Action Classification Fight with OpenPose

Salvatore Arienzo
Orlando Napoli
Giovanni Cammarano
Giuseppe Emanuele Pezzillo
Università degli Studi di Salerno
{s.arienzo, g.cammarano15, o.napoli1, g.pezzillo1}@studenti.unisa.it

June 2, 2020

Abstract

L'obiettivo di questo paper, è quello di proporre una strategia di identificazione di situazioni di violenza, tramite skeleton detection. Attraverso un algoritmo di multi-pose estimation [1], viene stimata la posa delle persone presenti nella scena, le pose estratte vengono poi elaborate, e con i suddetti dati viene allenata una rete neurale che effettua la predizione. Un metodo automatico di fight detection troverebbe amplio utilizzo in molti sistemi, a partire da quelli relati alla sicurezza.

1 Introduzione

Negli ultimi anni sempre più aziende ed organizzazioni governative affidano vari aspetti di sorveglianza e prevenzione a sistemi che fanno uso di Intelligenza Artificiale. Lo sviluppo di nuove tecnologie di video analisi permette di automatizzare e rendere più efficace sistemi di prevenzione di furti ed altri atti criminali. Questi sistemi permettono di intervenire tempestivamente poiché sono in grado di capire da movimenti particolari compiuti da persone nella folla se questi sono in procinto di compiere azioni criminose. Aspetti come andatura o postura possono essere considerati come biometriche per la rilevazioni di scontri anche in contesti affollati (e dunque ad alto rischio) come stadi o aereoporti.

2 Lavori Correlati

Al fine di raggiungere l'obiettivo prefissato, è stata utilizzata l'implementazione di OpenPose [1] proposta da Michal Faber, che utilizza una Rete Convoluzionale MobileNetV2[2] per allenare un algoritmo di Multi-Pose Estimation. Per allenare la rete neurale, sono stati utilizzati i dati contenuti nel Real Life Violence Situations Dataset[3] di Mohamed Elesawy. Il dataset, al suo interno, contiene esattamente 2000 video divisi in due categorie, 1000 per la categoria Non Violence ed altrettanti per la cat-

egoria Violence. I video sono della durata approssimativa di 4 secondi e rappresentano scene di violenza urbana, di risse, e di simulazioni di situazioni di violenza per quanto riguarda la parte di Violence, mentre rappresentano normali scene di vita giornaliera, partite di calcio, o film, nella parte di Non Violence.

3 Metodo Utilizzato

Per raggiungere l'obiettivo, sono stati estratti i frame dai video presenti nel dataset ed è stata poi effettuata una stima della posa su ogni singolo frame. Successivamente, sono state calcolate le distanze tra le parti del corpo maggiormente interessate in una eventuale violenza. Infine, i dati ricavati sono stati dati in input ad una rete neurale per la classificazione.

3.1 Preprocessing

Il dataset iniziale è stato ridotto in 50 video per la categoria Violence e 50 per quella Non Violence, questa decisione è stata presa sulla base della necessità di ridurre i tempi di elaborazione, date le limitate capacità computazionali a disposizione. Dall'estrazione dei frame dai video, sono stati ricavati circa 15 mila immagini, normalizzando tutti i video alla stessa risoluzione.

3.2 OpenPose

Dopo aver estratto i frame, questi sono stati dati in input all'algoritmo di Multi-Pose Estimation che ne ha calcolato i keypoints. OpenPose prevede un totale di 18 keypoints di cui:

- 17 per indicare le parti del corpo rilevate
- 1 per identificare lo sfondo

E' stato controllato che OpenPose vedesse effettivamente due persone nell'immagine, solo in questo caso il frame viene salvato nel dataset. E' stato infatti verificato che OpenPose non rileva sempre tutte le parti del corpo. Risulta infatti poco accurato in alcuni scenari di utilizzo, come in situazioni in cui ci sono movimenti veloci, i corpi non sono interi o sono attaccati (cosa molto frequente, per via del combattimento). L'output di OpenPose consiste in una lista di valori indicanti per ogni persona:

- Nome della parte del corpo
- Coorinata X
- Coordinata Y

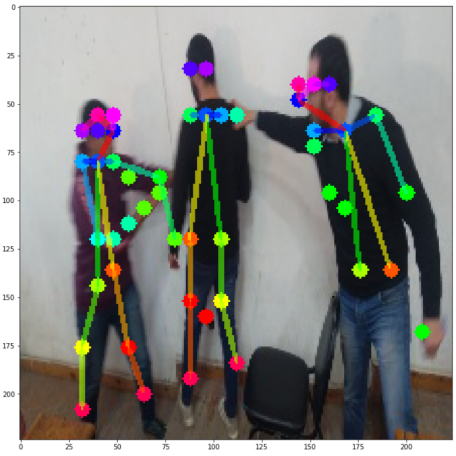

Figura 1: Output grafico di OpenPose.

3.3 Dataset

Uno dei motivi per cui abbiamo scelto di utilizzare video contenenti solo due persone riguarda la costituzione del

dataset: tra le features utilizzate troviamo infatti la distanza tra alcune parti del corpo delle persone rilevate ritenute significative Ne consegue che, se fossero presenti più persone, sarebbero necessarie, non solo le connessioni da Persona 1 a Persona 2, ma anche tutte le varie combinazioni che si genererebbero, ad esempio Persona 1 - Persona 3, Persona 2 - Persona 3 e cosi via, aumentando o diminuendo di volta in volta il numero di features in maniera esponenziale in base alle persone presenti nel frame. Per il calcolo della distanza tra le parti del corpo viene utilizzata una funzione per calcolare la distanza euclidea tra le parti del corpo che abbiamo ritenuto significative. La distanza tra le parti del corpo considerate significative, sono ad esempio quelle maggiormente coinvolte durante azioni di violenza, alcuni esempi potrebero essere il naso o la faccia in generale, o le articolazioni, a causa di pugni o calci. Alcuni esempi di distanze calcolate sono quindi: la distanza Polso-Naso o Ginocchio-Naso, ovviamente tutte le distanze vengono calcolate a coppie, per entrambe le persone, ad esempio Polso Sinistro della Persona 1 – Naso Persona 2, Polso Destro Persona 1 – Naso Persona 2, Polso Sinistro della Persona 2 – Naso Persona 1, Polso Destro Persona 2 – Naso Persona 1. Abbiamo anche previsto il calcolo della distanza tra collo ed anca della stessa persona, in modo da valutare quanto questa persona sia vicina o lontana dalla telecamera.

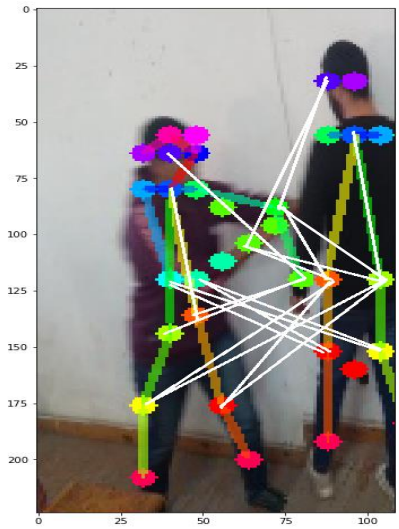

Figura 2: Distanze prese in considerazione.

Il dataset sarà quindi composto dalle seguenti features:

- Id del frame.
- Coorinata X ed Y per ogni parte del corpo, di ogni persona.
- Alcune distanze tra le parti del corpo delle due persone.
- Distanze tra le parti del corpo della stessa persona.
- Un valore booleano.

Essendo i keypoints localizzati da OpenPose 18, troveremo nel dataset 18 Coordinate X e 18 Coordinate Y per ogni persona, per un totale di 72 features dedicate alla posizione di tali parti. Alle distanze, sono invece dedicate un totale di 26 features di cui:

- 24 per le distanze tra parti del corpo di due persone diverse.
- Coorinata 2 per le distanze tra l'anca e il collo di entrambi i soggetti.

In ultimo, viene salvato un valore booleano per indicare la classe del frame. Si avranno quindi in totale, 100 features.

4 Esperimenti

Per esegurie il task di classificazione, si è scelto di utilizzare una rete neurale. Tutti i dati ricavati finora sono stati quindi salvati in un file csv e dati in input alla suddetta rete per l'addestramento. Le prime 99 features del dataset vengono utilizzate per la predizione, mentre una sola feature (il valore booleano) è la feature predetta. Il 70% del dataset è stato utilizzato per il training set e la restante parte è stata utilizzata in ugual misura per test e validation set. Dopo numerosi test effettuati, si è scelto di utilizzare una rete neurale con 4 layers, e con soli 32 neuroni: avendo utilizzato una rete relativamente "piccola" sono state applicate delle tecniche di regolarizzazione[3] come il Dropout[4], per ridurre l'overfitting.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 32)	3168
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 8)	136
dropout_3 (Dropout)	(None, 8)	0
dense_4 (Dense)	(None, 1)	9

Figura 3: La struttura della rete neurale utilizzata.

5 Risultati

Dopo aver effettuato vari test sulla rete neurale, aggiungendo o rimuovendo layers, testando le varie funzioni di attivazione e migliorando i vari iperparametri, la migliore combinazione è risultata l'utilizzo della funzione di attivazione ReLU [5] per il primo layer, Tanh [6] per il secondo e terzo, mentre per il layer di output è stata utilizzata una funzione Sigmoid[7]. Il modello ha ottenuto un accuracy dell'82% dopo un training di 15 epoche.

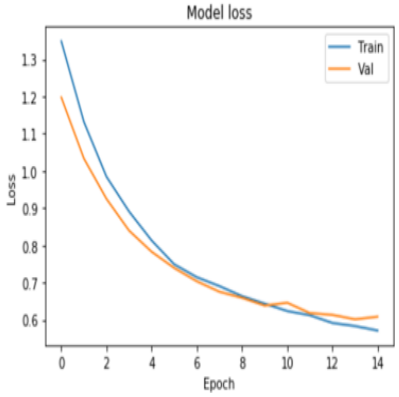

Figura 4: Grafico di loss per il modello descritto.

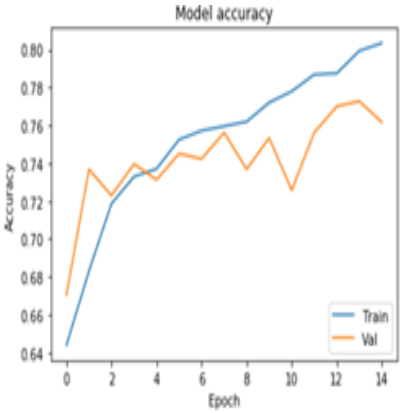

Figura 5: Grafico di accuracy per il modello descritto.

Dopo aver effettuato la classificazione su ogni frame del video, se nel suddetto video almeno il 20% dei frame risulta categorizzato come Violento, allora l'intero video viene categorizzato come Violento, Non Violento altrimenti.

Il modello proposto riesce quindi ad eseguire il task richiesto con una buona percentuale di accuracy. Alcune migliorie sono tuttavi possibili: appurati i limiti di OpenPose in situazioni di vicinanza dei corpi, sarebbe interessante testare altri algoritmi di skeleton detection oppure adattare i parametri di OpenPose stesso per svolgere al meglio questo task. Un altra possibile migliorìa riguarda i dati: avere dei dati studiati appositamente per la skeleton detection migliorerebbe molto le fasi di training. Nei dati utilizzati infatti, molti frames venivano scartati perché OpenPose non visualizzava due persone anche se queste erano presenti: questo è dovuto al fatto che il dataset [3] presenta situazioni reali di scene di violenza, in cui le inquadrature non sono ottimali, e spesso i corpi escono fuori dall'obiettivo della telecamera, rendendo difficile l'individuazione per OpenPose. Di seguito mostriamo alcuni frame correttamente categorizzati[9].

Prediction: Image 6 - Predicted Label: Violence

Prediction: Image 10 - Predicted Label: Non Violence

Referenze

[**1**] Multi-Person Pose Estimation project for Tensorflow 2.0 with small and fast model based on MobilenetV2 - https://github.com/michalfaber/tensorflow_Realtime_Multi-Person_Pose_Estimation

[**2**] Mobilenet - https://keras.io/api/applications/mobilenet/

[**3**] Real Life Violence Situations Dataset - https://www.kaggle.com/mohamedmustafa/real-life-violence-situations-dataset

[**4**] Layer weight regularizers - https://keras.io/api/layers/regularizers/

[**5**] Dropout Neural Network Layer In Keras - https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab

[**6**] Introduction to the Rectified Linear Unit (ReLU) - https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[**7**] Tanh function - https://theclevermachine.wordpress.com/tag/tanh-function/

[**8**] The Sigmoid Activation Function - https://www.allaboutcircuits.com/technical-articles/sigmoid-activation-function-activation-in-a-multilayer-perceptron-neural-network/

[**9**] Demo Examples - https://github.com/salvatore-arienzo/Action_Classification_with_OpenPose_Violence/tree/master/demo