ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Scienze
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

ANALISI DELLA CORSA TRAMITE IMAGE SYNTHESIS E HUMAN POSE ESTIMATION

Elaborato in
Deep Learning

Relatore
Prof. Guido Borghi

Co-relatore
Prof. Davide Maltoni

Presentata da
Marco Avagnano

Terza Sessione di Laurea
Anno Accademico 2022 – 2023

PAROLE CHIAVE

Computer vision

Deep Learning

Human Pose Estimation

Image Synthesis

Generative AI

Le nostre vite cominciano a finire il giorno in cui stiamo zitti di fronte alle cose che contano.

Martin Luther King

Sommario

Negli ultimi anni, l'intelligenza artificiale si é evoluta in maniera esponenziale soprattutto nella branca della Computer Vision. In particolare sono nati metodi di Human Pose Estimation (HPE) in grado di analizzare un'immagine con all'interno una o più persone ed estrarre la posizione dei loro giunti (keypoints). La precisione di questi modelli dipende fortemente dalla qualità dell'immagine e la visibilità delle parti del corpo. In questo lavoro viene proposta un'analisi basata su immagini, della corsa di un atleta su tapis roulant, da cui si estraggono i keypoints tramite tecniche di HPE. L'attenzione è focalizzata sul fatto che le immagini catturate mostrano solo la parte inferiore del corpo dell'atleta. Questa limitazione è dovuta alla necessità di posizionare le attrezzature di registrazione da una distanza specifica dal tapis roulant. I modelli di HPE eseguiti su questo tipo di immagine mostrano in molti casi una scarsa accuratezza. Per risolvere questo problema viene proposto l'utilizzo di modelli generativi di immagine, Palette e Stable Diffusion Infinity, con l'obiettivo di generare la parte superiore del corpo per tentare di migliorare l'accuratezza dei modelli di HPE sui keypoints della parte bassa. Vengono adottati due differenti task generativi, inpainting e outpanting, che consentono di ricostruire l'immagine partendo da una maschera di rumore gaussiano. Nel corso del lavoro vengono proposti diversi esperimenti che mostrano l'aumento della precisione delle tecniche di HPE sulle immagini generate, un confronto fra i modelli a diffusione proposti e una breve valutazione della complessità computazionale del processo di generazione.

Indice

Sommario **vii**

1 Introduzione **1**

2 Human Pose Estimation **3**
2.1 2D Human Pose Estimation 3
2.1.1 Introduzione 3
2.1.2 Human model 4
2.1.3 Metodo regressivo 5
2.1.4 Metodo heatmap-based 5
2.1.5 Multi person pose estimation 6
2.2 3D Human Pose Estimation 15
2.2.1 Single-view 15
2.2.2 Multi-view 17

3 Image generation **19**
3.1 Architettura UNet 21
3.2 Generative Adversarial Networks 22
3.2.1 StyleGAN 24
3.2.2 SAGAN 25
3.2.3 BigGAN 26
3.3 Modelli a diffusione 26
3.3.1 Denoising Diffusion Probabilistic Models 27
3.3.2 Latent Diffusion Models 30

4 Metodo proposto **35**
4.1 Descrizione del problema 35
4.1.1 Valutazione delle tecniche di HPE 35
4.2 Soluzione proposta 37
4.3 SimpleHRNet 37
4.4 Palette 38
4.4.1 Architettura adottata 38

4.4.2 Approccio utilizzato 39
4.4.3 Data preprocessing 40
4.4.4 Training 41
4.5 Stable Diffusion Infinity 43

5 Esperimenti **45**
5.1 Dataset 45
5.1.1 Raccolta dati 45
5.1.2 Suddivisione 46
5.2 Valutazioni e risultati 47
5.2.1 HRNet a diversi tagli 47
5.2.2 Palette 49
5.2.3 Stable diffusion infinity 52
5.2.4 Test sulle sequenze preliminari 54
5.2.5 Valutazione dei keypoints in uno spazio 3D 56
5.3 Tempi di esecuzione 56

6 Conclusioni **59**

Conclusioni **60**

Ringraziamenti **61**

Bibliografia **63**

Elenco delle figure

2.1 Illustrazione della pipeline top-down. (a) Immagine in input, (b) sono state rilevate due persone dallo human detector, (c) crop delle singole immagini, (d) risultato della single-person pose estimation (e) risultato della multi-person pose estimation. [1] . 7

2.2 Illustrazione della pipeline bottom-up. (a) Immagine in input, (b) sono stati rilevati i keypoints di tutte le persone nella foto, (c) risultato della multi-pose estimation. Source: [1] 9

2.3 Illustrazione dell'architettura della rete usata da OpenPose. Source: [2] 9

2.4 Confronto fra tecniche bottom-up allo stato dell'arte sul dataset MPII. Con "Ours" é indicato il metodo OpenPose. Per le reference dei metodi coinvolti fare riferimento a: [2] 11

2.5 Analisi dei tempi di esecuazione dei metodi proposti in base al variare del numero di persone nell'immagine. Source: [2] 12

2.6 Viene illustrato come i moduli di fusione aggregano le informazioni a diverse risoluzioni. Si può notare che per svolgere up-sampling si utilizzano convoluzioni 1×1 e per down-sampling convoluzioni 3×3. Source: [3] 13

2.7 Risultati sul COCO test-dev set e confronto del metodo proposto con approcci top-down e bottom-up.Per le reference dei metodi coinvolti fare riferimento a: [3] 14

2.8 Stima delle pose in uno spazio 3D utilizzando un approccio multiview con più camere. Source: [4] 16

3.1 Task text-to-image. Le immagini sono state generate a partire dalla didascalia mostrata in figura.
Source: `https://github.com/Stability-AI/StableCascade` . 20

3.2 Task image-to-image. La prima a sinistra é l'immagine di partenza a cui é stato aggiunta una percentuale di rumore. il modello a diffusione ha poi risolto il rumore generando le immagini seguenti.
Source: `https://github.com/Stability-AI/StableCascade` . 20

3.3 Task di inpainting. Viene ricostruito il volto della donna tramite diffusione di rumore.
Source: `https://github.com/Stability-AI/StableCascade` . 21
3.4 L'architettura della UNet con la tipica forma a U. I box bianchi rappresentano le featue maps copiate dai blocchi dell'encoder.
Source: [5] 22
3.5 Viene mostrato il nuovo generatore introdotto da StyleGAN. L'input viene mappato in uno spazio latente W che controlla il generatore attraverso una normalizzazione chiamata AdaIN in ogni layer convoluzionale. Alla fine di ogni convoluzione viene aggiunto del rumore gaussiano. 24
3.6 Rappresentazione del livello di self-attention per l'architettura SAGAN 25
3.7 Rappresentazione grafica del processo di aggiunta e rimozione del rumore direttamente sui pixel dell'immagine 27
3.8 Risultati sul datataset CIFAR10. NLL indica la negative log-likelihood. Per le reference dei metodi coinvolti fare riferimento a: [6] 28
3.9 Immagini casuali prese dal dataset MS-COCO e rigenerate usando diversi modelli. Source: [7] 30
3.10 Viene rappresentata l'architettura LDM, partendo dal encoding nello spazio latente, al processo di diffusione del rumore, a quello inverso, attraversando layer di cross-attention, per poi generare l'immagine originale nello spazio dei pixels. Source: [8] 31
3.11 Risultati ottenuti nel task di unconditional image synthesis. Addestramento effettuato in uno spazio latente regolarizzato KL. Per le reference dei metodi coinvolti fare riferimento a: [8] . . . 32
3.12 Risultati ottenuti nel task layout-to-image addestrato sul dataset OpenImages e valutato su ImageNet. Source: [8] 32

4.1 Esempi di valutazioni errate dei keypoints da parte di OpenPose (a sinistra) e HRNet (a destra) 36
4.2 Rappresentazione dei 17 keypoints stimati da HRNet in un modello skeleton 38
4.3 L'architettura di base utilizzata nei modelli a diffusione. Source [5] 39
4.4 Esempi di immagini prese dal dataset. 41
4.5 Immagine che mostra in pochi passaggi il processo di denoising del modello durante la fase di sampling. 42

5.1 Esempi di immagini da vicino e lontano e relative maschere . . . 46

5.2 Esempi di tagli all'immagine svolti durante gli esperimenti con conseguente stima dei keypoints 49
5.3 a) viene applicata una maschera dal ginocchio fino alla testa, b) Palette riesce a generare la posa partendo dal rumore applicato alla maschera 51
5.4 a) il frame originale estratto dalle sequenze preliminari a cui é stato applicato una patch nera per renderlo fruibile ad HRNet, b) l'immagine generata da Stable Diffusion 55
5.5 a) il frame originale estratto dalle sequenze preliminari a cui é stato applicato una patch nera per renderlo fruibile ad HRNet, b) l'immagine generata da Palette 56

Capitolo 1

Introduzione

L'analisi dei movimenti del corpo umano è fondamentale in molteplici ambiti, dall'ambito sportivo a quello medico, dall'ingegneria biomeccanica alla robotica. Negli ultimi anni l'evoluzione dell'intelligenza artificiale ha portato grandi innovazioni anche in questo campo. In particolare, tra gli strumenti utilizzati per l'estrazione delle informazioni riguardanti la postura e i movimenti del corpo, é emersa la Human Pose Estimation (HPE). E' una tecnica che presa un'immagine permette di stimare la posizione delle articolazioni umane (keypoints) come spalle, gomiti, polsi, ginocchia e caviglie, delle persone presenti nella foto. Nel corso degli anni sono stati proposti modelli come OpenPose e HRNet che sfruttano mappe di calore per stimare la probabilità dei giunti e che attualmente rappresentano lo stato dell'arte nella Human Pose Estimation 2D. Tuttavia, la precisione di questi modelli può essere compromessa da diverse variabili, tra cui la qualità delle immagini e la visibilità delle parti del corpo.

In questa tesi l'attenzione viene posta sull'analisi della corsa di un atleta su tapis roulant, sfruttando queste tecniche di HPE. Il vincolo principale in questo lavoro é che le sequenze video dell'attività sportiva possono inquadrare solamente la parte bassa del corpo. Tramite OpenPose e HRNet é stato svolta una prima analisi per verificare se fosse possibile applicare la HPE a questi tipi di immagine. I risultati di questi esperimenti preliminari hanno evidenziato in molti casi una bassa accuratezza dei modelli. Per superare le limitazioni delle tecniche di HPE é stato ipotizzato che generando artificialmente la parte superiore del corpo nelle immagini sarebbe stato possibile ottenere una maggiore accuratezza. Recentemente, con l'avvento dei modelli a diffusione c'è stata una forte evoluzione nel campo della generazione delle immagini. Queste tecniche, ispirate al concetto di diffusione probabilistica, si sono dimostrate estremamente efficaci nel sintetizzare immagini realistiche e dettagliate a partire da input rumorosi o incompleti. Sono stati selezionati perciò due

di questi modelli generativi per ricostruire la posa umana: Palette e Stable Diffusion. Palette é una tecnica di denoising basata sulla diffusione del rumore per generare immagini realistiche e dettagliate in un task di image-to-image. Il secondo metodo studiato, Stable Diffusion é un modello a diffusione latente che sfrutta un approccio simile al precedente per produrre immagini di alta qualità partendo da input testuali. Una volta scelti i metodi di generazione sono state raccolte immagini che raffigurano nella sua interezza un singolo individuo sul tapis roulant, ne sono stati calcolati i keypoints e per ogni immagine é stata prodotta una maschera nera in grado di oscurare la persona fino ad un'altezza (taglio) specifica rappresentata dai keypoints. Successivamente sono stati svolti due task generativi con entrambi i modelli, inpainting e uncropping/outpainting. Mascherando con del rumore gaussiano l'immagine nelle zone oscurate dal taglio, i modelli sono stati in grado di generare le parti del corpo mancanti. Le nuove immagini prodotte sono state nuovamente analizzate tramite HRNet e calcolata la precisione dei modelli rispetto alle immagini tagliate. I risultati di questo processo hanno evidenziato un incremento di precisione dei keypoints predetti sulla parte bassa del corpo. Inoltre grazie ad un addestramento specifico é stato possibile utilizzare il modello Palette anche su immagini diverse da quelle del dataset di addestramento.

In questa tesi, verrà esplorato nel dettaglio il processo di sviluppo, implementazione e valutazione dell'approccio generativo. Verrà analizzato anche il funzionamento delle tecniche di Human Pose Estimation e dei modelli a diffusione impiegati. I risultati ottenuti verranno discussi svolgendo un confronto sulle prestazioni dei due modelli generativi adottati e sulle possibili applicazioni future dell'approccio presentato.

Capitolo 2

Human Pose Estimation

2.1 2D Human Pose Estimation

2.1.1 Introduzione

La Human Pose Estimation (HPE) é una tecnica studiata nel campo della visione artificiale e consiste nella stima della posizione delle parti del corpo (keypoints) di un essere umano catturate da sensori, in particolare immagini e video. Fornisce informazioni geometriche e di movimento sul corpo che possono essere applicate per risolvere un'ampia gamma di sfide in molteplici settori, per citarne alcuni: human-computer interaction [9], motion analysis [10], virtual reality (VR) [11], ambiti sportivi [12] [13] e medici [14] [15].

La Human Pose Estimation ha attirato una forte attenzione all'interno della comunità scientifica, questo ha permesso una grande cooperazione tra i ricercatori. Le principali sfide a cui si viene sottoposti affrontando un task di questo tipo riguardano la corretta rilevazione di pose complesse e arti occlusi da oggetti o vestiti, oppure una stima accurata dei giunti di un gruppo di persone in una foto (multi-person pose estimation). Negli ultimi anni si é assistito a significativi progressi in questo campo, principalmente attribuiti all'impiego diffuso delle tecniche di Deep Learning. L'adozione di reti neurali profonde, reti convoluzionali e architetture specifiche come DeepPose [16], Hourglass [17] e HRNet [3], ha notevolmente migliorato l'accuratezza e robustezza nella stima delle articolazioni umane, soprattutto in immagini con occlusioni molto evidenti. In particolare hanno segnato un significativo passo avanti rispetto agli approcci tradizionali, come HOG [18], Edgelet [19] e pictorial structure model [20]. Tuttavia è importante considerare che il successo nell'applicazione di tecniche di deep learning spesso dipende dalla disponibilità di un ampio dataset di immagini correttamente annotato. Esistono diverse condizioni che ci

aiutano a classificare un task di HPE [21]:

- il tipo di immagine fornita in input: RGB, Depth, Infra-red;
- il numero di camere (singleview o multiview);
- il modello del corpo umano che si utilizza (kinematic, planar, o volumetric);
- la rappresentazione finale della posa che vogliamo ottenere (2D/3D pose estimation);
- il numero di persone presenti nell'immagine (single-person o multi-person).

In questa sezione andremo a trattare la Human Pose Estimation 2D e descriveremo gli approcci utilizzati per risolvere un task di single-person e multi-person pose estimation. In particolare ci soffermeremo sui due metodi di risoluzione più utilizzati: regressivo e heatmap-based.

2.1.2 Human model

Le tecniche adottate per risolvere un task di Human Pose Estimation sfruttano i cosiddetti human model, rappresentazioni visive del corpo umano che permettono di organizzare le posizioni dei giunti predetti dall'immagine. Questi keypoints sono combinati per costruire la posa che vogliamo catturare, pertanto la scelta dello human model diventa un aspetto cruciale. Esistono fondamentalmente tre tipi di modelli:

- Kinematic Model: conosciuto anche come skeleton viene utilizzato sia per la stima di pose 2D che 3D. E' molto flessibile, modella la relazione fra le varie parti del corpo, molto limitato per quanto riguarda la rappresentazione della struttura e la forma del corpo.
- Planar Model: chiamato anche contour based model viene utilizzato per la pose estimation 2D, riesce a modellare l'aspetto, la forma e i contorni della figura. Solitamente le parti del corpo vengono approssimate con dei rettangoli. Un esempio popolare é Active Shape Model [22].
- Volumetric Model: rappresenta una struttura tridimensionale detta mesh, modella la forma e la struttura di un corpo umano. E' utilizzata per la stima di pose 3D, un esempio sono GHUM e GHUML [23].

2.1.3 Metodo regressivo

Toshev [16] definisce i metodi regressivi come l'applicazione di framework end-to-end per apprendere un mapping diretto tra l'immagine e la posizione dei giunti o dei parametri del corpo. Il metodo DeepPose, precedentemente citato, segna il confine tra l'utilizzo dei metodi tradizionali e l'introduzione di DNN per svolgere HPE e dato l'impressionante aumento delle performance queste tecniche hanno totalmente rimpiazzato in poco tempo quelle classiche. Questi nuovi modelli apprendono in maniera automatica i parametri direttamente dal dataset di addestramento attraverso l'utilizzo di reti come le Convolutional Neural Network, in grado di estrarre feature dalle immagini in maniera molto più efficiente rispetto ai metodi "hand-crafted" adottati in precedenza. Rispetto ad una soluzione heatmap, l'approccio regressivo può risultare complessivamente più semplice da implementare e comprendere, ma anche meno robusto a variazioni di luce nella foto, pose troppo complesse e occlusioni. Oltre a DeepPose sono state proposte ulteriori architetture come la compositional pose regression [24] basata su ResNet50, la prima rete transformer-based a cascata [25] che sfrutta la self-attention per identificare la correlazione spaziale dei giunti.

2.1.4 Metodo heatmap-based

L'approccio heatmap-based consiste nel predire la posizione dei giunti del corpo producendo mappe di calore che rappresentano la probabilità della presenza di ciascuna articolazione in ogni punto dell'immagine. Invece che stimare le coordinate 2D direttamente, vengono applicati dei kernel gaussiani in ogni posizione del punto chiave etichettato con l'obiettivo di predire K mappe di calore per i K keypoints da identificare. Questo metodo é sicuramente più robusto rispetto a cambiamenti di luce e occlusioni nell'immagine, presenta però una difficoltà maggiore nell'implementazione paragonato ad un semplice task di regressione.

L'approccio a mappe di calore é stato inizialmente introdotto da Tompson [26], che proponeva la generazione di heatmap passando l'immagine in parallelo ad una ConvNet a diverse risoluzioni, per catturare feature su scale differenti. Questo lavoro é stato ripreso da Newell et al. [17] che hanno proposto un'architettura a clessidra, detta Stacked Hourglass poiché sfrutta le tecniche di up-sampling e down-sampling dell'immagine per estrarre feature a diverse risoluzioni. Newell nel suo approccio decide di rimuovere il modello grafico e rispetto a Tompson propone una singola pipeline che processa l'immagine e produce in output le heatmap. Seguendo la linea di questi lavori Sun et al.

presentano High-Resolution Net (HRNet) [3] un modello in grado di apprendere feature ad alta risoluzione connettendo sotto-reti a multi-risoluzione e conducendo fusioni su multi-scala. Questo approccio migliora ulteriormente le predizioni dei keypoints all'interno delle heatmap.

2.1.5 Multi person pose estimation

Il problema di multi-person pose estimation é sicuramente più complesso del single-person in quanto analizzando un'immagine non si conoscono né la posizione dei giunti né la posizione delle persone nella foto. Nascono perciò due importanti sfide, identificare tutte le persone nella foto e stimarne i keypoints. Nel corso degli anni sono nati due approcci per risolvere questo tipo di task: top-down e bottom-up.

Approccio top-down

Il metodo top-down é sicuramente quello più semplice fra i due e consiste nell'identificare una singola persona nell'immagine, stimarne i keypoints e calcolare la posa. Sarebbe quindi sufficiente ottenere un buon modello che riesca a segmentare o rilevare un bounding-box di ogni persona nella foto e sfruttare le tecniche usate per la single-person pose estimation, come mostrato in Figura 2.1. Questa pipeline soffre del cosiddetto "early committment": se il modello di person detector fallisce, cosa che accade spesso in presenza di un folto gruppo di persone, risulta impossibile calcolare la posa di ogni singolo individuo. Inoltre come spiegato da Cao et al. [27], potremmo avere un aumento di tempi e di risorse computazionali in quanto il runtime é proporzionale al numero di persone nelle immagini, per ognuna rilevata viene eseguito un estimatore di posa. Nella prima fase della pipeline top-down, vengono utilizzate tecniche di object detection come Fast R-CNN [28], Faster R-CNN [29], Mask R-CNN [30], e SSD [31] rifinite per il rilevamento delle persone. Faster R-CNN é strutturata in due fasi. La prima viene detta Region Proposal Network (RPN), prende in input un'immagine RGB e fornisce in output i bounding-box candidati a contenere la persona o l'oggetto cercato. La seconda fase é rappresentata dalla Fast R-CNN che sfrutta una tecnica chiamata Region of Interest (ROI) Pooling per estrarre le feature dalle regioni proposte e predire in output la classificazione e la regressione accurata dei bounding-box rilevati. Mask R-CNN é una tecnica di segmentazione di oggetti in un'immagine introdotta da Facebook. Ispirata alla Faster R-CNN adotta le stesse due fasi con la differenza che in parallelo viene prodotta una maschera binaria per ogni ROI [30].

Un importante lavoro svolto in ambito HPE top-down é stato presentato

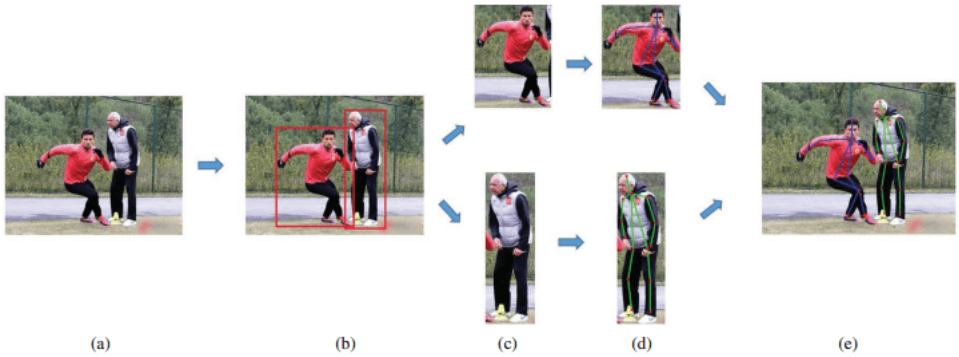

Figura 2.1: Illustrazione della pipeline top-down. (a) Immagine in input, (b) sono state rilevate due persone dallo human detector, (c) crop delle singole immagini, (d) risultato della single-person pose estimation (e) risultato della multi-person pose estimation. [1]

da Chen [32]: Cascaded Pyramid Network. L'architettura costruita per stimare i keypoints si suddivide in due sottoreti: GlobalNet e RefineNet. La prima si basa sulla struttura della Feature Pyramid Network (FPN) [33] che consente di mantenere le informazioni spaziali e semantiche delle feature estratte. La seconda é concatenata alla prima ed è costruita per gestire i cosiddetti "hard" keypoints, difficilmente stimabili dalla prima sottorete. Nel lavoro di Chen viene fatta un'importante considerazione sull'importanza dello human detector nella stima della posa. Si testano le performance di un pose estimator con differenti detector. Vengono utilizzate tecniche di "ensemble" e diverse FPN, per predire le regioni di interesse. Dai risultati ottenuti si nota un iniziale incremento di Average Precision (AP) in fase di stima dei keypoints solo quando il detector ha prestazioni basse, salendo con la precisione del detector non notiamo lo stesso fenomeno anche per il pose estimator che pian piano appiattisce la curva. Una possibile conclusione, come viene spiegato da Dang et al. [1], é che le persone non rilevate nell'immagine siano esempi difficili anche per lo human pose estimator.

Approccio bottom-up

A differenza della pipeline top-down, gli approcci bottom-up sono affascinanti [27] poiché offrono robustezza al early commitment e hanno il potenziale per rimuovere la dipendenza tra la complessità di esecuzione e il numero di persone presenti nell'immagine. Come possiamo notare in Figura 2.2, l'obiettivo é stimare direttamente i keypoints di tutte le persone nell'immagine e infine fornire le varie pose in output. Queste tecniche sono sicuramente più rapide

rispetto a quelle top-down in quanto non hanno bisogno di rilevare la posa di ogni persona separatamente, ma lo fanno in un'unica computazione.

Il primo approccio bottom-up basato sul deep learning si chiama DeepCut [34] e rivoluzionò il campo della HPE introducendo un modello in grado di partizionare ed etichettare le parti del corpo in un'immagine. Il modello si basa su una Fast R-CNN riadattata in grado di produrre in output la regione interessata e la classificazione di ogni parte del corpo di tutti gli individui nella foto. Viene poi formulato un "Integer Linear Program (ILP)" che esegue implicitamente una "non-maximum suppression" per raggruppare le parti del corpo trovate in una posa. Successivamente gli autori introducono DeeperCut [35], un approccio che va a migliorare DeepCut principalmente in tre aspetti: (1) si passa ad un'architettura basata su ResNet per il detector; (2) vengono introdotti i termini di "image-conditioned pairwise", sfruttando l'ampio campo recettivo della rete é possibile non solo predire le singole parti ad una posizione fissata, ma anche effettuare inferenze sulle relazioni tra le diverse parti vicine; (3) cambia il metodo di ottimizzazione riducendo i tempi di computazione.

Nel 2017 Cao et al. [27] introducono quello che é ad oggi una delle tecniche migliori per risolvere un task di Human Pose Estimation, le "Part Affinity Fields (PAFs)". Gli stessi autori successivamente propongono OpenPose, un sistema realtime per multi-person pose estimation che sfrutta le PAFs per associare le parti del corpo ai soggetti presenti nella foto. Secondo quanto riportato questo metodo riesce ad ottenere un'alta accuratezza ed elevate performance realtime a prescindere dal numero di persone nell'immagine. In particolare OpenPose introduce una pipeline dove in una prima fase vengono prodotte le "confidence maps" che stimano la presenza del keypoints in quell'area dell'immagine e le PAFs che mantengono le informazioni sia sulla posizione che sull'orientamento dell'arto. Nella seconda fase tramite queste informazioni viene ricostruita la posa di tutti gli individui tramite un algoritmo "bipartite graphs".

Figura 2.2: Illustrazione della pipeline bottom-up. (a) Immagine in input, (b) sono stati rilevati i keypoints di tutte le persone nella foto, (c) risultato della multi-pose estimation. Source: [1]

OpenPose

In questa sezione andremo a descrivere più nel dettaglio il framework bottom-up OpenPose.

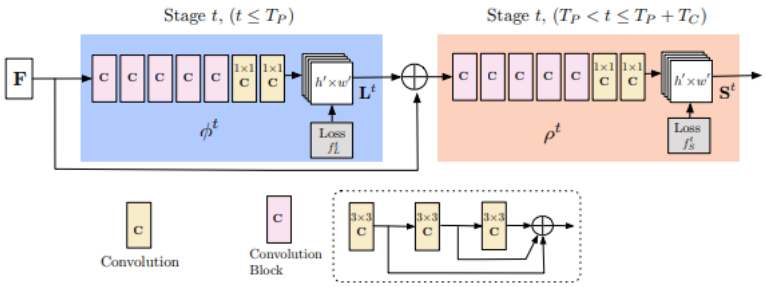

Figura 2.3: Illustrazione dell'architettura della rete usata da OpenPose. Source: [2]

L'architettura proposta dagli autori, come mostrato dalla Figura 2.3, si suddivide in due fasi multi-stage e segue l'approccio adottato per le Convolutional Pose Machines [36]. Inizialmente l'immagine RGB viene analizzata da i primi 10 layers di una VGG-19 generando in output delle feature map F che saranno l'input della prima fase. L'obiettivo di questa fase é predire le Part Affinity Fields che saranno nuovamente concatenate alle feature estratte precedentemente e rifinite per un numero TP di iterazioni:

$$L^t = \phi^t(F, L^{t-1}), \forall 2 \leq t \leq TP$$

ϕ^t si riferisce all'inferenza della CNN allo stage t. Questo procedimento viene applicato anche per la fase successiva, dopo aver predetto le PAFs si passa alle confidence maps:

$$S^t = \rho^t(F, L^{TP}, S^{t-1}), \forall TP < t \leq TP + TC$$

dove ρ^t indica l'inferenza della CNN, TC il numero di iterazioni svolte e L^{TP} l'output della fase precedente. Secondo gli autori é fondamentale mantenere questo tipo di approccio in quanto intuitivamente dalle PAFs si riesce a dedurre la posizione dei giunti che hanno però bisogno di essere assegnate correttamente alle persone nell'immagine. Come si può notare dalla Figura 2.3 viene utilizzato un approccio supervisionato per le due fasi descritte attraverso l'utilizzo di una loss L2, in particolare:

$$\sum_{c=1}^{C} \sum_{p} W(p) \cdot ||L_c^{t_i}(p) - L_c^*(p)||_2^2$$

dove $L_c^*(p)$ sono le etichette, W una maschera binaria che viene utilizzate per evitare di penalizzare i true positive durante l'addestramento e p una specifica posizione nell'immagine 2D. Come specificato in precedenza la pipeline si compone di una prima fase dove vengono prodotte le confidence maps, ovvero una rappresentazione 2D della confidenza che una particolare parte del corpo sia collocata in specifici punti dell'immagine. Visivamente é rappresentata da una mappa di calore intorno alle parti del corpo. Per poter procedere ad un addestramento supervisionato é necessario generare le mappe di confidenza reali partendo dai keypoints annotati. La confidenza in ogni specifica posizione p nell'immagine viene calcolata tramite la seguente formula:

$$S_{j,k}^*(p) = exp(\frac{-||p - x_{j,k}||_2^2}{\theta^2})$$

θ controlla la diffusione della mappa, j indica il giunto, k identifica l'individuo nella foto e $x_{j,k}$ indica la posizione reale del giunto. Ottenuta una matrice $h \times w$ per ogni keypoints annotato, se ne otterrà una unica che andrà a rappresentare la confidence map $S * j$ reale, tramite un'operazione di max fra le confidenze di ogni matrice generata. Predette le posizioni di ogni singola parte del corpo l'attenzione si sposta su come assemblarle in una posa. Il problema riscontrato dagli autori é la difficoltà di associazione fra i vari punti quando si presentano immagini in cui i keypoints di persone diverse sono molto vicini fra di loro. Entrano qui in gioco le PAFs che semplificano enormemente il problema mantenendo le informazioni sia sulla posizione che sull'orientamento degli arti. Per ogni arto viene disposta una matrice $L_{c,k}^*(p)$ dove in ogni posizione p dell'immagine esiste un vettore unitario v bidimensionale:

$$v = (x_{j2,k} - x_{j1,k})/||x_{j2,k} - x_{j1,k}||_2$$

se p é in corrispondenza di un arto specifico, 0 altrimenti. Ottenute le confidence maps e tramite non-maximum suppression le parti del corpo candidate,

l'associazione fra esse si riduce ad un problema di maximum weight bipartite graph, dove i nodi del grafo sono i keypoints e gli archi sono tutte le possibili connessioni. Gli autori utilizzano l' "Hungarian algorithm" [37] per risolvere il problema in tempo polinomiale.

Il metodo proposto é stato valutato in tre benchmark: dataset MPII suddiviso in 3844 gruppi di individui per il training e 1758 per il test per un totale complessivo di oltre 25000 immagini [38]; COCO keypoints challenge dataset [39]; il loro FOOT dataset per la valutazione dei keypoints rilevati nei piedi. In seguito analizzeremo i primi due benchmark. La metrica di valutazione utilizzata é mean Average Precision (mAP) su tutti i keypoints rilevati seguendo la metrica PCKh.

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
Subset of 288 images as in [1]									
Deepcut	73.4	71.8	57.9	39.9	56.7	44.0	32.0	54.1	57995
Iqbal et al.	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	10
DeeperCut	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	230
Newell et al.	91.5	87.2	75.9	65.4	72.2	67.0	62.1	74.5	-
ArtTrack	92.2	**91.3**	80.8	71.4	**79.1**	72.6	67.8	**79.3**	**0.005**
Fang et al.	89.3	88.1	80.7	75.5	73.7	**76.7**	**70.0**	79.1	-
Ours	**92.9**	**91.3**	**82.3**	72.6	76.0	70.9	66.8	79.0	**0.005**
Full testing set									
DeeperCut	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al.	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Levinko et al.	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
ArtTrack	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	**0.005**
Fang et al.	88.4	86.5	78.6	**70.4**	74.4	**73.0**	**65.8**	76.7	-
Newell et al.	**92.1**	89.3	78.9	69.8	76.2	71.6	64.7	77.5	-
Fieraru et al.	91.8	**89.5**	**80.4**	69.6	**77.3**	71.7	65.5	**78.0**	-
Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
Ours	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	**0.005**

Figura 2.4: Confronto fra tecniche bottom-up allo stato dell'arte sul dataset MPII. Con "Ours" é indicato il metodo OpenPose. Per le reference dei metodi coinvolti fare riferimento a: [2]

Sul dataset MPII (vedere Figura 2.4) sono stati effettuati due esperimenti per confrontare le prestazioni con i precedenti metodi bottom-up, il primo su un sottoinsieme di 288 immagini dal test set e il secondo su tutto esso. Per quanto riguarda il primo gli autori registrano prestazioni allo stato dell'arte e tempi di esecuzione sei volte inferiori rispetto alle altre tecniche. Il benchmark sul dataset COCO ha comparato i risultati di approcci top-down e bottom-up e in entrambe le categorie OpenPose si é classificato tra i metodi con precisione più alta. Gli autori hanno inoltre presentato un'auto-analisi sul dataset COCO tra il nuovo modello body+foot e l'approccio adottato nel loro primo lavoro sulle PAFs [27], notando un aumento di accuratezza del 6.9% .

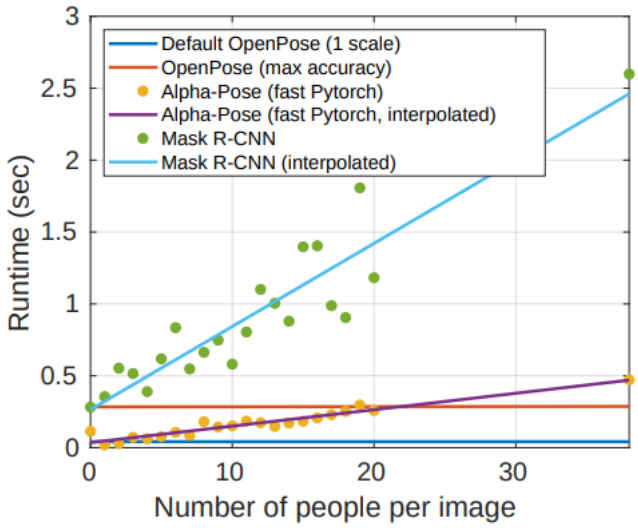

Figura 2.5: Analisi dei tempi di esecuazione dei metodi proposti in base al variare del numero di persone nell'immagine. Source: [2]

Un'ultima interessante analisi riguarda i tempi di inferenza rispetto al numero di persone presenti nell'immagine. Come confronto sono stati presi due metodi a quel tempo allo stato dell'arte Mask-RCNN e Alpha-Pose [40]. I test sono stati effettuati su un dataset comune su una GPU NVIDIA 1080 Ti. I risultati hanno evidenziato una forte dipendenza da parte dei metodi top-down rispetto al numero di persone presenti nell'immagine. In particolare il tempo di esecuzione é proporzionato al numero di RoI proposte. Il tempo di esecuzione di OpenPose viene valutato suddividendo la computazione in due fasi. La prima riguarda il runtime CNN che presenta una complessità O(1) costante. La seconda riguarda il parsing multi-person di complessità O(n^2) dove n é il numero di persone nell'immagine che presenta però un tempo due volte inferiore al primo runtime. Come si può notare dalla Figura 2.5 rispetto agli altri modelli confrontati, OpenPose offre invarianza nell'inferenza all'aumentare delle persone nell'immagine.

HRnet

In questa sezione andremo a descrivere HRNet [3] uno dei metodi attualmente allo stato dell'arte nella Human Pose Estimation che é inoltre stato utilizzato in questo lavoro di tesi. Il modello di deep learning proposto dagli autori connette stream di convoluzioni "high-to-low" in parallelo per poter mantenere rappresentazioni ad alta risoluzione durante tutto il processo di estrazione delle mappe di calore dall'immagine. Nel dettaglio la rete presenta inizialmente due convoluzioni 3×3 che diminuiscono la risoluzione dell'im-

magine di 1/4, il corpo centrale racchiude diverse tecniche che andremo ad analizzare: sequential multi-resolution subnetworks, parallel multi-resolution convolutions e repeated multi-resolution fusions. Gli autori definiscono "stage" una sottorete composta da un insieme di convoluzioni che viene collegata in serie ad una successiva effettuando il compito di down-sampling delle feature. Ad esempio consideriamo N_{sr} come una sottorete allo stage s con un indice di risoluzione r, la sua risoluzione é definita come $\frac{1}{2^{r-1}}$ rispetto a quella del primo stage. Una rete high-to-low con ad esempio 4 stage può essre definita in questo modo:

$$N_{11} \longrightarrow N_{22} \longrightarrow N_{33} \longrightarrow N_{44}$$

La tecnica di parallel multi-resolution convolutions consiste nell'aggiungere insiemi di convoluzioni ad alta risoluzione per il primo stage e gradualmente inserire fasi di media e bassa risoluzione. Queste operazioni di riduzione o aumento di up-sampling o down-sampling vengono effettuate in parallelo fondendo ad una certa fase i risultati (vedi Figura 2.6).

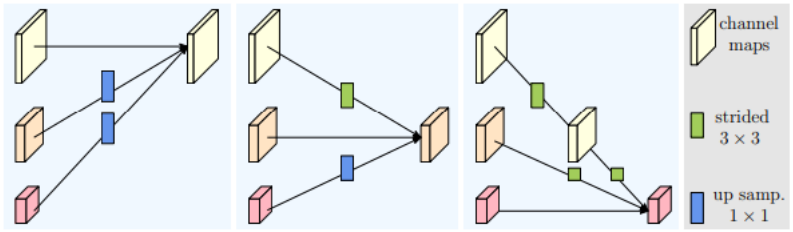

Figura 2.6: Viene illustrato come i moduli di fusione aggregano le informazioni a diverse risoluzioni. Si può notare che per svolgere up-sampling si utilizzano convoluzioni 1×1 e per down-sampling convoluzioni 3×3. Source: [3]

L'obiettivo di queste fusioni é quello di aggregare informazioni ottenute parallelamente da risoluzioni differenti tramite le cosiddette "exchange units", convoluzioni 1×1 o 3×3. Descriviamo più nel dettaglio questo processo: consideriamo come inputs s response maps $X_1, X_2, ..., X_s$ e come output $Y_1, Y_2, ..., Y_s$. Ogni output viene definito come:

$$Y_k = \sum_{i=1}^{s} a(X_i, k)$$

la funzione $a(X_i, k)$ consente di fare up-sampling o down-sumpling dalla risoluzione i a k. In particolare viene utilizzata la tecnica del "nearest neighbor" per le convoluzioni 1×1.

La stima dei keypoints finali viene svolta tramite l'estrazione di heatmap

dall'ultima exchange unit. Le mappe di calore reali vengono calcolate applicando filtri gaussiani 2D con deviazione standard 1 pixel centrati sul keypoint annotato. L'addestramento viene svolto definendo una funzione di loss MSE. In totale vengono addestrate due reti, una più piccola HRNet-W32 e una più grande HRNet-W48, la differenza tra le due risiede nella larghezza delle features estratte negli ultimi tre stage ad high-resolution.

La fase di training consiste in una prima fase di rilevamento della persona nell'immagine tramite l'approccio descritto in SimpleBaseline [41], dopodiché il box prodotto viene ridimensionato a 256×192 o 384×288, viene impostata una pipeline di preprocessing che prevede random flip, random scale e random rotation. L'addestramento dei modelli é stato effettuato su un sottoinsieme del dataset COCO che racchiude 57.000 immagini etichettate con 17 keypoints. La metrica di valutazione è quella standard di COCO ovvero Object Keypoints Similarity (OKS):

$$\frac{\sum_i exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}$$

dove d_i è la distanza euclidea tra keypoints rilevati e reali, v_i é un flag che indica se la parte del corpo é visibile o meno nell'immagine (0 non visibile, 1 o 2 visibile), s indica la scala della persona, k_i é una costante per-keypoint calcolata sul dataset COCO.

Method	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^M	AP^L	AR
Bottom-up: keypoint detection and grouping										
OpenPose	–	–	–	–	61.8	84.9	67.5	57.1	68.2	66.5
Associative Embedding	–	–	–	–	65.5	86.8	72.3	60.6	72.6	70.2
PersonLab	–	–	–	–	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet	–	–	–	–	69.6	86.3	76.6	65.0	76.3	73.5
Top-down: human detection and single-person keypoint detection										
Mask-RCNN	ResNet-50-FPN	–	–	–	63.1	87.3	68.7	57.8	71.4	–
G-RMI	ResNet-101	353×257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression	ResNet-101	256×256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	–
G-RMI + extra data	ResNet-101	353×257	42.6M	57.0	68.5	87.1	75.5	65.8	73.3	73.3
CPN	ResNet-Inception	384×288	–	–	72.1	91.4	80.0	68.7	77.2	78.5
RMPE	PyraNet	320×256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	–
CFN	–	–	–	–	72.6	86.1	69.7	78.3	64.1	–
CPN (ensemble)	ResNet-Inception	384×288	–	–	73.0	91.7	80.9	69.5	78.1	79.0
SimpleBaseline	ResNet-152	384×288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNet-W32	HRNet-W32	384×288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNet-W48	HRNet-W48	384×288	63.6M	32.9	**75.5**	**92.5**	**83.3**	**71.9**	**81.5**	**80.5**
HRNet-W48 + extra data	HRNet-W48	384×288	63.6M	32.9	**77.0**	**92.7**	**84.5**	**73.4**	**83.1**	**82.0**

Figura 2.7: Risultati sul COCO test-dev set e confronto del metodo proposto con approcci top-down e bottom-up.Per le reference dei metodi coinvolti fare riferimento a: [3]

Dai risultati mostrati in Figura 2.7 si nota una netta superiorità di HRNet su tutte le altre metodologie prese in considerazione sul dataset COCO. Ulteriori benchmark sono stati proposti sul dataset MPII e su PoseTrack2017 [42], dove è stato svolto un task di single-person pose estimation, e anche in quest'ultimi vengono riportati risultati allo stato dell'arte.

2.2 3D Human Pose Estimation

Come descritto da Zheng et al. [43] la Human Pose Estimation 3D é un task di visione artificiale che consiste nel stimare la posizione e l'orientamento dei keypoints da immagini o video, in uno spazio 3D. Può essere applicato in diversi ambiti come l'industria cinematografica 3D, realtà virtuale, analisi sportiva e medica. Sebbene in ambito 2D siano stati raggiunti grandi traguardi, la HPE 3D rimane un'importante sfida in ambito di ricerca.

Molte tecniche esistenti oggi cercano di stimare la posa 3D da immagini monoculari, quindi da un solo punto di vista, ad esempio traslando i keypoints calcolati su una posa 2D in uno spazio 3D. Questo approccio presenta il problema dell'occlusione, ovvero, se una parte del corpo o un altro oggetto sono sovrapposti ad un'altra parte del corpo é impossibile stimare, da un singolo punto di vista, la profondità di quest'ultima. E' possibile risolvere in parte questo problema utilizzando camere multi-view. Altre difficoltà riscontrate nella ricostruzione della posa da un'immagine monoculare sono ambiguità della profondità e insufficienti dati per l'addestramento del modello.

Nel corso di questi anni nell'ambito Human Pose Estimation sono stati utilizzati diversi sensori per la facilitare la stima della posa quali: sensori di profondità, Werable Inertial Measuraments Units (IMUs), sensori a radiofrequenza. Tuttavia gli sforzi maggiori si sono concentrati nell'utilizzo di approcci con immagini RGB che si possono categorizzare in: single-view single person, single-view multi-person e multi-view.

2.2.1 Single-view

Affrontando un problema single-view single person in ambito HPE 3D possiamo voler estrarre solo la posa rappresentata dallo human model skeleton oppure recuperare le informazioni che ci permettono di rappresentare una mesh 3D della persona. Nel primo caso ci si approccia in due modi: per stima diretta o per lifting da 2D a 3D. La stima diretta comporta la predizione in output di coordinate tridimensionali senza passare da una posa bidimensionale sfruttando

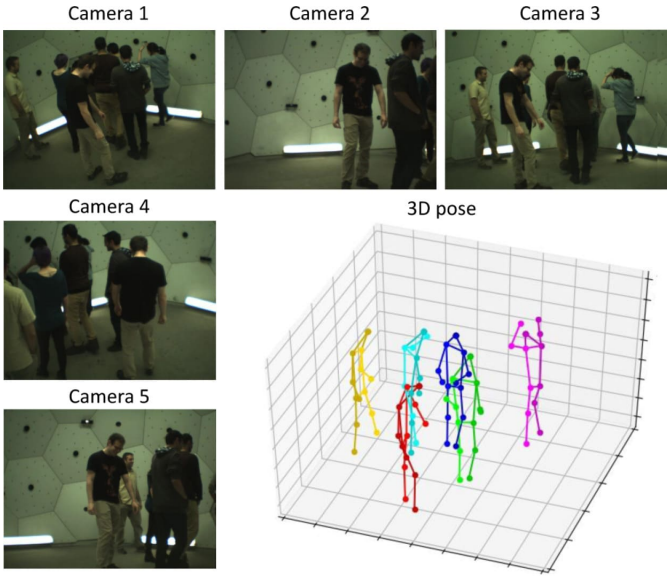

Figura 2.8: Stima delle pose in uno spazio 3D utilizzando un approccio multiview con più camere. Source: [4]

le tecniche di heatmap e regressione descritte precedentemente.

Nel corso degli anni sono state proposte diverse tecniche per risolvere il problema come ad esempio " Compositional human pose regression" [24] che adotta una riparametrizzazione della posa sfruttando le ossa invece che i keypoints definendo una "composotional loss" che codifica le dipendenze a lungo raggio. Nel caso del lifting invece si ottengono i keypoints in uno spazio bidimensionale e si cerca di ricondurli ad una posa 3D. Generalmente questi metodi funzionano meglio rispetto a stimare direttamente i punti, grazie alle prestazioni allo stato dell'arte dei modelli per HPE 2D.

Una tecnica molto promettente proposta per risolvere il task di lifting é la Graph Convolutional Networks [44] dove la posa viene trattata come un grafo, i giunti come nodi e le ossa come archi. Sono stati svolti dei tentativi anche utilizzando video al posto di immagini monoculari sfruttando le reti LSTM per catturare informazioni temporali da sequenze di pose.

Esistono poi una tipologia di approcci basati su Human Mesh Recovery (HMR) che sfruttano modelli SMPL (Skinned Multi-Person Linear) [45] per ricostruire una rappresentazione volumetrica di un corpo umano partendo da immagini RGB. Le human mesh riescono a catturare aspetti come la forma e gli angoli tridimensionali dei giunti.

Multi-person 3D

Anche in questo caso quando ci si approccia ad un task multi-person si sfruttano tecniche top-down e bottom-up. Nel primo caso viene svolta una detection delle persone nell'immagine, preso un singolo individuo si stima il punto centrale del corpo (root coordinate) e la posa tramite un modello di root-relative 3D single-person HPE. Nell'approccio bottom-up prima si predicono tutti i giunti e le depth maps poi si associano i keypoints agli individui basandosi su queste informazioni.

Gli aspetti positivi e negativi dei due approcci sono quelli descritti per la HPE 2D con la differenza che dovendo ricostruire una human mesh i metodi top-down funzionano meglio in quanto sfruttano tecniche di 3D single-person human mesh recovery, mentre i bottom-up necessitano di un modello aggiuntivo di regressione [43].

2.2.2 Multi-view

Precedentemente abbiamo citato il problema dell'occlusione e spiegato che non era risolvibile analizzando una persona in un'immagine da un singolo punto di vista, non avendo sufficienti informazioni per stimare la profondità del giunto coperto. Una possibile soluzione é stimare una posa 3D con approccio multi-view. L'obiettivo principale di quest'approccio é combinare le informazioni sulla posizione dei keypoints raccolte da camere con diverse view (vedi Figura 2.8). Recentemente sono state proposte soluzioni come Multi-view Pose transformer (MvP) [46] che sfrutta le embeddings query per combinare informazioni provenienti da viste differenti e Multi-view fusion model (MetaFuse) [47] un modello pre-addestrato adattabile facilmente a nuove impostazioni della camera svolgendo un fine-tuning con pochissimi dati.

Capitolo 3

Image generation

Il task di image generation ad opera di modelli di intelligenza artificiale é emerso negli ultimi anni grazie alla continua evoluzione delle reti convoluzionali e delle tecniche di deep learning. Questi modelli sono in grado di apprendere le distribuzioni di probabilità dei dati di addestramento e generare nuove istanze che condividono caratteristiche simili a quelle del set di dati originale. La generazione di immagini ha una vasta gamma di applicazioni pratiche, tra cui cinema, videogames, realtà virtuale, arte e design. Nel corso degli anni sono emerse due metodologie di sintesi di un'immagine:

- Conditional image generation: il modello generativo produce immagini basandosi su una specifica condizione o contesto. Questa condizione può essere rappresentata da un'etichetta di classe, una descrizione testuale o un'altra immagine: text-to-image (vedi Figura 3.1), layout-to-image e image-to-image (vedi Figura 3.2). Esempi di tecniche in grado di risolvere questo task includono: Conditional Generative Adversarial Networks (cGANs) [48], Conditional Variational Autoencoders (cVAEs) [49] e i modelli a diffusione [6] condizionali.

- Unconditional image generation : il modello generativo produce immagini senza alcuna condizione esterna apprendendo le caratteristiche principali del set di dati di addestramento e generando immagini coerenti e realistiche. Un esempio sono le Generative Adversarial Networks (GANs) [50] e i modelli a diffusione non condizionali.

Inoltre, sono nati diversi sotto-task che possono ricadere in entrambe le categorie sopracitate, come:

- Image traslation: questo task riguarda la manipolazione delle caratteristiche dell'immagine, come il cambio di stile, la modifica dell'illuminazione, la trasformazione di oggetti, ecc. I modelli sono in grado di imparare le trasformazioni desiderate e applicarle all'immagine in input;

- Inpainting: il modello basandosi sul contesto e la struttura, cerca di ricostruire parti mancanti o danneggiate di un'immagine. In Figura 3.3 mostriamo un esempio di inpainting;

- Super-resolution: questo sotto-task consiste nel generare immagini ad alta risoluzione a partire da immagini a bassa risoluzione, preservando i dettagli e migliorando la qualità;

Figura 3.1: Task text-to-image. Le immagini sono state generate a partire dalla didascalia mostrata in figura.
Source: `https://github.com/Stability-AI/StableCascade`

Figura 3.2: Task image-to-image. La prima a sinistra é l'immagine di partenza a cui é stato aggiunta una percentuale di rumore. il modello a diffusione ha poi risolto il rumore generando le immagini seguenti.
Source: `https://github.com/Stability-AI/StableCascade`

Figura 3.3: Task di inpainting. Viene ricostruito il volto della donna tramite diffusione di rumore.
Source: `https://github.com/Stability-AI/StableCascade`

La generazione di immagini rappresenta un campo di ricerca in continua evoluzione, con nuove tecniche e applicazioni emergenti che promettono di migliorare ulteriormente la qualità e la varietà delle immagini generate. Le più promettenti sorte fin'ora sono: le Generative Adversarial Networks (GANs) [50] e i Diffusion Models [6].

3.1 Architettura UNet

Prima di entrare nel dettaglio delle tecniche e dei modelli generativi per le immagini, é bene illustrare brevemente l'architettura neurale che accomuna molti di essi. Viene detta UNet [5] ed é stata presentata inizialmente per il task di Biomedical Image Segmentation. Unet é un autoencoder che consiste in due parti distinte. La prima viene detta "contracting path" ed é un classico processo di convoluzione composto da 4 encoder che dimezza le dimensioni spaziali dei dati e duplica il numero di feature channels ad ogni blocco. Ogni blocco corrisponde a due convoluzioni 3×3 seguite da una funzione di attivazione ReLU e un layer di max-pooling 2×2. Quest'ultima operazione viene svolta per ridurre il numero di parametri addestrabili e diminuire il costo computazionale. Ogni blocco dell'encoder é direttamente connesso tramite "skip connections" ai blocchi del decoder fornendo informazioni aggiuntive che possono aiutare a generare feature semanticamente migliori. La seconda parte é detta "expansive path" é costituita da layer convoluzionali trasposti che formano un decoder. Ogni blocco, composto da due layers 3×3 e uno 2×2, accoglie le features dalle skip connections concatenandole a quelle prodotte da tutto il processo di encoding. L'ultimo layer passa attraverso ad una convoluzione 1×1 con attivazione sigmoide che produce la maschera semantica. Gli aspetti chiave di quest'architettura sono il numero elevato di canali che trasmettono informazioni

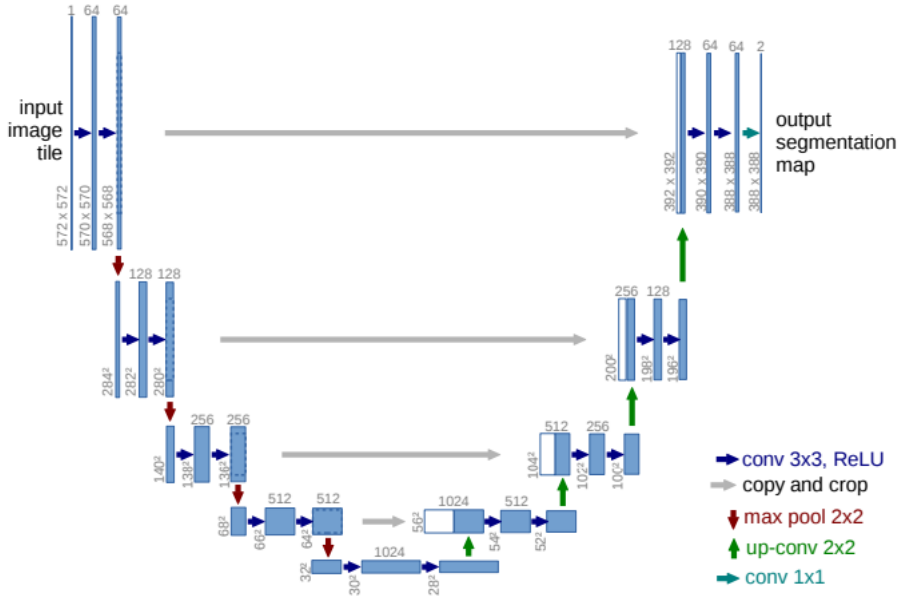

Figura 3.4: L'architettura della UNet con la tipica forma a U. I box bianchi rappresentano le featue maps copiate dai blocchi dell'encoder. Source: [5]

sul contesto dell'immagine, la overlap-tile strategy che permette di segmentare immagini di qualsiasi dimensione e la specchiatura dell'immagine di input per predire i pixels ai bordi dove il contesto é mancante. L'architettura viene mostrata in Figura

3.2 Generative Adversarial Networks

Le GANs sono modelli generativi che sfruttano due reti neurali profonde che competono una contro l'altra per generare dati simili visti durante l'addestramento. Come descritto da Costa et al. [51] quest'architettura combina un discriminatore D e un generatore G. Il generatore riceve solitamente del rumore in input che trasforma in un campione fake in output. Il discriminatore riceve sia i dati generati dal generatore che gli esempi reali dal dataset e durante l'addestramento deve imparare a classificarli in modo corretto. Questi componenti sono addestrati in maniera simultanea l'uno contro l'altro fino a che il discriminatore non é più in grado di distinguere fra esempi reali e fake. Nelle fasi iniziali dell'addestramento il generatore produce esempi facilmente classificabili come fake quindi il discriminatore impara in fretta a distinguerli. La funzione di loss del discriminatore é definita da una binary cross-entropy:

$$L_D = -E_x[log(D(x))]] - E_z[log(1 - D(G(z)))]$$

dove x rappresenta un dato reale e z l'output del generatore. Mentre la funzione di loss del generatore é una semplificazione della precedente e mira a minimizzare la probabilità che il discriminatore classifichi i dati ricevuti dal generatore come fake: $L_G = -E_z[log(1 - D(G(z)))$.

L'addestramento delle due componenti procede a periodi alternati, prima si addestra il discriminatore per qualche epoca mantenendo bloccati i pesi del generatore e poi si svolge l'operazione inversa. E' necessario prestare attenzione poiché col tempo il feedback del discriminatore diventa meno significativo e potrebbe condizionare negativamente il generatore riducendo la qualità dell'output generato.

Esistono altri due problemi comuni nelle GANs il primo riguarda il vanishing gradient e si verifica quando il discriminatore diventa troppo accurato e azzera la funzione di loss. Il gradiente dunque non passa nella rete del generatore e la GAN non fa progressi. Il secondo detto mode collapse coinvolge il generatore e accade quando quest'ultimo cattura solo una piccola parte del dataset e non riesce a produrre un output vario. Le Wasserstein GAN (WGAN) [52] e Unrolled Generative GAN (UGAN) [53] vengono proposte per alleviare questi problemi offrendo un addestramento più stabile ed evitando che il discriminatore rimanga bloccato in minimi locali costringendo per cui il generatore ad esplorare nuove aree del dominio dei dati.

Nel corso degli anni sono state presentate diverse implementazioni di GAN, ci focalizzeremo sui modelli più rilevanti recentemente sviluppati per svolgere il task di image generation.

3.2.1 StyleGAN

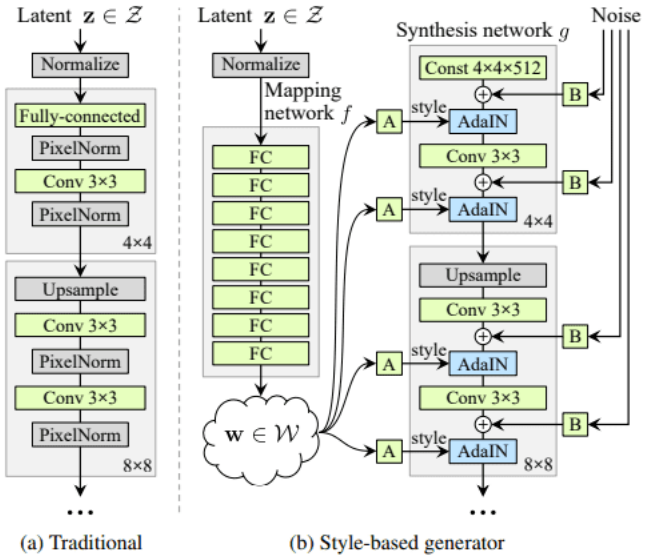

Figura 3.5: Viene mostrato il nuovo generatore introdotto da StyleGAN. L'input viene mappato in uno spazio latente W che controlla il generatore attraverso una normalizzazione chiamata AdaIN in ogni layer convoluzionale. Alla fine di ogni convoluzione viene aggiunto del rumore gaussiano.

L'architettura StyleGAN [54] introdotta da NVIDIA é progettata per generare immagini realistiche ad alta risoluzione manipolando e controllando gli stili e le features all'interno dell'immagine. Gli autori propongono una nuova architettura per il generatore rappresentata in Figura 3.5 che si compone di due reti. La prima é una "mapping network" che si occupa di mappare un vettore di input casuale in uno spazio latente W ad alta dimensionalità. In questa fase il modello impara le variazioni semantiche e stilistiche dell'immagine. La seconda rete é chiamata "synthesis network" ed é responsabile della generazione vera e propria dell'immagine. Riceve in input il vettore latente generato in precedenza e sfrutta la tecnica di AdaIN [55] per adattare lo stile alle features:

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

dove x_i sono le feature maps e vengono normalizzate in maniera indipendente, y é il vettore di stile estratto a partire dal vettore latente w, $\mu e \sigma$ sono la media e la varianza. Alla fine di ogni convoluzione viene aggiunto del rumore gaussiano per generare dettagli stocastici all'interno dell'immagine.

Recentemente sono state pubblicate diverse versioni di StyleGAN migliorando via via le prestazioni e l'architettura, fino ad arrivare a StyleGAN-T [56], una conditional GAN per il text-to-image generation che presenta risultati alla pari con diversi modelli a diffusione migliorando in molti casi il tempo di inferenza.

3.2.2 SAGAN

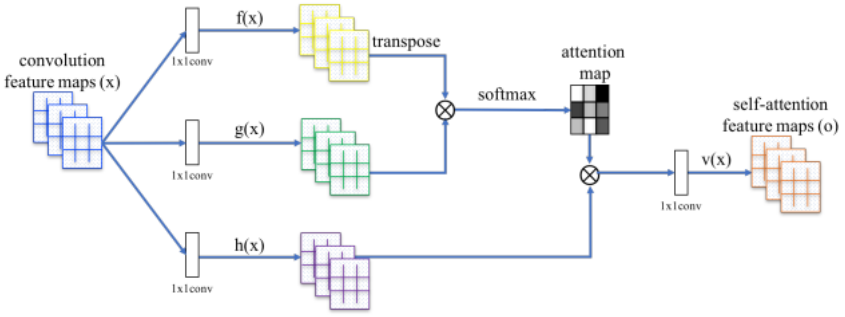

Figura 3.6: Rappresentazione del livello di self-attention per l'architettura SAGAN

Iglesias et al. [57] evidenziano nel loro studio che le architetture delle GAN incontrano difficoltà nel catturare informazioni correlate che sono spazialmente distanti all'interno delle immagini, a causa delle dimensioni ridotte del campo recettivo. Con l'avvento dell'architettura Transformer [58] viene proposta Self Attention GAN (SAGAN) [59] una rete neurale che sfrutta i layers di self-attention per permettere sia al generatore che al discriminatore di catturare dipendenze a lungo raggio nello spazio delle features. Il layer di self-attention viene posto subito dopo un layer convoluzionale (vedi Figura 3.6). Le features dell'immagine vengono prima di tutto trasformate in due spazi f, g per calcolare l'attenzione, dove $f(x) = W_f x$ e $g(x) = W_g x$,

$$\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} exp(s_{ij})}, s_{ij} = f(x_i)^T g(x_j).$$

Il parametro β_{ij} indica in che misura il modello presta attenzione alla posizione i-esima delle feature maps durante il processo di sintesi della j-esima regione dell'immagine. L'output del layer di self-attention é una matrice R^{CxN} dove N é il numero di feature e C il numero di canali e ogni elemento o_j della matrice viene calcolato tramite la seguente formula:

$$o_j = v(\sum_{i=1}^{N} \beta_{j,i} h(x_i)), h(x_i) = W_h x_i, v(x_i) = W_v x_i.$$

Ogni matrice W riportata nella formula é parametro addestrabile del modello implementato come convoluzione 1×1. Questo tipo di modello ottiene ottimi risultato nel task di class conditional image generation sul dataset ImageNet.

3.2.3 BigGAN

Sulle orme di SAGAN, BigGAN [60] sfrutta la stessa architettura per sintetizzare immagini ad alta dimensionalità partendo da dataset molto complessi come ImageNet. Gli autori mostrano come l'aumentare del numero di canali e del batch size delle immagini usate permetta di ottenere un punteggio di Inception Score (IS) nettamente superiore rispetto al modello base SAGAN. Questo perché si aumenta la diversità nel batch che permette al modello di coprire più stili, caratteristiche o categorie presenti nei dati di input. Viene introdotto inoltre il "truncation trick" che permette di ridurre il range di valori assunti nella distribuzione normale dal vettore latente $z \sim N(0, I)$. In questo modo la qualità delle immagini aumenta perdendo però di varietà.

3.3 Modelli a diffusione

I modelli a diffusione, una recente innovazione nel campo della visione artificiale, hanno raggiunto risultati allo stato dell'arte nel task di sintesi delle immagini scalzando le GANs. L'approccio utilizzato da questi modelli segue una prima fase di perturbazione delle immagini con del rumore in un "forward diffusion process". Consiste in steps multipli, come si può notare dalla Figura 3.7, dove progressivamente i dati vengono distrutti fino ad ottenere puro rumore gaussiano. La seconda fase viene chiamata "backward denoising process", in quest'ultima il modello iterativamente cerca di distruggere il rumore per ottenere l'immagine originale. Generalmente quest'ultimo processo viene svolto da una rete neurale, tipicamente un'architettura U-Net [5], per preservare le dimensioni spaziali dell'input.

Come spiegato da Rombach et al. [8], questi modelli basandosi sulla stima della massima verosimiglianza non presentano i problemi di mode-collapse e training instabile delle GANs e sfruttando la condivisione dei parametri possono modellare distribuzioni molto complesse di immagini naturali. Nel corso degli anni sono nate diverse formulazioni di modelli a diffusione, in particolare in questo lavoro tratteremo: Denoising Diffusion Probabilistic Models (DDPMs) [6] e Latent Diffusion Models (LDMs) [8] .

3.3.1 Denoising Diffusion Probabilistic Models

Quest'approccio é stato presentato nell'omonimo lavoro di Jonathan Ho et al. [6] e introduce i concetti generali adottati in ogni implementazione dei modelli a diffusione. Viene definito come forward process una catena di Markov che gradualmente aggiunge del rumore in accordo con una varianza β in t steps:

$$q(x_1 : T|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), q(x_t|x_{t-1}) := N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

dove $x_1, ..., x_t$ sono le variabili dello spazio latente di dimensione uguale all'input non corrotto x_0 e I é la matrice identità. Gli iperparametri β possono essere appresi sfruttando la tecnica di riparametrizzazione introdotta con i Variational AutoEncoder [61] oppure come proposto dagli autori fissati a costanti.

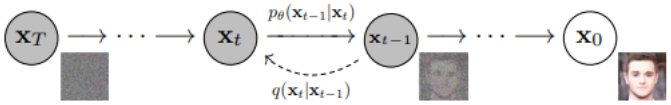

Figura 3.7: Rappresentazione grafica del processo di aggiunta e rimozione del rumore direttamente sui pixel dell'immagine

Il modello una forma $p0(x_0) := \int p_0(x_0 : T)dx_1 : T$ dove $p_0(x_0 : T)$ é il processo inverso che segue anch'esso una catena di Markov definita come:

$$p_\theta(x_0 : T) := p(xT) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), p_\theta(x_{t-1}|x_t) := N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

in particolare questa formula definisce la distribuzione di probabilità congiunta dei dati $x_0, ..., x_T$, $p(x$ dove $\Sigma_0(x_t, t))$ Potremmo pensare di addestrare una rete neurale per compiere l'operazione di denoising con una funzione obiettivo maximum likelihood in modo tale che la probabilità assegnata dal modello $p0(x0)$ ad ogni esempio visto nel training sia più grande possibile [62]. Tuttavia il calcolo della probabilità $p0(x0)$ richiederebbe di considerare tutte le sequenze di passaggi inverse che portano al dato di addestramento $x0$. Per affrontare questo problema, si ricorre all'uso di un variational lower bound, che fornisce un modo approssimato per valutare la probabilità tramite negative log-likelihood:

$$E[-logp_\theta(x_0)] \leq E_q \left[-log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] = E_q \left[-logp(x_T) - \sum_{t \geq 1} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] =: L$$

.

Di seguito riportiamo alcuni risultati ottenuti sul dataset CIFAR10 nel task unconditional image synthesis.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 3.75 (3.72)

Figura 3.8: Risultati sul datataset CIFAR10. NLL indica la negative log-likelihood. Per le reference dei metodi coinvolti fare riferimento a: [6]

Come si evince dalla Figura 3.8 l'approccio proposto ottiene risultati allo stato dell'arte sul task di unconditional image synthesis e genera immagini di qualità superiore rispetto a molti modelli condizionali. Come metrica di valutazione é stato adottato il Fréchet inception score (FID) e inception score, la differenza fra i due é che il primo compara la distribuzione delle immagini reali con la distribuzione delle immagini generate, mentre il secondo calcola il punteggio basandosi solo sulle immagini generate.

GLIDE

Una tecnica appartenente al mondo dei DDPM che si é distinta nel task di text-to-image con risultati allo stato dell'arte é GLIDE [7] . Gli autori di GLIDE mettono a confronto due tecniche di condizionamento del modello: classifier-free guidance [63] e CLIP guidance [64]. Prima di descrivere i due approcci é bene citare il lavoro di Dhariwal & Nichol [65] dove viene dimostrato che i modelli condizionali a diffusione possono ottenere grossi vantaggi in termini di generazione, affiancati ad un classificatore. In particolare si vede che perturbando la media $\mu_\theta(x_t|y)$ del modello in questo modo:

$$\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y) \nabla_{x_t} log p_\phi(y|x_t)$$

dove y é la classe predetta dal classificatore addestrato, s é lo "scale guidance" e $\Sigma_\theta(x_t|y)$ la varianza, con il gradiente della probabilità logaritmica $logp_\phi(y|x_t)$, si ottengono grandi vantaggi in fase di campionamento. Gli autori di GLIDE integrano quest'apporccio alle tecniche di guidance sopracitate.

Un modello CLIP é composto da due encoder $f(x)$ e $g(c)$. Il primo codifica le immagini, il secondo le caption testuali. I dati di addestramento sono rappresentati dalle coppie (x, c) e la loss corrisponde ad una "contrastive cross-entropy" che sfrutta il prodotto scalare tra $f(x)$ e $g(c)$, se la didascalia corrisponde all'immagine il valore sarà più alto. In sostanza CLIP fornisce uno score di similarità fra i due spazi vettoriali. GLIDE applica questa tecnica nella fase di denoising modificando la media della distribuzione condizionata da c, con i vettori di output dei due encoder addestrati:

$$\hat{\mu}_\theta(x_t|c) = \mu_\theta(x_t|c) + s \cdot \Sigma_\theta(x_t|c)\nabla_{x_t}(f(x_t) \cdot g(c)).$$

L'approccio classifier-free invece é leggermente più complesso, non prevede un classificatore esterno e viene implementato sostituendo in un modello condizionale $\epsilon_\theta(x_t|y)$ l'etichetta y con una variabile nulla durante la fase di addestramento. In fase di campionamento si condiziona l'output verso la caption c nel seguente modo:

$$\hat{\epsilon}_\theta(x_t|c) = \epsilon_\theta(x_t|\varnothing) + s \cdot (\epsilon_\theta(x_t|c) - \epsilon_\theta(x_t|\varnothing)).$$

Questo approccio offre diversi vantaggi, tra cui la capacità di utilizzare un singolo modello per guidare la generazione delle immagini e semplificare il processo quando si usano prompt complessi come il testo. Gli autori utilizzano il metodo classifier-free come fine-tuning per permettere al loro modello text-conditional di svolgere anche il task di image synthesis unconditional.

L'architettura base usata per il denoising é descritta nel lavoro di Dhariwal & Nichol [65]. Per condizionare sul testo viene usato un modello transformer. Di seguito riportiamo alcune immagini generate dallo stesso prompt testuale, evidenziando le differenze fra i due approcci di guidance descritti precedentemente.

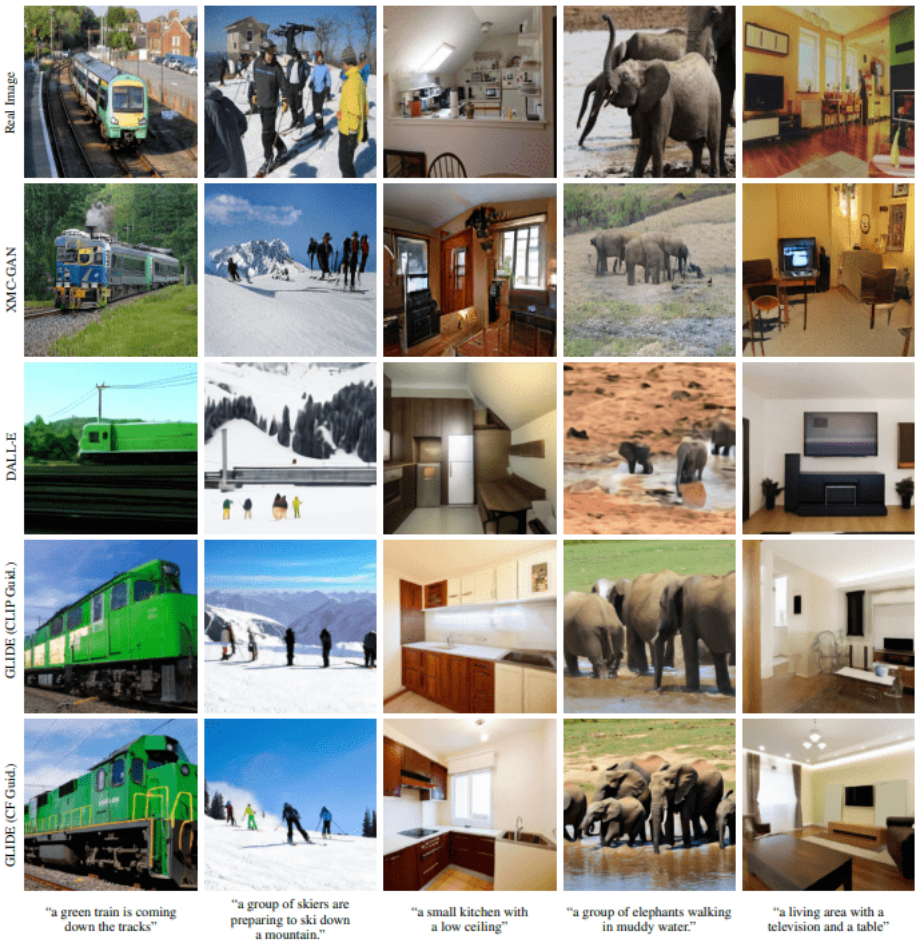

Figura 3.9: Immagini casuali prese dal dataset MS-COCO e rigenerate usando diversi modelli. Source: [7]

3.3.2 Latent Diffusion Models

Da un'attenta analisi svolta su un modello basato sulla stima della massima verosimiglianza si nota che l'addestramento si sviluppa in due fasi: la prima viene detta "perceptual compression" dove vengono rimossi i dettagli ad alta frequenza dall'immagine, come piccole variazioni di intensità luminosa o colore dei pixel che richiedono una capacità di elaborazione elevata per essere modellati in modo accurato. La seconda é detta "semantic compression" dove il modello impara la composizione semantica dei dati. Gli autori dei modelli a diffusione latente [8] si pongono come obiettivo quello di sostituire la perceptual compression con un approccio computazionalmente più rapido.

Come si può notare in Figura 3.10, viene proposto l'addestramento di un encoder ϵ che permetta il mapping dell'immagine di input in uno spazio latente di bassa dimensionalità z. Viene ridotta la complessità, si catturano le features importanti in una forma più compatta e tramite l'operazione di upsample si può ottenere una rappresentazione ad alta risoluzione dell'immagine. L'architettura riprende i principi base dei modelli a diffusione, con la differenza che il rumore viene applicato al vettore latente e non direttamente ai pixels dell'immagine.

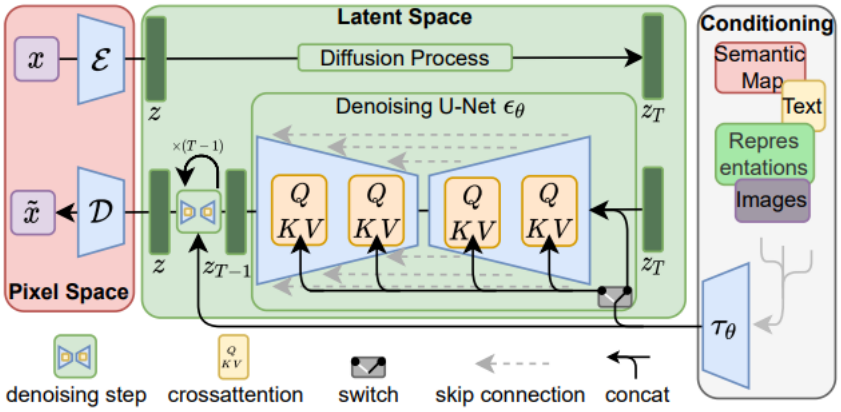

Figura 3.10: Viene rappresentata l'architettura LDM, partendo dal encoding nello spazio latente, al processo di diffusione del rumore, a quello inverso, attraversando layer di cross-attention, per poi generare l'immagine originale nello spazio dei pixels. Source: [8]

Per la fase di denoising viene utilizzata la rete UNet con l'aggiunta di layers di cross-attention [58] fruibile dai diversi tipi di input condizionali che si vogliono sottomettere alla rete. La fase di condizionamento, ovvero quel passaggio che serve a indirizzare il generatore verso l'output desiderato, viene svolta da un encoder T_θ che processa l'input y in una rappresentazione $T_\theta(y)$ concatenata al vettore latente zT . La fase di mapping via cross-attention durante il denoising viene svolta nel seguente modo:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$$

dove

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t), K = W_k^{(i)} \cdot T_\theta(y), V = W_v^{(i)} \cdot T_\theta(y).$$

Nella formula abbiamo le tre matrici $W_Q, W_K e W_V$ che sono parametri della UNet e $\varphi_i(z_t)$ denota una rappresentazione intermedia della rete implementando

ϵ_θ . La funzione di loss L_{LDM} basata su image-conditioning viene definita come:

$$L_{LDM} := E\left[||\epsilon - \epsilon_\theta(z_t, t, T_\theta(y))||_2^2\right] .$$

CelebA-HQ 256 × 256				FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DC-VAE	15.8	-	-	ImageBART	9.57	-	-
VQGAN+T.	10.2	-	-	U-Net GAN (+aug)	10.9 (7.6)	-	-
PGGAN	8.0	-	-	UDM	5.54	-	-
LSGM	7.22	-	-	StyleGAN	4.16	0.71	0.46
UDM	7.16	-	-	ProjectedGAN	**3.08**	0.65	0.46
LDM-4 (ours, 500-s†)	**5.11**	0.72	0.49	*LDM-4* (ours, 200-s)	4.98	**0.73**	**0.50**

LSUN-Churches 256 × 256				LSUN-Bedrooms 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DDPM	7.89	-	-	ImageBART	5.51	-	-
ImageBART	7.32	-	-	DDPM	4.9	-	-
PGGAN	6.42	-	-	UDM	4.57	-	-
StyleGAN	4.21	-	-	StyleGAN	2.35	0.59	0.48
StyleGAN2	3.86	-	-	ADM	1.90	**0.66**	**0.51**
ProjectedGAN	**1.59**	0.61	0.44	ProjectedGAN	**1.52**	0.61	0.34
*LDM-8** (ours, 200-s)	4.02	**0.64**	**0.52**	*LDM-4* (ours, 200-s)	2.95	**0.66**	0.48

Figura 3.11: Risultati ottenuti nel task di unconditional image synthesis. Addestramento effettuato in uno spazio latente regolarizzato KL. Per le reference dei metodi coinvolti fare riferimento a: [8]

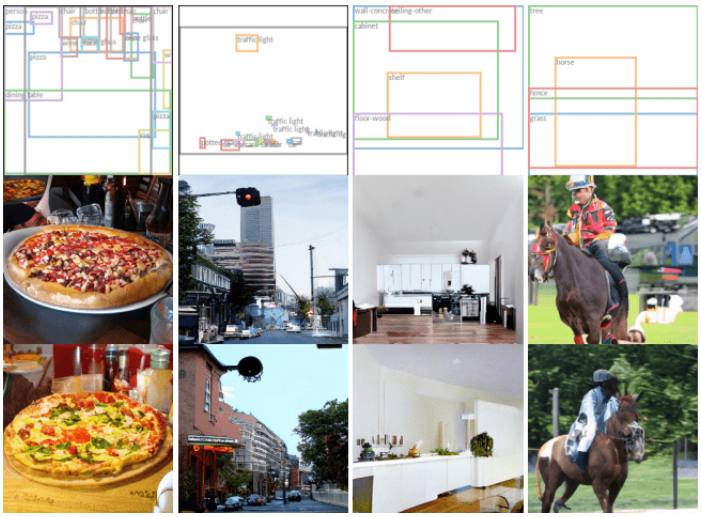

Figura 3.12: Risultati ottenuti nel task layout-to-image addestrato sul dataset OpenImages e valutato su ImageNet. Source: [8]

In Figura 3.11 riportiamo alcuni risultati ottenuti dal modello proposto nel task di unconditional image synthesis evidenziando soprattutto la differenza di prestazione rispetto ai DDPM in termini di FID. Sono stati effettuati dei test nel task di text-to-image synthesis addestrando un modello con un encoder KL-regularized, mentre l'encoder T_θ che si occupa del condizionamento tramite testo, é stato implementato come transformer. Questa combinazione ha mostrato una forte generalizzazione su prompt testuali molto complessi inseriti dall'utente. I test sono stati effettuati su MS-COCO ottenendo risultati alla pari con gli altri modelli a diffusione riducendo sostanzialmente il numero di parametri.

Per valutare le prestazioni e la flessibilità dei livelli di cross-attention vengono svolti dei test di layout-to-image generation su ImageNet. Per sintetizzare l'immagine non viene fornito del testo ma uno schema che comprende le etichette corrispondenti alle classi da generare (vedi Figura 3.12).

Vengono addestrati due modelli di latent diffusion: uno sul dataset COCO e un altro su OpenImages. Il primo ha ottenuto risultati allo stato dell'arte per il task di layout-to-image generation. Questo lavoro ha portato allo sviluppo di "Stable Diffusion" uno dei migliori modelli a diffusione latente per il task di general text-to-image.

Capitolo 4

Metodo proposto

In questo capitolo andremo a descrivere il problema trattato nella tesi, gli steps che hanno portato alla soluzione e i dettagli tecnici sulle metodologie implementate per risolvere il problema.

4.1 Descrizione del problema

Si vuole sviluppare un sistema per analizzare la corsa di un atleta su tapis roulant estraendo da immagini e video i parametri posturali che consentono di valutare se la persona stia eseguendo correttamente l'attività fisica. L'ipotesi iniziale per risolvere questo problema é stata quella di applicare tecniche di Human Pose Estimation per stimare la posizione dei giunti nella parte inferiore del corpo.

4.1.1 Valutazione delle tecniche di HPE

La prima fase del lavoro quindi é stata quella di raccogliere le migliori tecniche di Human Pose Estimation presenti letteratura ed eseguire dei test preliminari su qualche video eseguito riprendendo da dietro la persona. L'azienda ha posto un vincolo di distanza della telecamera dal tapis roulant, il che permette solo la ripresa della della parte inferiore del corpo. Gli stimatori di posa valutati sono stati tre: OpenPose [2], HRNet [3] e Mediapipe [66]. I primi due rappresentano lo stato dell'arte, in particolare HRNet, che come evidenziato nel Capitolo 2 raggiunge risultati anche superiori a OpenPose. Mediapipe invece é un approccio presentato da Google che mira ad ottenere buone prestazioni in real-time su dispositivi mobili e laptop e permette l'utilizzo su piattaforme web. Tutti e tre gli estimatori sono stati testati sui video raccolti preliminarmente e mappati sulle immagini in input i keypoints predetti per una valutazione visiva dei risultati. Questi test hanno mostrato

un problema comune, ovvero valutando solo la parte inferiore del corpo e non tutta la figura molti keypoints venivano predetti lontano dalla posizione dei loro giunti peggiorando l'accuratezza del modello.

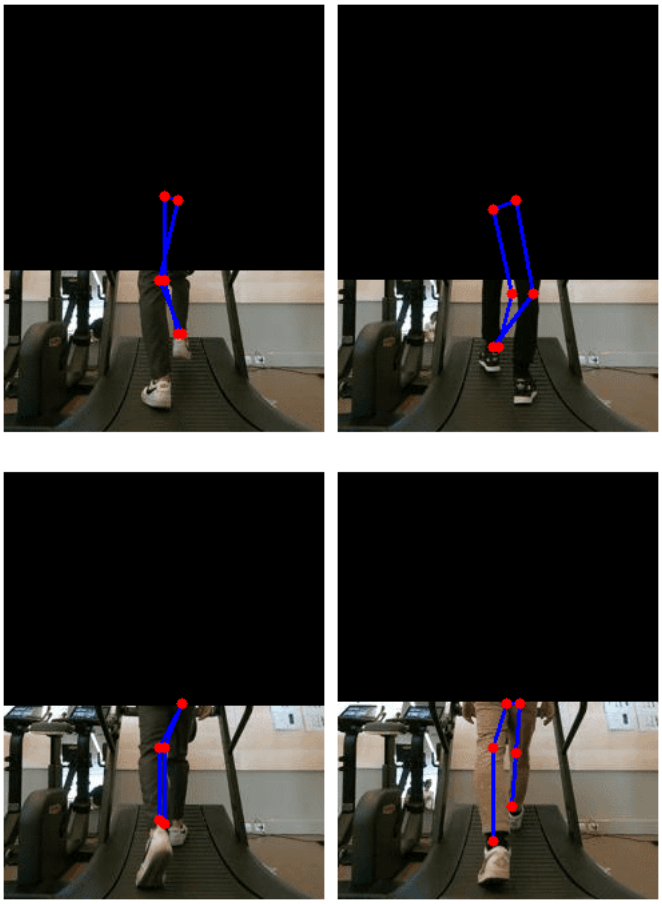

Figura 4.1: Esempi di valutazioni errate dei keypoints da parte di OpenPose (a sinistra) e HRNet (a destra)

La Figura 4.1 mostra alcuni casi in cui i modelli predicono in posizioni errate i keypoints di caviglie ginocchia e anche, ricevendo in input immagini in cui sono visibili solo le gambe.

HRNet é stato scelto come tecnica di HPE di riferimento per gli esperimenti svolti durante le fasi successive del lavoro, in quanto OpenPose é un modello relativamente pesante e dai test effettuati impiega il doppio di tempo per calcolare i keypoints sull'intera figura umana rispetto ad HRNet. Inoltre come mostrato nel Capitolo 2, in particolare in Figura 2.7, le prestazioni di OpenPose

sul test-dev di COCO sono decisamente peggiori, perciò si é deciso di optare per HRNet.

4.2 Soluzione proposta

Nel corso degli ultimi anni il campo della visione artificiale ha visto nascere modelli generativi di immagine molto accurati e che rendono praticamente impossibile ad occhio umano riconoscere la differenza con quelle reali. Dati i vincoli imposti dal problema trattato, si é deciso di sfruttare alcuni di questi modelli per cercare di ricostruire la parte superiore del corpo con l'ipotesi che, rieseguendo HRNet sugli output generati, migliori l'accuratezza dei keypoints predetti.

Come accennato in precedenza esistono diversi task di generazione, nel nostro caso si tratta di ricostruire un'immagine partendo da gambe e bacino, si é quindi scelto un approccio basato su inpainting e uncropping. La tecnica di inpainting consiste nel ricostruire una porzione interna di immagine cancellata, avendo a disposizione il contesto circostante. Uncropping o outpainting é una tecnica che consiste nel generare una regione d'immagine esterna e affine a quella di partenza. Prima di passare alla descrizione tecnica dei modelli usati per questo scopo, forniamo alcuni dettagli sulla tecnica di HPE adottata.

4.3 SimpleHRNet

L'implementazione dell'architettura di HRNet é stata presa dal seguente repo Github `https://github.com/stefanopini/simple-HRNet`. L'autore ci fornisce una versione non ufficiale della tecnica di HPE in Python, ottenendo risultati alla pari con quella ufficiale e compatibile al 100% con i pesi dei modelli pre-addestrati. La classe SimpleHRNet facilita di molto l'uso di questa tecnica, in poche linee di codice é possibile ottenere i 17 keypoints calcolati sull'immagine. Il modello utilizzato é HRNet a 48 canali e con risoluzione 384×288, molto simile a quella delle immagini utilizzate in questo lavoro di tesi. Essendo un approccio multi-pose il risultato é un array che contiene le stime di tutte le persone nell'immagine. Il nostro task ne prevede solo una riducendosi ad una single-person pose estimation. La libreria mette a disposizione YOLOv3/v5 [67] come person detector per un task di multi-person HPE.

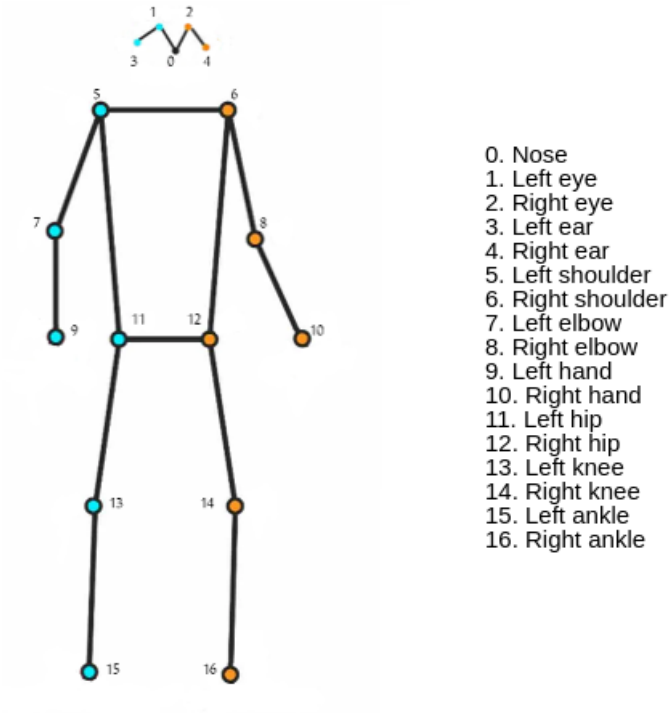

Figura 4.2: Rappresentazione dei 17 keypoints stimati da HRNet in un modello skeleton

In Figura 4.2 possiamo osservare il 17 keypoints rilevati da HRNet rappresentati da un modello skeleton.

4.4 Palette

Il primo approccio risolutivo adottato é un Denoising Diffusion Probabilistic Model chiamato Palette e specializzato per i task di inpainting. E' modello che si basa sulla generazione condizionata image-to-image. E' stato addestrato un modello da zero sul nostro dataset raccolto. L'implementazione é stata presa da `https://github.com/Janspiry/Palette-Image-to-Image-Diffusion-Models` che fornisce un framework per la pipeline dei modelli a diffusione e permette di svolgere training visualizzando i progressi, testing e mettendo a disposizione una serie di metriche per la valutazione dei risultati.

4.4.1 Architettura adottata

Palette propone una rete UNet (Figura 4.3) rimodulata per la fase di denoising. Segue l'approccio introdotto nei modelli a diffusione [6] riadattato

grazie a Dhariwal & Nichol [65]. La rete UNet, usata per il processo di denoising, usa uno stack di residual layers e convoluzioni che svolgono il downsampling connessi con skip connections a stack di convoluzioni di upsampling (come descritto nel Capitolo 3). L'input della rete é una concatenazione delle features di due immagini per un totale di 6 canali. La prima y_{cond} é un'ulteriore condizionamento, la seconda y_0 é l'input con il rumore aggiunto. La rete prende in input anche un vettore di embeddings sinusoidali che guiderà la rimozione del rumore. Ad una certa risoluzione nel nostro caso 16×16, dopo il blocco residuale della UNet, viene calcolata l' Attention sull'output del precedente layer. Come funzione di attivazione non viene usata la ReLU ma la SiLU. Definita come $SiLU(x) = x \cdot sigmoid(x)$, è una funzione non monotona, che può consentire una maggiore flessibilità nell'approssimare funzioni complesse rispetto a ReLU. Viene utilizzata la Group Normalization [68] che divide i canali in gruppi calcolandone per ognuno la media e la varianza per la normalizzazione. In casi come questo in cui la dimensione del batch é molto piccola si preferisce questo tipo di approccio rispetto alla batch normalization classica che genererebbe stime inaccurate su campioni così ristretti.

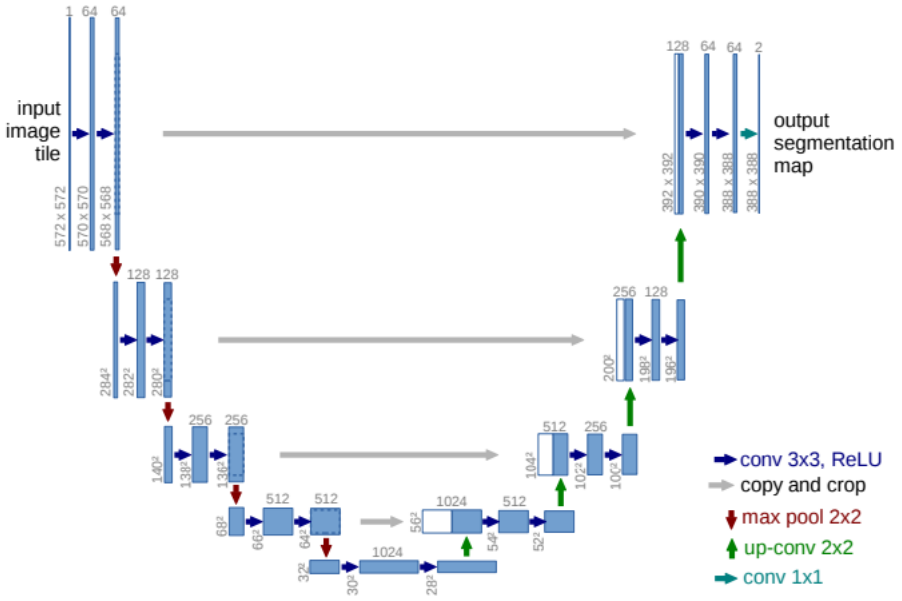

Figura 4.3: L'architettura di base utilizzata nei modelli a diffusione. Source [5]

4.4.2 Approccio utilizzato

Per ricostruire la parte superiore del corpo vengono svolti due task di ricostruzione dell'immagine diversi: uncropping e inpainting. Uncropping o

outpainting é una tecnica che consiste nel generare una regione d'immagine esterna e affine a quella di partenza. La tecnica di inpainting consiste nel ricostruire una porzione interna di immagine cancellata, avendo a disposizione il contesto circostante. Si tratta di due operazioni che tentano di rigenerare un'immagine a cui é stata applicata una maschera di rumore gaussiano. La prima é sicuramente più complicata in quanto a diversi tagli del corpo si cerca di ricostruire l'intera immagine superiore compreso lo sfondo. La seconda, sicuramente più semplice, basandosi sul contesto circostante si cerca di generare solo le parti del corpo coperte dal rumore.

4.4.3 Data preprocessing

Come si può notare dalla Figura 5.1 i dati sono immagini RGB 640×480 scalati a 320×240 per agevolare l'addestramento e mantenere le proporzioni in un formato più piccolo, e raffigurano una persona di spalle che corre su un tapis roulant. Da questi dati sono state estratte delle maschere rettangolari che rappresentano i punti in cui verrà aggiunto il rumore gaussiano. Questi rettangoli coprono interamente la figura umana e in fase di training o test tramite i keypoints raccolti si riesce a rimuovere la maschera dove non necessaria. Per il task di uncropping la maschera viene generata nel dataloader durante l'addestramento.

La pipeline inizia sottoponendo al framework il dataset di training, per ogni immagine viene ricavata una matrice di uguali dimensioni dove con 0 é rappresentata la porzione da non generare (rimane invariata), e con 1 la parte a cui sarà aggiunto il rumore. Questa matrice viene poi moltiplicata per l'immagine in input in questo modo:

$$x_{cond} = x \cdot (1 - mask) + mask \cdot random_noise(x)$$

dove x é l'immagine di partenza, $random_noise(x)$ é una funzione che applica del rumore gaussiano casuale.

Figura 4.4: Esempi di immagini prese dal dataset.

4.4.4 Training

L'addestramento é composto dalle due fasi descritte precedentemente per i DDPMs. La prima consiste nel perturbare l'immagine originale seguendo una distribuzione normale a parametri fissati e non addestrabili, per il numero di timesteps definiti a priore. Il risultato di questo processo é l'input della rete UNet che insieme all'immagine condizionale x_{cond} ricevuta in precedenza entra nel processo di denoising. Utilizzando delle maschere per i nostri task, il modello deve risolvere solo la porzione dell'immagine che abbiamo indicato con

la maschera:

$$\tilde{y} = (\gamma y_0 + 1 - \gamma\varepsilon) \cdot mask + (1 - mask) * y_0$$

dove y_0 é l'immagine di train, ϵ il rumore aggiunto all'immagine scalato da γ. La funzion di loss é indicata come:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{(x,y)}\mathbb{E}_{\varepsilon,\gamma}\left[f_\theta(x_{cond}, \tilde{y}, \gamma) - \varepsilon\right]_2^2$$

dove f_θ rappresenta il modello UNet che ha come obiettivo quello di avvicinarsi al rumore ε per rimuoverlo e ottenere un risultato tanto più vicino all'immagine di partenza. Gli autori di Palette fanno delle considerazioni rispetto a una loss L_1 o L_2.

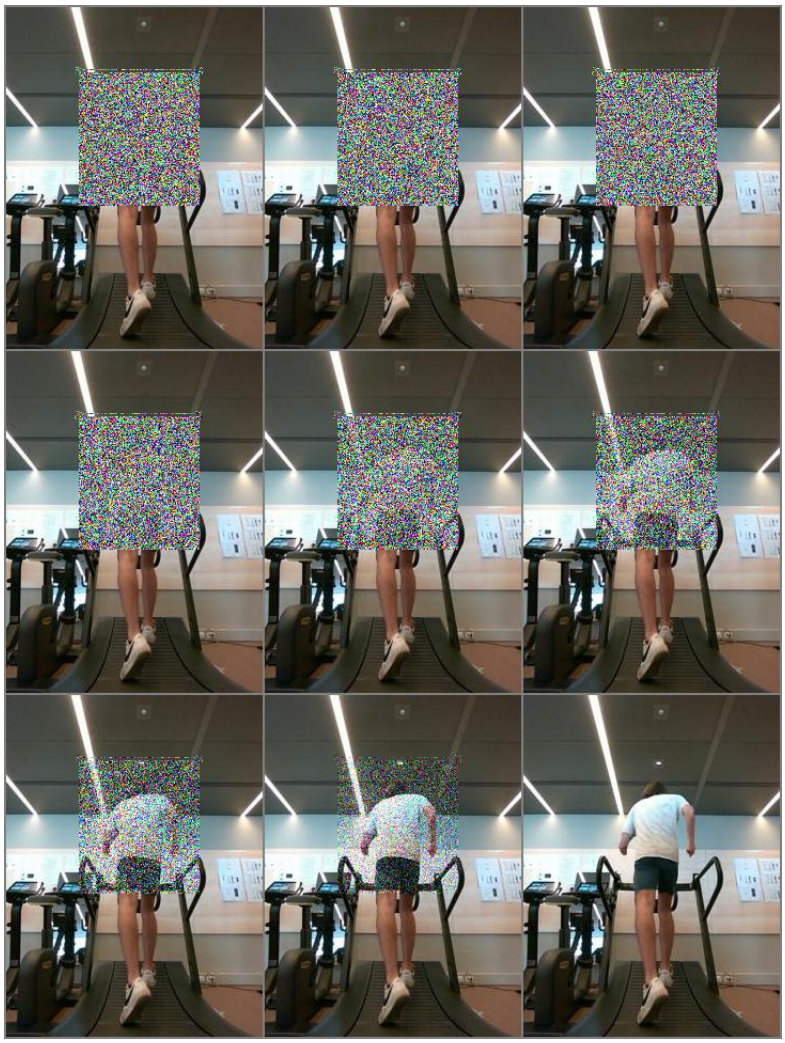

Figura 4.5: Immagine che mostra in pochi passaggi il processo di denoising del modello durante la fase di sampling.

Questa scelta impatta notevolmente sulla diversità della distribuzione prodotta. In particolare si nota che con L_1 si ha una generazione di campioni poco diversificati ma si riducono le allucinazioni, mentre con L_2 si aumenta la varietà [69]. In Figura 4.5 viene mostrato tutto il processo di denoising di Palette. E' un processo su più steps che segue una catena di Markov. Per ogni step, il modello cerca di cancellare il rumore rimanente generando le parti mancanti fornendo un output sempre più simile all'immagine originale.

4.5 Stable Diffusion Infinity

In questo lavoro di testi é stato preso in considerazione un secondo metodo: Stable Diffusion Infinity. L'implementazione fa riferimento al seguente repo Github: `https://github.com/lkwq007/stablediffusion-infinity`. Questa tecnica sfrutta un modello Stable Diffusion generale text-to-image per il task di inpainting, ideale per il problema trattato in questo lavoro. A causa della mancanza di dati, tempistiche brevi e della complessità computazionale non sono state svolte ulteriori operazioni di fine-tuning. Il modello é stato perciò utilizzato solo per fare inferenza sul test-set.

Capitolo 5

Esperimenti

In questo capitolo verranno presentati e spiegati nel dettaglio tutti gli esperimenti svolti durante il lavoro di tesi.

5.1 Dataset

5.1.1 Raccolta dati

Il dataset utilizzato per svolgere gli esperimenti é stato raccolto durante lo svolgimento di questo lavoro di tesi. Questo dataset è composto da un insieme di immagini che raffigurano singole persone in corsa o camminata su un tapis roulant, riprese da dietro. Lo strumento utilizzato per catturare le sequenze video é la Intel Realsense, fotocamera avanzata progettata per fornire una vasta gamma di funzionalità di visione artificiale. Dotata di tecnologie come la profondità 3D e il rilevamento dei movimenti, consente di catturare non solo immagini ad alta risoluzione, ma anche informazioni sulla profondità degli oggetti nell'ambiente circostante.

Sono state effettuate sostanzialmente tre tipi di riprese due da dietro, posizionando la fotocamera a distanze diverse, e una da davanti. Durante gli esperimenti sono state usate solo immagini con vista posteriore, in quanto il sistema dovrà funzionare dal retro del tapis roulant. L'obiettivo era quello di raccogliere un insieme di dati più simile possibile alle sequenze preliminari registrate prima dell'inizio del progetto, inquadrando anche la parte superiore del corpo.

Le immagini sono state salvate in un formato *raw* in modo da mantenere al 100% la qualità e catturare informazioni sulla profondità utili a calcolare parametri in uno spazio tridimensionale.

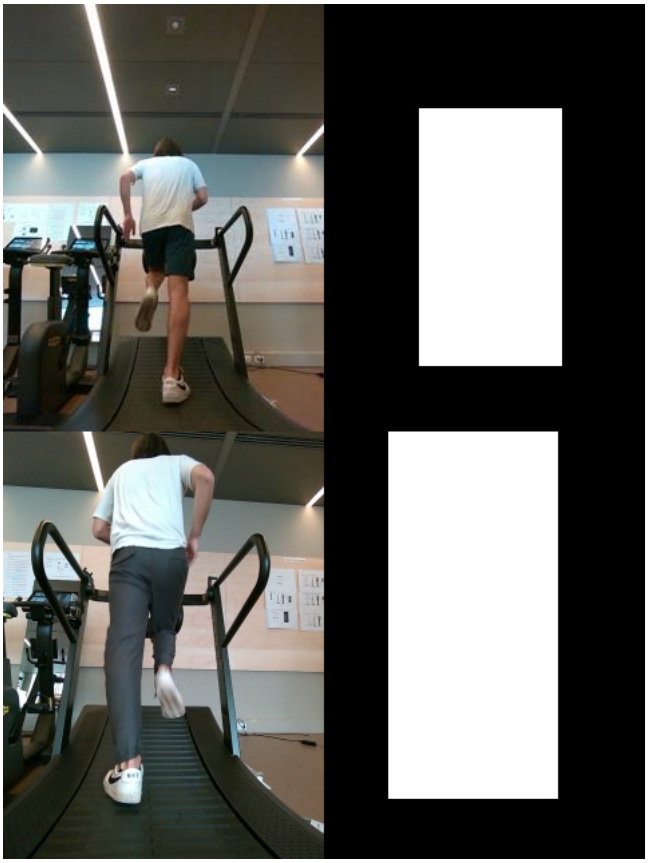

Figura 5.1: Esempi di immagini da vicino e lontano e relative maschere

5.1.2 Suddivisione

Le immagini estratte dal formato *raw* sono state suddivise per distanza dalla fotocamera: vicino e lontano. Il dataset principale contiene solo le immagini riprese da lontano e ne racchiude circa 10000 suddivise in 7500 immagini per il train e 2500 per il test. In un secondo momento ne sono state aggiunte altre, prese da una visuale più vicina. Eseguendo HRNet e stimando i punti, sono state selezionate solo le immagini con una confidenza media maggiore a 0.5 rispetto ai keypoints di testa e spalle (in molti frame queste due parti venivano tagliate fuori e avrebbero condizionato negativamente Palette). Il nuovo set di dati comprende un totale di 14000 immagini. Per ogni immagine é stata calcolata una maschera rettangolare dai giunti stimati per svolgere successivamente il task di inpainting (vedi Figura 5.1).

5.2 Valutazioni e risultati

I keypoints rilevati salvati e utilizzati come punto di riferimento per i successivi esperimenti. Come metrica di comparazione si é scelto di usare la Percentage of Correct Keypoints (PCK) che consiste nel calcolare la percentuale dei keypoints che risiedono ad una certa distanza da quelli di riferimento:

$$PCK_i = \frac{1}{N} \sum_{n=0}^{N} dist(kpred_i, ktrue_i) \leq 0.5$$

dove *dist* é la distanza euclidea, N il numero di immagini nel dataset e PCK_i la percentuale corretta del i-esimo keypoint su tutte le immagini raccolte. Infine si é calcolato un punteggio aggregato che é la media delle percentuali sia sulla parte inferiore che superiore del corpo.

5.2.1 HRNet a diversi tagli

			Lower body				
	Left hip	**Right hip**	**Left knee**	**Right knee**	**Left ankle**	**Right ankle**	**LB**
Neck cut	69.92	74.09	**88.27**	**88.27**	83.74	81.86	**81.02**
Shoulder cut	58.56	64.88	**84.54**	84.34	76.86	75.81	**74.16**
Chest cut	51.23	53.23	**80.26**	79.26	75.94	72.81	**68.79**
Hips cut	0.6	0	41.45	33.07	**41.53**	39.45	**26.02**
Knee cut	0.16	0.3	17.42	22.58	37.21	**37.85**	**19.25**
			Upper body				
	Head	**Left shoulder**	**Right shoulder**	**Left elbow**	**Right elbow**	**Wrists**	**UB**
Neck cut	6.3	61.19	49.49	54.18	56.31	**63.92**	**48.57**
Shoulder cut	2.73	30.87	30.43	53.34	45.77	**62.41**	**37.59**
Chest cut	0.53	0.4	0	46.42	41.49	**60.25**	**24.85**
Hips cut	0	0	0	0	0	**5.52**	**0.92**
Knee cut	0	0	0	0	0	0	**0**

Tabella 5.1: Valutazione della correttezza dei keypoints su un'immagine tagliata all'altezza di keypoints specifici. LB = Lower Body, UB = Upper Body.

Dinanzi all'ipotesi delineata nel capitolo precedente, secondo la quale l'accuratezza di un modello di stima della posa umana nella parte inferiore risulta

peggiorare omettendo la parte superiore del corpo dall'immagine, é stata formulata la Tabella 5.1. Questi risultati descrivono un processo di stima dei keypoints su tutte le immagini raccolte che si può articolare in due fasi. La prima consiste nell'identificare cinque tipologie di tagli, rappresentate da coordinate y (altezza) utilizzate per definire una linea orizzontale da sinistra a destra. Tale linea maschera tutti i pixel situati al di sopra di essa:

- neck cut: HRNet non consente di stimare direttamente le coordinate del collo, perciò é stata presa la posizione del naso e aumentata di qualche pixel;

- shoulder cut: viene selezionata la coordinata y maggiore tra la spalla destra e sinistra per avere un taglio più basso possibile;

- chest cut: viene calcolato ottenendo la media delle coordinate $y_1 = max(y_{rs}, y_{ls}), y_2 = min(y_{rh}, y_{lh})$ dove y_{rs}, y_{ls} sono le altezze delle spalle e y_{rh}, y_{lh} quelle delle anche, spostata di qualche pixel in alto;

- hips cut: si utilizza il metodo precedente utilizzando i keypoints di anche e ginocchia;

- knee cut: viene presa la coordinata minore fra le ginocchia e diminuita di qualche pixel

La seconda fase consiste nel eseguire una stima con HRNet per ogni immagine a cui é stata applicata la maschera. I nuovi keypoints predetti sono stati comparati con quelli di riferimento tramite la metrica PCK descritta in precedenza. La nostra ipotesi iniziale viene confermata dalla notevole degenerazione dell'accuratezza del modello sulla parte bassa del corpo, all'aumentare del livello di taglio diminuisce la correttezza dei keypoints. Notiamo in particolare che il modello non riesce a stimare correttamente la posizione delle parti del corpo oscurate nell'immagine. Questo fenomeno si nota molto bene dalle stime dei keypoints superiori a diversi tagli. Vedendo solo le gambe l'accuratezza é praticamente nulla. Ciò si può ricondurre al fatto che questi modelli necessitano di vedere le parti del corpo connesse al giunto da predire, in modo da comprendere la posizione e direzione dell'arto (vedi Figura 5.2).

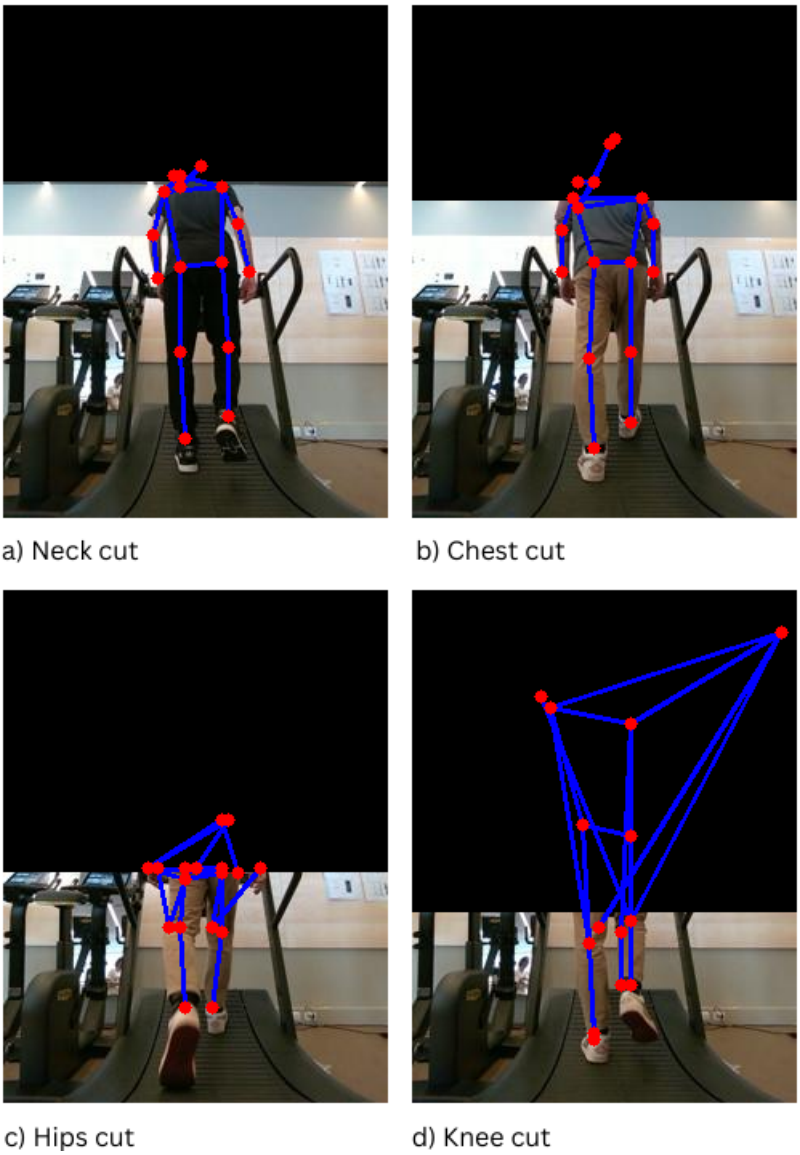

Figura 5.2: Esempi di tagli all'immagine svolti durante gli esperimenti con conseguente stima dei keypoints

5.2.2 Palette

In questa sezione descriveremo i due task di generazione svolti con il modello Palette e commenteremo i risultati ottenuti.

Uncropping

			Lower body				
	Left hip	**Right hip**	**Left knee**	**Right knee**	**Left ankle**	**Right ankle**	**LB**
Neck cut	67.16	67.24	71.96	**74.33**	64.99	67.08	**68.79**
Shoulder cut	65.92	65.99	72.37	**74.09**	64.56	66.08	**68.17**
Chest cut	64.68	65.12	72.73	**73.29**	64.6	66.52	**67.82**
Hips cut	44.81	43.17	66.96	**67.88**	61.71	63.92	**58.07**
Knee cut	30.03	32.59	62.43	**64.31**	60.27	62.67	**52.05**
			Upper body				
	Head	**Left shoulder**	**Right shoulder**	**Left elbow**	**Right elbow**	**Wrists**	**UB**
Neck cut	24.15	65.12	63.71	71.01	**73.77**	72.03	**63.12**
Shoulder cut	22.98	58.30	58.18	69.88	**72.04**	71.84	**60.73**
Chest cut	21.95	45.73	52.78	62.07	67.04	**70.41**	**55.77**
Hips cut	13.45	18.30	14.13	25.51	22.11	**36.45**	**23.78**
Knee cut	11.61	16.41	14.46	**18.67**	15.49	17.88	**16.59**

Tabella 5.2: Risultati della valutazione dei keypoints su un task di uncropping per ricostruire la parte superiore del corpo.

Il task di uncropping o outpainting consiste nel generare una grande porzione di immagine mascherata con del rumore gaussiano. A differenza dell'inpainting, questo tipo di task comporta una sfida aggiuntiva: bisogna considerare nella generazione non solo il corpo della persona, ma anche lo sfondo dell'immagine. Nel nostro caso, tuttavia, il background rimane costante in tutto il dataset. Pertanto, abbiamo inizialmente ipotizzato che questo fattore non influenzasse in modo significativo il modello.

Sono stati sfruttati i tagli dell'immagine descritti in precedenza per addestrare 5 modelli diversi a ricostruire la posa di una persona e in Tabella 5.2 si possono consultare i risultati delle generazioni. Il nostro obiettivo era quello di migliorare l'accuratezza di HRNet sulla parte inferiore del corpo generando la parte superiore, in quanto abbiamo visto che il modello é inaccurato se non ha a disposizione tutta la figura umana. Da questi risultati possiamo sicuramente affermare che le tecniche allo stato dell'arte per la Human Pose Estimation funzionano su immagini sintetizzate artificialmente e confrontandoli con i tagli della Tabella 5.1 notiamo un notevole aumento di precisione della

PCK all'altezza delle anche e ginocchia, predicendo correttamente in media il 30% in più dei keypoints nella parte bassa del corpo. In molti casi Palette é riuscito a generare una posa simile all'immagine originale, questo é dimostrato dalla seconda parte della Tabella 5.2. La percentuale di keypoints corretti nella parte alta del corpo é significativamente aumentata per ogni tipo di taglio.

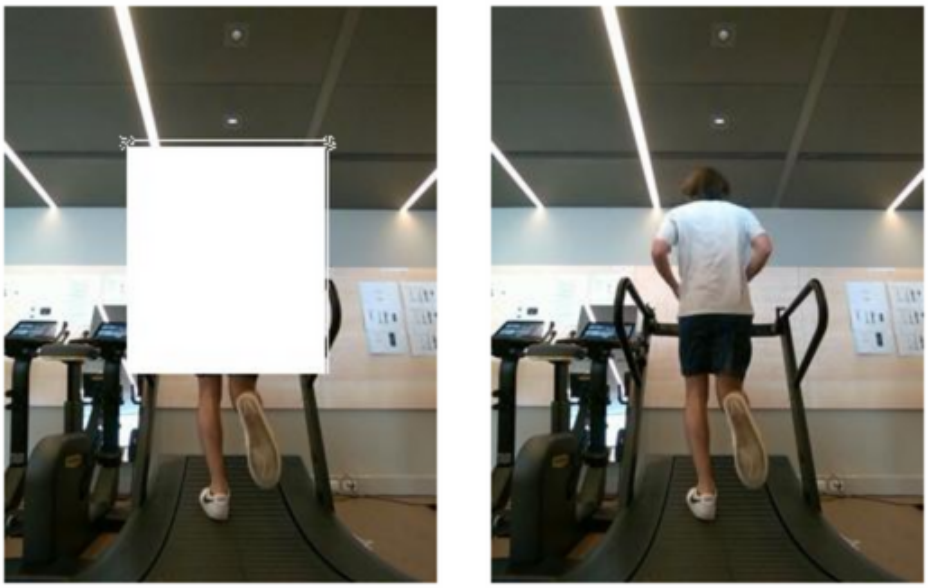

Figura 5.3: a) viene applicata una maschera dal ginocchio fino alla testa, b) Palette riesce a generare la posa partendo dal rumore applicato alla maschera

Inpainting

			Lower body				
	Left hip	**Right hip**	**Left knee**	**Right knee**	**Left ankle**	**Right ankle**	**LB**
Neck cut	69.12	67.53	75.36	**76.72**	68.91	69.56	**71.2**
Shoulder cut	67.21	65.43	73.27	**75.49**	67.05	68.12	**69.43**
Chest cut	65.19	64.75	72.21	**74.21**	65.39	66.27	**68**
Hips cut	36.32	39.57	69.36	**71.13**	65.16	65.48	**57.83**
Knee cut	27	31.23	59.83	62.47	61.43	**64.83**	**51.14**
			Upper body				
	Head	**Left shoulder**	**Right shoulder**	**Left elbow**	**Right elbow**	**Wrists**	**UB**
Neck cut	38.51	69.63	70.41	73.47	75.32	**78.91**	**67.61**
Shoulder cut	31.15	59.51	61.31	71.43	72.95	**74.57**	**61.82**
Chest cut	28.95	48.29	49.42	64.48	66.28	**70.69**	**56.98**
Hips cut	22.95	27.36	25.71	29.28	26.24	**38.65**	**29.83**
Knee cut	21.07	**23.03**	22.55	22.71	21.31	20.61	**21.71**

Tabella 5.3: Risultati della valutazione dei keypoints su un task di inpainting per ricostruire la parte superiore del corpo

Per svolgere il task di inpainting è stato necessario utilizzare delle maschere che riuscissero a oscurare solamente il corpo della persona nella foto (vedi Figura 5.3). Questo secondo approccio é stato pensato per velocizzare l'addestramento del modello e permettere una generalizzazione migliore su immagini mai viste. I risultati ottenuti sul test set sono molto simili a quelli del modello uncropping ottenendo una precisione maggiore nella generazione della parte alta. Questo si può ricondurre al fatto che Palette non deve generare lo sfondo e si può concentrare solo sul corpo.

5.2.3 Stable diffusion infinity

Ottenuti i primi risultati con Palette sono stati svolti ulteriori esperimenti con Stable Diffusion (SD-infinity).

	Lower body						
	Left hip	**Right hip**	**Left knee**	**Right knee**	**Left ankle**	**Right ankle**	**LB**
Neck cut	70.67	71.93	81.23	**80.55**	71.89	73.92	**75.03**
Shoulder cut	70.19	71.67	**81.77**	80.04	69.21	70.19	**73.85**
Chest cut	63.47	61.49	78.06	**78.54**	68.92	69.09	**69.93**
Hips cut	7.69	8.93	**50.47**	46.38	45.55	45.57	**34.09**
Knee cut	2.14	2.14	21.31	20	39.43	**41.52**	**21.09**
	Upper body						
	Head	**Left shoulder**	**Right shoulder**	**Left elbow**	**Right elbow**	**Wrists**	**UB**
Neck cut	17.41	60.94	62.11	75.77	74.19	**76.18**	**63.25**
Shoulder cut	12.81	43.35	42.12	71.19	69.54	**74.75**	**55.49**
Chest cut	7.39	26.06	21.97	54.46	57.86	**69.05**	**43.69**
Hips cut	0.56	1.21	0.76	1.21	0.49	**4.49**	**1.89**
Knee cut	0.05	**0.62**	0.32	0.52	0.26	0.13	**0.29**

Tabella 5.4: Percentuali di keypoints classificati come corretti sul task di outpainting di Stable Diffusion

Il modello é pre-addestrato su un dataset con più di 5 miliardi di coppie image-text. Per mancanza di dati annotati e per via della sua grandezza non é stato possibile svolgere un fine-tuning di Stable Diffusion sul nostro dataset. Il modello perciò é stato sfruttato solo per fare inferenza sulle immagini.

Il task svolto viene chiamato outpainting e prende in input due condizioni testuali specifiche. La prima é detta "prompt" e serve a condizionare positivamente il modello, mentre la seconda é detta "negative prompt" con la quale specifichiamo le caratteristiche dell'immagine che vogliamo evitare. L'immagine in input é uguale a quella utilizzata per il task di uncropping descritto sopra. Come prompt testuale é stato inserita la seguente frase : "Complete the picture of a man running on a treadmill". Potrebbe sembrare una descrizione generica ma non si é notato una sostanziale differenza utilizzando prompt più specifici.

Osservando la prima parte della Tabella 5.4 notiamo che le immagini generate sono poco coerenti con la posa originale, fino al taglio all'altezza del petto. Dopodiché Stable Diffusion diventa più preciso degli altri modelli. Questo probabilmente é dovuto alla mancanza di un fine-tuning ed al poco contesto che gli viene fornito nel caso dei tagli più bassi.

Spostando l'occhio sulla seconda sezione della tabella, in particolare sulle ultime due righe, notiamo una scarsa precisione nella generazione della parte alta del corpo. Era facilmente ipotizzabile in quanto esistono innumerevoli variazioni della posizione del busto, delle braccia o della testa e risulta complesso predirle correttamente se al modello non vengono fornite in input condizioni specifiche.

5.2.4 Test sulle sequenze preliminari

Come specificato in precedenza prima di iniziare questo lavoro di tesi sono state raccolte delle sequenze video i cui frame mostrano le gambe di un individuo che corre. Per dimostrare che il metodo e i task di generazione proposti possono essere utilizzati su queste immagini, raccolte in condizioni simili ma non uguali a quelle del training set, sono stati svolti ulteriori test su questo dataset. Purtroppo i tempi non hanno permesso l'annotazione manuale dei keypoints e dato che tecniche di Human Pose Estimation non funzionano correttamente in queste condizioni, non é stato possibile riportare alcun dato quantitativo.

SD-infinity

Stable Diffusion é stato applicato nello stesso modo e con gli stessi prompt descritti in precedenza. In Figura 5.4 possiamo vedere i risultati ottenuti. Attraverso la generazione artificiale dell'immagine, HRNet é riuscito a predire correttamente la posizione del ginocchio sinistro e delle anche. Nell'immagine a sinistra i keypoints della parte alta sono stimati in posizioni casuali, Stable Diffusion partendo dalla mano destra aggrappata ai sostegni riesce a generare una posa plausibile e permettere una stima più accurata dei giunti.

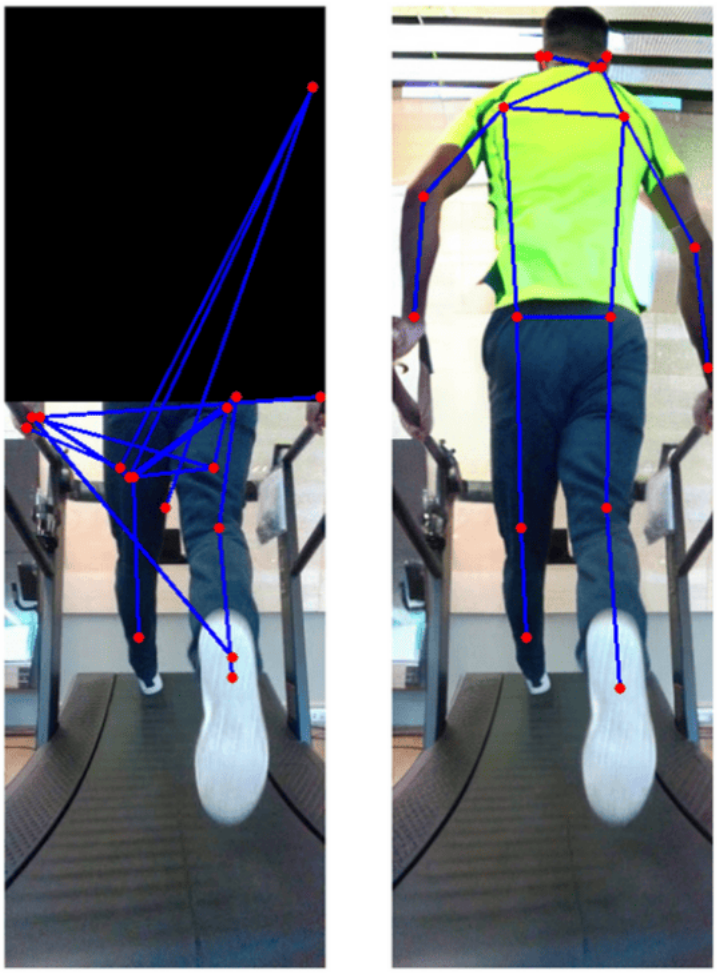

Figura 5.4: a) il frame originale estratto dalle sequenze preliminari a cui é stato applicato una patch nera per renderlo fruibile ad HRNet, b) l'immagine generata da Stable Diffusion

Palette

Palette non essendo un modello pre-addestrato, non avendo dunque una grande generalizzazione, ha avuto bisogno di alcuni accorgimenti. E' stato svolto un secondo addestramento sfruttando il dataset arricchito con le immagini riprese da vicino. Date le caratteristiche dei frame estrapolati dalle sequenze preliminari, durante il training sono state fornite al modello immagini tagliate sopra al ginocchio. Dopodiché é stato necessario svolgere qualche

altra epoca tagliando le immagini leggermente più in basso per permettere una generalizzazione migliore. Quest'approccio ha portato buoni risultati nel caso dell'inpainting, risparmiando a Palette la generazione dello sfondo intorno al corpo.

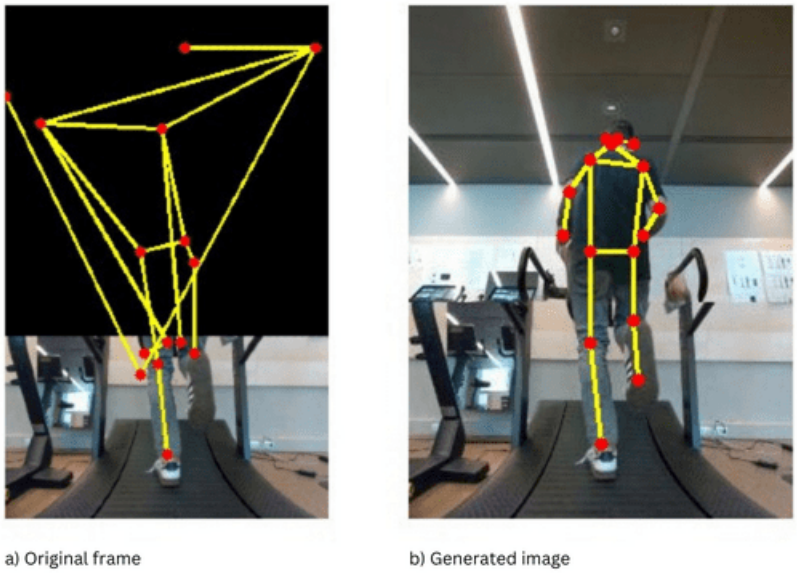

Figura 5.5: a) il frame originale estratto dalle sequenze preliminari a cui é stato applicato una patch nera per renderlo fruibile ad HRNet, b) l'immagine generata da Palette

In Figura 5.5 notiamo come dopo la sintesi della posa HRNet riesca a riconoscere esattamente la posizione del ginocchio destro, mentre nell'immagine di sinistra tenti di seguire la linea della gamba per stimare incorrettamente i keypoints delle anche. Si può intravedere una sfumatura di colori nella parte alta delle gambe dovuta alle tonalità dei pantaloni indossati dalle persone presenti nel dataset di addestramento.

5.2.5 Valutazione dei keypoints in uno spazio 3D

La strumentazione utilizzata per formare il dataset ha reso possibile la cattura di informazioni aggiuntive come la profondità. Grazie a queste informazioni é stato possibile proiettare i keypoints calcolati sulla persona da uno spazio 2D a uno 3D. Il problema principale é che la camera depth rispetto a quella RGB ha angolazioni diverse, quindi potrebbe riprendere una porzione più

ampia della scena. E' stato necessario allineare le immagini in modo che ogni pixel RGB combaciasse con la distanza nel frame depth. Dopodiché tramite le informazioni intrinseche della fotocamera che descrivono la relazione tra il sistema di coordinate 2D e 3D utilizzato e i dati raccolti, keypoints e profondità, é stato possibile ottenere un mapping dei pixel in punti 3D. Tramite queste nuove coordinate (x, y, z) sono stati estrapolati alcuni parametri informativi sullo stile di corsa riportati in Tabella 5.5.

	Left knee	**Right knee**	**Left groin**	**Right groin**	**Leg opening**
Run	**126.11**	122.47	94.11	88.51	49.19
Walk	**137.76**	133.85	93.31	89.19	46.07

Tabella 5.5: Angolatura media in gradi delle articolazioni inferiori in una sessione di corsa e camminata su tapis roulant

Questi i passaggi svolti per il calcolo degli angoli:

$$\vec{PQ} = (P_x - Q_x, P_y - Q_y, P_z - Q_z)$$

$$\vec{RQ} = (R_x - Q_x, R_y - Q_y, R_z - Q_z)$$

$$\alpha = \arccos\left(\frac{\vec{PQ}}{||\vec{PQ}||} \cdot \frac{\vec{RQ}}{||\vec{RQ}||}\right) \cdot \frac{180}{\pi}$$

dove P, Q, R sono i punti tridimensionali corrispondenti ai keypoints coinvolti nell'angolo. In particolare l'angolo del ginocchio é stato calcolato tramite i punti di anche e caviglie, quello dell'inguine con i punti di ginocchia e anche, mentre l'apertura delle gambe attraverso le ginocchia e il punto medio fra le anche.

Osservando i risultati ottenuti l'angolazione media del ginocchio é leggermente inferiore durante la corsa poiché viene piegato più spesso, mentre i restanti parametri sono simili fra loro.

5.3 Tempi di esecuzione

I test sono stati svolti su una singola GPU NVIDIA 1080 Ti con 12GB di RAM dedicata. Considerando la tipologia dei due modelli generativi presi in

esame e la loro grandezza ci aspettiamo tempistiche di inferenza molto più grandi per Stable Diffusion. Palette prendendo in input un batch size di 16 immagini e 1000 timesteps di denoising, riesce in 12 minuti a terminare la generazione ($\approx$ 45 secondi a immagine). Stable Diffusion invece con lo stesso numero di steps impiega $\approx$ 160 secondi a immagine. Tuttavia essendo un modello più generalizzato e migliore di Palette riesce in un numero 10 volte inferiore di timesteps ad ottenere gli stessi risultati.

Capitolo 6

Conclusioni

L'obiettivo di questo lavoro di tesi era quello di sviluppare un sistema in grado di monitorare in maniera automatica l'andamento della corsa di un atleta. Sfruttando le tecniche di Human Pose Estimation é possibile stimare la posizione delle articolazioni di una persona nell'immagine, parametri fondamentali per una corretta analisi dell'attività sportiva. Tuttavia, durante alcuni test preliminari é stato messo in luce un problema comune dei modelli di Human Pose Estimation: le predizioni dei keypoints si sono rivelate meno accurate se l'immagine mostra solo la parte bassa del corpo. Per risolvere questo problema é stato proposta la generazione artificiale tramite modelli a diffusione delle parti mancanti, con l'obiettivo di migliorare la stima dei giunti inferiori del corpo. I risultati ottenuti verificano la fattibilità di un utilizzo combinato dei modelli a diffusione nell'ambito della generazione delle immagini e della Human Pose Estimation. In particolare l'attenzione é stata rivolta sul confronto di due modelli: Palette e Stable Diffusion Infinity. Gli esperimenti hanno dimostrato che in entrambi i casi si riescono ad apportare miglioramenti significativi all'accuratezza delle predizioni di Human Pose Estimation. Emerge chiaramente che Stable Diffusion Infinity, grazie al suo ampio pre-addestramento, si distingue per un'ottima capacità di generazione partendo da immagini non presenti nel suo dataset di training. Nei casi dei tagli più bassi come quello al ginocchio e all'anca, si nota una difficoltà di Stable DIffusion Infinity nel generare una posa conforme a quelle presenti nel dataset raccolto durante questo lavoro. Questo fatto é probabilmente dovuto al poco contesto che viene fornito al modello e all'infinità di possibili variazioni della posizione del busto delle braccia, delle spalle e della testa. Al contrario Palette, che ha svolto un addestramento su immagini simili a quelle del test-set, ottiene risultati nettamente migliori generando su immagini tagliate così in basso. Per aumentare la qualità della generazione dunque sarebbe opportuno svolgere un ulteriore fine-tuning di Stable Diffusion sulle immagini del dominio applicativo. Durante gli esperimenti é

stato svolto un ulteriore confronto tra due tipi di generazione diverse, inpainting e uncropping. La seconda chiaramente é più facilmente applicabile in quanto non si necessita di ulteriore contesto oltre che a quello dell'immagine fornita in input. Con il metodo di inpainting e un aumento dei dati nel training-set, Palette é stato in grado di migliorare la qualità della generazione su immagini diverse da quelle su cui é stato addestrato. I dati inoltre suggeriscono che tramite inpainting si riescano ad ottenere delle pose più conformi alle immagini originali e un miglioramento della stima dei keypoints bassi nel caso dei tagli superiori al petto.

Quando le condizioni in input come la variazione della luminosità dell'immagine, il colore dei vestiti, le occlusioni degli arti, la posizione delle gambe e dei piedi, sono molto diverse da quelle del dataset di training la qualità della generazione di Palette peggiora. In un futuro sarebbe interessante raccogliere un dataset più ampio di immagini, per verificare se effettivamente una maggiore varietà di dati porti una generazione più flessibile su contesti diversi.

E' necessario svolgere un'ultima ma importante valutazione sui costi in termini di tempo e risorse necessari per adottare quest'approccio generativo presentano nel lavoro di tesi. Dagli esperimenti si evince che i due modelli impiegano dai 15 ai 45 secondi per singola immagine. Per un'applicazione real-time l'utilizzo di un modello generativo é complicato al giorno d'oggi in quanto per ottenere una durata della computazione simile a quella riportata, sarebbe necessario sfruttare dei dispositivi con schede video dedicate, e potrebbe essere molto costoso. In conclusione, l'obiettivo di questo lavoro di tesi é stato quello di mettere in luce il problema del calo di prestazione dei modelli HPE nel caso di immagini con parti del corpo oscurate e fornire una possibile soluzione che andasse effettivamente a migliorare le predizioni di quest'ultimi.

Ringraziamenti

Ringrazio tutte le persone che mi sono state vicine in questi anni e mi hanno supportato in questo lungo percorso di studi.

Bibliografia

[1] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, 2019.

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019.

[3] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[4] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9887–9895, 2019.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[7] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[9] Hai Liu, Tingting Liu, Zhaoli Zhang, Arun Kumar Sangaiah, Bing Yang, and Youfu Li. Arhpe: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction. *IEEE Transactions on Industrial Informatics*, 18(10):7107–7117, 2022.

[10] Jan Stenum, Cristina Rossi, and Ryan T Roemmich. Two-dimensional video-based analysis of human gait using pose estimation. *PLoS computational biology*, 17(4):e1008935, 2021.

[11] Dalila Burin, Yingxu Liu, Noriki Yamaya, and Ryuta Kawashima. Virtual training leads to physical, cognitive and neural benefits in healthy adults. *Neuroimage*, 222:117297, 2020.

[12] Jianbo Wang, Kai Qiu, Houwen Peng, Jianlong Fu, and Jianke Zhu. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In *Proceedings of the 27th ACM international conference on multimedia*, pages 374–382, 2019.

[13] Reza Afrouzian, Hadi Seyedarabi, and Shohreh Kasaei. Pose estimation of soccer players using multiple uncalibrated cameras. *Multimedia Tools and Applications*, 75:6809–6827, 2016.

[14] Kenny Chen, Paolo Gabriel, Abdulwahab Alasfour, Chenghao Gong, Werner K Doyle, Orrin Devinsky, Daniel Friedman, Patricia Dugan, Lucia Melloni, Thomas Thesen, et al. Patient-specific pose estimation in clinical environments. *IEEE journal of translational engineering in health and medicine*, 6:1–11, 2018.

[15] Felix Achilles, Alexandru-Eugen Ichim, Huseyin Coskun, Federico Tombari, Soheyl Noachtar, and Nassir Navab. Patient mocap: Human pose estimation under blanket occlusion for hospital monitoring applications. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part I 19*, pages 491–499. Springer, 2016.

[16] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016.

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[19] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International journal of computer vision*, 75:247–266, 2007.

[20] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61:55–79, 2005.

[21] Shradha Dubey and Manish Dixit. A comprehensive survey on human pose estimation approaches. *Multimedia Systems*, 29(1):167–195, 2023.

[22] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[23] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020.

[24] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression, 2017.

[25] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1944–1953, 2021.

[26] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[27] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[28] Ross Girshick. Fast r-cnn, 2015.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016.

[32] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.

[33] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[34] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation, 2016.

[35] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model, 2016.

[36] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines, 2016.

[37] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[38] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[40] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation, 2018.

[41] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[42] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking, 2018.

[43] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey, 2023.

[44] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[45] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

[46] Jianfeng Zhang, Yujun Cai, Shuicheng Yan, Jiashi Feng, et al. Direct multi-view multi-person 3d pose estimation. *Advances in Neural Information Processing Systems*, 34:13153–13164, 2021.

[47] Rongchang Xie, Chunyu Wang, and Yizhou Wang. Metafuse: A pre-trained fusion model for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13686–13695, 2020.

[48] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[49] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[50] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[51] Victor Costa, Nuno Lourenço, and Penousal Machado. *Coevolution of Generative Adversarial Networks*, page 473–487. Springer International Publishing, 2019.

[52] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[53] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks, 2017.

[54] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[55] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

[56] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis, 2023.

[57] Guillermo Iglesias, Edgar Talavera, and Alberto Díaz-Álvarez. A survey on gans for computer vision: Recent research, analysis and taxonomy. *Computer Science Review*, 48:100553, May 2023.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[59] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.

[60] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019.

[61] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[62] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[63] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[65] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[66] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines, 2019.

[67] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[68] Yuxin Wu and Kaiming He. Group normalization, 2018.

[69] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022.