lernen. Idealerweise entstehen am Ende dann Belief Maps die klare Maxima an den Stellen haben, die den Schlüsselpunkten im Originalbild entsprechen.

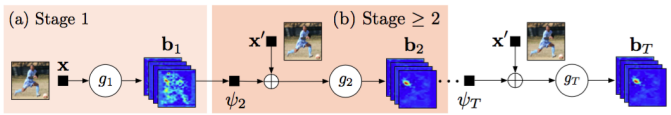


Abbildung 2. Pose Machine nach [1], aus [2]. Die beiden unterschiedlichen Architekturen der Stufen werden in Stufe Eins als $a$) und Stufe Zwei bis $T$ als $b$) unterteilt.

Formal ist die Pose Machine definiert als eine Reihe von Klassifikatoren $g_t$, wobei $g_1$ als Eingabe nur das Eingabebild $x$ erhält. Jeder Klassifikator erzeugt eine Tensor $b_t \in \mathbb{R}^{x \times w \times (P+1)}$. Dabei sind $h$ und $w$, Höhe und Breite des Bildes. P ist die Anzahl der Schlüsselpunkte (P wird um Eins erhöht, weil der Hintergrund ebenfalls seperat erkannt wird).

Die Ergebnisse $b_t$ dienen als Eingabe in die nächste Stufe. Um die Informationen aus allen Belief Maps zu verarbeiten, wird auf $b_t$ eine Kontext-Merkmalfunktion $\psi_t$ angewendet.

[1]Wenn hier von Klassifikatoren gesprochen wird, dann handelt es sich immer um einen Multi-Klassen-Klassifikator, der für alle benötigten Schlüsselpositionen Schätzungen durchführen kann.

Diese verbindet „Context Patch Features“ zum groben Erkennen der Konfidenz umliegender Schätzungen und „Context Offset Features“ zum genauen Aufnehmen von relativen Postionsinformationen (das heißt relativ zu anderen Schätzungen).

Die entstandene neue Belief Map wird dann mit dem Ausgangsbild addiert und bildet die Eingabe für den Klassifikator der nächsten Stufe. In der finalen Stufe werden die Belief Maps dann auf Maxima untersucht und anhand dessen die Positionen der Körperteile festgelegt.

Zusammenfassend produziert der Klassifikator $g_1$:

$$g_1(x_z) \rightarrow \{b_1^p(Y_p = z)\}_{p \in \{0 \dots P\}}$$

Das bedeutet, dass aus Merkmalen des Bildes $x_z$ Belief Maps $b$ für jeden Schlüsselpunkt (und Hintergrund) erstellt werden. Diese Belief Maps bilden die geschätzte Wahrscheinlichkeit ab, das die jeweilige Bildposition $Y_p \in Z$ der korrekten Schlüsselpunktpostion $z \in Z$ enspricht. Dabei sei $Z \subset \mathbb{R}_2$ die Gesamtheit aller möglichen Punkte im Bild.

Wie zuvor beschrieben, produzieren die weiteren Klassifikatoren $g_t$ eine Ausgabe mit gleichem Format und gleicher Bedeutung, erhalten aber zusätzlich zur Bildinformation $x_z$ noch die Ergebnisse der Kontext-Merkmalfunktion $\psi_t$:

$$g_t(x_z, \psi_t(z, b_{t-1})) \rightarrow \{b_1^p(Y_p = z)\}_{p \in \{0 \dots P\}}$$

In der Arbeit von Ramakrishna et al. [1] werden als Klassifikatoren $g$ boosted random forests [6], [7] gewählt. Zur Feature Extraction wird die Histogram of oriented Gradients (HoG) Methode [8] verwendet.

## IV. Convolutional Pose Machines

In der Arbeit von Wei et al. [2] wird der Klassifikator der Pose Machines durch eine Deep Convolutional Architecture ersetzt. Dadurch müssen keine Merkmale $x_z$ statisch aus dem Bild extrahiert werden und auch keine handgemachten Kontext-Merkmalfunktionen zum Einsatz kommen. Stattdessen ermöglicht die Architektur, sowohl Merkmale des Kontextes, als auch des Bildes selbst, direkt zu lernen. Außerdem können alle Stufen und damit alle Klassifikatoren zusammen trainiert werden, weil die Architektur komplett differenzierbar ist. Bei Random Forests ist das nicht möglich [1].

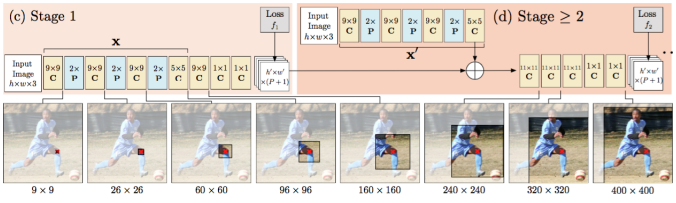


Abbildung 3. Convolutional Pose Machine aus [2]. Die unterschiedlichen Architekturen der ersten und weiteren Stufen werden in Abschnitte $c$) und $d$) unterteilt. Im Abschnitt $e$) wird das rezeptive Feld des Netzwerkes dargestellt. Es verdeutlicht wie viele räumliche Merkmale in die Schätzung einfließen.

In Abbildung 3 sieht man die Architektur der Convolutional Pose Machines. Insbesondere kann man sehen, dass die Merkmalextraktion durch Convolutional Layer und Pooling Layer ersetzt wurde. Dabei wird in der ersten Stufe eine andere Struktur zur Extraktion verwendet als bei den folgenden Stufen (siehe Abbildung 3, Teil $d$, Markierung $x'$). Bei einfachen Pose Machines bleibt die Extraktion immer gleich.

In der ersten Stufe werden nach den Layern, die die Extraktion ersetzen noch drei weitere Convolutional Layer angehängt, die anstelle des Klassifikators stehen. Zusammen kann dieses Faltungsnetzwerk eine Belief Map erstellen, indem die Funktion der Feature Extraktion und des Belief Mapping antrainiert wird. Dazu befindet sich nach jeder Stufe eine Verlustfunktion $f_t$.

Die Verlustfunktion bestraft einen großen euklidischen Abstand zwischen den geschätzten und idealen Belief Maps. Dabei ist eine ideale Belief Map $b_*^p(Y_p = z)$. Sie entsteht durch das Überlagern mit Gaußfunktionen an den Stellen, wo sich die Schlüsselpunkte im Ground Truth befinden. Die zu minimierende Verlustfunktion in jeder Stufe ist deshalb gegeben durch

$$f_t = \sum_{p=1}^{P+1} \sum_{z \in Z} ||b_t^p(z) - b_*^p(z)||_2^2 \,.$$

Da diese Verlustfunktion nach jeder Stufe angewendet wird, entstehen nach jeder Stufe interpretierbare Belief Maps. Außerdem helfen die Verlustfunktionen beim Trainieren des Netzwerkes (siehe Abschnitt IV-B). Für das Ende-zu-Ende Training des Netzwerkes werden einfach alle Verlustfunktionen zum Gesamtverlust

$$F = \sum_{t=1}^{T} f_t$$

addiert.

Die Kontext-Merkmalfunktion $\psi_t$ fällt bei den Convolutional Pose Machines weg, da auch dieses Verhalten antrainiert werden kann (siehe Abschnitt IV-A). Deshalb wird die Belief Map direkt mit dem Ergebnis der Merkmalextraktion $x'$ addiert und mit 5 weiteren Convolutional Layern zu einer neuen Belief Map verarbeitet.

### A. Merkmale im Kontext

Wie in den Pose Machines ist das Ziel der Convolutional Pose Machines, lokale Merkmale mit Umgebungsmerkmalen zu verbinden. Die neue Architektur kann Vorteile von Deep Learning Methoden [5] und Graph basierten Methoden [9], [10] verbinden. In Abbildung 3, Abschnitt $e$) sieht man, dass das rezeptive Feld des Netzwerkes mit jedem Layer wächst. So werden mehr und mehr Merkmale einbezogen, die nicht in unmittelbarer Nähe des untersuchten Punktes sind. Diese Aufgabe hatte bei den Pose Machines die handgemachte Funktion $\psi$. Im Idealfall kombiniert das Netzwerk die Informationen über Vorhersagen von anderen Schlüsselpunkten und Merkmalextraktion besser als $\psi$ zuvor.

Das rezeptive Feld eines Convolutional Neural Networks ist der Bereich auf dem Ausgangsbild, der Einfluss auf die Ausgabe des jeweiligen Layers hat. Dabei nimmt der Einfluss vom Zentrum zu den Randbereichen exponentiell ab [11]. Die Größe des Feldes kann mit den Formeln

$$r_{out} = r_{in} + (kernel - 1)\, d_{in}$$

$$d_{out} = d_{in}\, stride$$

berechnet werden [11]. Dabei ist $r$ das rezeptive Feld, und $d$ die Distanz zwischen zwei Merkmalen. $stride$ ist die Schrittweite und $kernel$ ist die Kernelgröße der Convolution im jeweiligen Layer. Die Feldgröße kann so rekursiv für alle Layer berechnet werden.

Ein größeres rezeptives Feld kann folglich durch unterschiedliche Anpassungen erreicht werden. Es können Pooling Operationen ausgeführt werden, die Kernelgröße oder Schrittweite kann erhöht werden, oder es werden mehr Convolutional Layer hinzugefügt. Pooling Layer werden nur zur Feature Extraction verwendet. Deshalb nennt sich das Netzwerk auch Fully Convolutional. Bei den Pooling Operationen werden immer Daten verworfen, was die Genauigkeit senkt. Es ist auch möglich die Größe des Feldes durch einen größeren Faltungskernel zu erhöhen. Ein größerer Kernel vergrößert allerdings die Anzahl der Parameter im Netzwerk und verlangsamt es. Aus diesem Grund wird eine Architektur mit vielen Convolutional Layern verwendet. Eine höhere Anzahl von Layern kann das Vanishing Gradient Problem hervorrufen. Da das Vanishing Gradient Problem durch mehrere Verlustfunktionen gedämpft werden kann (siehe Abschnitt IV-B), wird die Tiefe des Netzwerkes ausgenutzt.

Die Belief Maps werden trotzdem nach jeder Stufe um einen Faktor 8 herunterskaliert, um weniger Parameter im Netzwerk zu benötigen. Die Schrittweite der Faltungen haben die Autoren experimentell festgelegt. Es hat sich eine Schrittweite von 8 bewährt. Die Parameteranzahl wird weiterhin durch geteilte Gewichte reduziert. Weil alle Stufen, außer der Ersten, die gleiche Funktion übernehmen, können die Gewichte von korrespondierenden Faltungen der Layer geteilt werden.

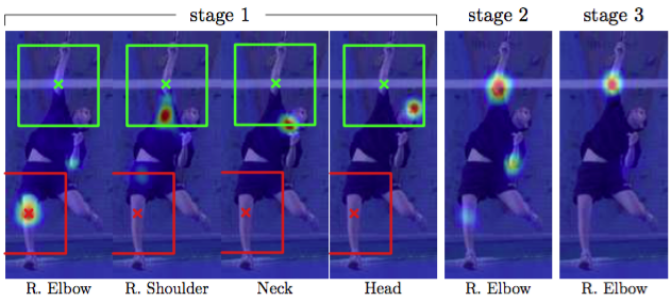


Abbildung 4. Belief Maps einer drei Stufen Convolutional Pose Machine. Eingeblendet sind zwei rezeptive Felder des kompletten Netzwerkes mit einer Größe von 160x160 Pixeln. In Rot sieht man das Feld zentriert auf der ersten Schätzung für den rechten Ellenbogen. In Grün sieht man das Feld zentriert auf der Stelle des Ellenbogen vom Ground Truth. Es ist zu sehen, dass die schlechte Schätzung der ersten Stufe korrigiert wurde.

Wie in Sektion II erläutert, sind Schlüsselpunkte, die weit vom Rumpf entfernt sind, schwer zu erkennen. In Abbildung 4 sieht man, dass das rezeptive Feld andere Peaks in den Belief Maps einschließt. Dadurch kann die Information über die Morphologie des Köpers genutzt werden, um schlechte Schätzungen zu korrigieren. In diesem Fall könnte zum Beispiel die korrekte Erkennung von Kopf, Schulter und Nacken dazu beigetragen haben, die Position des Ellenbogens in der zweiten Stufe zu verschieben. Man kann außerdem beobachten, dass der Peak, welcher sich an der falschen Stelle befindet, deutlich breiter ist als andere. Dies kann eine „Unsicherheit“ in der Schätzung bedeuten.

Diese qualitativen Beobachtungen wurden in [2] durch Experimente bestätigt, indem das rezeptive Feld bei gleicher Parameteranzahl vergrößert wurde. Es kam zu einer signifikanten Verbesserung der Genauigkeit.

### *B. Training und Vanishing Gradients*

Die Architektur einer Convolutional Pose Machine kann viele Convolutional Layer enthalten. Diese Tiefe ist ein Problem, denn es können sogenannte Vanishing Gradients auftreten. Das heißt, dass bei der Back-Propagation die Gradienten der Layer zwischen Ein- und Ausgabe immer kleiner werden und praktisch „verschwinden“. Ein effizientes Training ist dann nicht mehr möglich.

Durch die Architektur der Pose Machines kann die Anzahl der Layer zwischen Input und Output jedoch reduziert werden. Dazu dienen die Verlustfunktionen nach jeder Stufe. Sie erzwingen die Ausgabe der Belief Maps und sorgen damit dafür, dass auch in den Layern der mittleren Stufen Ergebnisse erzeugt werden und die Gradienten nicht verschwinden. Im Folgenden wird diese Taktik Intermediate Supervision genannt.

Das Problem der Vanishing Gradients kann auch experimentell gezeigt werden. In Abbildung 6 kann man Gradienten von 9 beispielhaft ausgewählten Layern sehen und deren Veränderung über drei Epochen des Trainings. In Schwarz ist die normale Architektur und in Rot ein Trainingslauf ohne Intermediate Supervision dargestellt. Man kann deutlich erkennen, dass die Varianz der Gradienten im Fall ohne Intermediate Supervision sehr klein ist. Die Gradienten bewegen sich um Null. Das heißt: Die Neuronen dieser Layer werden wenig Informationen weitergeben und tragen wenig zur Klassifikation bei. Veränderungen finden hautpsächlich in den letzten Layern statt. Im Fall mit Intermediate Supervision ist die Varianz deutlich größer und scheint nur langsam über die Epochen zu sinken. Das gilt für alle dargestellten Layer, auch jene welche nicht direkt vor einer Verlustfunktion stehen. Die Intermediate Supervision scheint folglich in einem guten Abstand eingesetzt zu sein.

### *C. Die beste Konfiguration*

Nach den bereits diskutierten Designentscheidungen wurde die Anzahl der Stufen experimentell optimiert. Bei Tests auf dem Leeds Sports Pose Dataset[2] [12] ergab sich, dass mehr Stufen grundsätzlich eine höhere Genauigkeit ergeben. Allerdings verbessert sich die Genauigkeit immer geringer, je mehr Stufen es gibt. Die Autoren haben sich entschieden eine 6 stufige Architektur zu evaluieren. Sie benutzen normalisierte Eingabebilder mit der Größe 368x368 Pixel. Dabei werden die Ursprungsbilder zu erst skaliert und anhand von Größe und Schlüsselpunkten zugeschnitten oder erweitert. Es wurden außerdem unterschiedliche Trainingsmethoden getestet, wobei sich das Ende-zu-Ende Training als bestes heraustellte.

[2]mit Person Centric Annotations (PC), d.h. rechts und links bedeutet von der abgebildeten Person aus rechts und links

Abbildung 5. Qualitative Resultate auf dem MPII Datensatz. Man kann beobachten, dass die Schlüsselpunkte für unterschiedliche Posen und Winkel erkannt werden. [2]

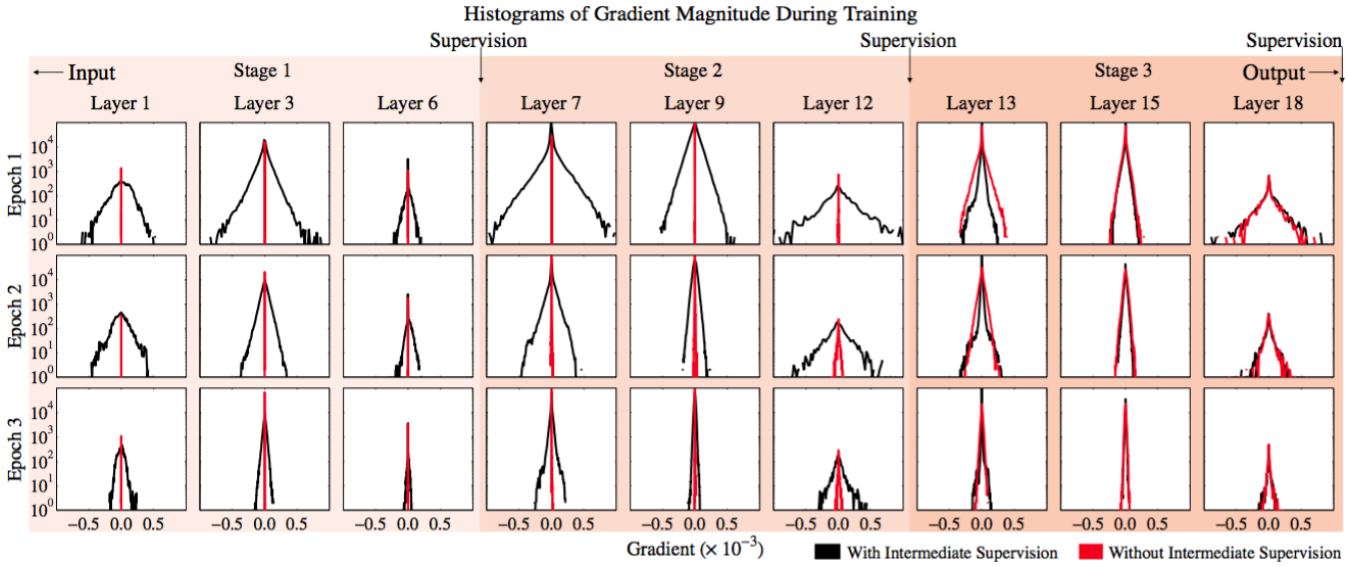


Abbildung 6. In den Histogrammen wird die Größe der Gradienten über drei Epochen des Trainings abgebildet. [2]

## V. Evaluation

Im Folgenden werden Testergebnisse auf dem MPII Human Pose Dataset [3] dargelegt. Der Datensatz besteht aus mehr als 28000 Trainingsbildern. Für das Training wurden die Daten augmentiert. Dabei wurden zufällige Rotationswinkel $[-40°,40°]$, Skalierungsfaktoren [0.7,1.3] und horizontales Spiegeln verwendet. Als Metrik wird die PCKh[3] Metrik verwendet. Weil in diesem Datensatz auch mehrere Personen auf einem Bild sichtbar sein können, wurden zwei Sets von Belief Maps für das Training erstellt. Eine enthält alle Peaks für jede Person in der Nähe der Hauptperson. Die Zweite enthält nur Peaks für die Schlüsselpunkte der Hauptperson. Die erste Art von Belief Maps wird für die Verlustfunktion der ersten Stufe verwendet, weil hier nur lokale Zusammenhänge erlernt werden. In allen weiteren Stufen wird das zweite Set verwendet, um geometritsche Relationen zwischen den Peaks richtig abzubilden.

Das Ergebnis nach der PCKh Metrik war zur Zeit der Veröffentlichung State of the Art mit einer Detektionsrate von 88.0%. Wenn zusätzlich noch der Leeds Sport Pose Datensatz zum Training benutzt wurde, kam das Netzwerk auf ein Ergebnis von 88.5%. 2016 war dieses Ergebnis mehr als 6% besser als vergleichbare Methoden. Zur Zeit dieser Seminararbeit befindet sich die Convolutional Pose Machine auf Platz 12 in der offiziellen Rangliste [3]. State of the Art ist Tang et al. [13] mit einer Wertung von 92.3%

[3]Percentage of correctly labeled Keypoints. „h“ bedeutet, dass 50% der Länge des Kopfsegmentes als Schwellewert für die Übereinstimmung festgelegt wird [3].

## VI. Fazit und aktuelle Arbeiten

Die Convolutional Pose Machines entstanden aus der Kombination von neuen Deep Learning Techniken und der Idee, Kontext über ein Stufenmodell erlernen zu lassen. Die Autoren haben im Jahr 2017 ein bedeutendes Paper im Bereich des Multiple People Tracking veröffentlicht, das die hier dargestellten Ansätze weiter verfolgt [14]. Die Ideen der Intermediate Supervision sind immer noch relevant. Allerdings gibt es auch Methoden, wie zum Beispiel Batch Normalization, die nicht im Paper über Convolutional Pose Machines betrachtet wurden. Da sich das Feld des Maschinellen Lernens sehr schnell bewegt, sind die Ergebnisse seit 2016 vielmals überboten worden. Das Konzept der Convolutional Pose Machines ist jedoch weiterhin von Bedeutung.

## Literatur

[1] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh, “Pose machines: Articulated pose estimation via inference machines,” in *European Conference on Computer Vision*. Springer, 2014, pp. 33–47.

[2] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.
[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele.
[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
[5] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.
[6] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[7] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.
[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.
[9] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1014–1021.
[10] T. Lan, Y. Wang, and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *International Conference on Computer Vision (ICCV)*, 2011.
[11] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.
[12] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *Proceedings of the British Machine Vision Conference*, 2010, doi:10.5244/C.24.12.
[13] W. Tang, P. Yu, and Y. Wu, "Deeply learned compositional models for human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 190–206.
[14] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.