

**UNIVERSITÀ DEGLI STUDI DI BARI   
“ALDO MORO”**

Dipartimento di Informatica

Corso di Laurea in Informatica

*Tesi di Laurea in Modelli e Metodi per la Sicurezza delle applicazioni*

**RICONOSCIMENTO DI ANOMALIE**

**DA TELECAMERE DI SORVEGLIANZA**

*Relatore*:  
Prof. Donato Impedovo

*Laureando/a*:  
Marco Cappiello

Anno Accademico 2021/2022

**Abstract**

In questo lavoro di tesi si è cercato di identificare anomalie da video catturati da telecamere di sorveglianza utilizzando tecniche di apprendimento debolmente supervisionato.

Il problema viene affrontato come un problema di regressione, dividendo ogni video presentato al sistema in 32 segmenti temporali a cui vengono assegnati punteggi rappresentanti la probabilità che ci sia un’anomalia.

Il sistema proposto utilizza features estratte attraverso una convolutional neural network con kernel di tre dimensioni che rappresentano caratteristiche spazio-temporali dei video.

La sperimentazione è stata effettuata utilizzando due diversi dataset, l’UCF CRIME e il Real life Violence situations, raggiungendo un AUC del 70% e dell’80%.

**Ringraziamenti**

Eventuali ringraziamenti, se proprio si vuole ringraziare qualcuno…

**Indice**

Sommario

[Capitolo 1 – Introduzione 9](#_Toc120177857)

[Capitolo 2 – Stato dell’arte 11](#_Toc120177858)

[Capitolo 3 – Dataset 17](#_Toc120177859)

[Capitolo 4 – Sperimentazione 27](#_Toc120177860)

[4.1 Features Extraction 28](#_Toc120177861)

[4.1.1 Convolutional 2D Neural Network 29](#_Toc120177862)

[4.1.2 Convolutional 3D Neural Network 32](#_Toc120177863)

[4.1.3 Architettura Rete C3DNN 33](#_Toc120177864)

[4.2 Segmentazione dei video e normalizzazione dei segmenti 33](#_Toc120177865)

[4.3 Training 35](#_Toc120177866)

[4.3.1 Multiple Instance Learning 35](#_Toc120177867)

[4.3.2 Funzione obbiettivo 36](#_Toc120177868)

[4.3.3 Architettura Rete 38](#_Toc120177869)

[4.3.4 Dettagli implementazione 39](#_Toc120177870)

[4.4 Inferenza 40](#_Toc120177871)

[Capitolo 5 – Risultati 41](#_Toc120177872)

[5.1 Prima sperimentazione 43](#_Toc120177873)

[5.2 Seconda sperimentazione 44](#_Toc120177874)

[5.3 Terza sperimentazione 45](#_Toc120177875)

[Capitolo 6 – Conclusioni 47](#_Toc120177876)

[Capitolo 7 – Sviluppi Futuri 49](#_Toc120177877)

# Capitolo 1 – Introduzione

L’avanzamento tecnologico ha automatizzato e migliorato molti aspetti della vita dell’uomo, due delle invenzioni più affascinanti e innovative degli ultimi due secoli sono state indubbiamente la macchina fotografica e la telecamera, entrambe permettono all’uomo di salvare un’immagine o una sequenza di immagini per il futuro.

Ovviamente come ogni scoperta queste due invenzioni portarono a loro volta delle innovazioni in molteplici aspetti della vita dell’uomo, e presto la nostra società ha visto nascere il cinema, la televisione e più recentemente i social network. Oltre a una rivoluzione nel mondo dell’intrattenimento le telecamere sono state artefici di una rivoluzione nell’ambito della sicurezza: al giorno d’oggi, infatti, è molto comune l’utilizzo di telecamere in aeroporti, uffici, negozi e luoghi pubblici per prevenire o individuare crimini.

I vantaggi dell’utilizzo di telecamere nell’ambito della sicurezza e della sorveglianza sono evidenti, questa rivoluzione tecnologica ha infatti portato alla nascita di un nuovo modo di fare sorveglianza, non vi è infatti più bisogno di un grosso numero di personale addestrato che deve passare ore sul posto a sorvegliare, il tutto è completamente decentralizzato, più telecamere possono sorvegliare registrare diverse località e i video risultanti possono essere consultate comodamente da una postazione centrale.

D’altra parte, il dispiego di un alto numero di telecamere generano un’enorme mole di dati che devono comunque essere supervisionate e controllate in tempo reale da personale qualificato, ciò comporta comunque un alto dispendio di risorse umane e un costo economico consistente, oltre a questo problema stiamo comunque affidando la sicurezza di persone, proprietà e beni a degli uomini e quindi non bisogna sottovalutare l’errore umano dovuto alla stanchezza o alla distrazione.

Al momento esistono in commercio telecamere capaci di rilevare il movimento o addirittura di riconoscere un viso ma non esiste nessun sistema capace di capire capire autonomamente cosa sta accadendo nel filmato.

A tal proposito possiamo sfruttare gli avanzamenti nel campo del machine learning, che si pone l’obbiettivo di creare sistemi capaci di apprendere e ragionare, e nell’ambito della computer vision, che si pone come obbiettivo quello di segmentare immagini, rilevare azioni e identificare individui, per cercare di automatizzare questo lavoro e cercare di creare un sistema di sorveglianza affidabile e con un basso livello di falsi allarme (per minimizzare per l’appunto l’intervento umano).

In questo lavoro di tesi viene quindi proposto un metodo per individuare automaticamente anomalie in un video registrato da telecamere di sorveglianza cercando di minimizzare il numero di falsi negativi e quindi il numero di falsi allarmi, azioni normali segnalate come anomale.

Per prima cosa bisogna capire cos’è un’anomalia: le anomalie sono azioni infrequenti che vengono effettuate pubblicamente.

Questa affermazione evidenza un aspetto fondamentali da affrontare per poter risolvere questo problema: un’azione è anomala se effettuata in un contesto in cui l’azione è infrequente, per esempio, correre in un parco non è un’azione anomala mentre correre sull’autostrada lo è, così come fare a botte su un ring non è anomalo mentre farlo per strada lo è.

Proprio per questo motivo, utilizzando un classico e semplicissimo classificatore dell’apprendimento supervisionato, sarebbe necessario avere a disposizione un numero elevato di esempi di azioni anomale e azioni normali in un altrettanto elevato numero di contesti diversi, oltretutto questi esempi dovrebbero essere ben catalogati temporalmente e spazialmente.

La difficoltà nel reperire tale dataset ha spinto all’utilizzo di un modello d’apprendimento debolmente supervisionato, il Multiple Instance Learning, inoltre, in questo lavoro di tesi viene considerato il problema del rilevamento di anomalie da telecamera di sorveglianza un problema di regressione a differenza dei più comuni modelli analizzati nel capitolo dello studio dello stato dell’arte che lo affrontano come problema di classificazione.

Ogni video presentato al nostro sistema verrà diviso in 32 segmenti temporali di lunghezza variabile e verrà assegnato un punteggio rappresentante la probabilità che il segmento sia anomalo a ciascuno di questi segmenti.

Un video sarà quindi anomalo se almeno uno dei segmenti del video sarà anomalo, ovvero se il punteggio assegnato dal nostro sistema a uno dei segmenti di cui è diviso il video supererà un treshold variabile da video in video, alternativamente se ogni segmento è non anomalo il video è non anomalo.

Una volta rilevato un video anomalo si potrà agire come opportuno segnalando la clip ad un agente di polizia o a un sorvegliante.

# Capitolo 2 – Stato dell’arte

Zaidi, Jagadeesh et al.[1] propongono un metodo semplice ma efficace per individuare anomalie da un *video stream*, una telecamera di sorveglianza che raccoglie video solo in presenza di movimento che poi verranno pre-processati e mandati al sistema costruito.

Per prima cosa i video raccolti vengono convertiti in formato AVI, successivamente i video vengono ridimensionati ad un’altezza di 240 pixel e una larghezza di 320 pixel (320x240 px) e convertiti in frames. Successivamente verrà applicata la *background subtraction*, metodo che riesce a rilevare oggetti in rilievo (*foreground object*) in un video attraverso l’utilizzo di un frame di riferimento che per l’appunto rappresenta lo sfondo del video.

L’applicazione della *background subtraction* servirà ovviamente per identificare l’oggetto in movimento in un video, per attuare questa operazione è necessario avere un frame di riferimento e senza oggetti in movimento per effettuare il confronto, il risultato della background subtraction possono essere sia oggetti in movimento, animali o nel caso di questo paper figure umane utili per la classificazione.

Successivamente viene applicato un filtro di Kalman, filtro che ha molteplici applicazioni, in questo caso viene utilizzato per tener traccia della locazione futura degli oggetti rilevati, per diminuire il rumore nelle immagini e per associare più oggetti rilevati.

Per quanto riguarda la classificazione dei video in anomali o normali viene effettuata la seguente distinzione: i video in cui una persona si nasconde, striscia per terra, si gira di scatto, si muove velocemente dietro una persona o cerca di uscire dall’inquadratura in modo innaturale vengono segnati come anomali, i restanti video sono invece normali e quindi non anomali.

La classificazione dei video ricavati dal filtro di Kalman viene effettuata da una Support Vector Machine, ovvero una tecnica di apprendimento supervisionato che rappresenta gli esempi di training come punti nello spazio e si pone come obbiettivo quello di trovare un iperpiano (*hyperplane*) che divida gli esempi, in questo caso i video, in due classi, in questo caso normali o anormali come mostrato in figura.

Nandini, Mathivanan et al.[2] propongono un metodo per identificare individui da video di sorveglianza e identificare le azioni svolte in ambiente scolastico, e quindi con un database formato da visi conosciuti da diverse angolazioni e azioni sospette con cui effettuare il confronto.

Per prima cosa i video registrati da una telecamera vengono convertiti in frames, successivamente i frame vengono convertiti in *grayscale*, ovvero in diverse scale di grigio, e successivamente viene applicato il gaussian blur, ciò è necessario per diminuire il rumore nelle immagini, e infine viene applicata la canny edge detection per identificare gli oggetti in rilievo.

I frame risultanti da questo metodo vengono successivamente utilizzati per effettuare il confronto con i visi e le azioni nel database e allertare le autorità in caso di azioni sospette.

Sultani, Chen et al.[3] vedono il problema del riconoscimento di anomalie come un problema di regressione, viene posto come obbiettivo quello di riuscire a dare un punteggio alto ai frame in cui è presente un’anomalia e un punteggio basso ai frame normali.

Vengono estratte features dai video attraverso una convolutional neural network con kernel di tre dimensioni e viene utilizzato *multiple instance learning* per addestrare un modello capace di assegnare punteggi di anomalia ai video. Il modello costruito raggiunge un AUC del 75.4%.

Nale, Sawarbandhe et al.[4] utilizzano le relazioni geometriche tra le articolazioni per identificare l’attività umana tramite una combinazione di Pose Estimation e LSTM.

Dai video vengono identificate 25 articolazioni sulla figura umana, di queste per minimizzare la complessità dell’algoritmo vengono selezionate quelle relative alla testa, alle mani e alle gambe, successivamente vengono identificate 39 rette che passano per le articolazioni che rappresentano mani, gambe, spalle, fianchi, piedi, testa e spina dorsale, in fine vengono selezionati 5 piani: mani, gambe e testa.

Come già accennato la classificazione avviene attraverso l’uso di un LSTM, ovvero una variante di una rete neurale ricorrente capace di memorizzare lunghe sequenze di dati e la loro variazione nel tempo.

Il dataset utilizzato è il NTU RGB + D action recognition dataset formato da video RGB con una risoluzione di 1920 x 1080 pixel, mappe di profondità e video IR di una risoluzione di 512 x 424 pixel e include i dati scheletrici 3D in ogni fotogramma, viene raggiunta un accuracy del 74 %.

Babiker et al. [5] utilizzano una telecamera che inquadra un punto fisso, quindi identificatno il background del video come il primo frame presente nella registrazione, a questo punto per ottenere la figura umana basterà effettuare la background subtraction e successivamente per minimizzare la varianza intraclasse verrà applicato il filtro Otsu.

L’immagine risultante da questa prima fase di preprocessing sarà una figura umana in bianco e il background in nero. I valori delle features dei pixel bianchi nell’immagine vengono calcolate attraverso la blob analysis, le features in questione sono l’ordinata, l’ascissa, l’altezza e la lunghezza del blob, oltre a queste quattro informazioni vengono memorizzate anche il centroide del blob e l’area del rettangolo che lo contorna.

Il dataset usato è composto da 1000 esempi divisi in 5 classi di azioni (boxing, salutare, camminare, sedersi e lo stendersi) registrate in ambiente controllato e quindi con la stessa inquadratura, di cui il 70% usate per il training, il 15% per la validazione e un altro 15% per il test. La classificazione in questo caso viene effettuata da una multylayer neural network ottenendo un accuracy del 94%.

Salem et al. [6] usano un approccio originale per affrontare questo task, utilizzano video in formato MOV a 23 frame per secondo, il primo frame viene usato come frame di riferimento per effettuare la background subtraction.

L’immagine ottenuta da questa operazione viene poi riprocessata con operazioni morfologiche di apertura, chiusura e riapertura per eliminare il rumore nell’immagine. In questo approccio si semplifica il problema distinguendo tra attività sospette (correre) e attività non sospette(camminare).

Inoltre, questo approccio non utilizza nessuna tecnica del machine learning, viene fatta una distinzione tra oggetti che si avvicinano o allontanano dalla telecamera e oggetti che si muovono da sinistra verso destra (o destra verso sinistra).

Nel primo caso l’anomalia viene identificata attraverso il cambio di dimensione dovuto alla prospettiva dell’oggetto tra un frame all’altro, nel secondo caso invece viene identificata grazie alla velocità di spostamento dell’oggetto, e quindi in base alla velocità del cambiamento della coordinata X.

Questo approccio è estremamente semplicistico e quindi non adeguato ai nostri scopi.

Sivasakthi et al[7], invece utilizza una teacheable machine, algoritmo offerto da Google per task di identificazione di azioni in un video, e video di sorveglianza in tempo reale per identificare anomalie. Viene utilizzato per il confronto un dataset appositamente costruito contenente azioni come il combattere, lo stare fermi in piedi, il camminare o l’impugnare una pistola, in questo caso la classificazione delle azioni avviene in base a features scheletriche.

Questo approccio ha però il non sorvolabile difetto di dipendere da riprese di singole persone e con un’inquadratura frontale a mezzo busto.

Amrutha et al. [8] propongono un sistema di rilevamento di attività sospette in un campus attraverso registrazioni da un sistema a circuito chiuso che verranno convertite in frames. Il sistema è stato addestrato utilizzando una combinazione del dataset KTH, composto da 6 classi di azioni normali e 100 video per classe, e il dataset CAVIAR per quanto riguarda le azioni sospette. Viene utilizzata una convolutional 2d neural network addestrata sul dataset Imagenet, la VGG-16, per estrarre le features dai video, e una LSTM per effettuare la classificazione.

In questo caso la classificazione viene effettuata sui singoli frame e non a livello di video, inoltre essendo una rete specializzata per il rilevare anomalie all’interno di un campus identifica azioni come l’utilizzare il telefono o il correre come sospette, per questo motivo non risulta particolarmente utile per un utilizzo più generico. Il modello costruito in questo approccio raggiunge un accuracy dell’87%.

Bordoloi et al. [9] ,invece, utilizzano un dataset composto da 15 azioni divise in 3 classi (furto, intrusione e borseggio) registrati in contesti diversi e l’algoritmo YOLOv3 (*you only look once*).

In questo caso ogni video del dataset è stato annotato con coordinate e azione presente nel video e successivamente si è usata una Convolutional Neural Network formata da 53 layer di convoluzione per estrarre features spazio temporali dalle clip. I risultati ottenuti da questo metodo nonostante le dimensioni ridotte del dataset sono molto buone, viene raggiunto un’accuracy del 95%.

Nell’ approccio proposto da Kranthi Kumar et al. [10] viene preso come input un video che successivamente verrà diviso in frames, dopodiché verrà effettuata la background extraction e la foreground extraction per identificare oggetti che si muovono tra frames.

Anche in questo caso l’estrazione delle features viene effettuata attraverso una convolutional Neural Network mentre la classificazione viene effettuata tramite una discriminative Deep Belief network preaddestrata. Il sistema costruito in questo modo ha il grosso vantaggio di riuscire ad elaborare video in tempo reale

L’ accuracy del sistema costruito in questo modo è del 74%.

Liu, Ying et al.[11] affrontano il problema del riconoscere azioni anomale in due modi, utilizzando una Convolutional Neural Network e un classificatore Bayesiano.

Per quanto riguarda il classificatore Bayesiano anche in questo caso si ha utilizzato il filtro di Kalman per riconoscere e segmentare le figure umane in movimento tra i frame, mentre le features utilizzate per la classificazione sono la length-width ratio, ovvero il rapporto tra la larghezza e l’altezza del segmento estratto, il valore di entropia del segmento e l’*hu5 invariant moment* ovvero il valore medio dell’intensità dei features che compongono l’immagine segmentata.

Il dataset utilizzato è il già citato KTH dataset e viene raggiunto un’accuracy media del 93% usando il classificatore Bayesiano e del 97% usando una convolutional 3d neural network.

Le azioni classificate in questo caso sono il correre, il camminare, l’inciampare e il dare pugni a vuoto registrati in contesti controllati, per questo motivo il sistema non è abbastanza robusto per essere utilizzato in un contesto reale.

Dileep et al.[12] affrontano il problema creando un proprio dataset distinguendo tra azioni come l’intrusione e il cadere.

Per prima cosa dividono i video in frames che successivamente vengono convertiti in BGR e a cui verrà cambiata la risoluzione in 200x200 px. Successivamente viene utilizzata la pose estimation, e nello specifico Mediapipe, un framework capace di identificare la figura umana in un video e nello specifico di estimare le coordinate X e Y dei punti focali della figura umana (testa, spalle, gomiti, ginocchia, etc.).

In questo caso la classificazione viene invece effettuata attraverso una convolutional neural network di due dimensioni formata da due layer di convoluzione. Non viene indicata alcuna una metrica di valutazione, il sistema riesce però ad effettuare classificazioni in tempo reale.

Anche Vrskova et al. [13] propongono un nuovo dataset, in questo caso le azioni classificate sono rapina, lotta, rapimento, abuso e azioni normali, raggiungendo in totale 469 video.

La rete utilizza una convolutional long short term memory neural network (CONVLSTM), rete capace di identificare le caratteristiche spaziotemporali di una sequenza di frames.

Il sistema costruito in questo modo raggiunge un’accuracy del 97.64%.

Inoltre, è importante notare che il sistema ha avuto maggiori difficoltà nel classificare azioni provenienti dalla classe “abuso”, questo è dovuto alle numerose interpretazioni di quest’ultima presenti nel dataset.

Iqbal et al. [14] usano una Thermal camera per creare un dataset di azioni sospette che successivamente verrà convertito in frames su cui verrà applicata la background subtraction e successivamente varie morphological post processing operations.

Le azioni umane hanno un rateo di 0.92 a 0.2, quest’informazione ci permette di distinguere gli esseri umani dagli oggetti nell’immagine foreground ottenuta. Tra le varie operazioni morfologiche c’è la “bwmorph” che individua lo scheletro umano e nello specifico le varie giunture e le terminazioni.

La classificazione delle azioni verrà attuata da una rete neurale utilizzando le feature trovate nella fase precedente. L’accuracy del sistema proposto è del 90%.

Atikuzzaman et al. [15] invece affrontano il problema della classificazione dell’azione in un video, propongono un dataset composto da 5648 immagini divise in 5 classi (corsa, camminata, stare in piedi, stare seduti e stare sdraiati).

Per prima cosa i frame vengono convertiti in grayscale, successivamente viene usata una rete neurale per identificare la figura umana in un frame. Il risultato di questa rete saranno le coordinate X e Y del punto in cui è stata rilevata la figura umana e la larghezza e l’altezza della sagoma rilevata.

Dopo aver identificato la sagoma umana l’immagine iniziale viene ritagliata e ne viene cambiata la risoluzione a 64x64 px per poi utilizzare una convolutional neural network per classificare l’azione.

Il sistema in questo caso raggiunge un accuracy del 99.86%, ma non risulta utile ai nostri scopi in quanto classifica azioni molto distinte tra loro in singole immagini.

Nell’approccio proposto da Manaf et al.[16] si usa una una 2D-CNN, formato da sei convolutional layers e 3 max pooling layers, per estrarre le features dai frame di cui è composto un video, successivamente si utilizzano SPDB-LSTM (Stacked parallel Bi dimensional LSTM) per apprendere le relazioni temporali tra le features estratte.

Il modello proposto è stato testato su due dataset, lo SPHAR, raggiungendo un’accuracy del 94.55 e l’UCF-101, raggiungendo un’accuracy e del 94.5%

Arunnehru et al. [17] introducono l’utilizzo di un kernel di tre dimensioni per classificare una sequenza di frame, effettuano sperimentazioni su due dataset, il KTH raggiungendo un’accuracy del 94.9 e il Weizman raggiungendo un accuracy del 97.2.

La rete utilizzata in quest’ultimo approccio è formata da due convolutional layer seguiti da layer di max-pooling, in questo caso la rete necessiterà di una sequenza di frame e l’operazione matematica di convoluzione verrà effettuata con un kernel tridimensionale, l’utilizzo di questa architettura permette di creare una features map contenente features provenienti da frame intrinsecamente collegati tra di loro riuscendo a rappresentare le caratteristiche spazio temporali di un video.

# Capitolo 3 – Dataset

Per questo lavoro di tesi si è scelto di utilizzare due dataset, l’UCF crime dell’università della Florida Centrale e il Real Life Violence Situations Dataset, il primo è composto da 950 video anomali divisi in 13 categorie (abuso, arresto, incendio, assalto, scasso, esplosione, combattimento, incidente stradale, rapina, sparatoria, taccheggio, furto e vandalismo) e 950 video normali mentre il secondo è composto da 1000 video contenenti violenza di diverso tipo e 1000 video contenenti azioni normali.

|  |  |
| --- | --- |
| **Categorie UCF CRIME** | **Numero di video per categoria** |
| Abuse | 50 |
| Arrest | 50 |
| Arson | 50 |
| Assault | 50 |
| Burglary | 100 |
| Explosion | 50 |
| Fighting | 50 |
| Normal Videos | 950 |
| Road Accidents | 150 |
| Robbery | 150 |
| Shooting | 50 |
| Shoplifting | 50 |
| Stealing | 100 |
| Vandalism | 50 |
| #Totale | 1900 |

|  |  |
| --- | --- |
| **Categorie RLVSD** | **Numero di video per categoria** |
| Normal Videos | 1000 |
| Violence | 1000 |
| #Totale | 2000 |

Il modello utilizzato in questo progetto necessità di confrontare, in fase d’addestramento, un video anomalo con un video normale, oltre a questi sono necessari dei video per valutare il sistema, per questo motivo si ha creato tre partizioni dei due dataset. L’UCF crime è stato diviso nel seguente modo:

* 810 video anomali come training set di video anomali;
* 810 video normali come training set di video normali;
* 280 video (140 anomali+140 normali) come test set.

|  |  |  |
| --- | --- | --- |
| **Categoria** | **Numero di video nel train set** | **Numero di video nel test set** |
| Abuse | 48 | 2 |
| Arrest | 45 | 5 |
| Arson | 41 | 9 |
| Assault | 47 | 3 |
| Burglary | 87 | 13 |
| Explosion | 29 | 21 |
| Fighting | 45 | 5 |
| Normal Videos | 810 | 140 |
| Road Accidents | 127 | 23 |
| Robbery | 145 | 5 |
| Shooting | 27 | 23 |
| Shoplifting | 29 | 21 |
| Stealing | 95 | 5 |
| Vandalism | 45 | 5 |
| #Totale | 1620 | 280 |

L’RLVSD invece è stato diviso nelle seguenti tre partizioni:

* 700 video anomali come training set di video anomali;
* 700 video normali come training set di video normali;
* 600 video (300 anomali+300 normali) come test set.

|  |  |  |
| --- | --- | --- |
| **Categorie RLVSD** | **Numero di video nel train set** | **Numero di video nel test set** |
| Normal Videos | 700 | 300 |
| Violence | 700 | 300 |
| #Totale | 1400 | 600 |

L’UCF crime presenta alcune peculiarità che lo hanno reso un dataset particolarmente problematico nell’ambito del machine learning, infatti ogni video è stato raccolto da vere telecamere di sorveglianze sparse per il mondo e riprende crimini o scene di vita quotidiana di un ambiente realistico e non controllato a differenza di altri dataset, registrati in ambienti controllati, tipicamente utilizzati per questo task e descritti brevemente nel capitolo sullo stato dell’arte.

Il realismo dei video implica a sua volta implica alcune caratteristiche:

* i video hanno lunghezza eterogenea, alcuni video durano pochi secondi mentre altri durano una decina d’ore;
* i video hanno risoluzione diversa tra loro;
* ogni video è stato registrato a diversi frame per secondo;
* alcuni video sono stati registrati da telecamere in movimento;
* alcuni video presentano dei cambi di scena improvvisi (ciò influirà negativamente sulle prestazioni del modello)
* Le classi presenti nel dataset sono molto generiche, per esempio l’azione del fare intrusione potrebbe essere intesa come scassinare una serratura o anche come lo sfondare una vetrata di un negozio con una macchina o il forzare una serratura con un piede di porco o un trapano (figure 1, 2 e 3);

Immagine che contiene testo, interni, tavolo, scrivania

Descrizione generata automaticamente

Figura

Figura



Figura

* alcuni video possono presentano delle scritte in sovrimpressione (perché utilizzati da telegiornali o programmi televisivi) (figura 4);



Figura

* alcuni video utilizzano dei filtri come telecamere termiche o notturne (figura 5);

Immagine che contiene testo, bianco, casco, luce

Descrizione generata automaticamente

Figura

* alcuni video presentano effetti particellari (insetti, fumo, cenere, neve, pioggia) che verranno rilevati come pixel in movimento (figura 6 e 7);

Immagine che contiene testo, via, strada, scena

Descrizione generata automaticamente

Figura

Immagine che contiene testo, scena, via, strada

Descrizione generata automaticamente

Figura

* alcuni video riprendono azioni svolte in pubblico dove l’anomalia è presente solo in una piccola porzione dello schermo ed è difficile da identificare anche ad occhio nudo (figura 8).



Figura

Ogni video nel dataset è stato ridimensionato a 240x320 pixel e il framerate è stato fissato a 30 fps.

Di ogni video è disponibile la lunghezza e annotazioni riguardanti i frame in cui sono presenti anomalie, che tuttavia vengono usate esclusivamente per validare i risultati conseguiti dalla rete.

Le annotazioni a loro volta sono molto imprecise e si limitano ad indicare intervalli di frame in cui avviene sicuramente un’anomalia, ciò significa che i restanti frame potrebbero comunque contenere un’anomalia non indicata (figura 9), questa inattendibilità ha influito sulla scelta del modello d’apprendimento e nello specifico nella scelta di un metodo di weakly supervised learning.

Immagine che contiene testo, interni, tavolo da pranzo

Descrizione generata automaticamente

Figura (frame segnalato come non anomalo)

Il Real Life Violence Situations Dataset a differenza dell’UCF crime invece è composto da singole clip di pochi secondi e nessun tipo di annotazione.

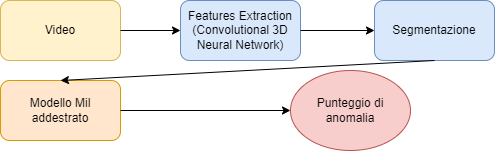
Per ovviare a questa mancanza si ha generato un file contenente il nome di ogni video, la sua lunghezza in frame e si è deciso di prendere come intervallo dove è presente l’anomalia la clip nella sua interezza.

# Capitolo 4 – Sperimentazione

La maggior parte degli approcci esplorati nel Capitolo 2 risulterebbero inefficaci se applicati al dataset utilizzato, infatti, anche conoscendo i frame anomali sarebbe difficile identificare un’azione sospetta in un’immagine senza conoscerne il contesto intuibile guardando una clip, inoltre anche se si conoscesse quando un’azione sospetta sta avvenendo sarebbe comunque impossibile capire dove nel video sta avvenendo rischiando di addestrare il video utilizzando features provenienti da settori di video superflui.

Per questo motivo piuttosto che cercare di classificare un’azione in un video si è scelto di cercare un modo per distinguere tra video normali e video anomali, per far ciò si ha scelto di utilizzare il multiple instance learning utilizzando clip sicuramente normali e clip sicuramente anomale cercando di “allontanare” il più possibile le prime dalle seconde.

Di seguito vengono riportate le varie fasi necessarie per creare il nostro modello (figura 10).

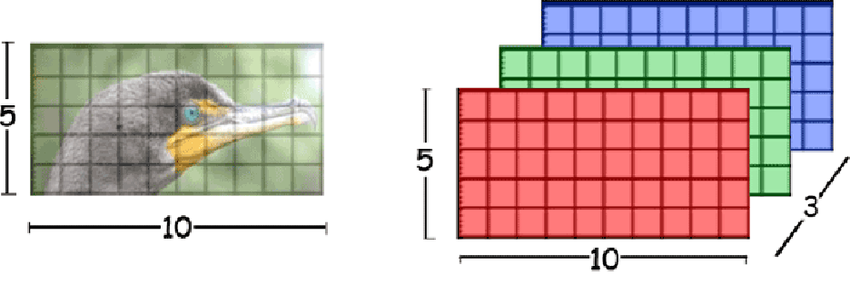


Figura

## 4.1 Features Extraction

Prima di poter creare la nostra rete e di poter addestrare il nostro modello è stato necessario elaborare i video contenuti nel dataset, è infatti necessario avere dati rappresentanti caratteristiche temporali e spaziali per ogni clip presente nel dataset. Come è ben noto un video non è altro che una sequenza di frames in rapida successione, cosa ben meno nota invece è che questi frame vengono rappresentati all’interno dei computer come una matrice di bit rappresentanti l’intensità dei tre colori principali, rosso, giallo e blu.

Ciò significa che possiamo rappresentare ogni frame di un video come tre matrici con valori compresi tra 0 e 255 (2^8-1) come mostrato dalla figura 11.



Figura

Questa rappresentazione rigorosa e numerica di ogni video ci è utile per estrarre le caratteristiche di ogni frame, questo task è stato eseguito sfruttando una rete neurale convoluzionale con kerneldi tre dimensioni, che d’ora in poi chiameremo convolutional 3D neural network (o C3DNN).

### 4.1.1 Convolutional 2D Neural Network

Una convolutional Neural Network è un tipo di rete feed-forward particolarmente performante per i task di riconoscimento di immagini, come si evince dal loro nome si basano sull’operazione matematica di convoluzione.

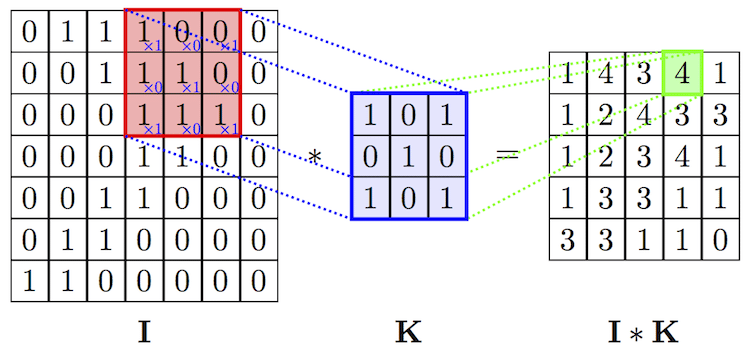
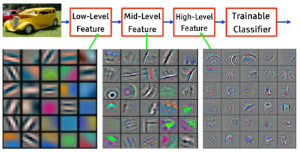


Figura rappresentazione grafica dell'operazione di convoluzione

Matematicamente questa è un’operazione molto semplice, come mostrato in figura 12 l’input della nostra rete è un’immagine che come è stato spiegato precedentemente viene rappresentata come una matrice di pixel, il kernel K (o filtro), invece è il vero protagonista di questa funzione, egli infatti scorre lungo l’immagine effettuando il prodotto tra due matrici, il risultato di questa operazione è a sua volta una matrice che viene chiamata features map.

A un livello puramente teorico invece il funzionamento di una CNN è anche più semplice, il kernel, chiamato anche filtro, per l’appunto filtra l’immagine cercando pattern riconoscibili, ciò significa che per ogni convolutional layer viene cercata una specifica features che rappresenta una caratteristica diversa dell’immagine, come per esempio una forma ricorrente o un colore come mostrato in figura 12.

.

Figura

Dopo ogni convolutional layer vengono rilevate più features fino ad arrivare a features di alto livello utilizzabili per una classificazione.

In figura 14 viene riportata invece la tipica architettura di una rete neurale convoluzionale:



Figura architettura di una CNN

Possiamo distinguere nell’architettura:

* Convolution Layer: layer che effettua l’operazione matematica di convoluzione;
* Max Pooling layer: effettua l’operazione di *downsampling*, ovvero attraverso l’utilizzo di un secondo kernel scorre la matrice ricevuta in input dal layer di convoluzione selezionando soltanto il valore massimo all’interno della sottomatrice analizzata (chiamata pool) e memorizzandola in una seconda matrice di grandezza minore (figura 15) riducendo di molto la complessità della rete e migliorandone le prestazioni;
* Fully connected layer: ovvero il penultimo layer dell’immagine, simile a quello di una classica rete neurale, ogni nodo è collegato a tutti i nodi del layer finale.

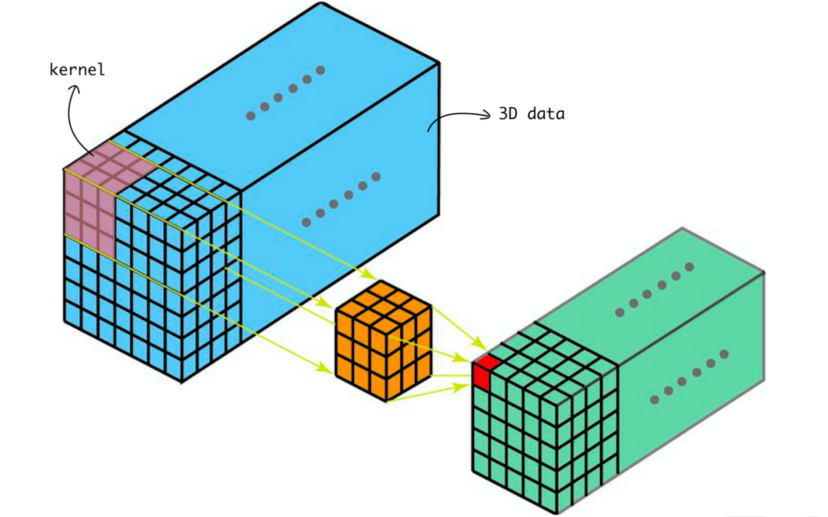


Figura

### 4.1.2 Convolutional 3D Neural Network

Una Convolutional Neural Network classica estrae features da un singolo frame, ciò non sarebbe particolarmente utile per il nostro task in quanto per riconoscere un’anomalia è necessario analizzare un’intera clip, per questo motivo si ha utilizzato una versione avanzata di una classica CNN ovvero una Convolutional 3D Neural Network.

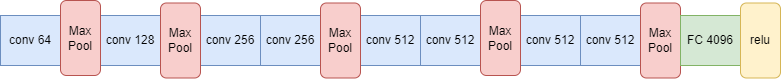
A inizio Capitolo 4 è stata spiegata la rappresentazione di un’immagine sottoforma di matrice di pixel, viene naturale quindi intendere un video come una sequenza di matrici o meglio ancora come una matrice tridimensionale, in questo caso il nostro Kernel, anch’esso tridimensionale scorrerà sul parallelepipedo rappresentante la clip generando una features map tridimensionale come rappresentato in figura 16.



Figura

La tridimensionalità di queste features rappresenta features spazio-temporali perfette per task di Human activity Recognition (HAR) in un video.

### 4.1.3 Architettura Rete C3DNN



Figura

La rete utilizzata (figura 17) è stata addestrata sul dataset Sports1M, dataset contenente un milione di clip di sport.

La features extraction viene effettuata per clip lunghe sedici frame, il risultato di questa rete per ogni video è quindi un numero variabile di tensori di 4096 dimensioni rappresentanti caratteristiche spazio-temporali di quella porzione di video.

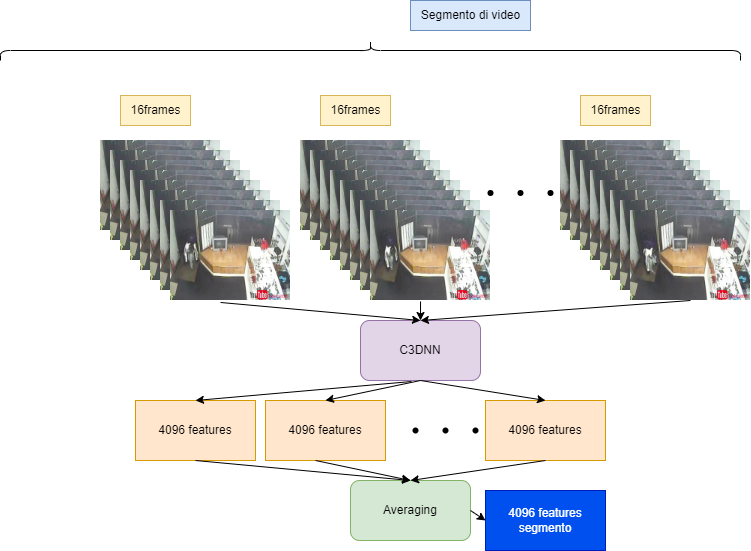
## 4.2 Segmentazione dei video e normalizzazione dei segmenti

Come già spiegato nel capitolo 3 il dataset utilizzato è composto da video di lunghezza variabile, ciò è risultato un problema durante la fase di addestramento del nostro sistema durante la quale bisogna confrontare features provenienti da clip anomale con feauteres provenienti da clip normali, per questo motivo è stato necessario normalizzare le features ottenute durante la fase di features extraction.

Per ogni video nel dataset di lunghezza in frames si ha 32 segmenti di lunghezza :

Dopodiché per ogni clip all’interno di un singolo segmento viene generato un nuovo tensore contenente la media dei valori delle features delle clip in quel segmento. Il risultato di questa operazione è un file contenente 32 tensori lunghi 4096D che rappresentano le nostre features spazio-temporali.

Questa fase viene illustrata in figura 18.



Figura

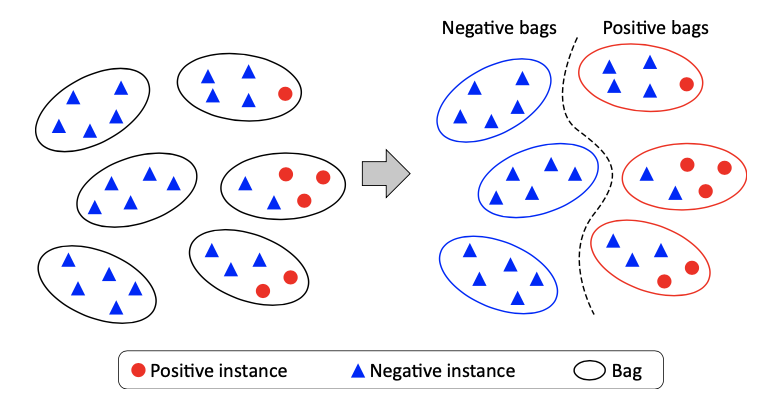
## 4.3 Training

I segmenti ottenuti nelle fasi precedenti rappresentano istanze in una borsa, questa rappresentazione è necessaria per il nostro approccio che si basa per l’appunto sul concetto di bag positiva e bag negativa.

### 4.3.1 Multiple Instance Learning

Il multiple instance learning si basa sull’idea di utilizzare borse contenenti istanze, che da questo momento in poi chiameremo con il loro nome inglese bags, per la fase di addestramento.

Una bag è negativa se tutte le istanze all’interno di essa sono negative, mentre sarà positiva se almeno una delle sue istanze è positiva (figura 19). In questo caso il sistema conoscerà soltanto il label di una borsa e non delle singole istanze di cui è formata.



Figura

Ovviamente per il nostro caso specifico una bag corrisponde a un video mentre i segmenti di cui quest’ultimo è formato rappresentano le istanze. Il vantaggio di questo approccio è evidente: non sono necessarie annotazioni a livello temporale, basta soltanto sapere quali video sono anomali e quali normali.

### 4.3.2 Funzione obbiettivo

Come già citato il task viene inteso come problema di regressione, per questo motivo il nostro obbiettivo è fare in modo che video anomali abbiano segmenti con punteggi di anomalia più alti rispetto a segmenti anomali:

Dove Va e Vn rappresentano video anomali e video normali, e sono i punteggi assegnati agli i-esimi segmenti dei video.

Il nostro obbiettivo è quindi quello di distanziare il più possibile il segmento che ha un punteggio più alto per quanto riguarda i video anomali dal segmento con il punteggio più alto per quanto riguarda i video normali (un falso positivo).

Possiamo quindi costruire la nostra rank loss function nel seguente modo:

Questa funzione nasce da una reinterpretazione della hinge loss, funzione di ottimizzazione per problemi di classificazione usando una Support Vector Machine.

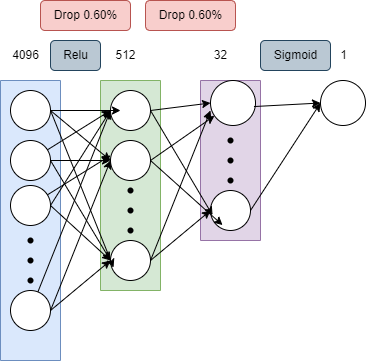
Possiamo ampliare la formula facendo due considerazioni, la prima è che il punteggio dei segmenti temporalmente precedenti a un segmento anomalo dovrebbero avere un punteggio vicino a quello del segmento anomalo, la seconda è che in un video un’anomalia dovrebbe avvenire raramente.

Possiamo quindi aggiungere due termini alla nostra funzione per rappresentare queste caratteristiche richieste dalle nostre previsioni:

Il valore delle due costanti nella nostra implementazione sono dello 0.00008.

### 4.3.3 Architettura Rete

Il modello è sato addestrato utilizzando una fully connected neural network illustrata in figura 20.



Figura

L’architettura utilizzata è molto semplice, si tratta di una fully connected neural network formata da quattro layer.

Vengono utilizzate due funzioni di attivazione, una RELU e una funzione sigmoidea, la prima corrisponde a:

Con output del layer precedente, mentre la seconda corrisponde a:

Con numero di Nepero. Questa funzione ha dominio compreso tra 0 e 1, il risultato di quest’ultimo layer corrisponde al punteggio di anomalia assegnato ad un segmento di video risultante dall’utilizzo della rete.

### 4.3.4 Dettagli implementazione

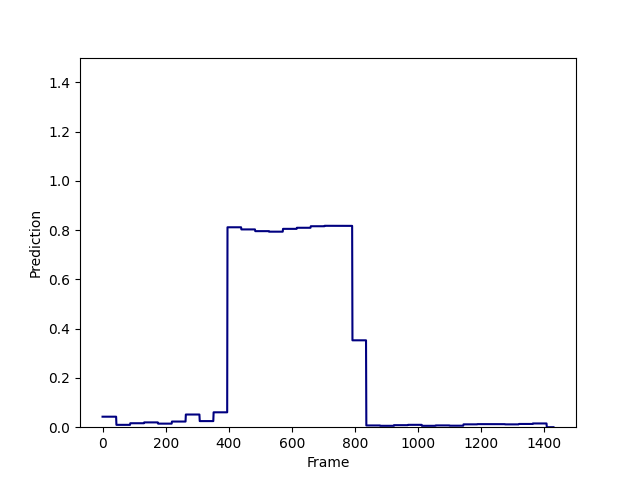
La rete è stata creata in python3 utilizzando la libreria pyTorch, è stata addestrata per 40 epoche e 1000 iterazioni ad epoca con una batch size di 60 elementi con un learning rate dello 0.001 e utilizzando l’optimizer Adadelta.

La fase di addestramento è stata effettuata su una macchina che montava una GTX 1050TI ed è durata in totale 2 ore

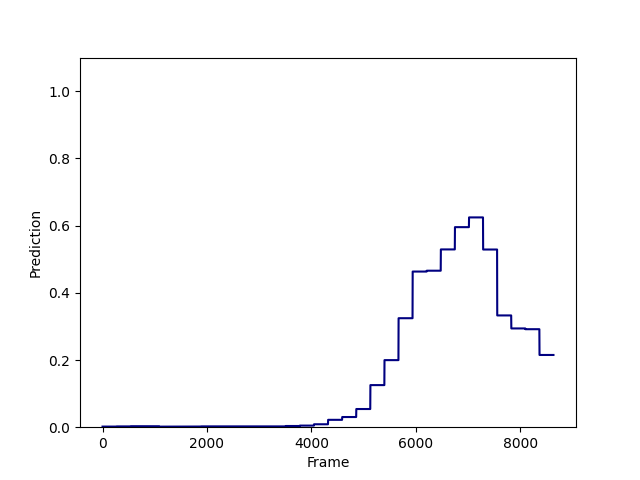
## 4.4 Inferenza

L’architettura illustrata nel capitolo 4.3.3 viene riutilizzata per effettuare previsioni su nuovi video presentati al sistema, in questo caso le fasi di features extraction e di segmentazione devono ovviamente essere effettuate anche per il nuovo video.

Le previsioni vengono effettuate sui singoli segmenti, quindi i risultati saranno 32 punteggi di anomalia differenti riportati su un grafico che ha per ascissa il punteggio di anomalia del segmento e per ordinata i frame corrispondenti (figura 21 e figura 22).



Figura

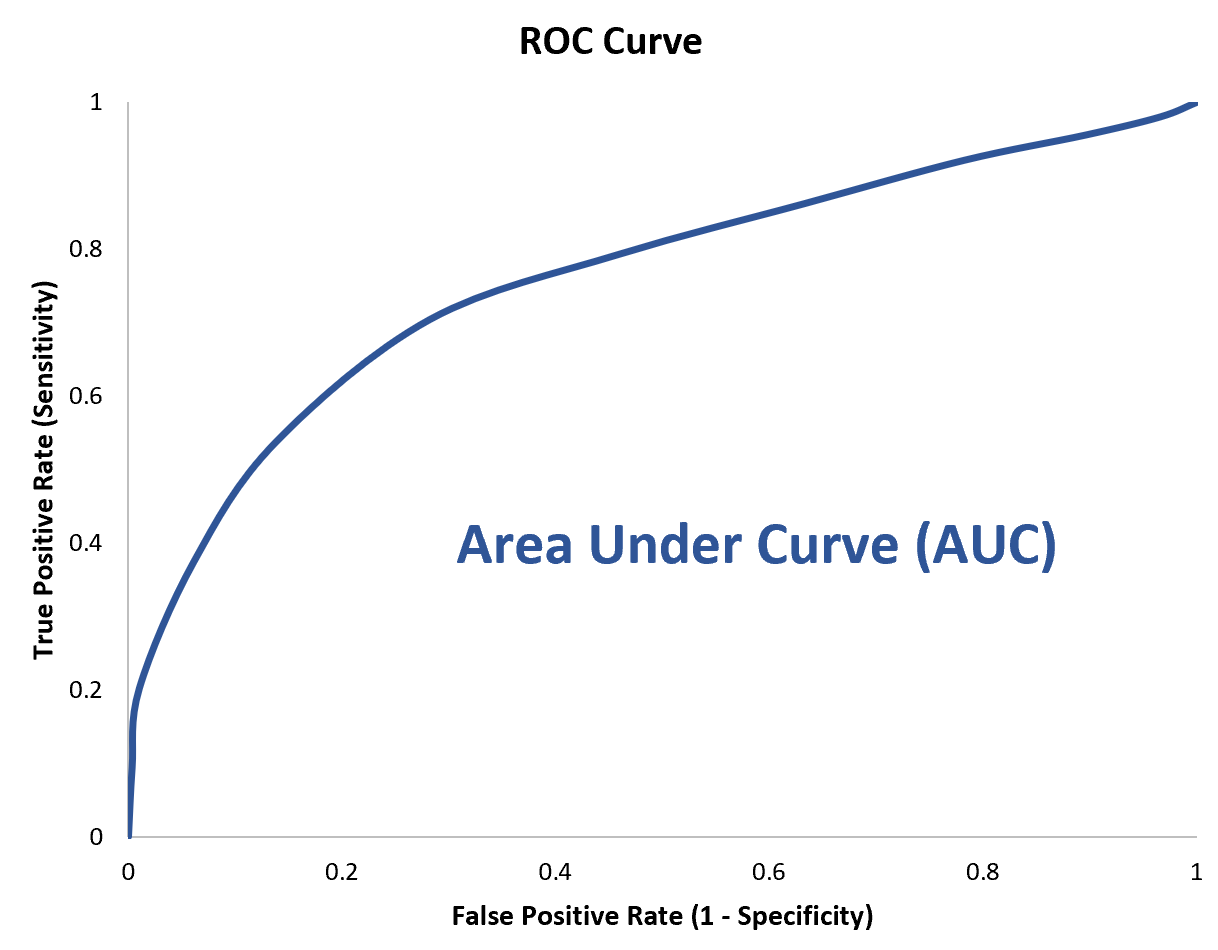


Figura

# Capitolo 5 – Risultati

Nonostante il sistema creato faccia di fatto regressione, il nostro obbiettivo è comunque quello di distinguere tra video anomali e video non anomali, per questo motivo si è deciso di utilizzare come metrica di valutazione la curva Receiver Operating Characteristic (ROC).

La curva ROC (figura 23) viene disegnata utilizzando il tasso di Veri Positivi (*True Positives*) e il tasso di Falsi Positivi (*False Positives*), questi due valori sono stati calcolati utilizzando la libreria sklearn di pyton e le annotazioni riguardanti i segmenti anomali di entrambi i dataset.



Figura

Le formule del True positive Rate e del False Positive Rate sono rispettivamente:

Fortuitamente la libreria utilizzata ha permesso di calcolare questi due valori utilizzando i punteggi di anomalia assegnati dal nostro sistema e i punteggi aspettati attraverso la funzione:

fpr, tpr, thresholds = roc\_curve(y\_true=y\_trues, y\_score=y\_preds)

Dove *y\_preds* è una lista composta dai punteggi assegnati a tutti i frame di cui è composto ognuno dei 32 segmenti in cui è diviso il nostro sistema mentre *y\_trues* è una lista composta dai valori aspettati dal nostro sistema.

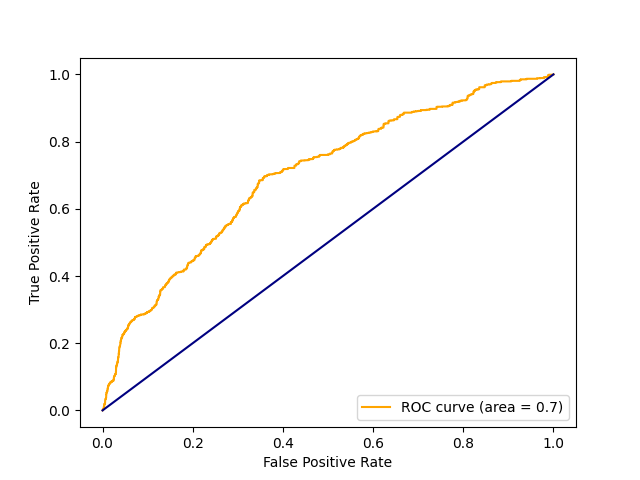
Nel nostro caso ogni frame segnato come non anomalo viene contrassegnato con uno zero mentre ogni frame appartenente ad un intervallo segnato come anomalo viene contrassegnato con un uno.

### 5.1 Prima sperimentazione

Durante la prima sperimentazione il sistema è stato addestrato su 1620 video appartenenti all’UCF crime. Questo train set è formato da video contenenti abusi, arresti, incendi, assalti, intrusioni, esplosioni, combattimenti, incidenti stradali, rapine, sparatorie, taccheggi, furti e vandalismo.

Il modello generato su questo train set è stato testato sui 280 video dello stesso dataset riservati per il testing.

La curva ROC risultante da questa sperimentazione è la seguente (figura 24):



Figura

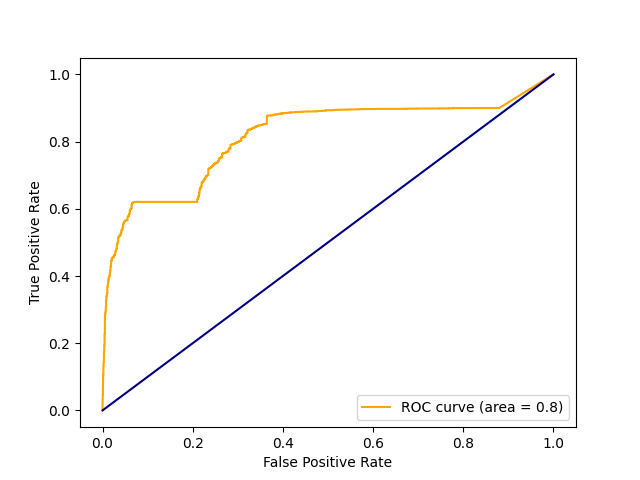
L’area sottostante la curva disegnata, l’area under the curve, in questo caso è del 70%.

### 5.2 Seconda sperimentazione

Durante la seconda sperimentazione il sistema è stato addestrato su 1400 video appartenenti al Real Life Violence Situations Dataset. Questo train set è formato da video contenenti risse, rapimenti, combattimenti e scene di violenza in generale.

Il modello generato su questo train set è stato testato sui 600 video dello stesso dataset riservati per il testing.

La curva ROC risultante da questa sperimentazione è la seguente (figura 25):



Figura

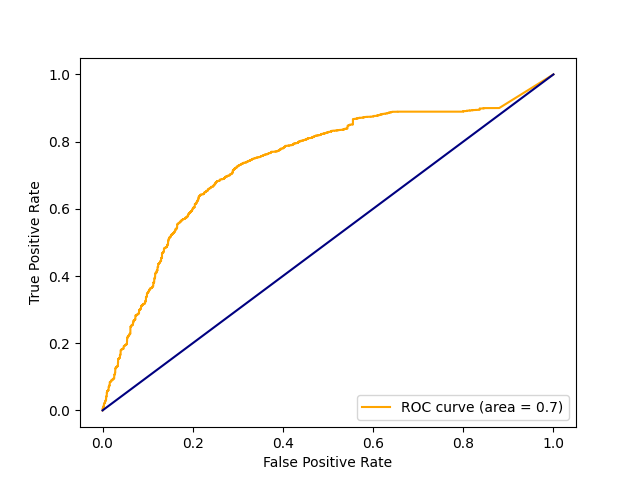
L’auc raggiunto durante questa sperimentazione è del 81%.

### 5.3 Terza sperimentazione

Durante la terza sperimentazione il sistema è stato addestrato su 1400 video appartenenti all’UCF crime.

Il modello generato su questo train set è stato testato sul dataset Real Life Violence Situations Dataset.

La curva ROC risultante da questa sperimentazione è la seguente (figura 26):



Figura

L’auc raggiunto durante questa sperimentazione è del 74%.

Non è stato possibile effettuare una quarta sperimentazione opposta alla terza dove viene utilizzato un modello addestrato sul Real Life Violence Situations Dataset e testato sull’UCF crime in quanto in quest’ultimo sono presenti molte classi di azioni non presenti nel primo impedendo al sistema di identificare azioni mai viste prima.

# Capitolo 6 – Conclusioni

Riuscire ad identificare anomalie è un task molto difficoltoso, il sistema costruito raggiunge comunque buoni risultati in tutte e tre le sperimentazioni effettuate. La terza sperimentazione effettuata utilizza lo stesso train set e test set di Sultani et al. [3] che hanno raggiunto un AUC del 74% utilizzando una diversa architettura per estrarre le Convolutional 3D features.

|  |  |
| --- | --- |
| **Modello** | **AUC** |
| Esperimento 1 | 70% |
| Esperimento 2 | 74% |
| Esperimento 3 | 84% |
| Stato dell’arte | 75% |

# Capitolo 7 – Sviluppi Futuri

La difficoltà principale affrontata è stata la mancanza di dataset per addestrare e testare il sistema, oltre che la quasi mancanza di dataset registrati in contesti non controllati e realistici che renderebbero il sistema inutile in un contesto reale, inoltre il sistema creato non riesce a rilevare anomalie in tempo reale per motivi puramente tecnici. Dei possibili sviluppi futuri comprendono:

* L’utilizzo di un nuovo dataset, registrato in contesti reali e contenente più classi di azioni.
* L’utilizzo di un diverso tipo di features, come per esempio le inflated 3D convolutional features.
* L’implementazione del sistema per rilevare anomalie in tempo reale,

**Bibliografia**

[1] Zaidi, S., Jagadeesh, B., Sudheesh, K. V., & Audre, A. A. (2018). Video Anomaly Detection and Classification for Human Activity Recognition. In International Conference on Current Trends in Computer, Electrical, Electronics and Communication, CTCEEC 2017 (pp. 544–548). Institute of Electrical and Electronics Engineers Inc.

[2] Nandini G., Dr. B. Mathivanan, Nantha Bala. R. S., Poornima P, (2018) Suspicious human activity detection. In International Journal of Advance Research and Development, IJARND 2018, VOLUME 3, ISSUE 4

[3] Sultani, W., Chen, C., & Shah, M. (2018). Real-World Anomaly Detection in Surveillance Videos. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 6479–6488). IEEE Computer Society.

[4] Nale, R., Sawarbandhe, M., Chegogoju, N., & Satpute, V. (2021). Suspicious human act vity detection using pose estimation and LSTM. 2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA).

[5] Babiker, M., Khalifa, O. O., Htike, K. K., Hassan, A., & Zaharadeen, M. (2018). Automated daily human activity recognition for video surveillance using neural network. In *2017 IEEE International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2017* (Vol. 2017-November, pp. 1–5). Institute of Electrical and Electronics Engineers Inc.

[6] Ibrahim Salem, F. G., Hassanpour, R., Ahmed, A. A., & Douma, A. (2021). Detection of Suspicious Activities of Human from Surveillance Videos. In *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, MI-STA 2021 - Proceedings* (pp. 794–801). Institute of Electrical and Electronics Engineers Inc.

[7] Sivasakthi. T, Dr. Brindha. S, Hariharasudhan. S. M, Vishal.V. S, Priyadharsan. M, (2022). Suspicious Activity Detecting Camera. In 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT). 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT). IEEE.

[8] Amrutha, C. V., Jyotsna, C., & Amudha, J. (2020). Deep Learning Approach for Suspicious Activity Detection from Surveillance Video. In *2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2020 - Conference Proceedings* (pp. 335–339). Institute of Electrical and Electronics Engineers Inc.

[9] Bordoloi, N., Talukdar, A. K., & Sarma, K. K. (2020). Suspicious Activity Detection from Videos using YOLOv3. In *2020 IEEE 17th India Council International Conference, INDICON 2020*. Institute of Electrical and Electronics Engineers Inc.

[10] K. Kranthi Kumar, B. Hema Kumari, T. Saikumar, U. Sridhar, G. Srinivas, G. Sai Karan Reddy (2022). SUSPICIOUS ACTIVITY DETECTION FROM VIDEOSURVEILLANCE, International Journal of Research Publication and Reviews, Vol 3, Issue 6, pp 2373-2377.

[11] Liu, C., Ying, J., Han, F., & Ruan, M. (2019). Abnormal human activity recognition using bayes classifier and convolutional neural network. In *2018 IEEE 3rd International Conference on Signal and Image Processing, ICSIP 2018* (pp. 33–37). Institute of Electrical and Electronics Engineers Inc.

[12] Dileep, A. S., S., N., Sreeju, Farhana, & Surumy. (2022). Suspicious human activity recognition using 2D pose estimation and convolutional neural network. *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*.

[13] Vrskova, R., Hudec, R., Kamencay, P., & Sykora, P. (2022). Recognition of human activity and abnormal behavior using deep neural network. *2022 ELEKTRO (ELEKTRO)*.

[14] Iqbal, J. L. M., & Arun, S. (2018). Intelligent Information System for Suspicious Human Activity Detection in Day and Night. In International Journal of Informatics and Communication Technology (IJ-ICT) (Vol. 7, Issue 3, p. 117). Institute of Advanced Engineering and Science.

[15] Atikuzzaman, Md., Rahman, T. R., Wazed, E., Hossain, Md. P., & Islam, Md. Z. (2020). Human Activity Recognition System from Different Poses with CNN. In 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI). 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE.

[16] A. Manaf F and S. Singh, "A Novel Hybridization Model for Human Activity Recognition using Stacked Parallel LSTMs with 2D-CNN for Feature Extraction," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1-7, doi: 10.1109/ICCCNT51525.2021.9579686.

[17] Arunnehru, J., Chamundeeswari, G., & Bharathi, S. P. (2018). Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos. Procedia Computer Science, 133, 471–477.