HUMAN POSE ESTIMATION

Cerca di andare a ricostruire la postura umana → identificare le parti e le articolazioni principali del corpo umano

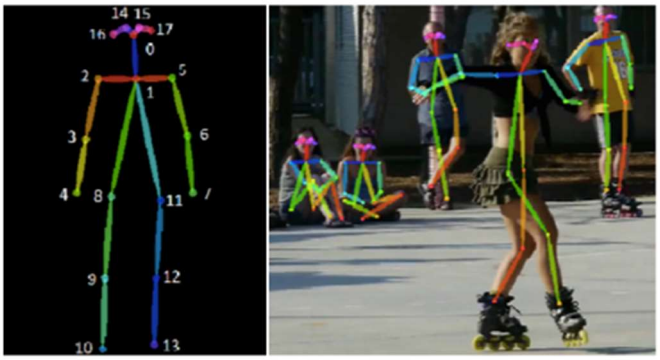
Posibili applicazioni:

- Human computer interaction -> azione fatta + movimento del corpo
- Virtual reality -> movimentare avatar
- Movies and animation -> catturo immagine di una persona reale per rimappare movimenti su animazione 3d
- Video surveillance -> riconoscendo le varie parti del corpo
- Sport motion analysis -> studio performance, analisi movimenti, ricostruzione virtuale.

Questo task è una cosa decisamente complicata perché il corpo umano non è rigido, è articolato e ci possono essere delle self-occlusions oltre che quelle ambientali, condizioni di illuminazione diverse, più persone che si muovono nell'ambiente (in quanto ambiente non controllato).
Inoltre, l'aspetto varia in base ai vestiti indossati.

Gli **step principali** per la HPE sono:

- Localizzare i keypoints del corpo umano
- Raggruppare i keypoints in una configurazioen valida



Il problema cambia anche se si tratta di dover fare la estimation per una singola persona o per più persone.

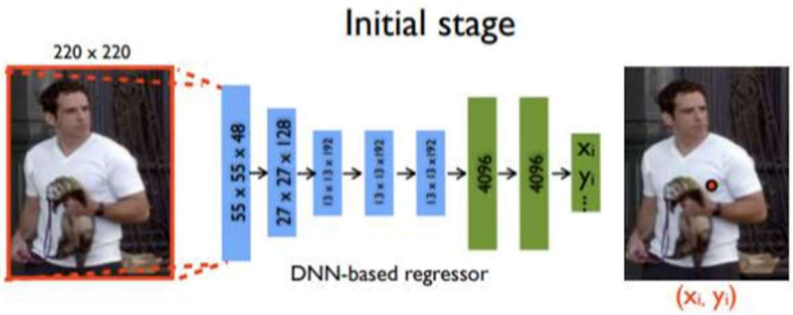
SINGLE-PERSON POSE ESTIMATION (SPPE)

DeepPose

È il primo modello che ha applicato in maniera efficace il deep learning.
Affronta il problema della body joint detection come un problema di regressione → stimare in maniera continua quella che è la posizione 2D dei giunti
Si basa sulle CNN che estraggono le features di ogni frame.
È un modello a cascata in cui usa più regressori a cascata: ogni regressore elabora i risultati del precedente.
Altra cosa interessante è che usa un approccio olistico: ragiona non in singoli giunti ma in insiemi di giunti → stimo giunto anche se non è visibile nell'immagine

DeepPose model

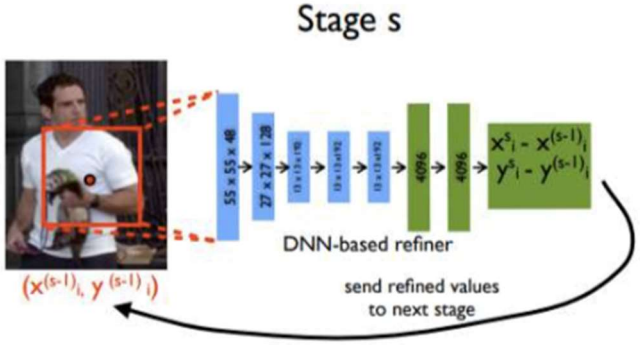
Il backend è un AlexNet a cui è aggiunto un layer per predirre la posizione dei body joints.
Il modello è allenato usando una loss L2 per la regressione.



N giunti, ultimo vettore ha n*2 giunti.

Cascading regressors

Si usa questo approccio a cascata perchè giunti che vengono identificati nella prima fase non sono del tutto precisi.
Il motivo principale per cui la stima iniziale non è precisa è dovuto alla dimensione dell'immagine in ingresso perché AlexNet riceve un input piccolo -> per questo devo ridurre l'imamgine di ingresso -> Riducendo quindi sottocampionando l'immagine si va a perdere informazione dall'immagine perché non sapendo ancora la posizione della persona sono costretto a darla tutta in input.



Al secondo stage ho già delle informazioni più dettagliate sulla posizione della persona nell'immagine e di conseguenza la stima dei giunti sarà migliore.

- Ri-applico un secondo step di regressione partendo dalle infromazioni del primo passo
- Livello di dettaglio migliore

DeepPose

Ha difficoltà nel collegare i giunti dell'immagine e poi le proprietà di generalizzazione del modello sono scarse -> overfitting sul training, poca generalizzazione.
Il passo che si può fare per migliorare è quello di usare le joint heatmaps:

- Spostare il focus da identificazione esatta della posizione dei giunti a stima della posizione dei giunti per una successivva elaborazione migliore

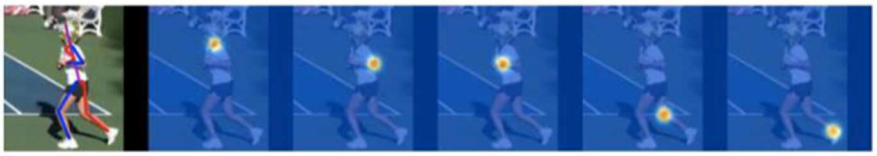
Joint heatmaps

Dato un giunto la heatmap è un'immagine in cui ogni pixel contiene la probabilità che il giunto si trovi in quel pixel.
In altri termini è la regione dell'immagine in cui è probabile trovare un giunto.

ConvNet pose

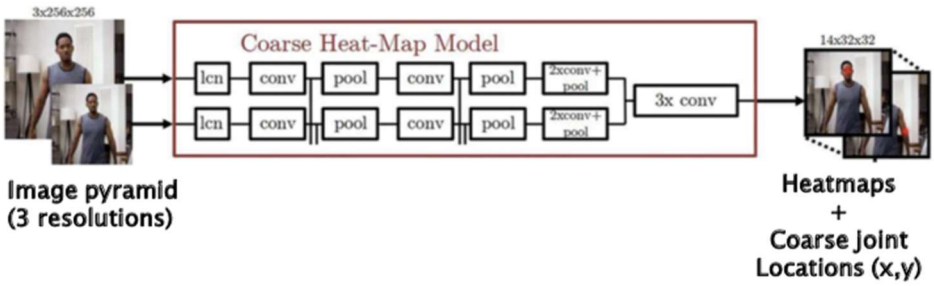
Questa architettura p composta da 3 moduli principali:

- Analisi multi-scale filtering che analizza l'immagine a differenti risoluzioni per far fronte al problema della scarsa stima iniziale dovuta alle limitazioni della grandezza delle immagini di input
- Heatmap estimation che va ad analizzare le zone in cui precedentemente era stata stimata la presenza di giunti
- Fine heatmaps che va a fare un lavoro di rifinitura sui giunti



Heatmap estimation

Rami diversi che lavorano a livello piramidale.



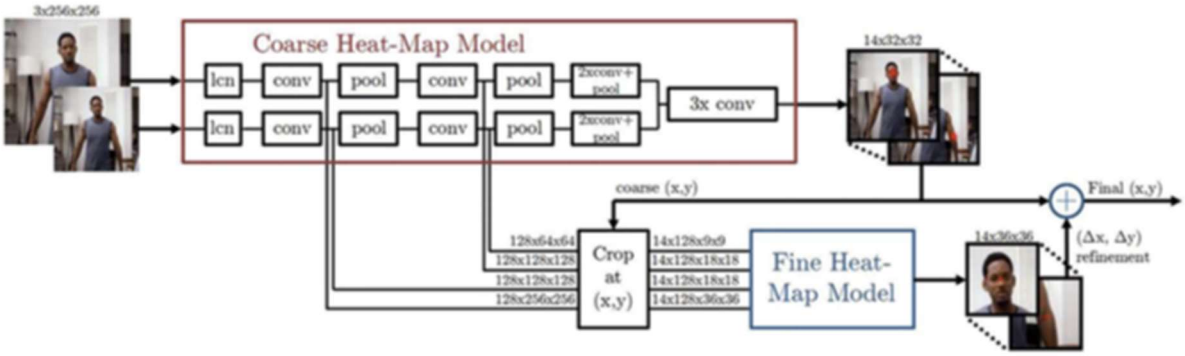
Si utilizza un approccio sliding window

Ci sono 3 rami che utilizzano versioni diverse dell'immagine per risoluzione quindi una versione piramidale dell'input.

- Lavoro sui 3 rami

Alla fine si ottiene una heatmap per ogni singolo giunto, dalla quale posso estrarre la posizione.

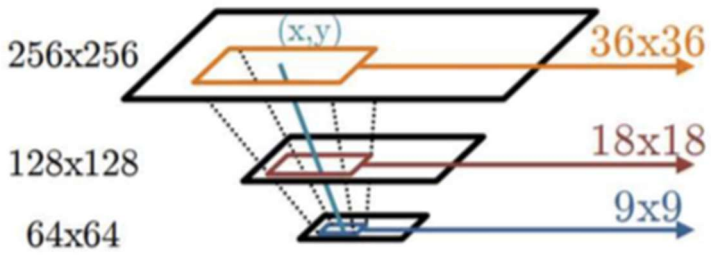
Fine heatmaps



Nello step precedente si ha ancora una stima grezza.

Vado ad utilizzare le feature del blocco precedente. Il fine **heat-map model** è una rete siamese.

Parto da una stima del giunto (x,y) per poi andare ad affinare la stima con le differenti **risoluzioni**



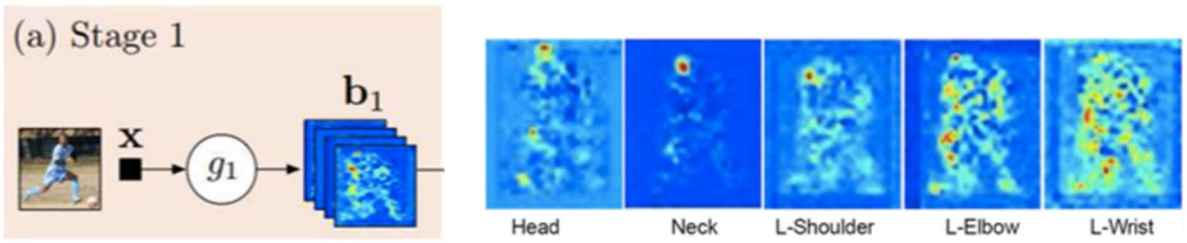
In questa rete rimane il problema della mancanza di un modello strutturale dei giunti umani.

- Non soffre almeno dei problemi di generalizzazione
- Non sfrutta bene le informazioni sulla struttura gerarchica del corpo -> servono relazioni strutturali

Convolutional pose machines

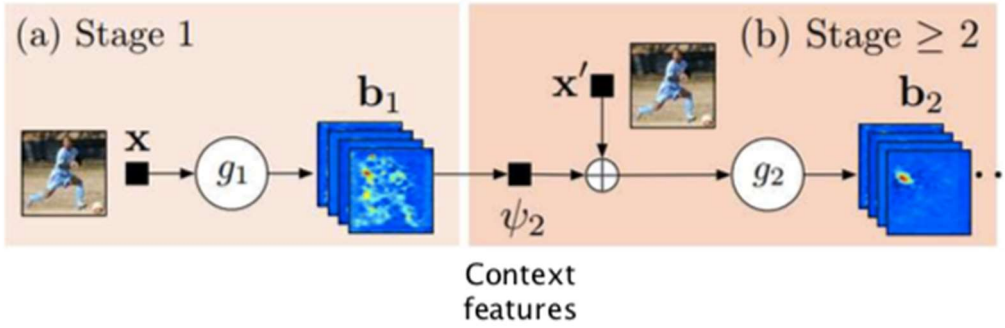
È un framework che procede in maniera iterativa. Per fare la stima sfrutta le informazioni su tutti i giunti per cercare di risolvere le ambiguità dell'elaborazione.
Anche se non utilizza un modello esplicito del corpo umano, si può dire che lo usa in maniera implicita perché integra nello stesso contesto le informazioni sui vari giunti.

Ogni passo dell'elaborazione contiene un predittore unico che provvede a calcolare quelle che sono le distribuzioni di probabilità dei singoli giunti del corpo. L'uscita del predittore sarà una **serie di heatmap, una per ogni giunto**.



Il modello è end-to-end con tutti i vantaggi nell'addestramento che potrà sfruttare le informazioni derivanti dai passi successivi.

I passi successivi alla prima stima consistono nell'andare ad elaborare le heatmap in uscita dal primo passo tramite una **funzione di contesto** per poi ottenere delle nuove heatmap più precise.

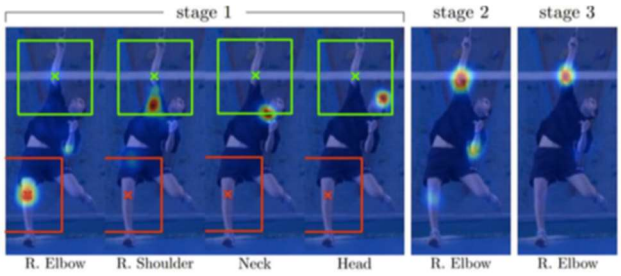


Ricalcolo le feature sfruttando le informazioni aggiuntivi provenienti da passi successivi.

Questo mi permette di avere delle feature diverse e magari migliori rispetto a quelle del passo precedente

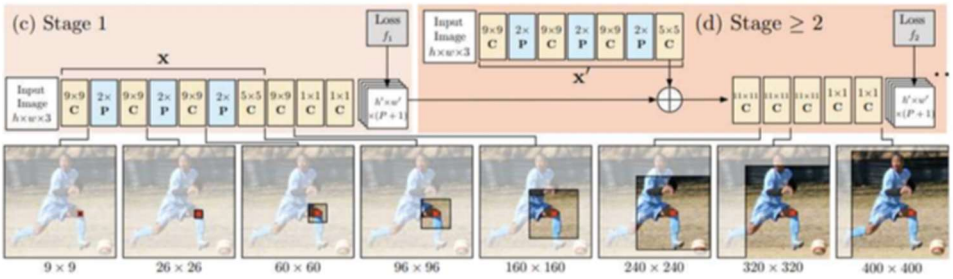
Context features

Lo stage 2 usando tutte le heatmap riesce a ricavarne un contesto per capire quante le stime precedenti erano corrette e/o precise.

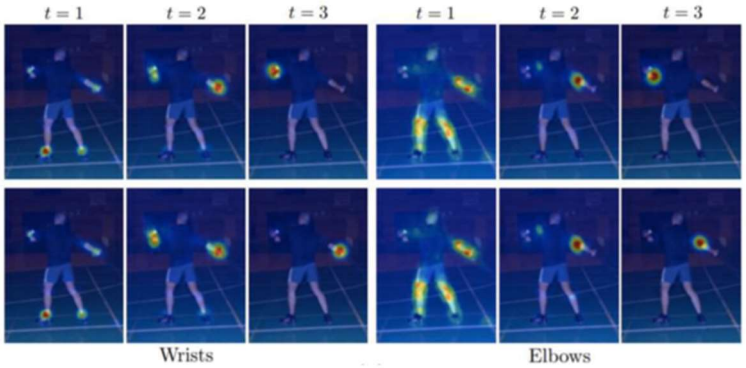


Receptive fields

L'utilizzo di un campo ricettivo (intorno delle aree convoluzionali che viene valutato) sempre più grande durante i passaggi della rete aiuta a catturare dipendenze spaziali a lungo raggio tra i giunti.



Results



Ottiene risultati e performance migliori rispetto ai modelli precedenti.

MULTI-PERSON POSE ESTIMATION MPPE

È un task molto più difficile rispetto al single-person pose estimation perché è sconosciuta la posizione e il numero di persone.
Si hanno due tipi di approcci: uno top-down e uno bottom-up

Top-down approaches

Sono caratterizzati dall'elaborazione in due passi:

1- Identificazione delle persone tramite bounding box
2- Da ogni bounding box candidato, trovo la posa della persona che eventualmente èp contenuta all'interno della regione.

All'interno della bounding box candidata, posso dire di non aver trovato la persona oppure posso sfruttare i moduli già conosciuti come quelli di object detection e di single-pose estimation per stimare la posa della persona presende nel bounding box.
L'accuratezza è dipendente da quella dell'object detector.
Il costo è proporzionale al numero di persone presenti.

Bottom-up approaches

Si cerca di identificare tutte le possibili parti di persone all'interno dell'immagine, e poi di raggruppare i singoli elementi a ciascun individuo.

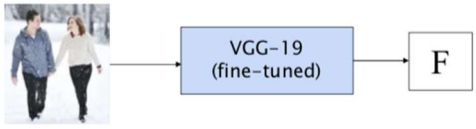
La parte difficile è la seconda in cui si cerca di raggruppare gli elementi singoli. Il costo è quasi indipendente dal numero di persone.

OpenPose

Metodo bottom-up open-source che può essere visto come un'estensione delle multi-branch Computation Pose Machines.
Usa una struttura a stage multipli per elaborare sia le heatmap (localizzare i giunti all'interno delle immagini) che una forma di struttura non parametrica del corpo umano chiamata Part Affinity Fields (cerca di andare a capire come possano essere collegati tra di loro i diversi giunti -> Serve per associare le varie parti del corpo con le rispettive persone.)
Ci sono tanti stadi successivi che cercano di migliorare il calcolo precedente.

1. extract frame features

Estrae le features dei frame usando una VGG-19 (fine-tuned) come feature extractor.



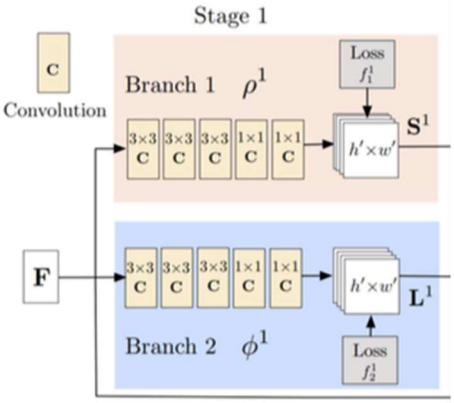
Questo si differenzia già dalle CPM perché queste feature non vengono più aggiornate.

- Vengono usate così come sono in qualsiasi stadio dell'architettura (*non vengono ricalcolate ad ogni passo come in altri casi)*

2. heatmap & PAFs

Passo le feature estratte F e le passo a due rami:

- Il primo genera una heatmap (o Silenty map) per giunto per tutte le persone
- Il secondo determina i PAFs, uno per arto, per tutte le persone (relazioni tra i vari aggiunti)

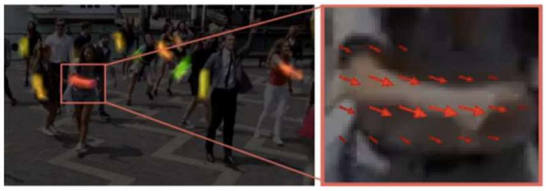


One heatmap per joint
(J joints in total)

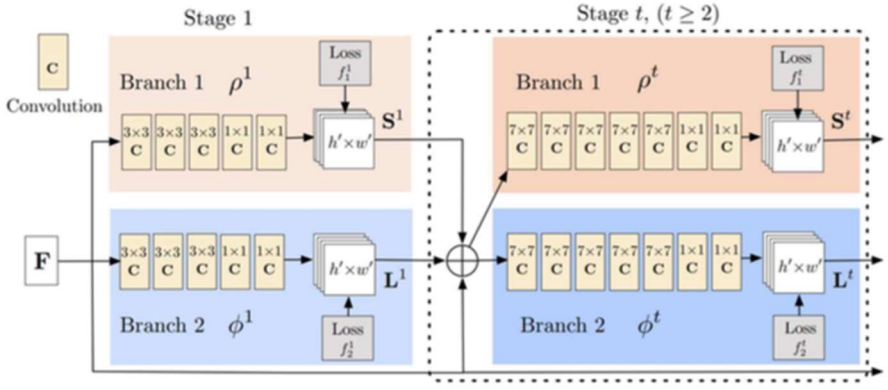
One PAF per limb
(C limbs in total)

Part Affinity Fields

Contengono la rappresentazione della probabilità/score di associazione tra due giunti dell'immagine. Vanno a codificare le relazioni tra le parti diverse del corpo (senza tener conto di tutto il corpo.)
Ci sono delle informazioni per ogni pixel: vettori 2d di flusso che cercano di codificare la posizione e la direzione di tutte le occorrenze dello specifico arto.



3. refining detections and associations



Combino per migliorare le informazioni:

- Feature, heatmaps, PAFs

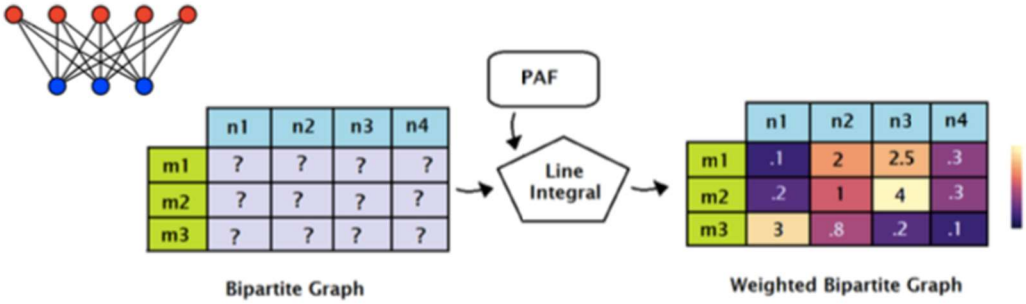
$$\mathbf{S}^t = \rho^t(\mathbf{F}, \mathbf{S}^{t-1}, \mathbf{L}^{t-1}), \ \forall t \geq 2,$$
$$\mathbf{L}^t = \phi^t(\mathbf{F}, \mathbf{S}^{t-1}, \mathbf{L}^{t-1}), \ \forall t \geq 2,$$

4. part association

Alla fine delle fasi precendenti avrò tutte le PAFs e un grafo con tutte le rispettive associazioni

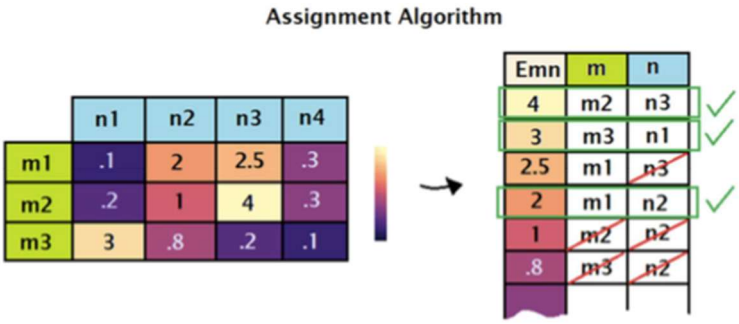
Per raggruppare i PAFs per le singole persone si possono usare i vettori direzionali degli stessi per andare a tirare fuori le informazioni strutturali che risolvono le ambiguità.

In pratica si va a calcolare l'integrale di linea sul PAF che va a sommare tutti i contributi dei vettori di flusso lungo la linea. Facendolo per tutti i collegamenti si hanno i pesi di tutti i collegamenti.



Assignment algorithm

Si vanno a mettere in ordine i collegamenti per peso e prendo iterativamente i pesi più alti corrispondenti ai punti che non sono stati ancora presi.



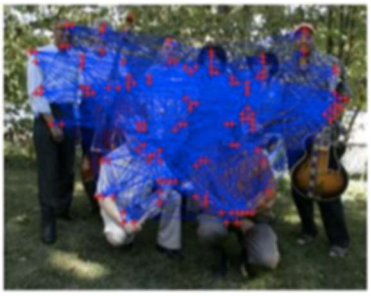
5. Merging

Lo step finale è di unire iterativamente le parti unendo i collegamenti che hanno giunti in comune capendo che appartengono allo stesso individuo. -> continuo finchè non sono possibili ulteriori connessioni.
Il risultato finale mi darà il numero finale.

- Start by creating a human from each detected part.
- If two humans share the same endpoint, they are indeed the same human, merge them into the same set, remove one human and repeat

DeepCut

Usa approccio bottom-up.
Usa un blocco che identifica tutti i potenziali giunti di tutte le persone usando un detector standard restituendo delle regioni con dei potenziali giunti al loro interno.



Si hanno quindi D candidati a cui si assegna un peso che indica la probabilità che appartengano ad una classe specifica.
Partendo da questi candidati, si ricava un grafo completo di tutti i possibili collegamenti.
Prima di tutto si cerca di capire quali sono realmente i giunti, di assegnare l'etichetta di uno specifico giunto e di clusterizzarli per la singola persona.
Si ottengono sottografi (ancora densi) appartenenti ad ogni singola persona.
Si fa infine l'ultimo taglio per individuare gli effettivi collegamenti dei giunti.

DeepCut si basa su ILP.

ILP

È un modello di programmazione lineare intera. Cerca di andare a trovare il minimo o un massimo di una funzione a più variabili che è soggetta a dei vincoli lineari.
Il modello si basa su 3 insiemi di variabili binarie x, y e z.

- If x(d,c) = 1 then it means that (joint) candidate d belongs to (joint) class c (→ D x C variables)
- If y(d,d') = 1 candidates d and d' belong to the same person (D x D variables)

- Variable z is used to partition pose belonging to different people
- z(d,d',c,c') = x(d,c) * x(d',c') * y(d,d')
- If the above value is 1, candidate d belongs to class c, candidate d' belongs to class c', and both candidates d,d' belong to the same person
  - Total of (D^2 x C^2) z variables

– uniqueness

$$\forall d \in D: \quad \sum_{c \in C} x_{dc} \leq 1$$

➔ Ogni candidato può appartenere al massimo asd una classe di giunti

– consistency

$$\forall dd' \in \binom{D}{2}: \quad y_{dd'} \leq \sum_{c \in C} x_{dc}$$

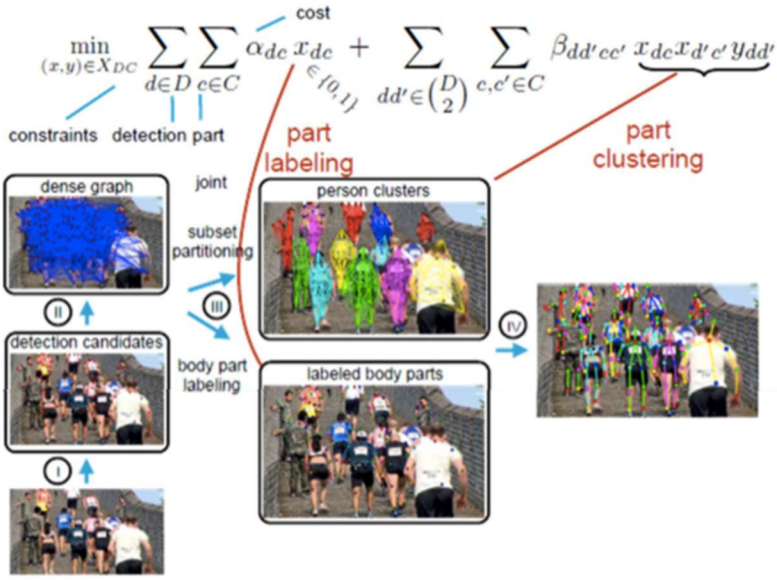
$$\forall dd' \in \binom{D}{2}: \quad y_{dd'} \leq \sum_{c \in C} x_{d'c}$$

➔ Consistenza di assegnazioni dei candidati alla stessa persona durante la fase di assegnazione dei cluster: per andare ad assegnare una cippia di candidati ad una persona, devo prima averli segnati come giunti validi e poi aver assegnato un'etichetta.

– transitivity

$$\forall dd'd'' \in \binom{D}{3}: \quad y_{dd'} + y_{d'd''} - 1 \leq y_{dd''}$$

Funzione obiettivo
Deve essere minimizzata.

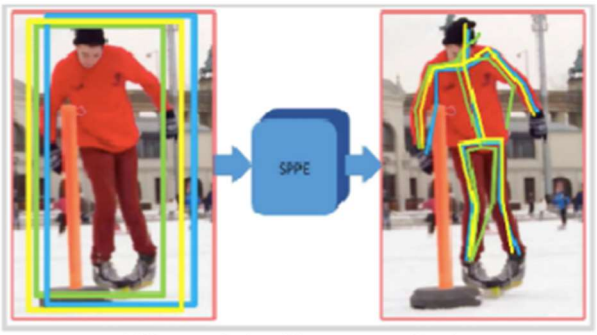


Primo termine: somma le variabili che definiscono quando un candidato d è un giunto reale di c (per il loro peso di appartenere alla classe).

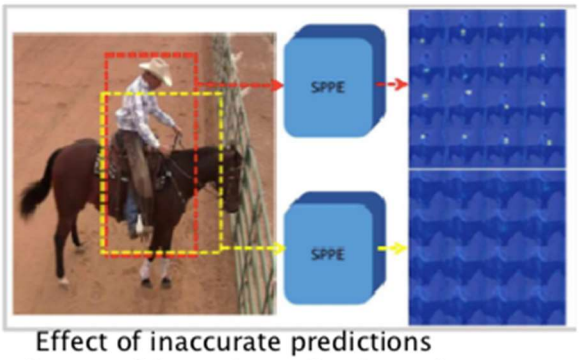
Seconda parte: clusterizzare i giunti -> coppia di candidati appartiene o no alla stessa persona, viene moltiplicato per peso Beta -> vado a pesare collegamento da giunti andando a valorizzare quelli tra stessa persona.

AlphaPose (RMPE)

È uno dei più famosi con approcci top-down. Si basa sull'osservazione che la parte problematica di questo tipo di approccio è data dalla parte di object detection.



Effect of duplicate predictions
(two skeletons)



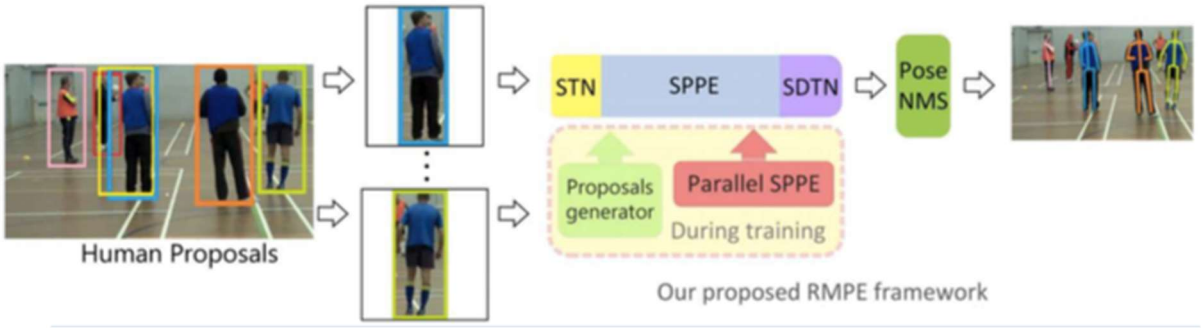
Effect of inaccurate predictions
(low confidence bounding boxes)

Prima si applica **un single-person pose extractor** (SPPE) che fornisce pose ridondanti e inaccurate e quindi si applica un ulteriore passo di regional multi-person pose estimation per sistemare le inesattezze precedenti.
Il vantaggio è che essendo un approccio generale si possono usare i moduli che si vogliono per le due parti.

- Struttura plug & play

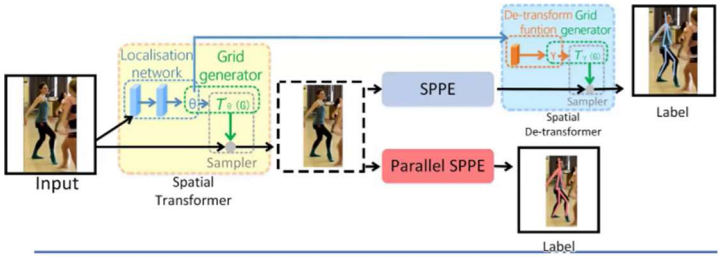
Alphapose architecture

STN + SPPE + SDTN: generano stima delle pose che vengono passate ad un blocco di NMS che va ad eliminar ele previsioni multiple su una persona.



Pose proposal

- STN (spatial transformer network) selects (and centers) the dominant human region in the bbox, simplifying the work of SPPE. SDTN (de-transformer...) remaps estimates to original coordinates
- The parallel SPPE branch is a regularizer that penalizes poses that are not well centered



➔ Ramo sppe parallelo verifica solo se centrato e propaga l'informazione indietro all stn.

Pose NMS

Cerca di rimuovere gli elementi non massimi:

- Seleziono la posa con più confidenza
  - Elimino le pose simili che si torvano a poca distanza

Estensione a 3D: si possono usare prodotti commerciali, stimando la posizione dei punti.

- Telecamere multiple
- Dispositivi commerciali (luci laser, etc)
- Varie problematiche