



UNIVERSITÀ DEGLI STUDI DI FIRENZE
SCUOLA DI SCIENZE MATEMATICHE FISICHE E NATURALI
DIPARTIMENTO DI INFORMATICA

Tesi di laurea Magistrale in Informatica - Data Science

RICONOSCIMENTO DI AZIONI UMANE USANDO TECNICHE DI APPRENDIMENTO PROFONDO PER LA STIMA DELLA POSA

Candidato
Andrea Moscatelli

Relatore
Marco Bertini

Correlatore
Correlatore 1

ANNO ACCADEMICO 2019 - 2020

Indice

Prefazione	ii
Introduzione	iii
1 Stima della posa	1
2 PoseNet	4
2.1 Stima dei key-points	5
2.2 Raggruppamento dei key-points in istanze di persona	7
3 Detectron2	9
4 Classificazione	12
4.1 Struttura della rete	12
4.2 Tecniche	12
4.2.1 Semplice	12
4.2.2 Tecnica dei centri	12
4.2.3 Tecnica delle differenze	12
5 Risultati ottenuti	13
6 Conclusioni	14
7 Sviluppi futuri	15
Bibliografia	16

Prefazione

prefazione

Introduzione

Introduzione

Capitolo 1

Stima della posa

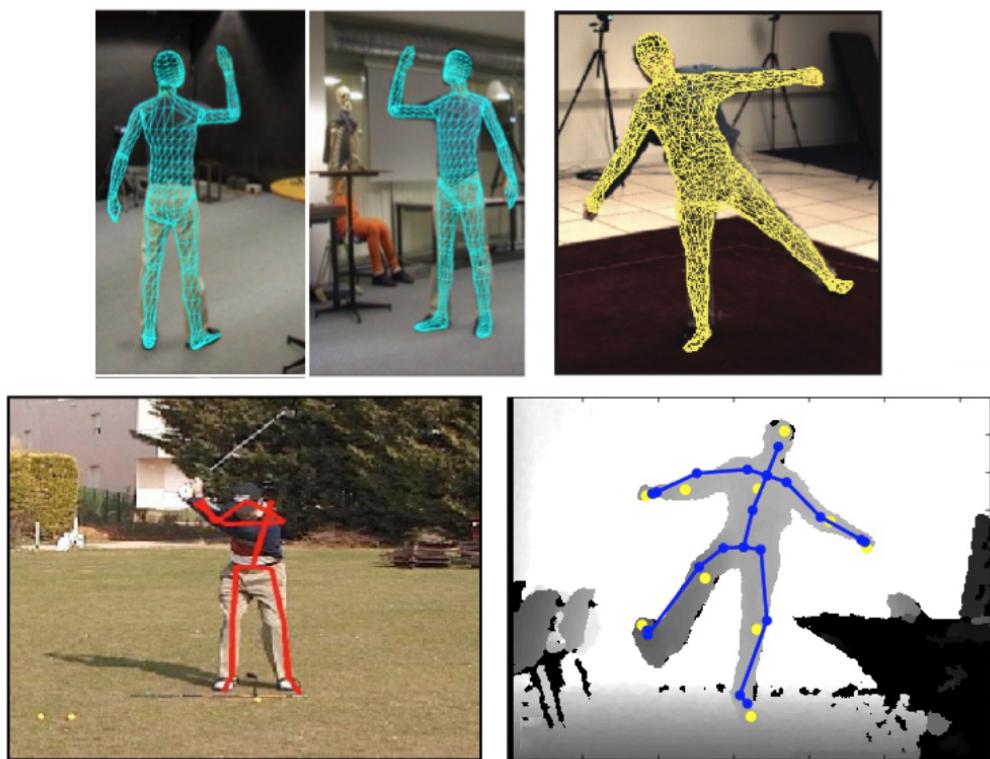


Figura 1.1. Esempi di stima della posa. In alto tre esempi di stima della posa utilizzando modelli di tipo volumetrico. In basso due esempi di stima della posa ottenuti utilizzando modelli di tipo scheletrici.

Cos è la stima della posa?

Quando parliamo di *stima della posa* ci riferiamo ad una tecnica di *computer vision* dedicata al riconoscimento di figure umane all'interno di video ed immagini, così da poter riconoscere ad esempio dove, all'interno dell'immagine, si trova la testa, il braccio, la gamba destra, etc.. del soggetto inquadrato.

Questa tecnica non va assolutamente confusa con tecniche di riconoscimento di persone, infatti la stima della posa è in grado solo di riconoscere dove le parti del corpo di un individuo sono situate all'interno dell'immagine, non *chi* è inquadrato.

I campi di applicazione della stima della posa sono i più svariati: software interattivi che reagiscono al movimento della posa, robotica, realtà aumentata, animazione, fotoritocco intelligente, fitness, riabilitazione, etc. Stiamo parlando di un problema tutt'altro che semplice, infatti la condizione di luce dell'immagine, la variabilità dell'ambiente circostante, l'inclinazione del soggetto inquadrato, rendono il riconoscimento della posa un problema non affatto banale.

Spinti dal crescente interesse, negli ultimi anni sono stati sviluppati diversi algoritmi per la stima della posa, raggiungendo in molti casi risultati davvero sorprendenti con un'accuratezza prossima alla perfezione.



Figura 1.2. Un esempio di utilizzo in campo medico della stima della posa

La maggior parte dei software in circolazione in grado di stimare in maniera sufficientemente corretta la posa di un individuo non sono liberamente accessibili.

Due fra i migliori algoritmi (ad oggi) di *pose detection* sono sicuramente *Posenet* [1] e *Detectron 2* [4], dei quali ci occuperemo in maniera più approfondita nei capitoli seguenti.

Capitolo 2

PoseNet

I recenti progressi nel campo della visione artificiale hanno permesso alla comunità scientifica di spostarsi verso problemi ancora più articolati rispetto a quelli classici, come ad esempio il riconoscimento facciale, con l'obiettivo di riconoscere figure umane in contesti non vincolati e molto variabili.

L'algoritmo *PoseNet* è stato ideato proprio con lo scopo di identificare una o più figure umane in qualsiasi contesto, anche in contesti “affollati”, ed essere in grado di identificare l'istanza di ogni persona stimandone i suoi *punti chiave* (o *key-points*).

Esistono due approcci principali per affrontare il rilevamento di più persone, la stima della posa e la segmentazione. L'approccio *top-down* inizia identificando e localizzando approssimativamente le singole istanze di persona identificando il riquadro dell'immagine dentro le quali sono contenute, seguito da una fase di stima della posa o di separazione “primo piano-sfondo” nell'area identificata. Al contrario, l'approccio *bottom-up* inizia localizzando entità semantiche individuali, come ad esempio gambe, braccia, mani, etc, seguito dal loro raggruppamento in istanze di persone complete. PoseNet adotta questo secondo approccio.

In particolare PoseNet utilizza una rete neurale convoluzionale nella quale il costo computazionale del riconoscimento delle pose è essenzialmente indipendente dal numero di persone raffigurate nella scena ma dipende esclusivamente dalla scelta delle features della rete.

L'approccio adottato in PoseNet è quello di identificare dapprima tutti i punti

chiave di ogni persona nell'immagine e successivamente raggrupparli in istanze utilizzando un processo “greedy”, ovvero partendo dal rilevamento “più sicuro”, e non come spesso accade da un punto fisso di riferimento (ad esempio il naso), avendo come vantaggio quello di funzionare bene anche se in disordine.

Oltre a stimare punti chiave sparsi, PoseNet stima anche maschere di segmentazione per ogni persona. Per fare ciò, viene allenata una seconda rete neurale con la quale viene associato ad ogni pixel x_i dell'immagine la probabilità di appartenenza di quel pixel ad ogni candidato j identificato. Se la probabilità è sufficientemente alta allora viene associato il pixel x_i al candidato j .

Questo algoritmo è stato allenato utilizzando il dataset COCO [2] che annota molte persone con 17 punti chiave (12 del corpo e 5 del volto), migliorando l'*AP* (average-precision) dal precedente miglior risultato da 0,655 a 0,687.

Questo metodo essendo molto semplice è anche quindi molto rapido, poiché non richiede alcuna fase supplementare di raffinamento dei risultati con tecniche di tipo *box-based* o *clustering*, facendo di PoseNet uno degli algoritmi più facilmente installabili su rete mobile.

2.1 Stima dei key-points

L'obiettivo di questa fase è quello di rilevare, in modo indipendente dall'istanza, tutti i key-points visibili appartenenti a qualsiasi persona dell'immagine. A tale scopo vengono prodotte delle *heat-maps*, ovvero dei canali della rete neurale dediti al riconoscimento di particolari caratteristiche dell'immagine (una canale per ogni key-point) e degli *offset* (due canali per ogni key-point per gli spostamenti in orizzontale e verticale). Sia x_i la posizione 2-D nell'immagine, dove $i = 1, \dots, N$ e N è il numero di pixels; $D_R(y) = \{x : \|x - y\| \leq R\}$ un disco di raggio R centrato in y e $y_{j,k}$ la posizione 2-D del k -esimo key-point della j -esima istanza di persona, con $j = 1, \dots, M$, dove M è il numero di istanze nell'immagine.

Per ogni tipo di key-point $k = 1, \dots, K$, viene impostato un task di classificazione binaria come segue. Viene generata una heat-map $p_k(x)$ tale che $p_k(x) = 1$ se $x \in D_R(y_{j,k})$ per qualsiasi istanza j , altrimenti $p_k(x) = 0$. Abbiamo quindi K tasks di classificazione binaria indipendenti, una per ogni tipo di key-point.



Figura 2.1. Generazione con PoseNet delle heat-maps per ogni tipologia di key-point

Ciascuno equivale a prevedere un disco di raggio R attorno a un tipo di key-point specifico di qualsiasi persona nell'immagine.

Oltre alle heat-maps, vengono anche usati vettori di offset a *corto raggio* $S_k(x)$ il cui scopo è quello di migliorare l'accuratezza della localizzazione dei key-points. Per ogni punto x all'interno dei dischi ricavati al passo precedente, il vettore di offset 2-D a corto raggio $S_k(x) = y_{j,k} - x$ rappresenta la distanza fra il punto x e il k -esimo key-point della j -esima persona. Vengono così generati K vettori per ogni punto x all'interno del disco definito che, combinati insieme in una *trasformata di Hough* $h_k(x)$, miglioreranno l'accuratezza della posizione predetta per ogni key-point. Solo i punti che superano una certa soglia di Hough (0.01, come indicato nel lavoro del team di PoseNet[1]) vengono considerati dei key-point, gli altri invece vengono scartati.

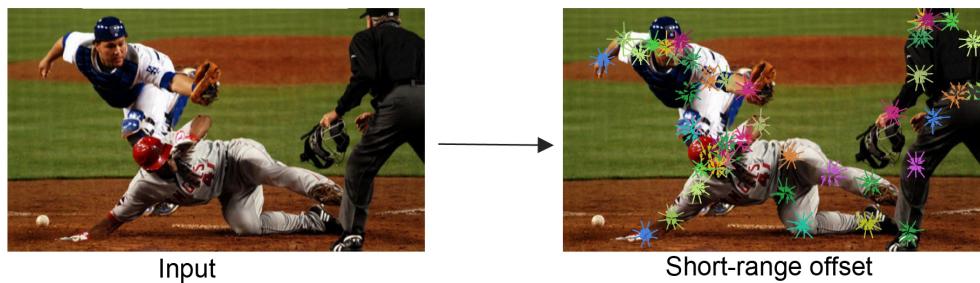


Figura 2.2. Esempio di stima degli offset a corto raggio con PoseNet

2.2 Raggruppamento dei key-points in istanze di persona

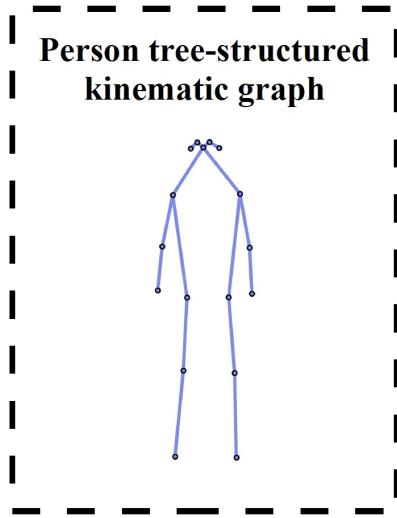


Figura 2.3. Struttura ad albero utilizzata da PoseNet per raggruppare i key-points appartenenti alla stessa persona

A questo punto è però necessario capire come associare ogni key-point alle persone raffigurate nell'immagine (nel caso ce ne sia più di una). Seguendo lo schema delle connessioni fra tipi di key-points (rappresentati in figura 2.3) la rete viene allenata per restituire in output anche i cosiddetti *offset a medio raggio*, ovvero probabilità di connessioni fra key-points, col lo scopo di raggruppare quelli appartenenti alla stessa persona. Un esempio di questa stima è raffigurato in figura 2.4. Una raffigurazione completa del sistema adottato da PoseNet per il riconoscimento delle pose di persone raffigurate in un'immagine è rappresentato in figura 2.5.

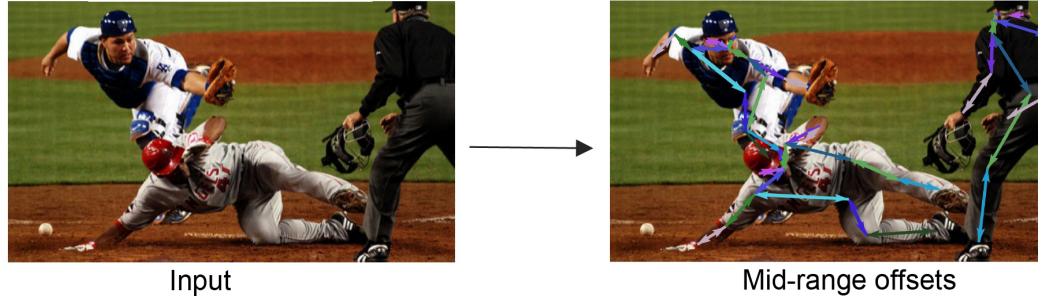


Figura 2.4. Esempio di stima degli offset a medio raggio con PoseNet. L'intento è quello di raggruppare i keypoints appartenenti alla stessa persona.

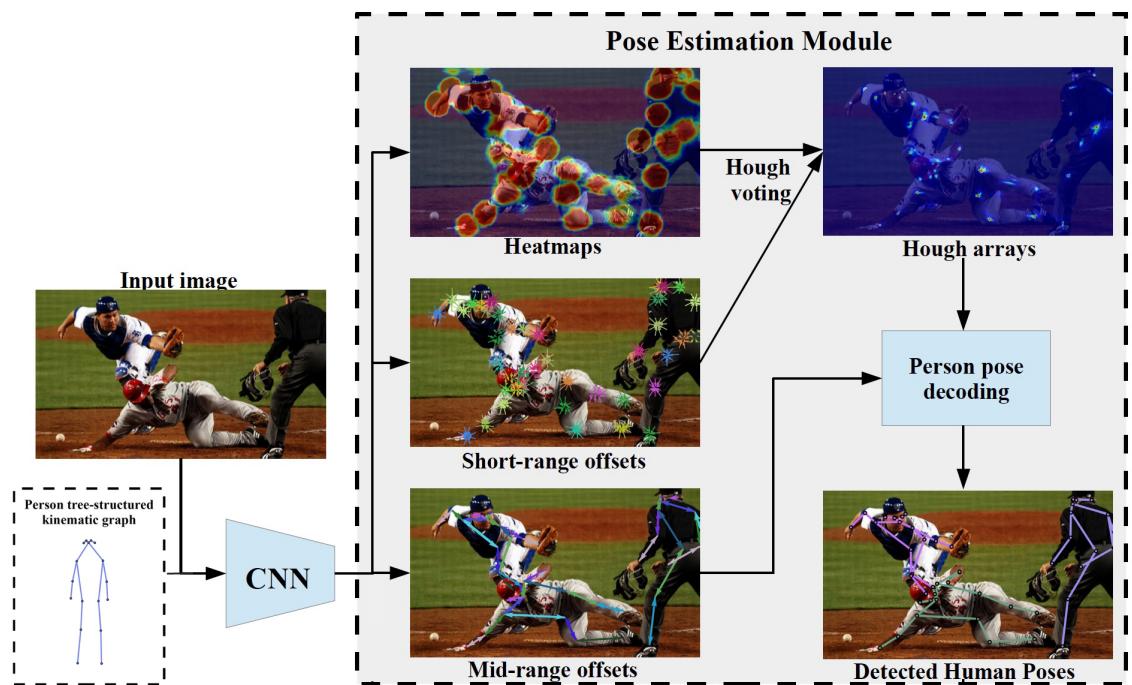


Figura 2.5. Combinazione delle fasi adottate da PoseNet per il riconoscimento della posa in un'immagine.

Capitolo 3

Detectron2

Detectron2 è un progetto open-source lanciato dalla *Facebook AI Research (FAIR)* ampiamente usato dalla comunità di ricerca in ambito *computer vision* che rappresenta, ad oggi, una piattaforma per il riconoscimento di oggetti allo stato dell'arte.

Il suo predecessore *Detectron* [5] fu un progetto cominciato nel 2016 con l'obiettivo di creare un sistema rapido e flessibile per il riconoscimento di oggetti in immagini originariamente basato su *Caffe2* [6] (un framework ideato per facilitare la sperimentazione e la divulgazione di nuovi modelli e algoritmi in ambito *deep learning*) e scritto in *Python*. Negli ultimi anni è stato perfezionato e supportato da una grande quantità di progetti, compreso “*Mask-R-CNN*” [7] e “*Focal Loss for Dense Object Detection*” [8], vincitori rispettivamente del *Premio Marr* e di *Miglior articolo scientifico studentesco* all'*Internation Conference on Compuer Vision (ICCV)* del 2017. L'intuitività e l'efficacia di questi algoritmi hanno permesso un notevole sviluppo nella risoluzione di problemi complessi nell'ambito della computer vision, come ad esempio l'*instance segmentation*, e hanno sicuramente giocato un ruolo rilevante nell'avanzamento tecnologico dei sistemi di riconoscimento visivo.

Detectron2 è adesso basato su *Pytorch*, una libreria open-source dedita al machine learning ed ampiamente usata nel campo della computer-vision che

ha inglobato in se anche il precedente framework Caffe2. Più nello specifico Detectron2 include le implementazioni dei seguenti algoritmi di object-detection:

- Cascade R-CNN [16]
- Panoptic FPN [17]
- TensorMask [18]
- Mask R-CNN [7]
- RetinaNet [8]
- Faster R-CNN [9]
- RPN [9]
- Fast R-CNN [10]
- R-FCN [11]

utilizzando le seguenti reti *backbone* (ovvero reti precedentemente allenate con lo scopo di estrarre in maniera efficiente le *features* di un'immagine):

- ResNeXt{50,101,152} [12]
- ResNet{50,101,152} [13]
- Feature Pyramid Networks (con ResNet/ResNeXt) [14]
- VGG16 [15]

e nel caso fosse necessario implementare nuove reti backbone, con Detectron2 è possibile farlo facilmente grazie alla struttura modulare di Pytorch, che permette di separare il nuovo modello dagli algoritmi di Detectron2.

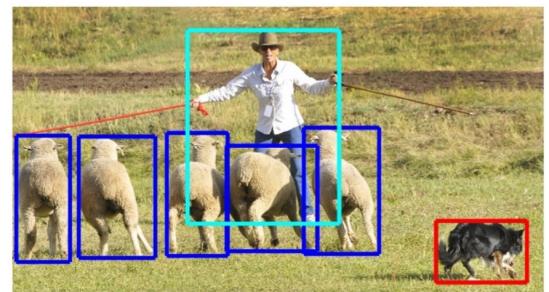
Essendo stato interamente riscritto in Pytorch, Detectron2 è più rapido del suo predecessore nei compiti di *object-detection*, *instance segmentation* e *human-pose prediction*, ed in più è in grado di fornire supporto per i nuovi task di

semantic segmentation e *panoptic segmentation*, ovvero la combinazione fra instance-segmentation e semantic-segmentation.

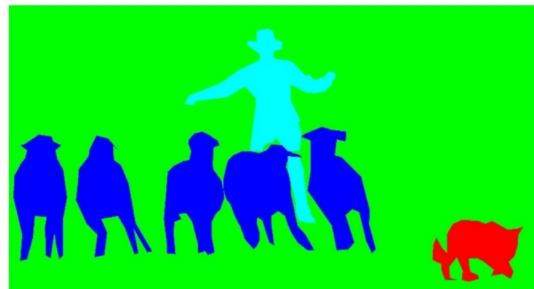
Oltre che nella ricerca, questa piattaforma viene usata anche da numerosi team di Facebook (e non solo) per l'addestramento di nuovi modelli in svariati campi della *computer vision*, come ad esempio la *realtà aumentata*, e in materia di sicurezza informatica, come ad esempio la *community integrity* (ovvero la difesa e la protezione di account su piattaforme social da contenuti maligni).



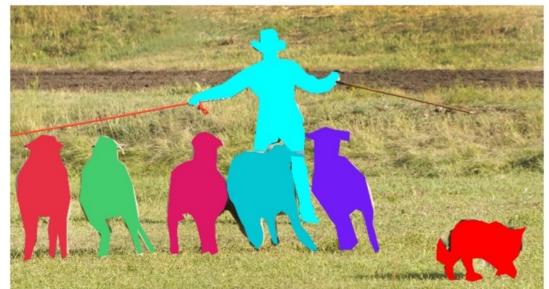
(a) Image classification



(b) Object localization



(c) Semantic segmentation



(d) This work

Figura 3.1. Combinazione delle fasi adottate da PoseNet per il riconoscimento della posa in un'immagine.

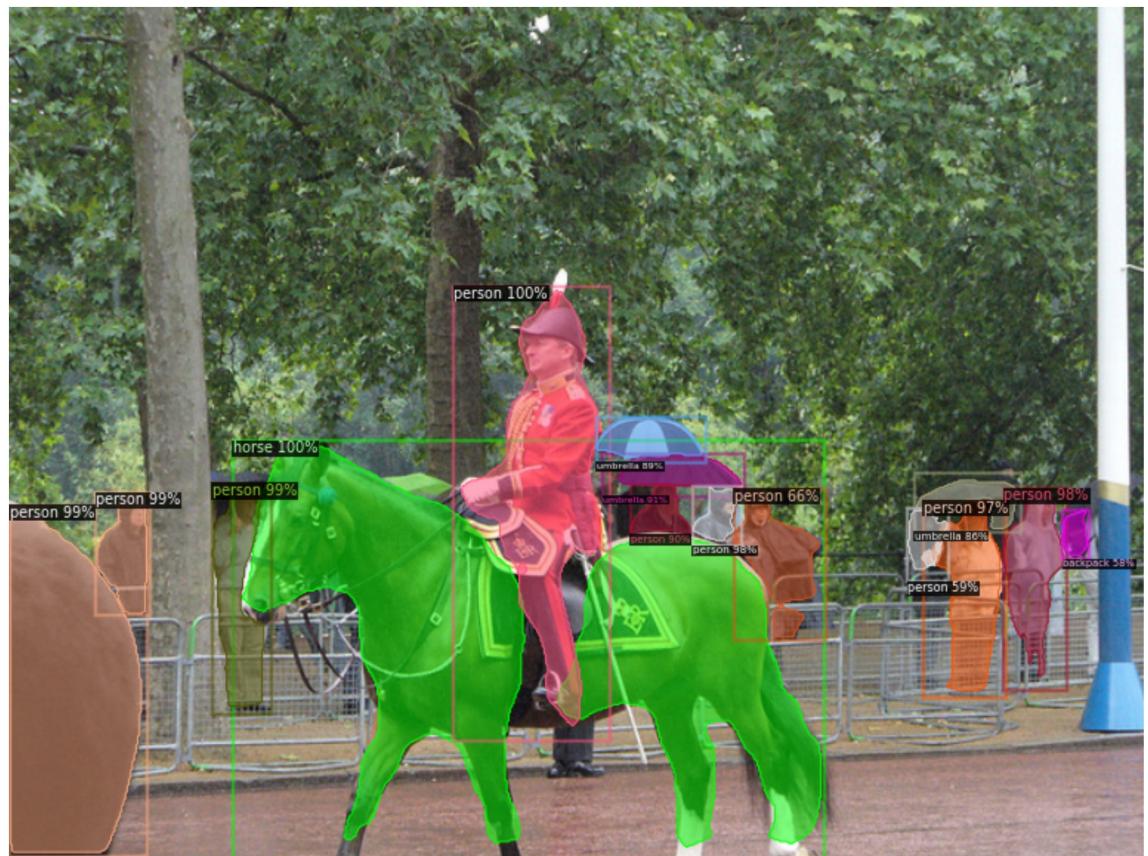


Figura 3.2. Combinazione delle fasi adottate da PoseNet per il riconoscimento della posa in un’immagine.

Capitolo 4

Classificazione

4.1 Struttura della rete

4.2 Tecniche

4.2.1 Semplice

4.2.2 Tecnica dei centri

4.2.3 Tecnica delle differenze

Capitolo 5

Risultati ottenuti

Capitolo 6

Conclusioni

Capitolo 7

Sviluppi futuri

Bibliografia

- [1] PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model - *George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, Kevin Murphy* - 2018
- [2] Coco 2016 keypoint challenge - Lin, T.Y., Cui, Y., Patterson, G., Ronchi, M.R., Bourdev, L., Girshick, R., Dollr,P. - 2016
- [3] PoseNet with TensorFlow.js - <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>
- [4] Detectron2 - *Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, Ross Girshick* - <https://github.com/facebookresearch/detectron2>, 2019
- [5] Detectron - *Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollàr, Kaiming He* - <https://github.com/facebookresearch/detectron>, 2018
- [6] Caffe2 - <https://caffe2.ai/docs>
- [7] Mask R-CNN - *Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick*, 2017
- [8] Focal Loss for Dense Object Detection - *Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár*, 2017
- [9] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks - *Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun*, 2015
- [10] Fast R-CNN - *Ross Girshick*, 2015

- [11] R-FCN: Object Detection via Region-based Fully Convolutional Networks - *Jifeng Dai, Yi Li, Kaiming He, Jian Sun*, 2016
- [12] Aggregated Residual Transformations for Deep Neural Networks - *Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He*, 2016
- [13] Deep Residual Learning for Image Recognition - *Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun*, 2015
- [14] Feature Pyramid Networks for Object Detection - *Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie*, 2016
- [15] Very Deep Convolutional Networks for Large-Scale Image Recognition - *Karen Simonyan, Andrew Zisserman*, 2014
- [16] Cascade R-CNN: High Quality Object Detection and Instance Segmentation - *Zhaowei Cai, Nuno Vasconcelos* - 2019
- [17] Panoptic Feature Pyramid Networks - *Alexander Kirillov, Ross Girshick, Kaiming He, Piotr Dollár* - 2019
- [18] TensorMask: A Foundation for Dense Object Segmentation - *Xinlei Chen, Ross Girshick, Kaiming He, Piotr Dollár* - 2019