UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Corso di Laurea Magistrale in Ingegneria Informatica

REPORT FINALE – Gruppo 8

ARTIFICIAL VISION CONTEST 2025

People Detection, Tracking, Classification and Behavior Analysis

DOCENTI

Mario VENTO

Antonio GRECO

STUDENTI

Valentina MICERA – 0622702379

Raffaele SBARDELLA – 0622702312

Ciro SETOLINO – 0622702171

Jacopo VOLPE – 0622702301

ANNO ACCADEMICO 2024/2025

Sommario

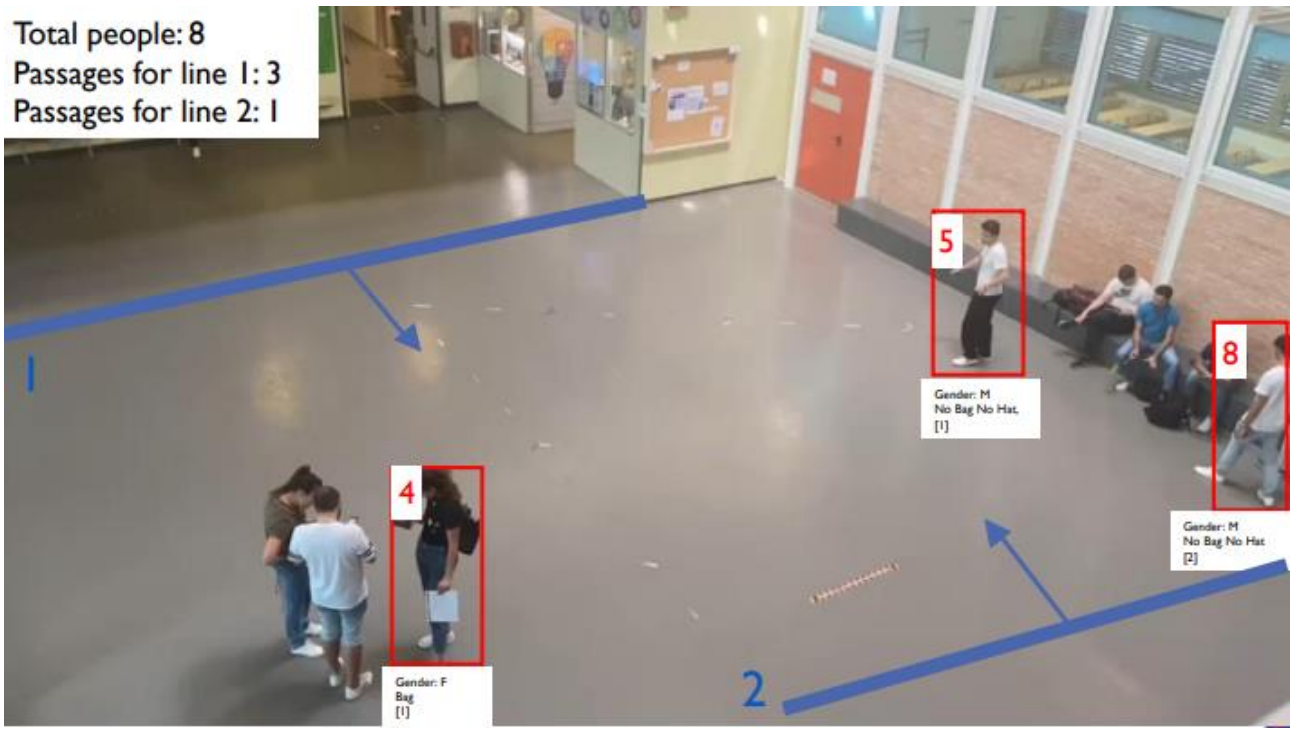
Introduzione

Nel presente report verranno descritte le attività necessarie per la progettazione e lo sviluppo di un sistema di visione artificiale in vista dell'*Artificial Vision Contest 2025*.

Il contest si svolgerà all'interno dell'atrio CUES, dove verranno registrati video in tempo reale per simulare scenari complessi di rilevamento e analisi del comportamento umano.

L'obiettivo del progetto è realizzare un software in grado di rilevare, tracciare e analizzare i movimenti delle persone presenti nella scena, identificando attributi specifici come genere, presenza di borse e cappelli, e monitorando i passaggi su linee virtuali predefinite.

Oltre a fornire un'interfaccia grafica per la visualizzazione in tempo reale, il sistema genererà un file di risultati dettagliato che sarà utilizzato per valutare le performance rispetto a un *groundtruth*. Questo report illustra le tecnologie adottate e i criteri di ottimizzazione per garantire un approccio efficace e accurato.



Mapping Geometrico

Il mapping geometrico è una tecnica fondamentale per la proiezione di punti tridimensionali su una superficie bidimensionale, come un'immagine.

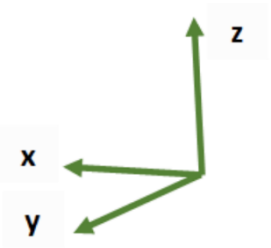
Il processo descritto è stato progettato per tradurre le coordinate 3D di un insieme di punti nello spazio in coordinate di pixel in un'immagine, utilizzando una telecamera virtuale e una rappresentazione matematica accurata.

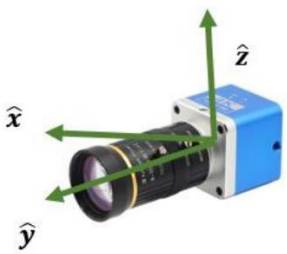
Di seguito è descritta l'implementazione passo-passo di questa tecnica.

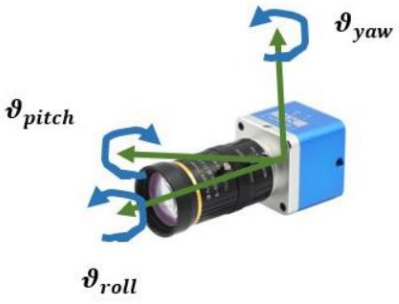
Parametri e Modello della Camera

La camera è stata modellata come un dispositivo pinhole con parametri generici:

- **Focale (f)**.
- **Dimensioni del piano immagine:** definite in termini di risoluzione (larghezza e altezza in pixel) e dimensioni fisiche (larghezza e altezza in metri).
- **Posizione della camera:** specificata rispetto a un sistema di riferimento globale, attraverso coordinate tridimensionali (x, y, z).
- **Orientamento della camera:** determinato dagli angoli di rollio (throll), beccheggio (thpitch) e imbardata (thyaw) rispetto alla propria direzione iniziale ([0 1 0]).







Proiezione dei Punti

La funzione project_points implementa l'intero processo di proiezione dei punti tridimensionali sul piano immagine, seguendo i passaggi descritti:

1. **Calcolo della direzione di inquadramento della camera:**

   La rotazione della camera è descritta da due angoli principali:

- imbardata (asse Z)

$$R_{yaw} = \begin{bmatrix} \cos(\theta_{yaw}) & -\sin(\theta_{yaw}) & 0 \\ \sin(\theta_{yaw}) & \cos(\theta_{yaw}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- beccheggio (asse X)

$$R_{pitch} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{pitch}) & -\sin(\theta_{pitch}) \\ 0 & \sin(\theta_{pitch}) & \cos(\theta_{pitch}) \end{bmatrix}$$

La matrice di rotazione risultante è il prodotto delle singole matrici di rotazione lungo questi assi.

$$R = R_{yaw} \cdot R_{pitch}$$

La **direzione di puntamento della camera** è stata quindi calcolata effettuando il prodotto tra la matrice di rotazione e la posizione di base della telecamera [0 1 0].

$$\vec{d}_{camera} = R \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Determinazione del Piano Immagine

Il piano immagine è posizionato davanti alla camera ad una distanza prestabilita.
Le sue dimensioni fisiche sono calcolate considerando il campo visivo orizzontale e verticale determinato dalla focale e dalla risoluzione dell'immagine.

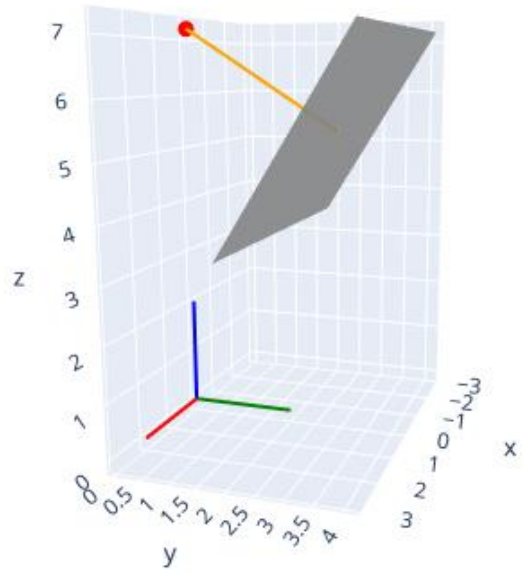
$$W_{plane} = distanza\_piano * \frac{s_w}{f}$$

$$H_{plane} = distanza_{piano} * \frac{s_h}{f} * PAR$$

Dove PAR (pixel aspect ratio) è un termine di correzione, calcolato come $\frac{s_{w/U}}{s_h/V}$, necessario per gestire la dimensione non quadrata, ma rettangolare dei pixels.

Il piano immagine è un piano cartesiano posizionato nello spazio tridimensionale con una normale che coincide con la direzione della telecamera.

Si ottiene utilizzando:



- **Un punto sul piano P**

  Ottenuto spostandosi dalla posizione della telecamera di una distanza fissa nella direzione di puntamento della camera.

- **La normale al piano $\vec{n}$**

  La normale del piano coincide con la direzione di puntamento della telecamera

L'equazione cartesiana del piano si presenta nella forma:

$$a \cdot x + b \cdot y + c \cdot z + d = 0$$

Dove:

- **(a, b, c)** è la normale del piano.
- **d** è determinato dalla posizione del punto P = (x_0, y_0, z_0)

$$d = -(a \cdot x_0 + b \cdot y_0 + c \cdot z_0)$$

Successivamente vengono calcolati i vertici del piano conoscendo la posizione del contro del piano **P** e la normale $\vec{n}$. Si calcolano due vettori ortogonali alla normale per calcolare le direzioni di spostamento per raggiungere le coordinate dei vertici:

$$\vec{right} = \frac{\vec{n} \times [0,0,1]}{\|\vec{n} \times [0,0,1]\|} \qquad \vec{up} = \vec{right} \times \vec{n}$$

Utilizzando i vettori $\overrightarrow{right}$ e $\overrightarrow{up}$, e le dimensioni ***Wplane*** e ***Hplane***, i vertici del piano sono determinati attraverso i seguenti calcoli:

$$\vec{P}_{TL} = \vec{P}_{plane} + \frac{W_{plane}}{2} \cdot \vec{right} + \frac{H_{plane}}{2} \cdot \vec{up}$$

$$\vec{P}_{TR} = \vec{P}_{plane} - \frac{W_{plane}}{2} \cdot \vec{right} + \frac{H_{plane}}{2} \cdot \vec{up}$$

$$\vec{P}_{BL} = \vec{P}_{plane} + \frac{W_{plane}}{2} \cdot \vec{right} - \frac{H_{plane}}{2} \cdot \vec{up}$$

$$\vec{P}_{BR} = \vec{P}_{plane} - \frac{W_{plane}}{2} \cdot \vec{right} - \frac{H_{plane}}{2} \cdot \vec{up}$$

Successivamente i vertici vengono ruotati intorno al centro per tenere conto del parametro ***throll*** traslando i vertici in modo che il centro del piano coincida con il centro del sistema di riferimento e utilizzando la **formula di Rodrigues**:

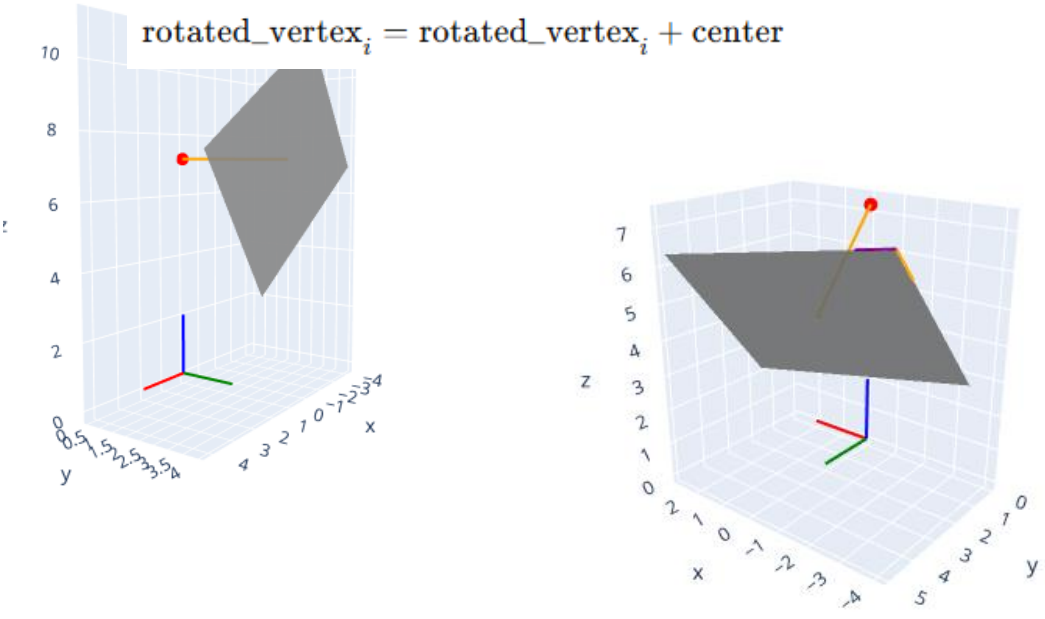
$$\text{translated\_vertex}_i = \text{vertex}_i - \text{center}$$

$$R = \begin{bmatrix} \cos(\theta) + u_x^2(1-\cos(\theta)) & u_x u_y(1-\cos(\theta)) - u_z \sin(\theta) & u_x u_z(1-\cos(\theta)) + u_y \sin(\theta) \\ u_y u_x(1-\cos(\theta)) + u_z \sin(\theta) & \cos(\theta) + u_y^2(1-\cos(\theta)) & u_y u_z(1-\cos(\theta)) - u_x \sin(\theta) \\ u_z u_x(1-\cos(\theta)) - u_y \sin(\theta) & u_z u_y(1-\cos(\theta)) + u_x \sin(\theta) & \cos(\theta) + u_z^2(1-\cos(\theta)) \end{bmatrix}$$

$$\text{rotated\_vertex}_i = R \cdot \text{translated\_vertex}_i$$

Infine si riportano i vertici nella loro posizione originale aggiungendo il centro:

$$\text{rotated\_vertex}_i = \text{rotated\_vertex}_i + \text{center}$$

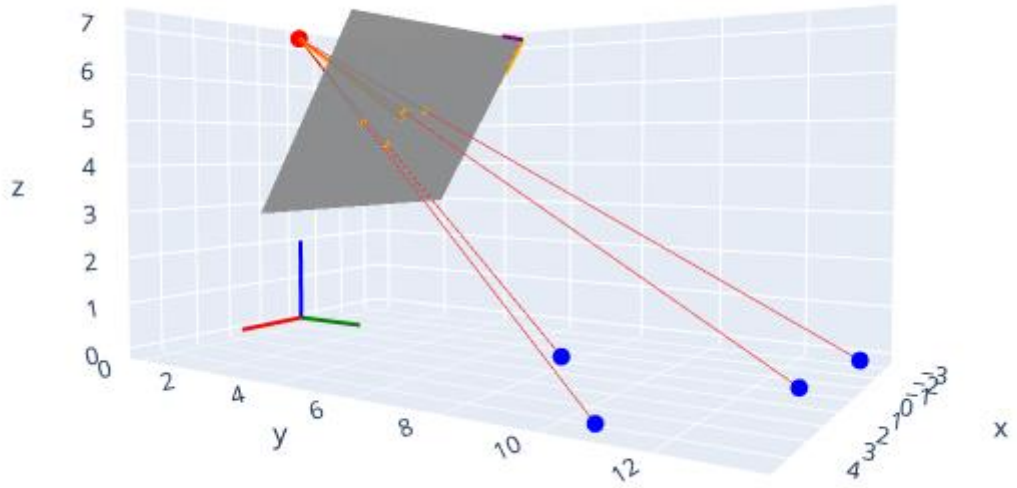


Esempio piano
roll=30°, pitch=0°, yaw=0°

Esempio piano
roll=10.5°, pitch=-36°, yaw=12°

2. Proiezione dei Punti sul Piano

Ogni punto tridimensionale è trasformato in un raggio che parte dalla camera e interseca il piano immagine.



L'intersezione determina le coordinate locali sul piano immagine, che vengono poi scalate per ottenere i corrispondenti pixel nell'immagine.

Viene identificato un nuovo sistema di riferimento con due assi con origine nel vertice in altro a sinistra (stesso riferimento utilizzato per l'origine delle coordinate in pixel) con gli assi calcolati come segue.

Gli assi del sistema sono definiti come:

- **Asse $\vec{w}$:** Diretto dal vertice in alto a sinistra verso il vertice in alto a destra.

$$\vec{w} = \frac{\text{vertice}_{\text{top\_right}} - \text{vertice}_{\text{top\_left}}}{\|\text{vertice}_{\text{top\_right}} - \text{vertice}_{\text{top\_left}}\|}$$

- **Asse $\vec{h}$:** Diretto dal vertice in alto a sinistra verso il vertice in basso a sinistra.

$$\vec{h} = \frac{\text{vertice}_{\text{bottom\_left}} - \text{vertice}_{\text{top\_left}}}{\|\text{vertice}_{\text{bottom\_left}} - \text{vertice}_{\text{top\_left}}\|}$$

Per ciascun punto proiettato $\overrightarrow{\boldsymbol{P_{intersect}}}$, la proiezione sui nuovi assi si ottiene utilizzando il prodotto scalare:

$$\text{proj\_w} = (\vec{P}_{intersect} - \text{vertice}_{\text{top\_left}}) \cdot \vec{w}$$

$$\text{proj\_h} = (\vec{P}_{intersect} - \text{vertice}_{\text{top\_left}}) \cdot \vec{h}$$

Le coordinate proiettate sono scalate in pixel in base alla risoluzione del piano immagine:

$$u = \text{int}\left(\frac{\text{proj\_w} \cdot U}{W_{plane}}\right) \qquad v = \text{int}\left(\frac{\text{proj\_h} \cdot V}{H_{plane}}\right)$$

Il risultato finale è un insieme di coordinate (u,v), che rappresentano i punti proiettati nello spazio immagine, espressi nel sistema di riferimento dei pixel del piano immagine.

Di seguito è mostrato un esempio di configurazione con annesso il risultato del mapping:

```
f = 0.003                                # Focale della camera (m)
U = 1280                                 # Risoluzione orizzontale sensore (px)
V = 720                                  # Risoluzione verticale sensore (px)
thyaw = 0 * math.pi / 180                # Angolo di Yaw (orientamento attorno all'asse Z)
throll = 0 * math.pi / 180               # Angolo di Roll (orientamento attorno all'asse X)
thpitch = -32 * math.pi / 180            # Angolo di Pitch (orientamento attorno all'asse Y)
xc = 0                                   # Posizione camera lungo X nel sistema di riferimento
yc = 0                                   # Posizione camera lungo Y
zc = 7.20                                # Posizione camera lungo Z
x_real = [-2.5, 0.5, 0.5, 4.6]           # Coordinate reali dei punti nel mondo
y_real = [13.41, 8.00, 13.00, 10.91]     # Coordinate reali Y dei punti
s_w = 0.00498                            # Larghezza fisica del sensore (m)
s_h = 0.00374                            # Altezza fisica del sensore (m)
```

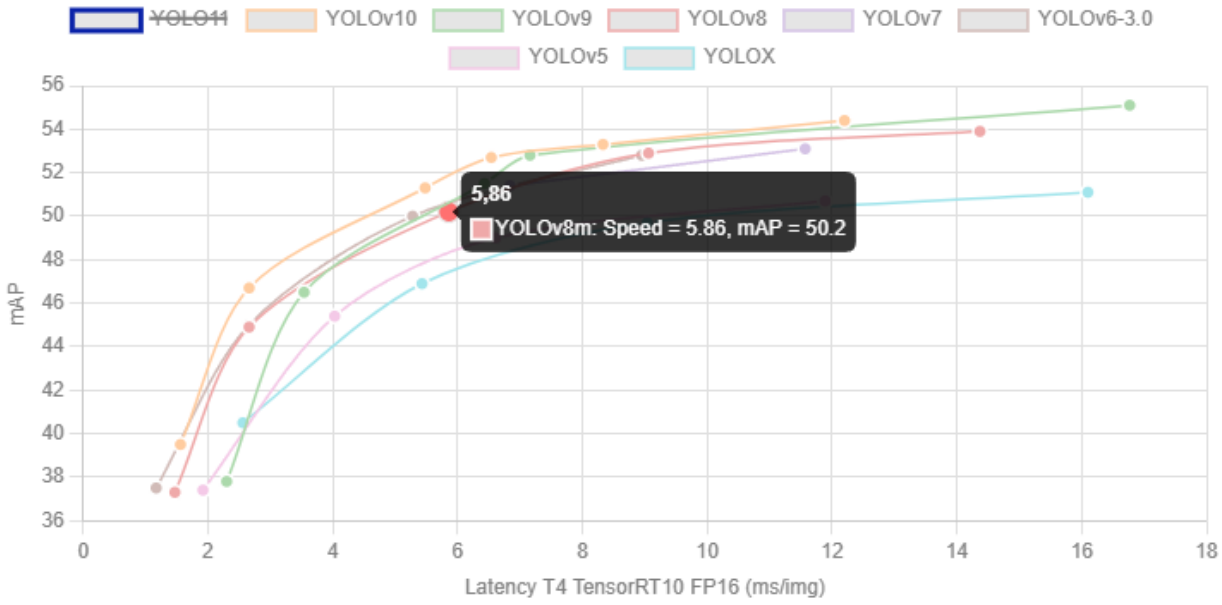
Esempio di utilizzo della tecnica di mapping 3D applicata su una fotografia all'atrio CUES dell'Università degli Studi di Salerno.

Detection

Per la fase di detection è stato utilizzato YOLOv8 (You Only Look Once, versione 8), uno dei modelli più avanzati e performanti per il rilevamento degli oggetti.
La sua principale forza risiede nella capacità di operare in tempo reale, rendendo possibile l'identificazione e la classificazione rapida degli oggetti all'interno di una scena. Rispetto alle versioni precedenti, YOLOv8 integra innovazioni e ottimizzazioni che ne migliorano ulteriormente efficienza e versatilità.

La famiglia di algoritmi di Object Detection su cui è stata focalizzata l'attenzione è quella contenente tutte le versioni Medium degli algoritmi di YOLO, in quanto gli algoritmi di questa famiglia rappresentano un buon punto di incontro tra la precisione degli algoritmi Large e l'efficienza computazionale degli algoritmi Small.

La scelta è ricaduta sull'utilizzo di YOLOv8m, in quanto tale algoritmo risulta molto più affermato e compatibile rispetto ai nuovi YOLO11m e YOLOv10m. Inoltre, YOLOv8m permette di ottenere un buon compromesso tra latenza e mean Average Precision; infatti, esso presenta una migliore mAP rispetto ai suoi "fratelli minori" YOLOv7m, YOLOv6m, YOLOv5l e YOLOXm, ed una latenza minore rispetto a YOLOv9m.



Per quanto riguarda i parametri utilizzati per eseguire l'Object Detection, cioè confidence e iou (intersection over union), si è scelto di impostare conf=0.3 e iou=0.9.

La decisione di questi parametri è legata alla scelta di non voler precludere eventuali rilevamenti di persone e quindi di bounding box da passare al Tracker, in quanto sarà poi quest'ultimo a scartare eventuali bounding box che non matchano con tracce esistenti o che non superano il threshold per diventare una nuova traccia.

Tracking

Integrare un algoritmo di Tracking con un modello di Object Detection è cruciale per migliorare l'efficacia del sistema nell'analisi dinamica delle scene. Mentre un modello di rilevamento degli oggetti si concentra sulla rilevazione e classificazione degli oggetti in singoli fotogrammi, l'aggiunta di un algoritmo di Tracking consente di mantenere la continuità nel monitoraggio degli oggetti attraverso frame successivi.
Questa coerenza è essenziale per tracciare il movimento degli oggetti nel tempo e ottenere informazioni significative sul comportamento dinamico della scena.
A tal fine, l'attenzione è stata rivolta all'integrazione di un algoritmo di Tracking avanzato chiamato BoT-SORT.

BoT-SORT è un sistema di tracciamento multi-oggetto che sfrutta il paradigma del tracking-by-detection, che combina tecniche di associazione spaziale, come il calcolo dell'IoU con metriche basate sulle feature dei rilevamenti, per abbinare le tracce in modo affidabile tra i frame.
BoT-SORT integra l'efficacia del Non-Maximum Suppression (NMS) per filtrare i falsi positivi e utilizza strategie di matching robusto per gestire occlusioni e variazioni rapide nei movimenti degli oggetti.
Al fine di ottenere un tracking ottimale delle persone nella scena, l'algoritmo BoT-SORT è stato valutato con diversi parametri di configurazione.

Il team ha avuto cura di registrare diversi video dell'atrio CUES, da diverse angolazioni e con diverse intensità di luce, da utilizzare poi per il testing dei parametri di BoT-SORT.
I parametri finali sono stati ottenuti tramite un processo di affinamento, verificando le performance sui tali video, in modo tale da non overfittare i suddetti parametri ad eventuali condizioni di luce, di affollamento o a determinate angolazioni della videocamera.

Di seguito la configurazione definitiva dei parametri.

1. **tracker_type: botsort**
   Questo parametro specifica il tipo di tracker utilizzato. In questo caso, BoT-SORT.
2. **track_high_thresh: 0.73**
   Soglia di confidenza per la prima associazione tra un rilevamento e una traccia esistente. Viene utilizzata per determinare se un oggetto rilevato è sufficientemente simile a una traccia già in corso. Valori più alti rendono il processo più selettivo, riducendo il rischio di associazioni errate.
3. **track_low_thresh: 0.3**
   Soglia di confidenza inferiore per una seconda associazione. Consente di ripristinare una traccia persa temporaneamente o di gestire situazioni in cui il rilevamento è ambiguo. Aiuta a migliorare la robustezza del sistema in condizioni difficili.

4. **new_track_thresh: 0.84**
   Soglia minima di confidenza per avviare una nuova traccia. Se un rilevamento non può essere associato a una traccia esistente con una confidenza superiore a 0.84, viene creata una nuova traccia. Questo parametro controlla la creazione di nuove tracce, evitando che rilevamenti non affidabili inneschino falsi positivi.
5. **track_buffer: 200**
   Numero massimo di frame memorizzati per ogni traccia. Il buffer viene utilizzato per gestire informazioni storiche delle tracce, come il tempo di vita e le condizioni per la loro eliminazione. Un valore più alto consente di mantenere attive tracce per periodi più lunghi.
6. **match_thresh: 0.999**
   Soglia per determinare la corrispondenza tra tracce. Stabilisce un livello per associare due tracce diverse, riducendo al minimo il rischio di unione errata di tracce appartenenti a oggetti distinti.
7. **fuse_score: True**
   Abilitazione della fusione delle informazioni di confidenza. Combina i punteggi di confidenza di rilevamento e apparenza visiva per migliorare la precisione complessiva del sistema di tracking. Questo consente una valutazione più affidabile delle associazioni tra tracce e rilevamenti.

Pedestrian Attribute Recognition (PAR)

La PAR è un problema appartenente al mondo del computer vision che mira a riconoscere e classificare diversi attributi di una persona nelle immagini o nei video.

Nel nostro progetto, l'obiettivo era, dato un **pedone**, **identificarne il genere**, **la presenza di borse** e **la presenza di cappelli**.

Dataset

Uno degli aspetti fondamentali per il Pedestrian Attribute Recognition (PAR) sono i dataset su cui addestrare e validare il modello ideato e progettato proprio per questo scopo.

Il dataset di riferimento per l'addestramento dei modelli di PAR è stato il MIVIA PAR Dataset, formato da training e validation set composti rispettivamente da 93.081 e 12.162 campioni.

Ad ogni campione sono associate 5 label:

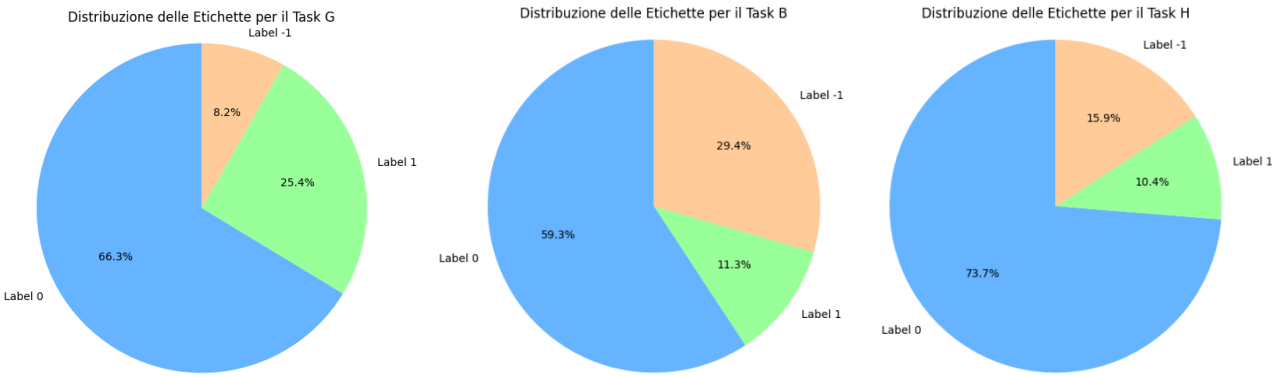
- le prime due sono irrilevanti per quelli che sono i nostri scopi in questo progetto, in quanto fanno riferimento ai colori della parte superiore e inferiore del corpo
- il genere della persona (0 Maschio, 1 Femmina)
- la presenza o meno della borsa (0 Non Presente, 1 Presente)
- la presenza o meno del cappello (0 Non Presente, 1 Presente)

Ogni etichetta, inoltre, può assumere il valore -1, che indica l'assenza di informazioni.

Analizzando il dataset sono emersi i seguenti problemi:

- La presenza di numerose etichette con valore "-1", che compromettesse la qualità complessiva del dataset, influenzando negativamente le prestazioni dei modelli.
- Uno sbilanciamento tra le classi, che rendesse difficile per i modelli apprendere correttamente le caratteristiche delle classi meno rappresentate.

Si è osservata, inoltre, la distribuzione delle label:



Per affrontare questi problemi, sono state esplorate diverse strategie con l'obiettivo di identificare un approccio che permettesse di sfruttare al meglio i dati disponibili senza compromettere la capacità del modello di generalizzare.

Rimozione dei campioni con etichette mancanti o incomplete

La prima strategia considerata è stata quella di eliminare dal dataset tutti i campioni che presentavano almeno un'etichetta con valore -1. Sebbene semplice da implementare, circa 40.000 campioni sono stati eliminati, con una conseguente perdita significativa di dati e di diversità all'interno del dataset. Questa strategia è stata quindi abbandonata, poiché la perdita di dati si è rivelata troppo penalizzante.

Utilizzo di una masked loss per le etichette mancanti

Questa tecnica consente di ignorare i campioni con etichette mancanti durante il calcolo della funzione di loss. In pratica, il modello aggiornava i suoi pesi solo per i task con etichette valide, evitando di essere penalizzato dalle informazioni mancanti.
In questo modo, i campioni privi di alcune etichette non hanno contribuito all'apprendimento per quei task specifici. Questa strategia ha però mantenuto una distribuzione sbilanciata dei dati, poiché alcuni task risultavano sottorappresentati rispetto ad altri.

Utilizzo di VQA per aggiungere le etichette mancanti

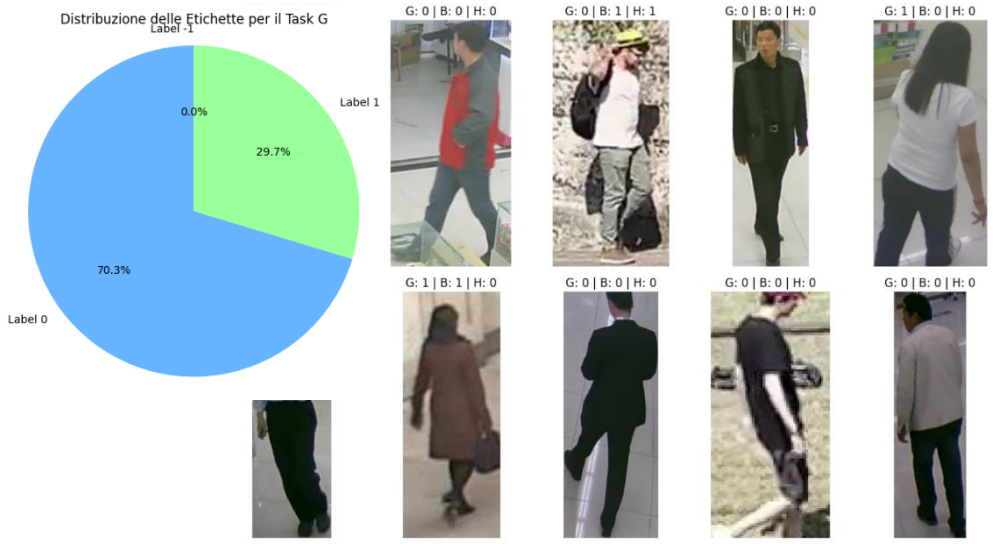
In questo step, si è cercato di migliorare il dataset fornito utilizzando un approccio basato su Visual Question Answering (VQA). Questo metodo prevede l'uso di un modello Vision-and-Language Transformer (VILT) per porre domande mirate e ottenere informazioni utili dalle risposte generate.

Prima di procedere con l'arricchimento del dataset, il modello VILT è stato sottoposto a un processo di fine-tuning sfruttando le etichette già presenti nel dataset "VQA". Quest'ultimo contiene 265.016 immagini e include per ciascuna immagine una media di oltre 5 domande e 10 risposte ground truth. Le etichette del dataset VQA sono state utilizzate per addestrare il modello a rispondere in modo accurato alle domande, rendendolo capace di interpretare correttamente il contenuto visivo e fornire risposte coerenti.

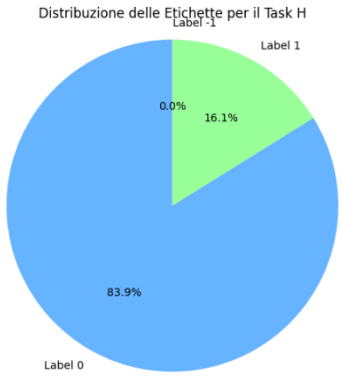
Dopo il fine-tuning, il modello è stato utilizzato per analizzare le immagini del training set attraverso tre domande specifiche:

- **"Is the person female?"**, per determinare il genere della persona.
- **"Does the person wear a hat?"**, per verificare se la persona indossa un cappello.
- **"Does the person wear a bag?"**, per sapere se la persona porta uno zaino.

Le risposte generate dal modello sono state memorizzate in un nuovo dataset e utilizzate per sostituire le etichette con valore "-1" nel dataset originale. Questo processo ha permesso di arricchire il dataset con etichette complete e accurate. Il nuovo dataset risultante è stato successivamente utilizzato per addestrare i modelli sottoposti a test.



Campioni prelevati casualmente dal dataset interamente etichettato, con le rispettive etichette



Distribuzioni dei campioni sul nuovo dataset labellato

Bilanciamento del dataset con tecniche di undersampling e oversampling

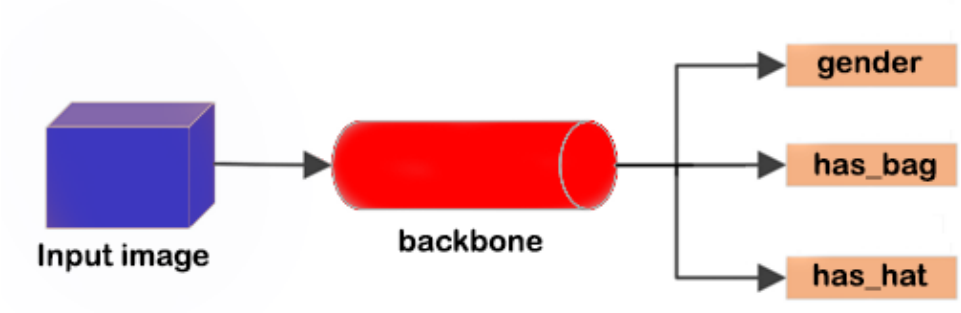
Un'altra strategia esplorata è stata quella di applicare tecniche di undersampling e oversampling per bilanciare il dataset, agendo specificamente sui campioni con etichette più o meno frequenti.

- Undersampling: È stato applicato sui campioni con etichetta più frequente, diminuendo il loro numero per ridurre lo sbilanciamento nel dataset, evitando che il modello fosse troppo influenzato dalla classe dominante.
- Oversampling: È stato invece applicato sui campioni con etichetta meno frequente, duplicando i dati o generando nuovi campioni sintetici per aumentarne il numero e il peso relativo nel dataset. Questo ha aiutato a dare maggiore importanza ai task con meno dati, evitando che il modello fosse troppo sbilanciato verso i task con più campioni

Architettura della rete

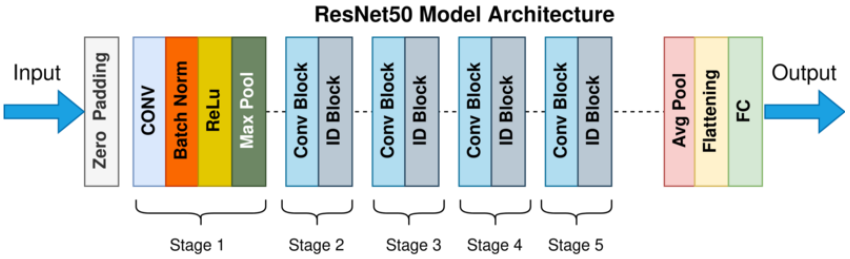
L'architettura proposta si basa su un approccio **multi-task**, progettato per affrontare la classificazione simultanea di più attributi pedonali sfruttando una rete neurale convoluzionale profonda. L'obiettivo è ottimizzare l'uso delle risorse computazionali condividendo le caratteristiche apprese dalla backbone tra i diversi task, pur mantenendo una specializzazione nelle predizioni per ciascun attributo.

La rete si compone di due componenti principali: una backbone condivisa per l'estrazione delle features e un branch specifico per ciascun task.



Backbone condivisa

La backbone della rete utilizza un modello pre-addestrato ResNet-50, una delle architetture di deep learning più consolidate per l'estrazione di caratteristiche visive da immagini.



L'ultimo blocco della rete ResNet-50, contenente AvgPool, Flattening e FullyConnceted layers, è stato rimosso, lasciando uno strato convoluzionale finale che genera una rappresentazione significativa per permettere alle nuove teste di effettuare i compiti di classificazione.

Utilizzare una rete pre-addestrata consente di sfruttare i pesi pre-addestrati, riducendo i tempi di addestramento e migliorando la generalizzazione.

Branch per i singoli task

Ogni task viene gestito da un branch indipendente.

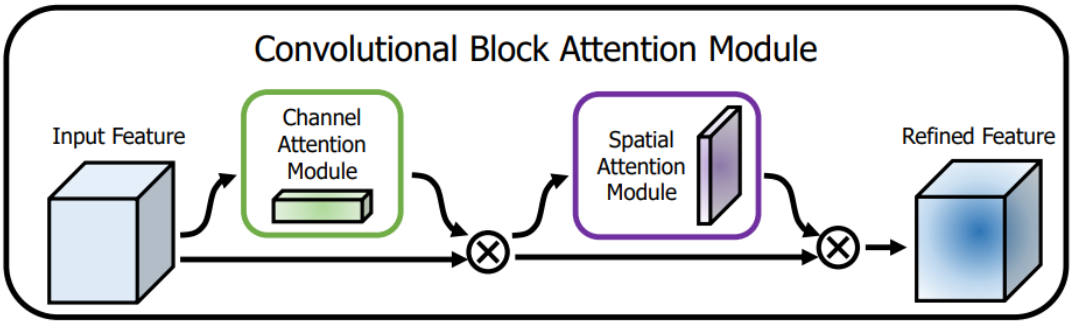
La struttura di ciascun branch include i seguenti componenti:

- Integrazione di un modulo di attenzione (CBAM) per enfatizzare le regioni chiave dell'immagine, migliorando la capacità di discriminare gli attributi.
- Pooling layer, per ridurre la dimensionalità dei dati.
- Flattening delle caratteristiche estratte dalla backbone, per trasformare le feature convoluzionali in un formato compatibile con i layer fully-connected, consentendo l'elaborazione da parte della testa di classificazione.
- Tre layer fully-connected con funzione di attivazione ReLU[CS1], alternati con batch-normalization layer e dropout layer, per prevenire l'overfitting.
- Ogni branch termina con un layer fully-connected seguito da una funzione di attivazione sigmoidale per fare predizione sull'attributo dello specifico task.

I branch sono stati progettati in maniera indipendente per garantire una maggiore specializzazione di ciascuna head. Ogni branch ottimizza, infatti, le proprie prestazioni in base alla natura del proprio task, senza entrare in conflitto con gli altri tasks.

Modulo di attenzione (CBAM)

Come modulo di attenzione è stato utilizzato CBAM (Convolutional Block Attention Module) per via delle sue caratteristiche distintive. Il CBAM è un componente progettato per ottimizzare le prestazioni delle reti neurali convoluzionali (CNN) implementando un efficace meccanismo di attenzione.



Questo modulo è composto da due sotto-moduli sequenziali, ognuno con una funzione specifica: uno dedicato all'attenzione per canale (channel-wise attention- CAM) e l'altro all'attenzione spaziale (spatial-wise attention - SAM). Il componente di attenzione per canale apprende i pesi relativi ai canali tramite operazioni di pooling e convoluzione, mentre il componente di attenzione spaziale utilizza queste informazioni per valutare dinamicamente l'importanza delle diverse posizioni spaziali nelle caratteristiche intermedie. La scelta di utilizzare CBAM al posto della SAM o della CAM è motivata dal

Modulo di attenzione CBAM

fatto che gli autori stessi consigliano di utilizzarle entrambe per poter ottenere i migliori risultati.

Nel design della rete neurale multi-task, il modulo di attenzione è stato posizionato strategicamente prima del livello di pooling.
Questa scelta è stata guidata da una serie di considerazioni tecniche e funzionali, mirate a massimizzare le prestazioni del modello nei compiti di classificazione multi-task.
Infatti, il livello di pooling, come l'Average Pooling, riduce la dimensionalità spaziale delle caratteristiche, comprimendo l'informazione. Posizionando il CBAM prima del pooling, il modulo può lavorare su una rappresentazione più ricca di dettagli spaziali, permettendo al meccanismo di attenzione di identificare e focalizzarsi su regioni specifiche dell'immagine con maggiore precisione.

Questo è particolarmente importante in un contesto multi-task, dove ogni testa potrebbe richiedere attenzione su diverse parti dell'immagine.

Training

Nel corso dell'addestramento del modello, sono stati esplorati diversi approcci di valutazione e ottimizzazione al fine di ottenere il miglior risultato possibile.

Inizialmente, è stata monitorata la loss sul training set, che mostrava un calo costante, indicativo del fatto che il modello stesse apprendendo correttamente dai dati. Tuttavia, un'analisi più approfondita sull'accuratezza (accuracy) del modello sul validation set ha messo in luce un problema significativo: sebbene l'accuracy fosse elevata per tutti e tre i task, il modello tendeva a predire costantemente l'etichetta "0".

Questo comportamento è stato attribuito alla natura sbilanciata del dataset, che presentava un numero significativamente maggiore di campioni con etichetta "0" rispetto a quelli con etichetta "1".

Per superare questa limitazione, è stato scelto l'F-score come metrica di valutazione, poiché a differenza dell'accuracy, l'F-score bilancia la precisione e il recall, risultando particolarmente utile in contesti con dataset sbilanciati. Questa metrica ha permesso di valutare in modo più accurato le prestazioni del modello, evidenziando la sua capacità di discriminare correttamente tra le due classi, nonostante lo squilibrio dei dati.

Inoltre, per cercare di ridurre ulteriormente l'effetto dello sbilanciamento, è stato applicato un approccio di undersampling sui campioni negativi, cercando di bilanciare la distribuzione delle classi e migliorare le performance complessive del modello e di utilizzare una Loss asimmetrica che desse maggiore peso ai campioni delle classi meno presenti.

FASE 1: Pre-training delle teste

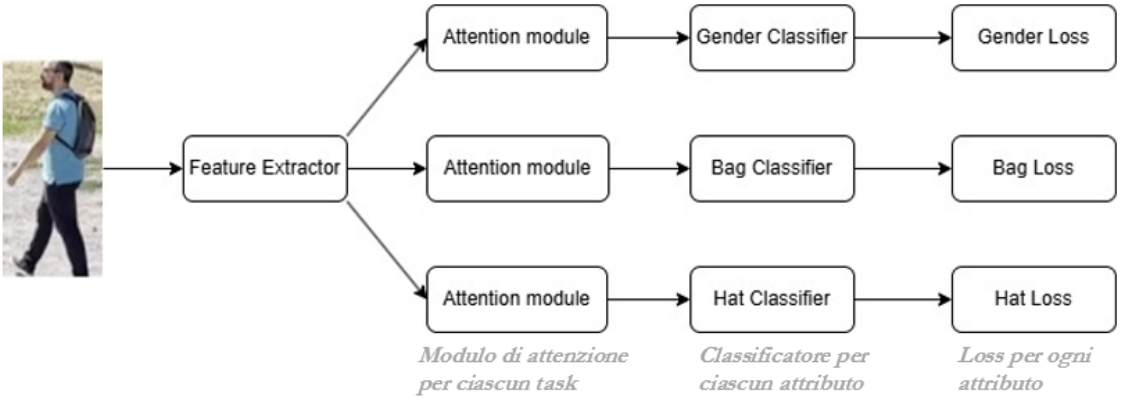
Come già accennato nel paragrafo precedente, inizialmente il training si è concentrato esclusivamente sull'aggiornamento dei pesi delle teste, consentendo loro di specializzarsi nei rispettivi task senza alterare i parametri della backbone.

Sono stati realizzati quindi 3 addestramenti separati seguendo le seguenti linee guida:

- Dataset parzialmente bilanciato: è stata effettuata un'operazione di undersampling per portare il training dataset in uno sbilanciamento massimo 65/35
- Learning rate più alto delle fasi successive: 1e-3
- Numero limitato di epoche: 10
- Utilizzata come funzione di loss la Binary Cross Entropy, adatta per i problemi di classificazione binaria.

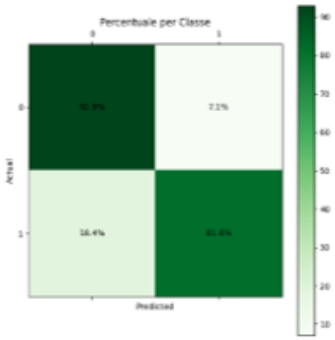
$$-(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

- Calcolata una loss indipendente per ogni task



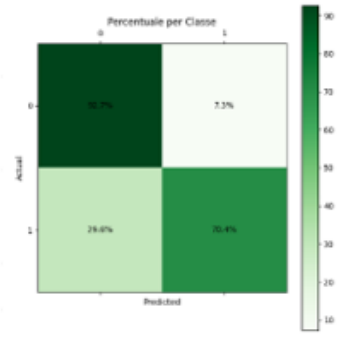
Alla fine dei 3 addestramenti sono state valutate, per ogni task, performance sul validation set come l'Accuracy, Precision, Recall e F-Score:

GENDER



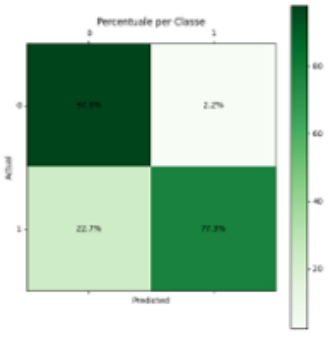
- Accuracy: 0.90
- Precision: 0.82
- Recall: 0.82
- F-Score: 0.83

BAG



- Accuracy: 0.88
- Precision: 0.69
- Recall: 0.70
- F-Score: 0.68

HAT



- Accuracy: 0.96
- Precision: 0.79
- Recall: 0.77
- F-Score: 0.81

Nel corso della prima fase di addestramento, è evidente che i pesi delle teste stiano iniziando a convergere. Tuttavia, osservando i risultati riportati:

1. **Hat:** L'accuracy è molto alta (0.96), ma il valore di F-Score (0.81) è più basso rispetto a quanto ci si aspetterebbe considerando la precisione e il recall. Questo

suggerisce che il modello sta imparando a predire "0" (assenza di hat) più frequentemente, sfruttando lo sbilanciamento parziale del dataset, dove la classe "0" è maggioritaria.

2. **Bag**: Anche qui l'accuracy (0.88) è alta, ma l'F-Score (0.68) è basso. La precisione (0.69) e il recall (0.70) mostrano che il modello fatica a identificare correttamente la classe "1" (presenza di bag), optando spesso per "0", seguendo una logica simile a quella riscontrata per la testa Hat.
3. **Gender**: In questo caso, il modello presenta una maggiore bilanciatura, con valori simili per accuracy (0.90), precisione (0.82), recall (0.82) e F-Score (0.83). Questo indica una capacità di generalizzazione migliore rispetto alle altre due teste.

I risultati per "Hat" e "Bag" riflettono un problema dovuto alla natura parzialmente sbilanciata del dataset. Questo porta il modello a preferire la predizione della classe dominante ("0"), consentendo all'F-Score di ottenere una valutazione più significativa delle prestazioni rispetto all'accuracy.

FASE 2: Fine-Tuning

Nella seconda fase del training, è stato effettuato un fine-tuning del modello, durante il quale sono stati aggiornati anche i pesi dell'ultimo blocco della backbone. Questo passaggio ha consentito di migliorare l'integrazione tra le caratteristiche apprese dal modello e i compiti specifici da eseguire.

Per ottimizzare ulteriormente le prestazioni, sono state adottate le seguenti configurazioni:

- Learning Rate ridotto: è stato scelto un valore di **1e-4** per il learning rate, al fine di evitare variazioni troppo brusche nei pesi durante l'ottimizzazione.
- Ottimizzatore: **AdamW**, che contribuisce a migliorare la generalizzazione del modello.
- Per adattare dinamicamente il learning rate durante il training, è stato utilizzato uno scheduler di tipo **ReduceLROnPlateau**, che riduce il learning rate quando la metrica osservata non migliora.
- Metrica osservata: Durante il fine-tuning, la metrica principale di valutazione è stata **l'F-score medio** sui tre task, che ha permesso di monitorare in modo bilanciato la performance del modello su ciascun task.
- Funzione di loss: Per risolvere i problemi evidenziati durante la prima fase è stata utilizzata una versione pesata della **BCELoss** (Binary Cross-Entropy Loss), con i pesi delle classi derivati dalla distribuzione delle etichette nei singoli task.

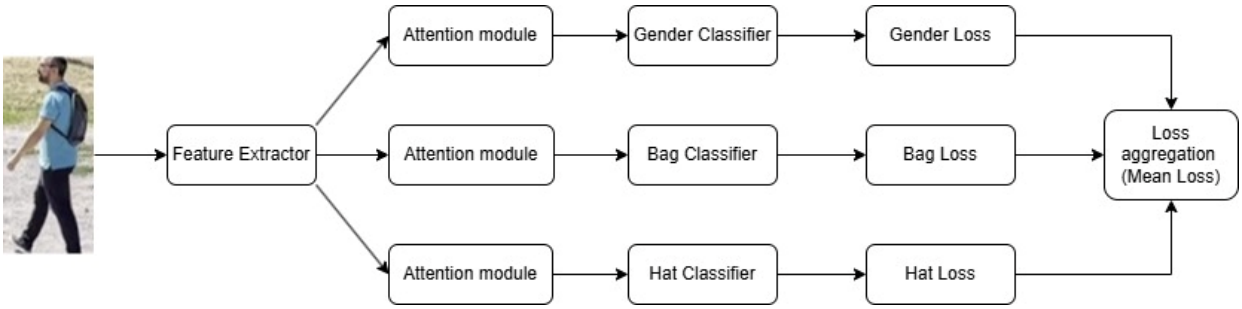
$$-\left[w_1 \cdot y \cdot \log(\hat{y}) + w_0 \cdot (1-y) \cdot \log(1-\hat{y})\right]$$

I pesi utilizzati per ogni classe sono stati i seguenti:

Gender: [3, 7] Has_bag: [3.2, 6.8] Has_hat: [1.7, 8.3]

Sono stati ricavati in base alla distribuzione delle classi: ad es. nel caso del gender si aveva il 70% di campioni 0 (maschi) e 30% di campioni 1 (femmine).
Quindi si è assegnato un peso 3 alla classe 0 e un peso 7 alla classe 1 (esattamente il reciproco della distribuzione). In questo modo le classi dominanti ricevono un peso minore rispetto a quelle meno presenti.

Le tre BCELoss sono state successivamente aggregate per procedere con il training ed effettuare l'operazione di backward nel modo corretto:

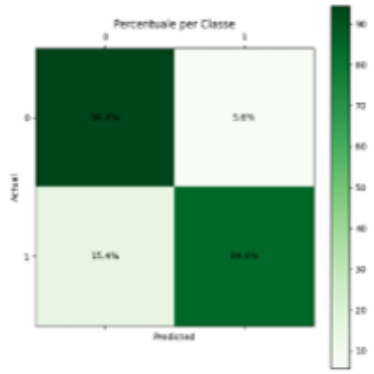


La loss aggregata è definita come la media delle loss individuali di ciascuna testa. Questo approccio ha garantito un bilanciamento tra i diversi compiti, evitando che uno di essi dominasse sugli altri durante l'ottimizzazione.

$$Total\ Loss = \frac{1}{N\_tasks} \sum_{i}^{N\_tasks} Loss_i$$

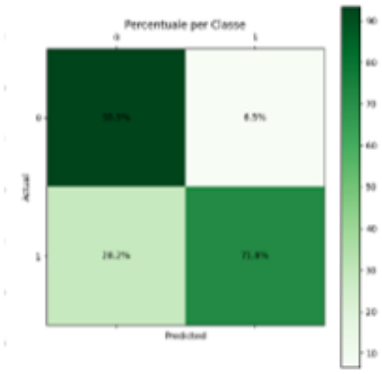
L'addestramento è durato 18 epoche con un picco massimo della metrica di valutazione (F-Score medio delle classi) sul validation set individuato nell'epoca 12.
I risultati ottenuti sono stati i seguenti:

GENDER



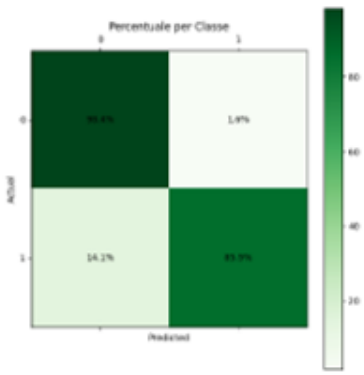
- Accuracy: 0.90
- Precision: 0.82
- Recall: 0.82
- F-Score: 0.83

BAG



- Accuracy: 0.89
- Precision: 0.72
- Recall: 0.72
- F-Score: 0.72

HAT



- Accuracy: 0.97
- Precision: 0.85
- Recall: 0.85
- F-Score: 0.85

Durante la seconda fase di addestramento, si sono osservati miglioramenti significativi nelle metriche di valutazione, in particolare per i task più problematici emersi nella prima fase. Ecco un'analisi dettagliata dei miglioramenti:

1. **GENDER:** Le metriche di precision, recall, F-score e accuracy rimangono stabili rispetto alla prima fase. Questo suggerisce che il modello era già ben bilanciato per questo task e ha mantenuto un'ottima capacità di classificazione. Non essendoci stato uno squilibrio evidente o un problema di performance, le modifiche apportate durante la seconda fase non hanno influenzato significativamente questa testa.

2. **BAG:** Il task Has_Bag ha mostrato il miglioramento più evidente:

   - **Precision** è passata da 0.69 a 0.72.
   - **Recall** è aumentata da 0.70 a 0.72.
   - **F-Score** è migliorato da 0.68 a 0.72.

   Questo incremento suggerisce che l'introduzione dei pesi durante l'addestramento ha permesso al modello di riconoscere meglio la classe con minor numero di campioni, riducendo l'impatto dello squilibrio presente nel dataset. Il modello è ora più equilibrato sia nella classificazione corretta della classe positiva (precision) che nella capacità di individuare tutte le istanze positive (recall), come confermato dall'aumento dell'F-Score.

3. **HAT:** Anche per il task Has_Hat si sono osservati miglioramenti:

- **Precision** è passata da 0.79 a 0.85.
- **Recall** è passata da 0.77 a 0.85.
- **F-Score** è migliorato da 0.80 a 0.85.

Questo dimostra che la testa dedicata a questo task è ora molto più efficace. Il miglioramento di precision e recall indica che il modello ha ridotto gli errori sia nelle predizioni false positive che false negative, rendendo la classificazione molto più accurata.

Utilizzo del modello PAR

L'utilizzo di YOLOv8 Pose nel sistema è stato scelto per garantire un tracciamento accurato delle persone nel video, con particolare attenzione a garantire che i risultati del modello PAR (Person Attribute Recognition) siano applicati solo ai frame in cui una persona è completamente visibile. Questa decisione è motivata da diverse ragioni:

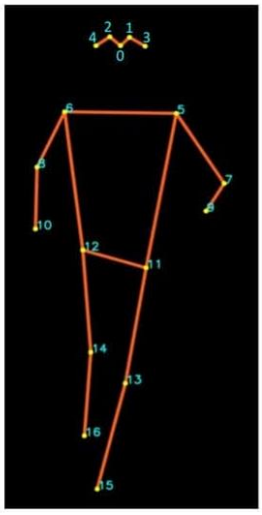
1. **Coerenza con il Dataset di Addestramento del Modello PAR**:

   Il modello PAR è stato addestrato su un dataset che comprende immagini di persone intere, senza occlusioni significative o sovrapposizioni. Per mantenere la coerenza tra i dati di addestramento e quelli di inferenza, è fondamentale escludere i frame in cui le persone non sono completamente visibili o appaiono parzialmente occluse.

2. **Eliminazione dei Tracciamenti Incompleti**:

   YOLOv8 Pose permette di identificare e tracciare l'intera persona nei frame. Utilizzando le coordinate dei bounding box e il numero di keypoint visibili, è possibile filtrare i tracciamenti in cui il soggetto appare incompleto.

Index	Key point
0	Nose
1	Left-eye
2	Right-eye
3	Left-ear
4	Right-ear
5	Left-shoulder
6	Right-shoulder
7	Left-elbow
8	Right-elbow
9	Left-wrist
10	Right-wrist
11	Left-hip
12	Right-hip
13	Left-knee
14	Right-knee
15	Left-ankle
16	Right-ankle



Source: https://learnopencv.com/wp-content/uploads/2021/05/fix-overlay-issue.jpg

Questo filtro previene l'inclusione di frame non validi nell'inferenza del modello PAR, migliorando così l'accuratezza delle analisi.

3. **Ottimizzazione della Qualità dei Risultati**:

   Attraverso il controllo del numero di keypoint visibili (ad esempio, accettando solo frame con più di 14 keypoint visibili, sui 17), il sistema garantisce che i frame utilizzati per l'inferenza del modello PAR rappresentino persone ben definite.

   Ciò riduce il rumore nei dati e migliora l'affidabilità delle previsioni di attributi come genere, presenza di borsa o cappello.

   Un'altra scelta implementativa è consistita nel filtrare i bounding box su cui fare inferenza in base all'iou (intersection over union) con gli altri bounding box presenti nella scena.

Questo esempio illustra una situazione in cui il sistema di rilevamento restituisce un risultato di una donna parzialmente coperta da una figura in primo piano. Analizzando il bounding box associato alla donna, si otterrebbe erroneamente un risultato PAR che descrive un uomo con cappello e zaino.

Conclusioni

Uno degli aspetti critici emersi riguarda la struttura del dataset, che ha presentato diverse limitazioni. Un'analisi approfondita ha rivelato che il dataset fosse principalmente composto da immagini scattate in ambienti esterni, mentre il nostro contesto di applicazione è indoor, con condizioni di illuminazione diverse.
Questa discrepanza ha influito sulla qualità e sull'affidabilità del modello, poiché le variazioni di luce possono alterare il riconoscimento. Inoltre, sono state riscontrate etichette errate e alcune immagini di bassa qualità, che risultavano poco significative per l'addestramento. Lo sbilanciamento delle classi e la presenza di numerose etichette mancanti hanno complicato ulteriormente la gestione delle classi.

Oltre alle soluzioni già descritte nei capitoli precedenti, il team ha esplorato ulteriori approcci per affrontare queste problematiche, come l'adozione di tecniche di dataset

augmentation, in particolare per i campioni relativi agli "hat" positivi, e l'integrazione del dataset con altre raccolte di dati pubblici specifici per la PAR.
Tuttavia, a causa delle limitazioni di tempo e risorse, non è stato possibile implementare completamente tutte queste soluzioni.

Per quanto riguarda il tracking invece, si è deciso di impiegare un matching threshold elevato, in modo da migliorare l'accuratezza del riconoscimento. Tuttavia, in alcune situazioni di occlusione tra persone, si è preferito assegnare un nuovo ID ad una stessa persona piuttosto che avere uno switch degli ID tra i bounding boxes che si sovrappongono, garantendo così una gestione più stabile e coerente del tracking.

Per quanto riguarda la PAR, sono emerse diverse problematiche legate alle ambiguità che influenzano negativamente l'accuratezza delle etichette. In alcune situazioni critiche, il modello commette errori di classificazione, come nel caso di un ragazzo con i capelli lunghi che viene erroneamente etichettato come femmina. Altre problematiche si verificano quando una donna che ha i capelli dello stesso colore della maglia viene scambiata per un maschio.
Il problema della sovrapposizione dei bounding-box è stato risolto effettuando un filtro basato sul numero di punti rilevati da YOLO-pose e su una soglia di iou limite, in questo modo il modello PAR riceve solo frame completi sul quale fornisce risultati più affidabili.
Un altro aspetto problematico riguarda il riconoscimento dei capelli: in presenza di texture difficili da identificare o di immagini di bassa qualità, il modello tende a interpretare erroneamente i capelli come un cappello, portando a ulteriori errori di classificazione.

Questi fattori evidenziano la necessità di miglioramenti sia nel dataset che nelle tecniche di riconoscimento per ridurre al minimo le ambiguità e migliorare la precisione complessiva del sistema.

Nonostante queste criticità, è importante sottolineare che il sistema ha comunque ottenuto buoni risultati nel complesso, con performance accettabili in vari scenari di test.

Per migliorare ulteriormente le performance, sarebbe necessario migliorare il dataset, risolvere le ambiguità nelle etichette e affinare le tecniche di riconoscimento, ma, a causa delle limitazioni di tempo e risorse computazionali, non è stato possibile eseguire tutte le ottimizzazioni desiderate.

Il modello si è rivelato efficace per la maggior parte dei casi, con un buon livello di generalizzazione, una gestione stabile del tracking e un funzionamento complessivamente efficace sulla PAR.