



UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E  
DELL'AUTOMAZIONE

---

**Un sistema di rilevamento cadute basato  
su telecamere RGB per l'assistenza  
sanitaria all'anziano**

**A RGB camera-based fall detection  
system for elderly healthcare**

Laureando:  
**Stefano Perniola**

Relatore:

**Prof. Aldo Franco Dragoni**

Anno Accademico 2021-2022





UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE

FACOLTÀ DI INGEGNERIA  
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E  
DELL'AUTOMAZIONE

---

**Un sistema di rilevamento cadute basato  
su telecamere RGB per l'assistenza  
sanitaria all'anziano**

**A RGB camera-based fall detection  
system for elderly healthcare**

Laureando:  
**Stefano Perniola**

Relatore:  
**Prof. Aldo Franco Dragoni**

Anno Accademico 2021-2022

---

UNIVERSITÀ POLITECNICA DELLE MARCHE  
FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE  
Via Brecce Bianche – 60131 Ancona (AN), Italy

*Alla mia famiglia e a tutti coloro che mi sono vicini*



# **Ringraziamenti**

Una menzione particolare va al mio relatore, il prof. Aldo Franco Dragoni, per la sua disponibilità e competenza, che mi ha portato fino in fondo in questo percorso. Si ringraziano tutti i membri del laboratorio universitario AirtLab che hanno contribuito allo sviluppo dell'elaborato e mi hanno fatto appassionare alla tematica.

*Ancona, Febbraio 2023*

Stefano Perniola



# **Abstract**

Le cadute sono una delle cause più comuni di lesioni e complicazioni tra gli anziani. Negli Stati Uniti, questi eventi sono la principale causa di complicazioni correlata a lesioni non intenzionali negli adulti di età pari o superiore a 65 anni. La caduta di un anziano può comportare conseguenze fisiche come fratture, lesioni interne, dolore e disabilità. In alcuni casi, può anche causare conseguenze psicologiche come ansia, depressione e perdita di autonomia. Gli algoritmi di rilevamento delle cadute più tradizionali funzionano analizzando i dati dell'accelerometro e del giroscopio da un dispositivo per determinare se si è verificata una caduta. Alcuni sistemi incorporano anche altri sensori come un barometro per aiutare a determinare l'altezza della caduta. Purtroppo, tali rilevatori di caduta sono in genere basati su dispositivi indossabili e gli anziani spesso li dimenticano o non hanno intenzione di indossarli. Negli ultimi anni, il deep learning è stato applicato al rilevamento delle cadute per far fronte a questi problemi cercando inoltre di migliorare l'accuratezza e ridurre i falsi allarmi. In particolare, l'obiettivo consiste nell' ottenere sistemi affidabili addestrando i modelli di deep learning su grandi set di dati di cadute per riconoscere pattern e caratteristiche associate ad esse. Al momento, i rilevatori di caduta basati sulla visione artificiale non sono ancora disponibili sul mercato ma la ricerca continua a studiare metodologie sempre più applicabili in contesti reali. In questa tesi, si vuole presentare una proposta di rilevatore di caduta a basso costo per l'assitenza sanitaria all'anziano basato sull'utilizzo di reti neurali convoluzionali e algoritmi di Motion History Image. L'obiettivo della fall detection è stato modellato come un problema di classificazione binaria (fall, not fall) in cui la predizione delle label viene effettuata frame-by-frame oppure in batch. I test condotti su oltre 140 diversi video di caduta hanno mostrato un valore di accuracy totale superiore al 96%.



# Contents

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Contesto . . . . .	1
1.2	Motivazioni . . . . .	4
1.3	Edge AI . . . . .	7
1.4	Obiettivi di ricerca . . . . .	10
<b>2</b>	<b>Stato dell'arte</b>	<b>13</b>
2.1	Dispositivi indossabili . . . . .	14
2.2	Dispositivi di visione . . . . .	17
2.2.1	Motion History Image . . . . .	19
2.2.2	Pose Estimation . . . . .	22
2.2.3	Reti Neurali Convoluzionali . . . . .	25
<b>3</b>	<b>Metodologie utilizzata</b>	<b>29</b>
3.1	Tecnologie utilizzate . . . . .	31
3.1.1	TensorFlow . . . . .	32
3.1.2	Google Colab . . . . .	33
3.1.3	Raspberry Pi . . . . .	35
3.1.4	TensorFlow Lite . . . . .	36
3.1.5	Fallnet . . . . .	37
3.1.6	FDNet . . . . .	43
3.2	Dataset utilizzati . . . . .	45
3.3	Architettura delle reti . . . . .	48
3.4	Fase di training e test . . . . .	56
3.4.1	PushBullet . . . . .	62
<b>4</b>	<b>Risultati ottenuti</b>	<b>65</b>
4.1	Valutazione modelli . . . . .	71
<b>5</b>	<b>Conclusioni</b>	<b>77</b>
5.1	Sommario . . . . .	79
5.2	Limiti dei modelli . . . . .	81
5.3	Sviluppi futuri . . . . .	82



# List of Figures

1.1	Architettura dell'Edge AI . . . . .	9
1.2	Esempio applicazione dell'Edge AI . . . . .	9
2.1	Dispositivi indossabili . . . . .	16
2.2	Vision based fall detection . . . . .	18
2.3	Motion history image . . . . .	21
2.4	Pose estimation . . . . .	24
2.5	CNN Architecture . . . . .	27
3.1	TensorBoard . . . . .	33
3.2	Google Colab . . . . .	34
3.3	Raspberry Pi 3 b+ . . . . .	35
3.4	TF Lite . . . . .	37
3.5	UR Fall Dataset . . . . .	46
3.6	4 ambienti di Fall Dataset . . . . .	47
3.7	Architettura fallnet . . . . .	48
3.8	Esempio max pooling . . . . .	50
3.9	Convoluzioni della fallnet . . . . .	52
3.10	MobileNet v2 architecture . . . . .	55
3.11	Valutazione addestramento . . . . .	57
3.12	Predizioni della rete . . . . .	58
3.13	Processamento immagine in MHI . . . . .	59
3.14	Predizioni della FDNet . . . . .	61
3.15	Segnalazione caduta da PushBullet . . . . .	63
4.1	Formula Accuracy . . . . .	66
4.2	Formula Precisione . . . . .	67
4.3	Formula Recall . . . . .	68
4.4	Formula f1 score . . . . .	69
4.5	Matrice di confusione . . . . .	70
4.6	Matrice di confusione fallnet . . . . .	71
4.7	Matrice di confusione FDNet . . . . .	72
4.8	Scenario nella norma . . . . .	74
4.9	Scenario di caduta . . . . .	75



# List of Tables

3.1 Architettura della Fallnet . . . . .	51
4.1 Metriche del modello . . . . .	71



# **Chapter 1**

## **Introduzione**

### **1.1 Contesto**

Il rischio di caduta è uno dei problemi più diffusi affrontati dagli individui anziani. Uno studio pubblicato dall'Organizzazione Mondiale della Sanità [1] stima che tra il 28% e il 35% delle persone sopra i 65 anni subisce almeno una caduta ogni anno, e questa cifra sale al 42% per le persone di età superiore ai 70 anni. Secondo le analisi effettuate, le cadute rappresentano oltre il 50% dei ricoveri ospedalieri anziani e circa il 40% delle mortalità non naturali per questo segmento della popolazione. Dunque le cadute sono una fonte significativa di mortalità per gli individui anziani nei paesi sviluppati. Queste risultano particolarmente pericolose per le persone che vivono da sole perché può passare una notevole quantità di tempo prima di ricevere assistenza. Circa un terzo degli anziani (quelli di età superiore ai 65 anni) in Europa vive da solo [2] e si prevede che la popolazione anziana aumenterà significativamente nei prossimi vent'anni. Sono state sviluppate diverse tecnologie per il rilevamento delle cadute. Tuttavia, richiedono in gran parte agli anziani di indossare dispositivi con sensori. Alcuni anziani, specialmente quelli con demenza, tendono a dimenticare di indossare tali dispositivi. Questi richiedono cure speciali per mantenere condizioni di vita indipendenti. Le persone affette da demenza generalmente desiderano vivere nelle proprie case; Tuttavia, questo non è sempre possibile. L'uso di sistemi intelligenti nelle case dei pazienti anziani (in un contesto smart home) migliora la loro

indipendenza, comfort e sicurezza [3] e previene la depressione. Inoltre, libera i caregiver da alcune attività quotidiane di cura. In studi simili, i caregiver ritengono che questi progressi tecnologici possano essere molto utili se usati comodamente, ad esempio in settori come la sicurezza (gli anziani si sentono più sicuri) e il tempo libero (gli anziani non hanno bisogno di caregiver per essere intrattenuti). Semplicemente sapere che il loro paziente è al sicuro a casa dà ai caregiver un'importante tregua psicologica. Le case intelligenti consentiranno alle persone di prolungare i loro anni di vita indipendente e ridurre il tempo necessario ai caregiver per monitorare i loro anziani. I sistemi di rilevamento delle cadute come quello descritto in questa tesi sono un passo importante verso lo sviluppo della casa e assistenza sanitaria intelligente. Il sistema di rilevamento delle cadute proposto in questo documento si basa su un dispositivo a basso costo che comprende un computer (raspberry pi) e una fotocamera incorporati. Questo dispositivo può essere installato nelle pareti di un edificio (fungendo da telecamera frontale) o sui soffitti (fungendo da telecamera pavimentale) e monitorare una stanza senza l'intervento umano. Inoltre, le persone monitorate a casa non sono tenute a indossare dispositivi. Pertanto, il sistema è in grado di monitorare 24 ore su 24 l'ambiente circostante. Il sistema si basa su algoritmi di visione artificiale che monitorano la presenza di persone in una stanza e rilevano se una persona è caduta. Quando viene rilevata una caduta, viene inviato un messaggio di allarme attraverso smartphone con alcune informazioni fondamentali, come il dispositivo che ha rilevato la caduta e il timestamp dell'accaduto. Non vengono scambiate altre informazioni sulla privacy. Il contributo principale di questo articolo è quello di dimostrare che un sistema di rilevamento delle cadute in tempo reale basato su algoritmi di visione può essere eseguito in un dispositivo a basso costo come una Raspberry Pi 3 b+, ottenendo buoni valori prestazionali (cioè accuracy del 97%),

### *1.1 Contesto*

paragonabili ad altri sistemi che utilizzano hardware più costoso e più potente. Questo articolo è strutturato come segue: lo stato dell'arte nel rilevamento delle cadute è discusso nella Sezione 2, sia dal punto di vista delle tecnologie commerciali che dei progressi nelle tecnologie di ricerca correlate. La sezione 3 si concentra sulle tecniche di visione artificiale e descrive i concetti e le procedure comuni utilizzati per il rilevamento delle cadute. La sezione 4 mostra i risultati ottenuti e presenta le metriche principali per la valutazione dei modelli. La sezione 5 fa un sommario del lavoro svolto, esponendo le conclusioni, i limiti dei modelli e i possibili sviluppi futuri per eventuali migliorie.

## **1.2 Motivazioni**

Il rilevamento delle cadute utilizzando le reti neurali convoluzionali (CNN) è importante perché le CNN sono adatte per le attività di analisi di immagini e video e hanno dimostrato un'elevata precisione nel rilevare le cadute negli scenari del mondo reale. L'uso delle CNN può automatizzare il processo di rilevamento delle cadute e ridurre la dipendenza da interventi manuali, che possono richiedere molto tempo e sono soggetti a errori. Inoltre, le CNN possono operare in tempo reale e possono essere integrate in dispositivi indossabili o case intelligenti, fornendo un monitoraggio costante e garantendo una risposta tempestiva in caso di caduta. L'uso di CNN nel rilevamento delle cadute può anche migliorare l'accuratezza e l'affidabilità dei risultati, riducendo il rischio di falsi allarmi o rilevamenti mancati. Nel complesso, il rilevamento delle cadute con le CNN può fornire una soluzione più efficace ed efficiente per rilevare le cadute e garantire la sicurezza degli anziani e delle persone con problemi di mobilità.

La tecnologia basata sulla visione offre una soluzione più flessibile, precisa e confortevole per la rilevazione delle cadute rispetto ai sensori indossabili.

Rispetto a questi ultimi infatti, possiamo considerare i seguenti vantaggi per quanto riguarda l'applicazione di tecniche di visione:

1. Larga copertura: la tecnologia basata sulla visione può coprire un'area più estesa rispetto ai sensori indossabili, che sono limitati alla posizione del paziente.
2. Accuratezza: le tecniche che fanno utilizzo di visione possono rilevare la caduta in modo più preciso rispetto ai sensori indossabili che possono essere influenzati dalla posizione o dai movimenti del paziente.
3. Maggiore flessibilità: gli algoritmi di visione sono flessibili e possono essere utilizzati in una varietà di ambienti e contesti, men-

tre i sensori indossabili sono spesso limitati ad una sola posizione o ambiente.

4. Comfort: i sensori indossabili possono essere scomodi da indossare per il paziente, mentre un approccio vision-based non presuppone alcun dispositivo indossabile, essendo la telecamera installata sulle pareti o soffitti.

5. Costo: la tecnologia basata sulla visione può essere più economica rispetto ai sensori indossabili, che possono essere costosi da produrre e mantenere.

6. Facilità di installazione: Una volta che il sistema di visione viene messo in produzione, può essere facilmente installato in un ambiente, mentre i sensori indossabili possono richiedere una configurazione più complessa.

7. Interoperabilità: sistemi vision-based per la fall detection possono essere integrati con altri sistemi, come sistemi di allarme, di sorveglianza o sistemi audio. Invece i sensori indossabili sono spesso limitati a funzionare da soli.

D'altra parte, è importante notare che sistemi basati sulla visione possono sollevare problematiche sul loro utilizzo che invece non emergono nei dispositivi indossabili. Tra questi possiamo citare:

1. Privacy: l'utilizzo di telecamere, nonostante non vengono elaborate e scambiate informazioni sensibili, potrebbe sollevare preoccupazioni riguardo alla privacy, poiché possono registrare immagini o video dell'ambiente e delle persone in esso presenti. I sensori indossabili, al contrario, non presentano questi rischi.

2. Affidabilità: a volte i sistemi basati sulla visione potrebbero essere influenzati da fattori esterni come la luce o la polvere, che possono interferire con la qualità delle immagini o dei video registrati. I sensori indossabili, al contrario, sono meno influenzati da questi fattori.

3. Accessibilità: la tecnologia basata sulla visione potrebbe essere meno accessibile per alcune persone, come coloro con disabilità

*Chapter 1 Introduzione*

visive, che potrebbero non essere in grado di utilizzare il sistema. I sensori indossabili, al contrario, possono essere utilizzati da una gamma più ampia di persone.

### 1.3 Edge AI

L'Edge AI (Intelligenza Artificiale on Edge) è un tipo di intelligenza artificiale che viene eseguita direttamente sul dispositivo che raccoglie i dati, anziché inviare i dati a un server centrale per l'elaborazione. Ciò significa che le decisioni possono essere prese rapidamente e localmente, riducendo la quantità di dati che devono essere trasmessi e elaborati su una rete remota. Questo rende l'Edge AI particolarmente adatto per le applicazioni che richiedono una bassa latenza e una risposta rapida, come i sistemi di controllo industriale, le automobili autonome, le apparecchiature medicali mobili e in questo caso il rilevamento di cadute. L'Edge AI offre molti vantaggi rispetto all'elaborazione dei dati in un centro di dati remoto. Ad esempio:

- Latenza ridotta: Poiché i dati vengono elaborati localmente, le decisioni possono essere prese più rapidamente, riducendo la latenza e migliorando la reattività del sistema.
- Privacy e sicurezza dei dati: Poiché i dati non vengono inviati a un centro di dati remoto, c'è una maggiore protezione della privacy e della sicurezza dei dati.
- Scalabilità: L'Edge AI è adatto per l'elaborazione di grandi quantità di dati su un gran numero di dispositivi, rendendolo ideale per l'adozione su larga scala.
- Affidabilità: Poiché l'Edge AI non dipende da una connessione di rete affidabile per funzionare, è più affidabile e resistente alle interruzioni di rete.
- Risparmio di risorse: Poiché l'Edge AI utilizza hardware più piccolo e più efficiente rispetto ai centri di dati remoti, è più economico e sostenibile dal punto di vista energetico.

Negli ultimi anni l'Edge AI sta diventando un settore sempre più promettente. Le soluzioni in questo contesto sono in rapida evoluzione e stanno diventando sempre più importanti per una vasta gamma di settori, dall'automazione industriale all'Internet delle cose (IoT).

La rilevazione delle cadute è un argomento che rientra in questo

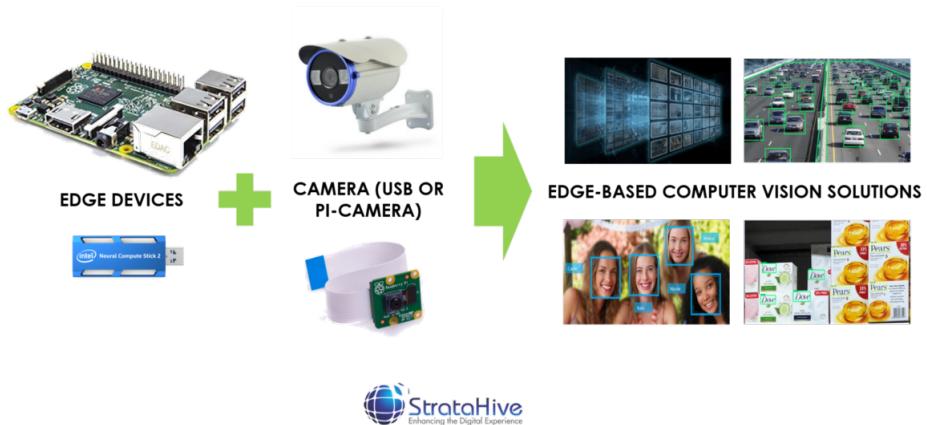
contesto e possiamo considerare tale attività come un’applicazione comune dell’Edge AI. In questo scenario, i sensori e le apparecchiature di rilevamento dei movimenti vengono posizionati nell’ambiente circostante, e i dati vengono elaborati localmente per determinare se la persona è caduta o meno. Se una caduta viene rilevata, un segnale può essere inviato a un dispositivo di emergenza, come un dispositivo di allarme o un telefono cellulare, per fornire assistenza tempestiva.

Nello stesso ambiente possono partecipare una o più telecamere. Ognuna di queste viene utilizzata per monitorare una specifica area e rilevare eventuali cadute. L’utilizzo di più telecamere aumenta l’affidabilità in quanto nel caso in cui una o più camere risultano offuscate o occluse da un oggetto, è possibile ottenere l’immagine dalle altre disponibili.

L’utilizzo di più telecamere consente una visualizzazione chiara e dettagliata dell’ambiente, il che può aiutare a fornire informazioni aggiuntive sulle circostanze della caduta e a identificare eventuali cause.

Altra caratteristica importante dell’Edge AI è la capacità di elaborare i dati in tempo reale e questo consente ai dispositivi di fornire una risposta tempestiva in caso di emergenza.

## Artificial Intelligence (AI) Solutions on Edge Devices



 StrataHive  
Enhancing the Digital Experience

Figure 1.1: Architettura dell'Edge AI

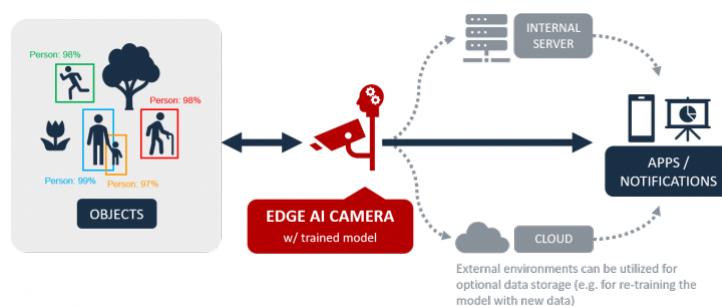


Figure 1.2: Esempio applicazione dell'Edge AI

## **1.4 Obiettivi di ricerca**

L’obiettivo di questa tesi è quello di costruire un sistema affidabile per la rilevazione delle cadute basato su visione. La definizione del problema è molto semplice. Si tratta di cercare di capire quando una persona sta cadendo o è caduta. Nonostante il concetto sia molto semplice, nella realtà dei fatti però per un sistema artificiale può essere complessa l’interpretazione della posa delle persone, in quanto queste possono compiere azioni apparentemente sospette ma che non hanno niente a che fare con una caduta (ad esempio abbassarsi per terra per raccogliere un oggetto, sdraiarsi sul letto, sedersi per terra e così via). L’attività che rileva le cadute si chiama fall detection ed è una tecnologia che monitora i movimenti di una persona, rileva eventuali cadute che si verificano ed eventualmente genera un allarme. A seguito dell’identificazione dell’evento, si può avvisare l’operatore sanitario o l’assistente della persona mediante smartphone. I sistemi di fall detection non sono limitati solo alle persone anziane. Possono anche essere utilizzati per proteggere atleti, lavoratori o chiunque sia a maggior rischio di caduta. Le telecamere possono avere un posizionamento frontale rispetto alla posizione della stanza, oppure possono essere poste sul soffitto, in modo tale da acquisire la pianta dell’edificio che si intende monitorare. Si può anche adottare una soluzione mista in cui vengono usate molte camere insieme, con un approccio Multi-View, in cui le telecamere collaborano fra loro per decidere in base alle singole acquisizioni se far scattare l’allarme o meno. In questo scenario si devono poi applicare tecniche di decisione efficaci. Per esempio se il numero di camere è dispari, si può adottare la semplice strategia del voto di maggioranza, in cui viene stabilito l’esito di un evento in base alla decisione più scelta fra gli agenti. Se il numero di telecamere è pari invece, il voto di maggioranza non è più applicabile, in quanto l’esito potrebbe

risultare incerto. In questi casi si potrebbero utilizzare soluzioni basate su score di confidenza delle singole camere o comunque si può scegliere di attivare a prescindere una segnalazione di caduta. Questo perchè, ammettendo di dover commettere un errore sulla valutazione delle posizioni, è meno grave generare un falso allarme piuttosto che non segnalare una vera caduta. Per quanto riguarda il calcolo dello score di confidenza, possiamo presupporre che alcune camere, per le caratteristiche di configurazione e posizionamento, sono statisticamente più affidabili rispetto altre. Ovvero riescono a predire e rilevare in maniera più precisa rispetto telecamere posizionate diversamente. Da questa supposizione, si attribuisce quindi un punteggio positivo per le camere più affidabili e uno negativo per quelle non affidabili. Lo score viene aggiornato continuamente in base alle inferenze che saranno fatte nei tempi successivi, seguendo un modello basati su premi e penalizzazioni. Le reti neurali utilizzano il concetto di premi e punizioni per addestrarsi e migliorare la loro capacità di effettuare previsioni o classificazioni corrette. Durante il processo di addestramento, la rete viene presentata con esempi di input e il suo output viene confrontato con la risposta corretta, nota come *label*. Se la rete produce una previsione sbagliata, viene penalizzata attraverso una riduzione della sua loss function. D'altra parte, se la rete produce una previsione corretta, viene premiata attraverso una riduzione della loss function. Questo processo viene ripetuto migliaia di volte per aiutare la rete a imparare dai suoi errori e migliorare la sua precisione nel tempo.

Per quanto riguarda la realizzazione del sistema di visione, è stata modellata una rete neurale convoluzionale tramite la piattaforma TensorFlow. Successivamente per poter adattare la rete su un sistema con capacità hardware più limitate, è stata fatta la conversione del modello tramite TensorFlow Lite. TensorFlow è una libreria software open source per il deep e machine learning.

Fornisce una piattaforma flessibile per sviluppare, addestrare e valutare i modelli. TensorFlow Lite è una versione leggera di TensorFlow progettata per dispositivi mobili e embedded. Consente agli sviluppatori di eseguire modelli TensorFlow su dispositivi con risorse computazionali limitate, come smartphone e microcontroller. È possibile utilizzarlo grazie al fatto che supporta un sottoinsieme dell'API TensorFlow e include ottimizzazioni per le prestazioni sul dispositivo e un ingombro di memoria ridotto. Ogni modello viene addestrato su un set di dati differente per posizionamento (in questo caso si distinguono dati da immagini frontali e da immagini pavimentali) Tali modelli uniti fra di loro costituiscono un unico sistema Multi-view per un rilevamento più efficace delle cadute.

Sono stati utilizzati due dataset differenti. Il primo si chiama "UR Fall Detection Dataset", e consiste in 140 video totali proposti sia in forma frontale che pavimentale. Il dataset fornisce anche dati sensoriali come quelli relativi all'accelerometro, tuttavia sono stati utilizzati soltanto i dati utili per l'approccio camera-based. Il secondo dataset è stato usato per estendere i dati del modello frontale e comprende 4 diversi ambienti di esecuzione cadute. I quattro ambienti sono: Casa, ufficio, sala caffè, e aula di lezione. Grazie all'estensione di questo nuovo dataset, la rete non solo comprende che le cadute sono indipendenti dalle caratteristiche dell'ambiente in background, ma concepisce diverse modalità e movimenti di caduta da persone diverse.

# **Chapter 2**

## **Stato dell'arte**

Il rilevamento delle cadute è una sfida importante per la salute e la sicurezza delle persone. Negli ultimi anni, c’è stata una crescente attenzione sull’utilizzo di tecnologie avanzate per la rilevazione delle cadute, in particolare per le persone anziane e vulnerabili. In generale ci sono due tipi di categorie di sistemi per la fall detection[4]. La prima concerne i dispositivi indossabili dalla persona da monitorare. Tali devices sono pensati per rilevare le cadute mediante strumenti di sensoristica come accelerometri o giroscopi che rilevano quei cambiamenti di ambiente e movimenti che suggeriscono una caduta. Questi sistemi possono risultare utili in molti casi, però presentano anche alcuni svantaggi. Spesso infatti, l’anziano non intende indossare il dispositivo per ragioni psicologiche o di comfort, oppure perché è estraneo a quella tecnologia specifica e non ha interesse ad utilizzarla. D’altra parte ci sono i dispositivi basati su telecamera che invece permettono all’assistito miglior libertà, in quanto il sistema di rilevamento cadute non invade la sfera fisica della persona ma rimane esterno e lontano.

## **2.1 Dispositivi indossabili**

Le tecnologie più comuni presenti in questi tipi di sensori sono accelerometri e giroscopi. Questi sono dispositivi facili da indossare, ma presentano alcuni inconvenienti come il consumo energetico (limitandone l'usabilità) e la sensibilità al movimento del corpo (che può causare falsi allarmi). Inoltre, una quantità considerevole di questi dispositivi si basa sulla capacità di un utente di attivare manualmente un allarme dopo un evento di caduta. Inoltre, anche se incorporano la tecnologia di rilevamento automatico delle cadute, questi tipi di dispositivi hanno generalmente molti falsi positivi, in base all'esperienza dell'autore. Tuttavia, da un punto di vista commerciale, la tecnologia dei sensori indossabili è la più comune. Questi dispositivi commerciali si presentano in genere sotto forma di ciondolo, cintura o orologio. In questa particolare categoria, Nella ricerca, è possibile trovare diversi approcci interessanti [5]. In una ricerca hanno presentato 13 algoritmi basati esclusivamente su accelerometri e hanno riportato un tasso medio di rilevamento dell'83% e un tasso di rilevamento di caduta del 98% per l'algoritmo più performante. Il problema principale con i rilevatori di accelerometri risiede nel discriminare le cadute reali dai movimenti bruschi, che possono generare falsi avvisi di caduta. Per risolvere questo problema, in [6] propongono di posizionare un accelerometro all'interno della testa di chi lo indossa. Lindemann et. [7] propone una soluzione simile, con l'applicazione del sensore all'interno dell'orecchio di chi lo indossa. I dispositivi indossabili più avanzati incorporano tecnologie di sensori. Il sistema presentato di Mathie et al. [8] utilizza un singolo sistema montato in vita di giroscopi e accelerometri per acquisire dati sull'inclinazione e il movimento di un soggetto. L'interessante approccio di Bianchi et al. [9] in cui vengono aggiunti sensori barometrici in grado di rilevare le variazioni di altezza causate dalle cadute. Segnalano un

tasso di successo di circa il 71%. Ghasemzadeh et al. [10] presenta una serie di sensori in grado di leggere la postura di un paziente e contemporaneamente ottenere delle letture dell'attività muscolare utilizzando sensori elettromiografici (EMG) con un tasso di rilevamento delle cadute del 98%. Lo sviluppo delle tecnologie di telefonia mobile, e dei sensori da esse incorporati, implica un'opzione molto interessante per soluzioni di rilevamento delle cadute lontano da casa. Abbate et al. [11] riportano un 100% del tasso di rilevamento delle cadute utilizzando un algoritmo basato su accelerometri che si trovano comunemente nei telefoni cellulari. Hanno addestrato i loro algoritmi in modo da scartare i falsi positivi generati da diverse attività comuni e raggiunto il 100% di specificità. Il play store ufficiale di Android offre attualmente applicazioni con queste funzionalità; Tuttavia, queste applicazioni forniscono poche o nessuna informazione sulla loro affidabilità. Una combinazione di sensori indossabili e telefoni cellulari è considerata da [12]. Il primo propone un sistema di monitoraggio delle cadute umane costituito da un'unità di sensori altamente portatile comprendente un accelerometro triaxis, un giroscopio triassiale e un magnetometro triassiale e un telefono cellulare per i dati di elaborazione, rilevamento delle cadute e messaggistica. In [13], vengono utilizzati telefoni cellulari dedicati. In questo caso gli accelerometri vengono utilizzati non solo per rilevare una caduta, ma anche per classificare automaticamente il tipo di caduta.

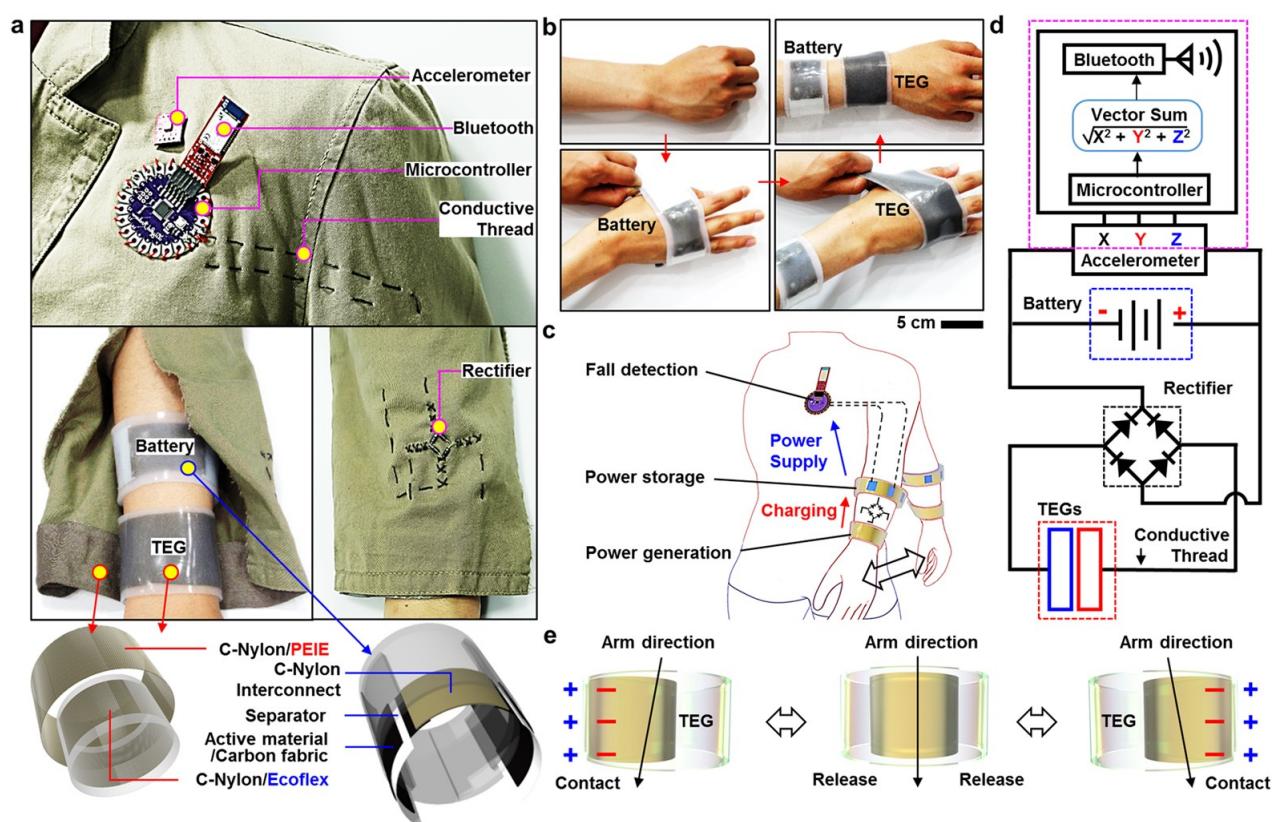


Figure 2.1: Dispositivi indossabili

## 2.2 Dispositivi di visione

La rilevazione delle cadute basata su telecamera è un'area in rapida evoluzione e il suo stato dell'arte sta continuamente migliorando. Ci sono numerosi sviluppi e tendenze nel campo che stanno prendendo il largo. Innanzitutto le tecnologie di elaborazione delle immagini stanno diventando sempre più avanzate e precise, rendendo possibile una rilevazione più affidabile delle cadute. A questo si aggiunge il fatto che l'utilizzo di modelli di intelligenza artificiale sta diventando sempre più comune nella rilevazione delle cadute con aprroccio vision based [14]. Questi modelli possono essere addestrati su dati di addestramento per rilevare e classificare i movimenti associate alle cadute. Lo sviluppo di sistemi intelligenti in questo contesto prevede una forte integrazione con altre tecnologie. Ad esempio, i sensori di movimento e i dispositivi indossabili possono fornire una copertura più completa e affidabile in aggiunta ai dispositivi di visione. Successivamente, la rilevazione della caduta può essere segnalata on edge oppure tramite applicazioni in Cloud, offrendo ai consumatori una maggiore flessibilità e scelta nella loro implementazione. Tra le tecniche basate sulla visione, ci sono numerosi approcci per affrontare il problema. Dal punto di visto dell'analisi del movimento ad esempio, vengono utilizzati algoritmi per monitorare i movimenti delle persone tramite i punti chiave dello scheletro umano e identificare eventuali cadute [15]. Le tecniche di rilevamento del corpo invece utilizzano tecniche di elaborazione delle immagini per identificare e tracciare le parti del corpo umano nei dati video. Successivamente attraverso delle tecniche di classificazione si utilizzano modelli di intelligenza artificiale addestrati su dati di addestramento per classificare i movimenti come cadute o non cadute. Altre tecniche riguardano ad esempio il tracking e il tracciamento per seguire le persone e i loro movimenti durante l'intero flusso del video [16].

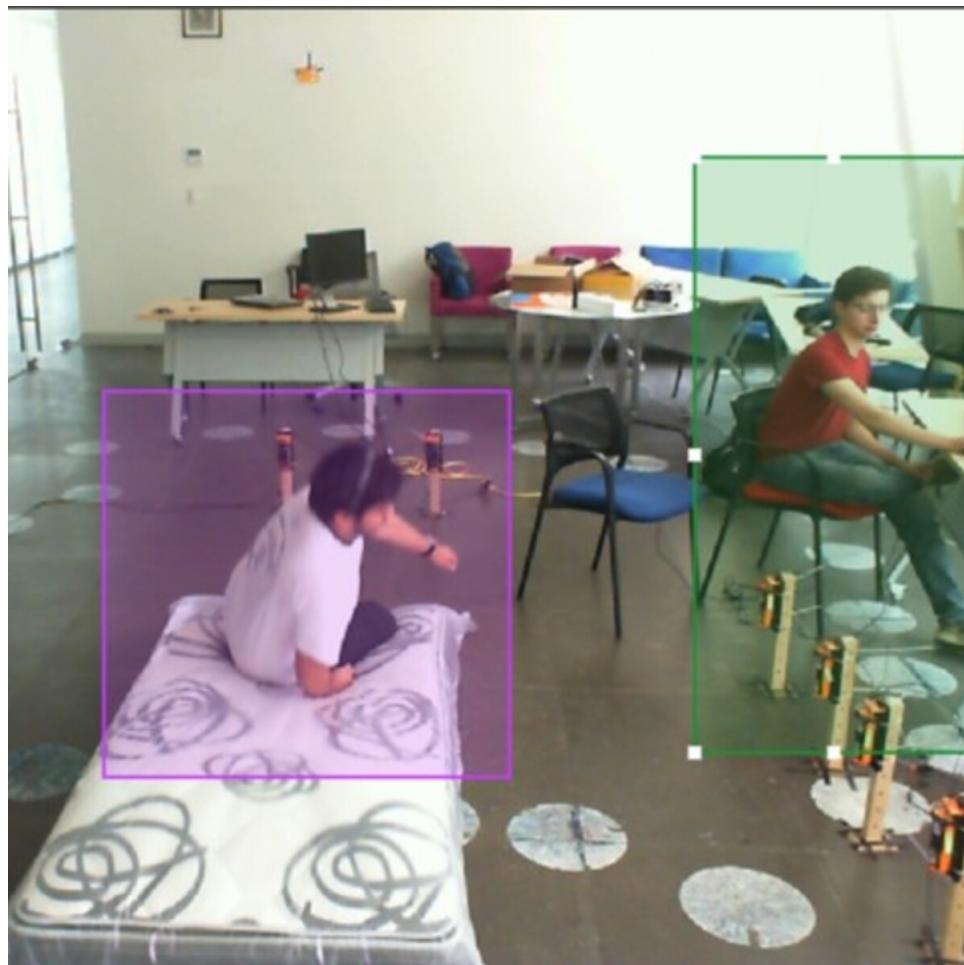


Figure 2.2: Vision based fall detection

In generale, le tecniche basate sulla visione sono molto efficaci nella rilevazione delle cadute, poiché possono utilizzare informazioni visive dettagliate e accurate per analizzare il movimento e identificare eventuali cadute. Tuttavia, possono anche essere soggette a falsi positivi e falsi negativi, a seconda della qualità e della quantità dei dati video disponibili.

### 2.2.1 Motion History Image

Per Motion History Image (MHI) [17] si intende una rappresentazione visiva utilizzata nella rilevazione delle cadute basata sulla visione. L'MHI rappresenta la quantità di movimento in un'immagine nel tempo, utilizzando una mappa di intensità in cui i pixel più scuri rappresentano i punti in cui c'è stato meno movimento nel tempo e i pixel più luminosi rappresentano i punti in cui c'è stato più movimento.

L'MHI viene utilizzato nella rilevazione della caduta poiché le cadute spesso causano un aumento della quantità di movimento in una scena, che può essere rilevato nell'MHI come un aumento dell'intensità in un'area specifica [18].

Per creare un MHI, prima si acquisisce una serie di immagini video e quindi si utilizza un algoritmo per identificare e tracciare i movimenti nell'immagine. L'MHI viene quindi creato accumulando informazioni sul movimento nel tempo e rappresentando queste informazioni come una mappa di intensità.

Per ricavare una precisa rappresentazione visiva della quantità di movimento in una scena video nel tempo, vengono accumulate informazioni sul movimento nel tempo e rappresentate queste informazioni come una mappa di intensità, in cui i pixel più scuri rappresentano i punti in cui c'è stato meno movimento nel tempo e i pixel più luminosi rappresentano i punti in cui c'è stato più movimento. Dalla mappa di intensità si cerca di individuare i punti in cui si nota un aumento dell'intensità in un'area specifica, il che significa un aumento della quantità di movimento nel video (potenziale caduta).

In generale, l'utilizzo di un MHI è una tecnica efficace nella rilevazione della caduta poiché permette di rappresentare la quantità di movimento in una scena nel tempo e di identificare i momenti in cui c'è stata una maggiore quantità di movimento, come potenziali

cadute. Tuttavia, come con qualsiasi tecnica basata sulla visione, può essere soggetto a falsi positivi e falsi negativi, a seconda della qualità e della quantità dei dati video disponibili.

L'algoritmo di Motion History Image per la rilevazione della caduta può essere suddiviso in tre fasi principali:

1. Acquisizione del video: la prima fase consiste nell'acquisire una serie di immagini video da una telecamera.
2. Tracciamento dei movimenti: la seconda fase consiste nel tracciare i movimenti nell'immagine, utilizzando un algoritmo di tracciamento del movimento come ad esempio Optical Flow o altri algoritmi di computer vision.
3. Creazione dell'MHI: la terza fase consiste nella creazione dell'MHI accumulando informazioni sul movimento nel tempo. Per ogni immagine del video, si identificano i movimenti e si accumulano informazioni sul movimento nel tempo, creando una mappa di intensità che rappresenta la quantità di movimento in una scena nel tempo.

Una volta creato l'MHI, l'algoritmo può essere utilizzato per identificare i momenti in cui c'è stata una maggiore quantità di movimento, come potenziali cadute. Ad esempio, si potrebbe utilizzare una soglia sull'intensità per identificare i punti nell'MHI che rappresentano un'elevata quantità di movimento, e quindi considerare questi punti come potenziali cadute.

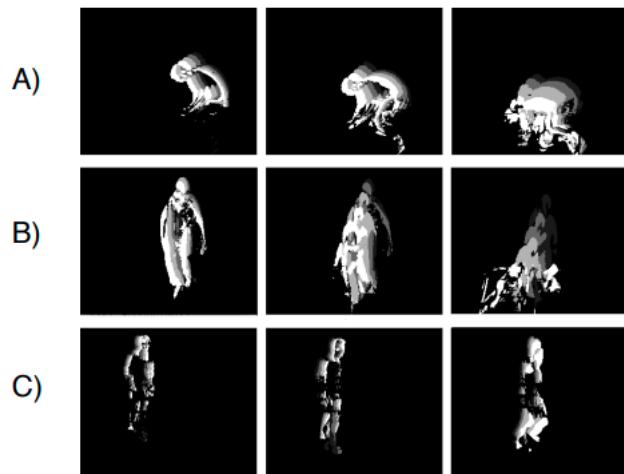


Figure 8: Shows generated MHIs. In A) a person is falling from a chair, B) a person is falling from standing position, C) a person walking around.

---

**Input :** A MHI  $mhi$ , the current image  $I_c$ , the previous image  $I_p$ , a duration  $\delta$ , and a threshold  $\xi$ .  
**Output:** An updated MHI  $mhi^*$ .

---

```

1  $I_d \leftarrow \text{DiffRGB}(I_c, I_p);$ 
2  $I_g \leftarrow \text{ConvertToGrayScale}(I_d);$ 
3  $I_t \leftarrow \text{BinaryThreshold}(I_g, \xi);$ 
4  $mhi^* \leftarrow \text{Decay}(mhi, \delta);$ 
5  $mhi^* \leftarrow \text{Add}(mhi^*, I_t);$ 
```

---

Figure 2.3: Motion history image

### **2.2.2 Pose Estimation**

La stima della posa (in inglese Pose Estimation [19]) è una tecnica di computer vision che si occupa di determinare la posizione e l'orientamento di un oggetto o di un individuo all'interno di un'immagine o un video. La posa viene rappresentata come un insieme di parametri che descrivono la posizione e l'orientamento dell'oggetto o dell'individuo.

La stima della posa è utilizzata in molte applicazioni di realtà aumentata, giochi, video e computer vision, tra cui la rilevazione del movimento umano, la tracciatura del volto e la modellizzazione 3D.

Esistono diversi algoritmi di stima della posa, come ad esempio l'algoritmo di tracciamento del movimento basato su optical flow, l'algoritmo di tracciamento del volto basato su modelli e l'algoritmo di stima della posa basato su deep learning. La scelta dell'algoritmo dipende dal tipo di applicazione e dalle specifiche esigenze di precisione e velocità.

La stima della posa può essere utilizzata come una componente importante per la rilevazione della caduta. In questo scenario, l'obiettivo è quello di analizzare la posa di un individuo nel tempo per identificare eventuali cadute.

Per questo scopo, l'algoritmo di stima della posa viene utilizzato per estrarre informazioni sulle caratteristiche della posa, come ad esempio la posizione delle articolazioni e l'orientamento del corpo, dalle immagini o dal video. Queste informazioni possono poi essere utilizzate per identificare eventuali cambiamenti nella posa che indicano una caduta.

Ad esempio, una caduta potrebbe essere identificata da un rapido cambiamento nella posizione delle articolazioni, come ad esempio la repentina flessione delle ginocchia o la perdita del controllo delle braccia.

La stima della posa può essere utilizzata insieme ad altri algoritmi, come ad esempio la motion history image, per aumentare la precisione della rilevazione della caduta e ridurre il tasso di falsi positivi. Tuttavia, come con qualsiasi tecnica basata sulla visione, l'algoritmo può essere soggetto a falsi positivi e falsi negativi, a seconda della qualità e della quantità dei dati video disponibili. Pertanto, è importante valutare attentamente i risultati dell'algoritmo e, se necessario, apportare modifiche o ottimizzare l'algoritmo per migliorare la precisione della rilevazione della caduta.

Per la fall detection, gli algoritmi di stima della posa possono utilizzare una combinazione di modelli pre-definiti e tecniche di apprendimento automatico. Ad esempio, un algoritmo potrebbe utilizzare un modello pre-definito delle articolazioni del corpo umano per identificare la posizione delle articolazioni nelle immagini o nei video. Queste informazioni possono poi essere utilizzate come input per un modello di apprendimento automatico, che potrebbe essere addestrato per identificare eventuali cambiamenti nella posa che indicano una caduta.

In alternativa, un algoritmo di stima della posa per la fall detection potrebbe utilizzare tecniche di apprendimento automatico end-to-end, che utilizzano una rete neurale per effettuare la stima della posa direttamente a partire dalle immagini o dai video senza la necessità di modelli pre-definiti. Queste tecniche possono essere addestrate su un vasto insieme di dati di formazione che includono diverse posizioni del corpo e diverse situazioni di caduta, in modo da acquisire una comprensione più profonda della posa e dei suoi cambiamenti.

In entrambi i casi, è importante che l'algoritmo di stima della posa sia in grado di analizzare con precisione la posa dell'individuo e identificare con precisione i cambiamenti nella posa che indicano una caduta. Questo richiede una combinazione di tecniche di

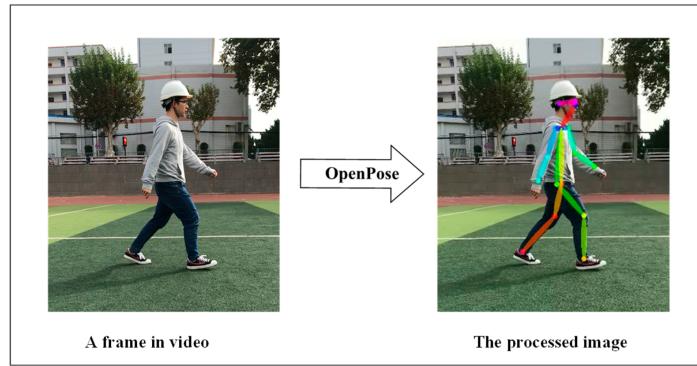


Figure 2.4: Pose estimation

elaborazione delle immagini e di apprendimento automatico che possono essere ottimizzate per ottenere la massima precisione nella rilevazione della caduta.

### 2.2.3 Reti Neurali Convoluzionali

Le reti neurali convolutionali (Convolutional Neural Networks, CNN) sono un tipo di reti neurali artificiali che sono state sviluppate per l'elaborazione di dati che hanno una struttura spaziale, come ad esempio immagini o segnali audio.

Le CNN utilizzano una combinazione di filtri convolutionali e funzioni di pooling per elaborare i dati a livello locale e costruire rappresentazioni sempre più complesse man mano che si avanza nella rete. I filtri convolutionali sono piccoli kernel che scorrono sull'input e calcolano le somme pesate di elementi locali per creare feature map più complesse. La funzione di pooling riduce la risoluzione spaziale dei dati per concentrarsi sulle informazioni più importanti e ridurre il numero di parametri da imparare.

Le reti neurali convolutionali possono essere addestrate utilizzando una vasta quantità di dati di formazione e tecniche di ottimizzazione come l'algoritmo di backpropagation. Queste reti sono state utilizzate con successo in molte applicazioni di elaborazione delle immagini, come la classificazione delle immagini, la rilevazione degli oggetti e la segmentazione delle immagini.

In uno strato convoluzionale di una rete neurale, un singolo neurone rappresenta un singolo filtro che viene applicato all'input. Questo filtro consiste in una matrice di pesi che viene moltiplicata per una porzione localizzata dell'input, che viene chiamata "receptive field". Questa operazione viene ripetuta per ogni porzione dell'input, utilizzando una tecnica nota come convolution, per produrre una "feature map".

Negli strati finali della rete ci sono i cosiddetti nodi, o neuroni, che lavorano insieme per elaborare l'input. Gli strati iniziali della rete utilizzano filtri convolutionali per estrarre caratteristiche a livello locale dall'input. Queste feature map vengono quindi passate attraverso strati di pooling che riducono la risoluzione spaziale dei

dati, mantenendo solo le informazioni più importanti.

Gli strati più avanzati della rete utilizzano una combinazione di tecniche di elaborazione per costruire rappresentazioni sempre più complesse dei dati. Questi strati possono essere costituiti da fully connected layers che effettuano una combinazione lineare di tutte le feature map precedenti, seguiti da funzioni di attivazione come la ReLU o la sigmoide per introducere non linearità nella rete.

La funzione del filtro e del neurone è quella di estrarre caratteristiche a livello locale dall'input, che possono essere utilizzate dagli strati successivi della rete per effettuare una classificazione o un'altra attività di elaborazione dei dati. I pesi del filtro vengono regolati durante il processo di addestramento della rete, in modo che la rete possa imparare a riconoscere le caratteristiche più importanti dei dati di formazione.

In generale, ciascun neurone in uno strato convolutionale lavora indipendentemente dagli altri, ma insieme costituiscono una rappresentazione complessiva delle caratteristiche del dato di input. Questa rappresentazione viene utilizzata dagli strati successivi della rete per produrre un'uscita più complessa.

Una volta che la rete è stata addestrata utilizzando un vasto set di dati di formazione, essa può essere utilizzata per elaborare dati di test. Il processo di addestramento consiste nel confrontare le previsioni della rete con i dati di formazione e nel regolare i pesi dei nodi per minimizzare l'errore. Questo processo viene ripetuto migliaia o milioni di volte per produrre una rete ben addestrata.

Le reti neurali convolutionali (ConvNets o CNNs) sono state ampiamente utilizzate per la rilevazione delle cadute basate su immagini e video. La loro architettura è adatta a gestire grandi quantità di dati e a riconoscere pattern complessi all'interno di questi dati.

Per la fall detection, le CNNs possono essere addestrate per riconoscere le posture associate alle cadute (ad esempio, una

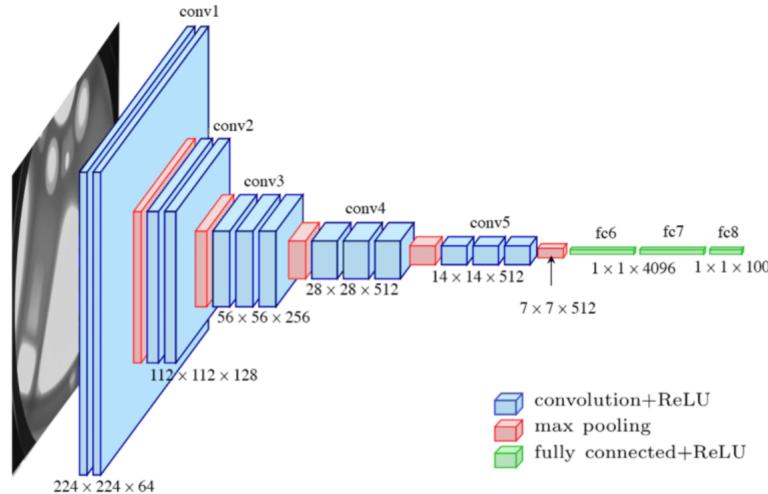


Figure 2.5: CNN Architecture

persona che cade all’indietro o in avanti) e distinguere queste posture da altre posture normali o sicure. Questo può essere ottenuto utilizzando una grande quantità di dati annotati di cadute e non-cadute, che vengono utilizzati per addestrare la rete a riconoscere queste posture.

Una volta addestrata, la CNN può essere utilizzata per analizzare il video in tempo reale e rilevare automaticamente le cadute. La precisione della rilevazione dipenderà dalla qualità dei dati di addestramento e dalla struttura della rete neurale scelta.

Inoltre, le reti neurali convolutionali possono essere combinate con altre tecniche di elaborazione delle immagini, come la pose estimation e la motion history image, per migliorare ulteriormente la precisione della rilevazione delle cadute.



# **Chapter 3**

## **Metodologie utilizzata**

Questa sezione riassume la metodologia utilizzata in questo studio per lo sviluppo di un sistema di rilevamento delle cadute umane in tempo reale basato su telecamera. Per quanto riguarda il modello di rilevamento, sono stati intrapresi due approcci diversi, uno basato su rete neurale convoluzionale (fallnet) e l'altro basato sull'algoritmo di motion history image (fdnet). Il primo approccio si caratterizza per il fatto che la classificazione delle cadute viene effettuata frame-by-frame. L'approccio basato su MHI invece, elabora la sequenza di immagini in batch.

Un approccio frame-by-frame e un approccio in batch sono due metodi diversi per elaborare i dati in un sistema di elaborazione del video o dell'immagine.

Un approccio frame-by-frame elabora ogni singolo fotogramma del video o immagine separatamente. Ciò significa che ogni fotogramma viene elaborato singolarmente e i risultati vengono restituiti immediatamente. Questo approccio è adatto per situazioni in cui i risultati devono essere disponibili il più presto possibile, ad esempio in sistemi di sorveglianza cadute in tempo reale.

Un approccio in batch, d'altra parte, elabora un insieme di fotogrammi o immagini alla volta. Ciò significa che vengono accumulate molte immagini o fotogrammi prima di eseguire l'elaborazione. Questo approccio è più adatto per situazioni in cui la velocità non

*Chapter 3 Metodologie utilizzata*

è così critica, ma è richiesta una maggiore potenza computazionale per elaborare molti dati contemporaneamente.

### *3.1 Tecnologie utilizzate*

#### **3.1 Tecnologie utilizzate**

Di seguito si presenteranno le tecnologie utilizzate per sviluppare il sistema di rilevamento cadute

### **3.1.1 TensorFlow**

TensorFlow [20] è una libreria open source di intelligenza artificiale sviluppata da Google. È utilizzata per costruire e addestrare modelli di apprendimento automatico e per eseguire operazioni matematiche su tensori, che sono array multidimensionali. TensorFlow viene utilizzato in molti ambiti, come la classificazione di immagini, la traduzione automatica e la previsione del tempo. Uno degli strumenti più utili forniti da Tensorflow è sicuramente la sua dashboard, chiamata TensorBoard. La TensorBoard è uno strumento basato sul Web per la visualizzazione, il monitoraggio e il debug delle esecuzioni di TensorFlow. Fornisce informazioni in tempo reale sulle tue corse di allenamento, tra cui precisione, perdita e altre metriche, oltre a visualizzare e confrontare i risultati di più corse. La dashboard di TensorFlow è anche in grado di visualizzare il grafico di calcolo e profilare le operazioni di TensorFlow. Può essere utilizzato in locale o in remoto ed è possibile accedervi tramite Colab. Fornisce informazioni in tempo reale sulle tue corse di allenamento, tra cui precisione, perdita e altre metriche, oltre a visualizzare e confrontare i risultati di più corse.

### 3.1 Tecnologie utilizzate

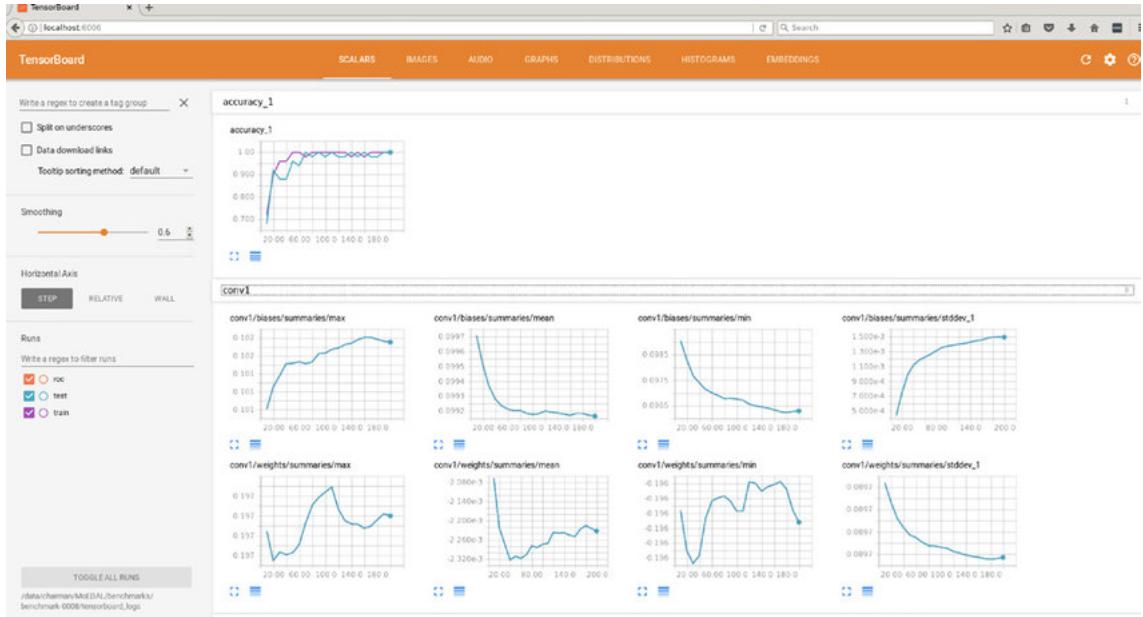


Figure 3.1: TensorBoard

#### 3.1.2 Google Colab

Google Colab [21] è un ambiente di sviluppo basato sul cloud che fornisce un Jupyter Notebook gratuito per l'esecuzione di codice Python. Colab è una piattaforma che rende semplice l'utilizzo di TensorFlow, PyTorch e altre librerie di machine learning senza dover installare nulla sul proprio computer. Con Colab, è possibile scrivere e eseguire codice, salvare e condividere documenti e utilizzare potenti risorse di calcolo come GPU e TPU.

Colab è particolarmente utile per la formazione, la ricerca e lo sviluppo di prototipi, in quanto offre una soluzione accessibile e conveniente per lavorare con grandi quantità di dati e modelli di machine learning.

Google Colab è integrato con Google Drive, che ti consente di salvare, condividere e collaborare sui tuoi progetti. Con Colab, puoi caricare i tuoi dati di addestramento e i tuoi notebook direttamente da Google Drive e puoi anche salvare i risultati dei tuoi progetti su Google Drive. Questo rende molto semplice la condivisione di progetti con altri, sia per la collaborazione che per la

*Chapter 3 Metodologie utilizzata*

presentazione. Inoltre, Google Drive offre spazio di archiviazione illimitato per i tuoi file, il che lo rende una soluzione ideale per la gestione dei dati e dei modelli di addestramento



Figure 3.2: Google Colab

### 3.1.3 Raspberry Pi

Il sistema di rilevamento delle cadute umane che è stato proposto in questa tesi è un dispositivo a basso costo basato sulla visione computerizzata, costituito da un computer embedded, quale la Raspberry Pi 3 Model B+ [22]. Questo sistema può essere installato su pareti o soffitti e avvia il rilevamento senza alcuna necessità di interazione manuale. La board Raspberry Pi è una serie di computer single-board a basso costo di dimensioni simili a quelle di una carta di credito sviluppati nel Regno Unito dalla Raspberry Pi Foundation. È stato creato per promuovere l'insegnamento della informatica di base nelle scuole e nei paesi in via di sviluppo. La Raspberry Pi può essere utilizzata per una varietà di scopi, come centro multimediale, console per il gaming, home theater PC, computer ad uso generale e molto altro ancora. Funziona con una varietà di sistemi operativi, tra cui Raspbian, un sistema operativo basato su Debian ottimizzato per Raspberry Pi. Il Raspberry Pi è conosciuto per la sua accessibilità, versatilità e semplicità d'uso, rendendolo una scelta popolare per appassionati, educatori e makers.

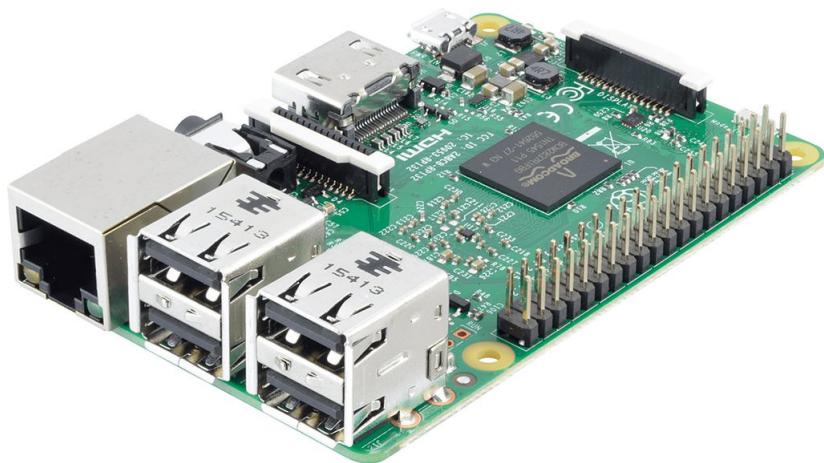


Figure 3.3: Raspberry Pi 3 b+

### **3.1.4 TensorFlow Lite**

TensorFlow Lite [23] è una versione leggera di TensorFlow, progettata per l'esecuzione di modelli di machine learning su dispositivi mobili e embedded con risorse limitate come smartphone, Raspberry Pi, e dispositivi Internet of Things (IoT). TensorFlow Lite fornisce un'interfaccia API per la creazione, l'addestramento e l'esecuzione di modelli di deep learning su dispositivi mobili con una minima latenza e una ridotta quantità di risorse di sistema. Ciò lo rende ideale per le applicazioni di realtà aumentata, l'elaborazione delle immagini e il riconoscimento vocale su dispositivi mobili.

TensorFlow Lite ha alcuni vantaggi rispetto a TensorFlow:

1. Leggero: TensorFlow Lite è progettato per funzionare su dispositivi con risorse limitate come smartphone e Raspberry Pi. Ha una minore quantità di risorse di sistema rispetto a TensorFlow.
2. Performance ottimizzate: TensorFlow Lite utilizza un'architettura di runtime ottimizzata per fornire prestazioni migliori rispetto a TensorFlow su dispositivi mobili.
3. Minima latenza: TensorFlow Lite è progettato per fornire risposte rapide ai comandi, rendendolo adatto per le applicazioni che richiedono una minima latenza, come la realtà aumentata e il riconoscimento vocale.
4. Supporto per la piattaforma: TensorFlow Lite è progettato per supportare una vasta gamma di piattaforme mobili, inclusi Android, iOS e Raspberry Pi, rendendolo accessibile a una vasta gamma di sviluppatori.
5. Interfaccia API semplificata: TensorFlow Lite fornisce un'interfaccia API semplificata per la creazione, l'addestramento e l'esecuzione di modelli di deep learning su dispositivi mobili.

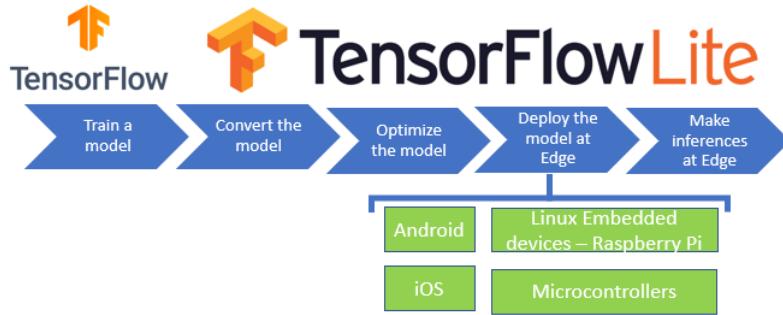


Figure 3.4: TF Lite

### 3.1.5 Fallnet

La fallnet è stata pensata come una rete neurale convoluzionale(CNN). Il modello viene utilizzato per l'elaborazione delle immagini e la classificazione delle stesse in maniera indipendente l'una dall'altra. La fallnet utilizza una serie di filtri che scorrono sull'immagine e catturano caratteristiche specifiche delle immagini. Questi filtri sono in grado di identificare forme, texture, oggetti e altre informazioni utili per la classificazione dell'immagine.

La rete è organizzata in strati, dove ogni strato utilizza i filtri per analizzare i dati. I filtri sono condivisi tra i diversi punti dell'immagine, che riducono la quantità di parametri che devono essere addestrati e migliorano la capacità di generalizzazione del modello.

Oltre ai filtri, sono presenti anche strati di pooling, che riducono la dimensione dei dati elaborati e aumentano la robustezza del modello. I dati vengono elaborati quindi attraverso una serie di strati di convoluzione e pooling, che vengono seguiti da strati densi, che forniscono la classificazione o la regressione finale.

Questo approccio rende molto efficace l'elaborazione delle immagini, poiché rende la rete in grado di catturare relazioni spaziali tra i pixel dell'immagine. Queste relazioni spaziali sono importanti per la comprensione delle immagini e possono essere utilizzate per

la classificazione delle immagini in cadute o non cadute.

La rete ottenuta è di semplice architettura ma risulta altrettanto facile da scalare e allenare, risultando quindi molto comoda da utilizzare su quantità di dati molto grandi.

### Convoluzione

La convoluzione [24] consiste nell'applicare un filtro su una porzione di dati, che viene utilizzato per estrarre caratteristiche specifiche dai dati. I filtri sono appresi durante il processo di addestramento del modello, in modo che possano identificare caratteristiche rilevanti per la classificazione o la regressione. La convoluzione è una operazione matematica utilizzata nei modelli di reti neurali convolutionali. In una rete neurale, la convoluzione viene utilizzata per estrarre caratteristiche a livello locale da un input, come un'immagine o un segnale. L'operazione consiste nel moltiplicare una porzione localizzata dell'input con un filtro, che è rappresentato da una matrice di pesi. Questa operazione viene ripetuta per ogni porzione dell'input, producendo una "feature map" che rappresenta la funzione filtrata dell'input. Il risultato finale è la riduzione della dimensionalità dell'input, ma al contempo la conservazione delle informazioni importanti. Ciò significa che la rete è in grado di identificare e utilizzare caratteristiche a livello locale all'interno dell'input, che possono essere combinate per formare una rappresentazione complessiva più alta. Questa rappresentazione viene quindi utilizzata dagli strati successivi della rete per effettuare una classificazione o un'altra attività di elaborazione dei dati.

L'algoritmo della convoluzione può essere descritto come segue:

1. Inizializzare una "feature map" vuota, che sarà utilizzata per memorizzare i risultati della convolution.
2. Definire una finestra di convoluzione (il filtro) con una matrice di pesi.
3. Scorrere la finestra sull'input, posizionandola su ogni porzione dell'input.
4. Per ogni posizione della finestra, moltiplicare i valori dell'input corrispondenti con i pesi del filtro.

5. Sommare i risultati delle moltiplicazioni e memorizzare il valore nella "feature map" corrispondente.
6. Ripetere i passaggi 4-5 per ogni posizione della finestra sull'input.
7. Ripetere i passaggi 3-6 per ogni filtro.

Il risultato di questi passi è una "feature map" che rappresenta la funzione filtrata dell'input. Questa "feature map" viene utilizzata come input per gli strati successivi della rete neurale. I pesi del filtro vengono regolati durante il processo di addestramento della rete, in modo che la rete possa imparare a riconoscere le caratteristiche più importanti dei dati di formazione.

### Max pooling

Il pooling è una tecnica utilizzata nelle reti neurali convolutionali per ridurre la dimensionalità dei dati e rendere la rete più robusta alle variazioni di posizione e scala delle caratteristiche.

Esistono diverse tecniche di pooling, ma la più comune è il max pooling [25]. In questo metodo, viene definita una finestra di pooling che scorre sulla "feature map" prodotta dalla convolution. Ad ogni posizione della finestra, viene selezionato il valore più alto e memorizzato nella "feature map" di pooling. Questo processo viene ripetuto per tutte le posizioni della finestra, portando a una "feature map" ridotta che rappresenta solo le informazioni più importanti della "feature map" originale.

Il pooling serve a ridurre la dimensionalità dei dati e a ridurre l'overfitting (quando un modello è troppo aderente ai dati di formazione), rendendo la rete più robusta alle variazioni di posizione e scala delle caratteristiche. Inoltre, il pooling rende la rete più veloce e meno computazionalmente intensiva, poiché riduce il numero di parametri che devono essere addestrati.

La tecnica di max pooling utilizza la seguente formula matematica:

$$\text{MaxPooling}(xi, j) = \max(xi, j)$$

dove  $xi, j$  è un elemento della "feature map" originale e  $\text{MaxPooling}(xi, j)$  è il corrispondente elemento nella "feature map" di pooling.

L'algoritmo funziona nel seguente modo:

Si definisce una finestra di pooling con una determinata dimensione (ad esempio 2x2 o 3x3). La finestra viene spostata sulla "feature map" originale, scorrendo su di essa in modo sincrono. Ad ogni posizione della finestra, viene selezionato il valore più alto e memorizzato nella "feature map" di pooling. Questo processo viene ripetuto per tutte le posizioni della finestra, portando a una "feature map" ridotta. È importante notare che la finestra di

### *Chapter 3 Metodologie utilizzata*

pooling viene generalmente spostata con uno "stride" predefinito (ad esempio, di 2 pixel), che determina il numero di pixel che verranno saltati durante lo scorrimento.

Il pooling a massimo è solo una delle tecniche di pooling disponibili, ma è la più comune e viene utilizzata in molte reti neurali convolutionali di successo.

### 3.1.6 FDNet

La motion history image (MHI) è una rappresentazione del movimento che viene utilizzata comunemente in computer vision per la rilevazione del movimento. La formula matematica per la creazione di una motion history image per ogni frame del video è la seguente:

- Inizializzare l'immagine MHI a zero:

$$MHI(x, y) = 0, \text{ per tutti i } x, y$$

- Estrarre il frame di differenza:

$$diff = abs(current\_frame - previous\_frame)$$

- Sogliare l'immagine di differenza:

$$diff = (frame\_diff > threshold)$$

- Aggiornare l'immagine MHI:

$$MHI(x, y) = max(MHI(x, y) * decay, frame\_diff(x, y))$$

Dove:  $decay$  è un fattore di decadimento che determina come la vecchia informazione sul movimento viene sbiadita nel tempo.  $threshold$  è una soglia per determinare quale parte del frame di differenza è considerata come movimento.

Questa formula crea un'immagine che rappresenta la durata e intensità del movimento in un particolare punto nello spazio. Più a lungo è stato presente il movimento in un punto, più scura sarà la sua rappresentazione nell'immagine MHI.

La rete neurale convoluzionale e l'algoritmo di Motion History Image sono entrambi utilizzati per la rilevazione di cadute, ma ci sono alcune differenze importanti tra loro.

La CNN è un tipo di rete neurale artificiale che è stata progettata per lavorare con immagini e video. Utilizza una serie di filtri per estrarre caratteristiche dalle immagini e poi utilizza queste caratteristiche per effettuare la classificazione. La CNN è generalmente più precisa rispetto all'MHI, ma anche più computazionalmente intensiva.

L'MHI è un algoritmo più semplice che utilizza una sola immagine per rappresentare il movimento nel tempo. Crea un'immagine che mostra l'accumulo di movimento nel tempo e quindi utilizza questa immagine per rilevare le cadute. L'MHI è meno preciso rispetto alla CNN, ma anche meno computazionalmente intensivo.

MHI non è perfetto e può essere soggetto a errori. È inoltre limitato dalla risoluzione della sequenza video, poiché l'MHI sarà limitato dalla risoluzione dei fotogrammi. Inoltre, MHI può essere computazionalmente costoso, poiché richiede una grande quantità di potenza di elaborazione per creare l'MHI. MHI è anche limitato dal numero di frame che possono essere utilizzati, poiché MHI sarà limitato dal numero di frame che possono essere elaborati. Inoltre, MHI può essere influenzato dal rumore, come le vibrazioni della fotocamera o i cambiamenti di illuminazione

### 3.2 Dataset utilizzati

Sono stati utilizzati due dataset differenti. Il primo di chiama "UR Fall Detection Dataset" [26], e consiste in 140 video totali proposti sia in forma frontale (chiamata "cam0") che pavimentale (chiamata "cam1"). In particolare, per entrambe le cam vi sono 30 sequenze di caduta e 40 sequenze che vengono definite activity day living, ovvero eventi quotidiani in cui le persone assumono posizioni naturali da non confondere con cadute (es. movimenti per sedersi, movimenti in cui ci si china per raccogliere un oggetto o stendersi sul letto). Il dataset fornisce anche dati sensoriali come quelli relativi all'accelerometro, tuttavia sono stati utilizzati soltanto i dati utili per l'approccio camera-based.

Il secondo dataset [27] è stato usato per estendere i dati del modello frontale e comprende 4 diversi ambienti di esecuzione cadute. I quattro ambienti sono: Casa, ufficio, sala caffè, e aula di lezione. Grazie all'estensione di questo nuovo dataset, la rete acquisisce importanti capacità di generalizzazione. Non solo comprende che le cadute sono indipendenti dalle caratteristiche dell'ambiente in background, ma concepisce diverse modalità e movimenti di caduta da persone diverse.

*Chapter 3 Metodologie utilizzata*

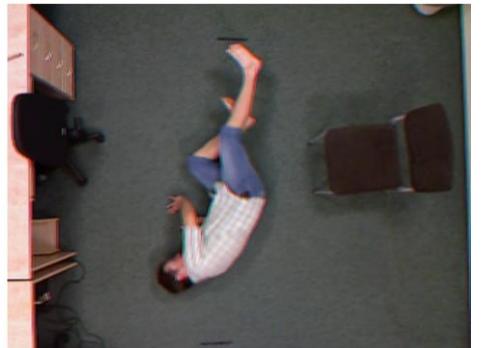
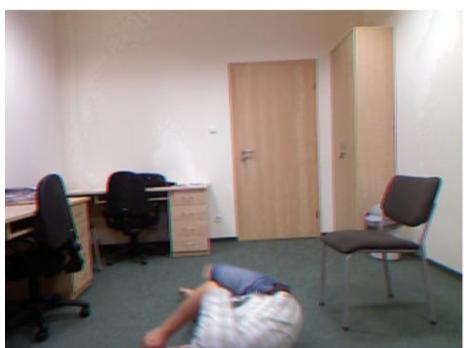
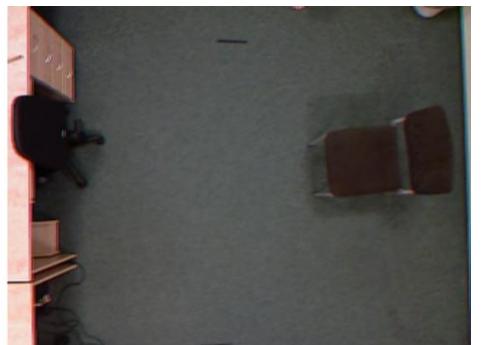


Figure 3.5: UR Fall Dataset

### 3.2 Dataset utilizzati

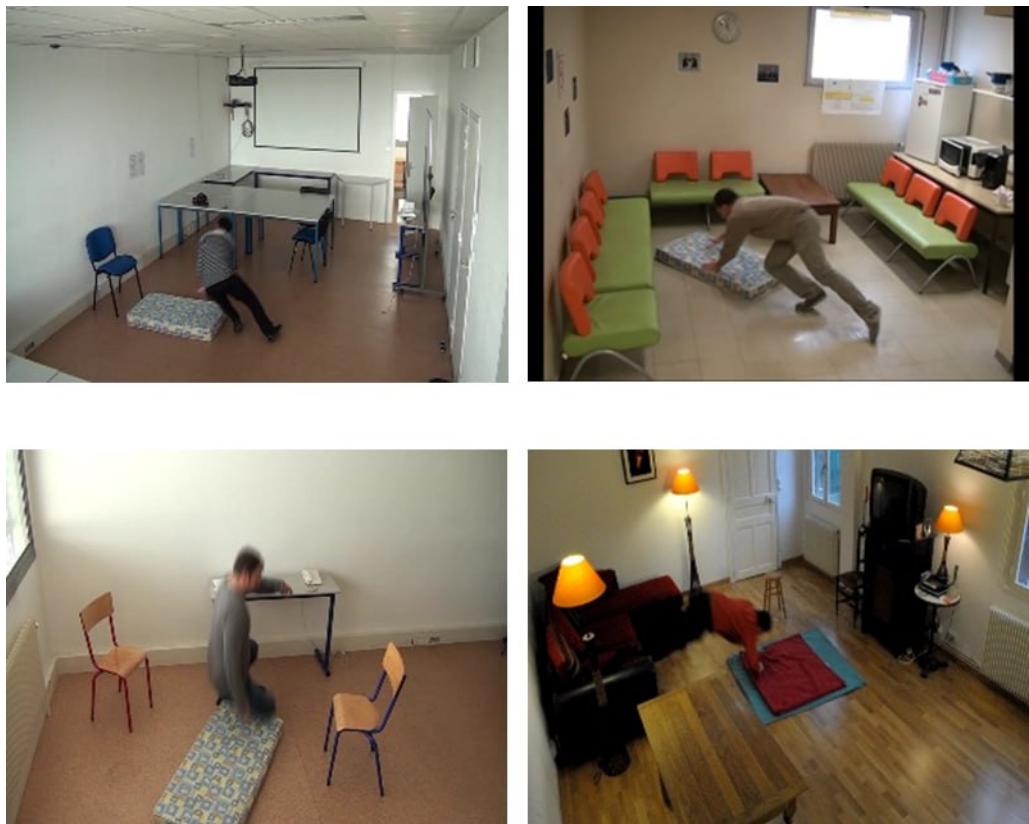


Figure 3.6: 4 ambienti di Fall Dataset

### 3.3 Architettura delle reti

#### Fallnet

La Figura mostra l'architettura complessiva della fallnet che ha tre componenti principali: il livello di convoluzione, il livello di maxpooling e livello di attivazione(fully connected). La rete è quindi un modello CNN che utilizza rappresentazioni visive basate su l'immagine RGB e la segmentazione e apprende le features di alto livello per il riconoscimento delle cadute. I singoli componenti della struttura concettuale in dettaglio sono mostrati di seguito.

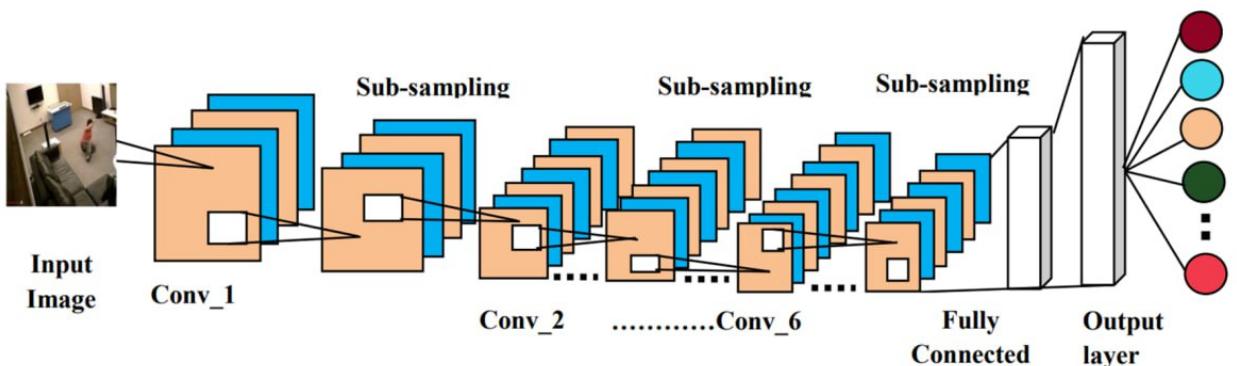


Figure 3.7: Architettura fallnet

Il livello di convoluzione effettua un'operazione matematica che fa scorrere un filtro sui dati di input (come un'immagine), calcolando un prodotto di punti tra le voci del filtro e l'input, producendo una feature map di output condensata. Nelle reti neurali convoluzionali (ConvNets), i livelli convoluzionali elaborano i dati di input utilizzando più filtri, generando più mappe di feature che catturano diversi aspetti dei dati di input.

Come funzione di attivazione è stata utilizzata la ReLU (Rectified Linear Unit). Si tratta di una funzione di attivazione ampiamente utilizzata nelle reti neurali, in particolare nelle ConvNets. Sostituisce tutti i valori negativi nei dati di input con zero e lascia invariati i valori positivi. Ciò introduce la non linearità nella

### *3.3 Architettura delle reti*

rete, consentendole di apprendere relazioni complesse tra input e output. La semplicità e l'efficacia dell'attivazione di ReLU lo hanno reso una scelta popolare per le attività di deep learning.

Il max pooling è un'operazione di down-sampling utilizzata per ridurre le dimensioni spaziali delle feature maps, mantenendo le informazioni più importanti. Questo ha diversi vantaggi per l'elaborazione dell'informazione: 1. Riduce il numero di parametri e il costo computazionale. 2. Aumenta l'invarianza della rete a piccole traslazioni e deformazioni nei dati di input. 3. Rende la rete più robusta ai cambiamenti spaziali e al rumore nei dati di input.

Il max pooling viene applicato dopo una serie di livelli convoluzionali e di attivazione, contribuendo a formare una rappresentazione gerarchica dei dati di input.

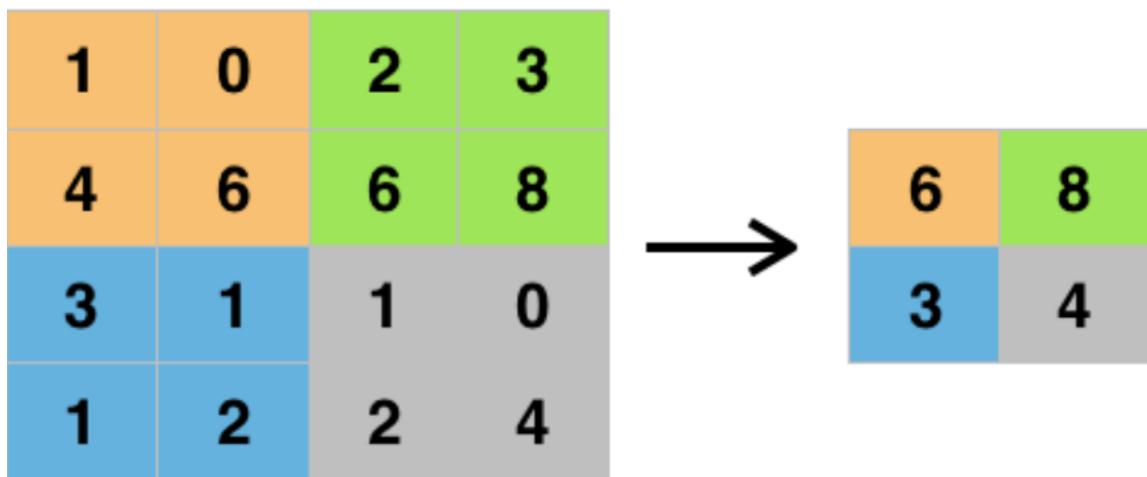


Figure 3.8: Esempio max pooling

Table 3.1: Architettura della Fallnet

Layer Type	Filter and Stride	Details	Output Shape
Conv1	3x3 and s=1	Conv1(16)	225,225,16
Activation	ReLU		225,225,16
MaxPooling		Pooling Size(2,2)	127,127,16
Conv2	3x3 and s=1	Conv2(16)	127,127,16
Activation	ReLU		127,127,16
MaxPooling		Pooling Size(2,2)	63,63,16
Conv3	3x3 and s=1	Conv3(32)	63,63,32
Activation	ReLU		63,63,32
MaxPooling		Pooling Size(2,2)	31,31,32
Conv4	3x3 and s=1	Conv4(32)	31,31,32
Activation	ReLU		31,31,32
MaxPooling		Pooling Size(2,2)	15,15,32
Conv5	3x3 and s=1	Conv5(64)	15,15,32
Activation	ReLU		15,15,64
MaxPooling		Pooling Size(2,2)	3,3,64
Conv6	3x3 and s=1	Conv6(64)	7,7,64
Activation	ReLU		7,7,64
MaxPooling		Pooling Size(2,2)	3,3,64
Flatten	Flatten to a vector		96,756
Dense	Dense Input=256		256
Dense	Input Classes=1		1
Activation	Sigmoid		1

Di seguito una rappresentazione grafica di come avviene la convoluzione in ognugno degli strati della rete:

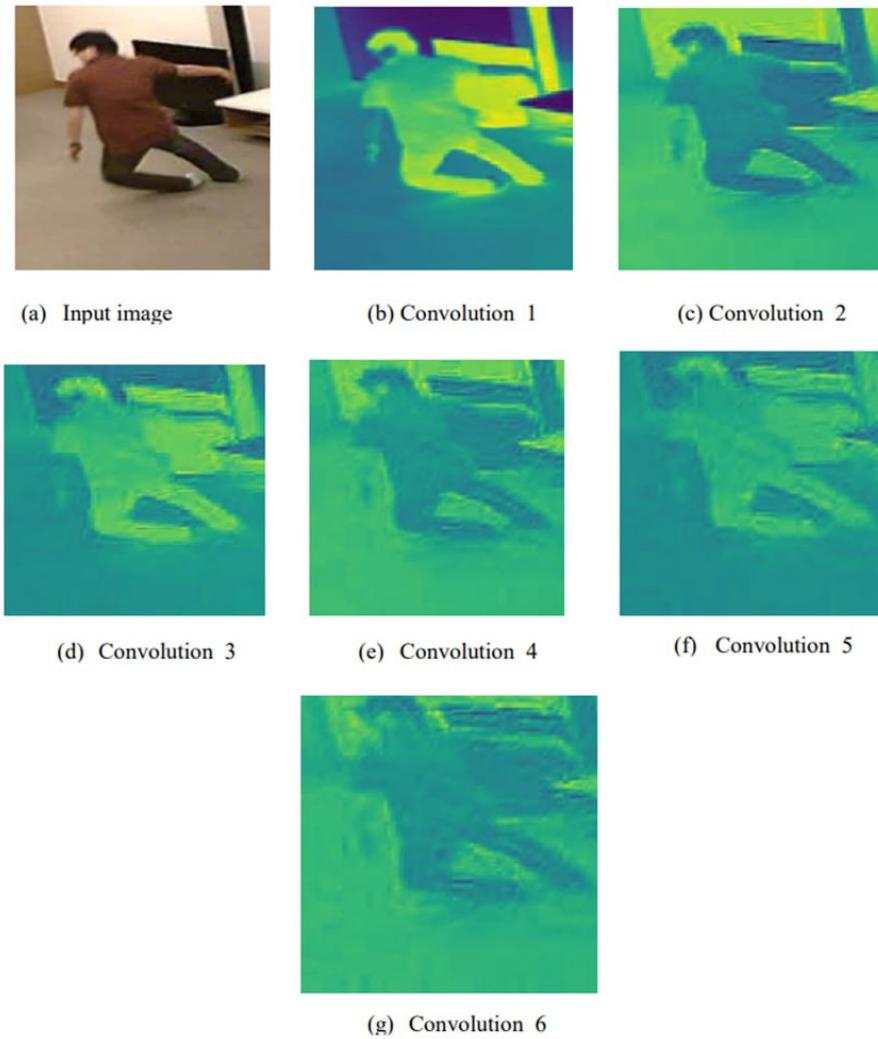


Figure 3.9: Convoluzioni della fallnet

Durante la progressione attraverso gli strati della rete neurale convolutionale, la convoluzione viene applicata sui dati di input con filtri diversi, che vengono imparati durante l'addestramento. La convoluzione produce una mappa di caratteristiche, che rappresenta le caratteristiche specifiche del dato di input che sono state rilevate dal filtro. Questa mappa di caratteristiche viene quindi inviata a un'altra serie di strati di attivazione e di pooling, che a loro volta la elaborano e la riducono in modo che la rete

### *3.3 Architettura delle reti*

possa concentrarsi sulla rappresentazione più importante del dato di input. Ciò continua per ogni strato successivo della rete, finché non si ottiene una rappresentazione finale del dato di input, che viene utilizzata per la classificazione

### Motion history image

Ogni volta che una persona cade, la caduta sarà associata a un grande cambiamento nel movimento. Quindi è importante estrarre e tenere traccia delle informazioni sul movimento. Il risultato di MHI è un'immagine che memorizza informazioni sull'attualità del movimento. I pixel più luminosi forniscono informazioni sul movimento più recente nella sequenza di immagini e raccontano così come la persona si è mossa nel corso di un'azione. Si tratta di una sequenza di immagini in una certa finestra temporale elaborate a cui applichiamo un valore di soglia per la generazione delle predizioni. Per quanto riguarda il calcolo dell'intensità di ogni pixel viene usata una funzione che rappresenta la storia temporale del movimento in quel determinato punto: Se il pixel è coinvolto nel movimento:  $I(x, y, t) = tau$

altrimenti:  $I(x, y, t) = max(I(x, y, t - 1) - 1.0)$

La rete che processa le immagini come MHI ha come backbone una mobilenet v2 [28] collegata ad uno strato fully connected per la classificazione.

La MobileNetV2 è una architettura di rete neurale che si concentra sulla riduzione della complessità computazionale e sulla scalabilità del modello per l'utilizzo in dispositivi mobili con limitate risorse di elaborazione. È una versione migliorata di MobileNetV1 e utilizza una nuova architettura di blocco di costruzione che migliora la sua precisione e le prestazioni.

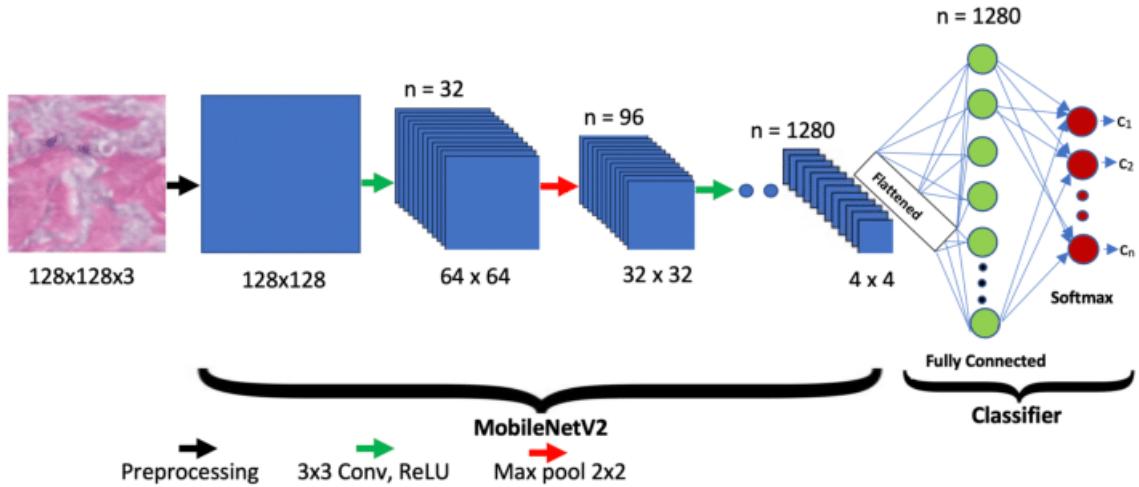


Figure 3.10: Mobilenet v2 architecture

L’architettura di MobileNetV2 è basata su una combinazione di convolution layer, depthwise convolution layer e layer di pooling. Utilizza una serie di blocchi di costruzione che incorporano questi strati in modo da ottenere un modello compatto ma preciso.

Uno dei principali elementi distintivi di MobileNetV2 è l’utilizzo di un blocco di costruzione chiamato "Inverted Residual", che consiste in una serie di strati di convolution profondi con una struttura di shortcut per mantenere la dimensionalità dei dati di input. Questa architettura permette una maggiore espansione della capacità di rappresentazione del modello mantenendo allo stesso tempo una bassa complessità computazionale.

In generale, l’architettura è ottimizzata per garantire una buona precisione in presenza di limitate risorse di elaborazione, rendendola adatta per un’ampia gamma di applicazioni su dispositivi mobili e IoT.

### **3.4 Fase di training e test**

Passando ora all’addestramento delle reti, la fall detection è stata trattata come un problema di classificazione binaria (fall/not fall). La classificazione viene eseguita frame-by-frame.

Adam (Adaptive Moment Estimation) è un algoritmo di ottimizzazione utilizzato nell’addestramento dei modelli di apprendimento automatico, in particolare nei modelli di deep learning. Funziona adattando i valori dei parametri del modello durante l’addestramento in modo da minimizzare la funzione di perdita. Adam è una combinazione di due tecniche di ottimizzazione: SGD (Stochastic Gradient Descent) e RMSProp. Si adatta automaticamente alle proprietà dei dati durante l’addestramento e utilizza informazioni sulla media del gradiente e sulla varianza del gradiente per ottenere un’aggiornamento più preciso dei parametri. Questo lo rende una scelta popolare tra gli sviluppatori di deep learning, in quanto spesso converge più velocemente rispetto ad altri algoritmi di ottimizzazione.

### 3.4 Fase di training e test

Di seguito i grafici dei risultati per le 15 epoche:

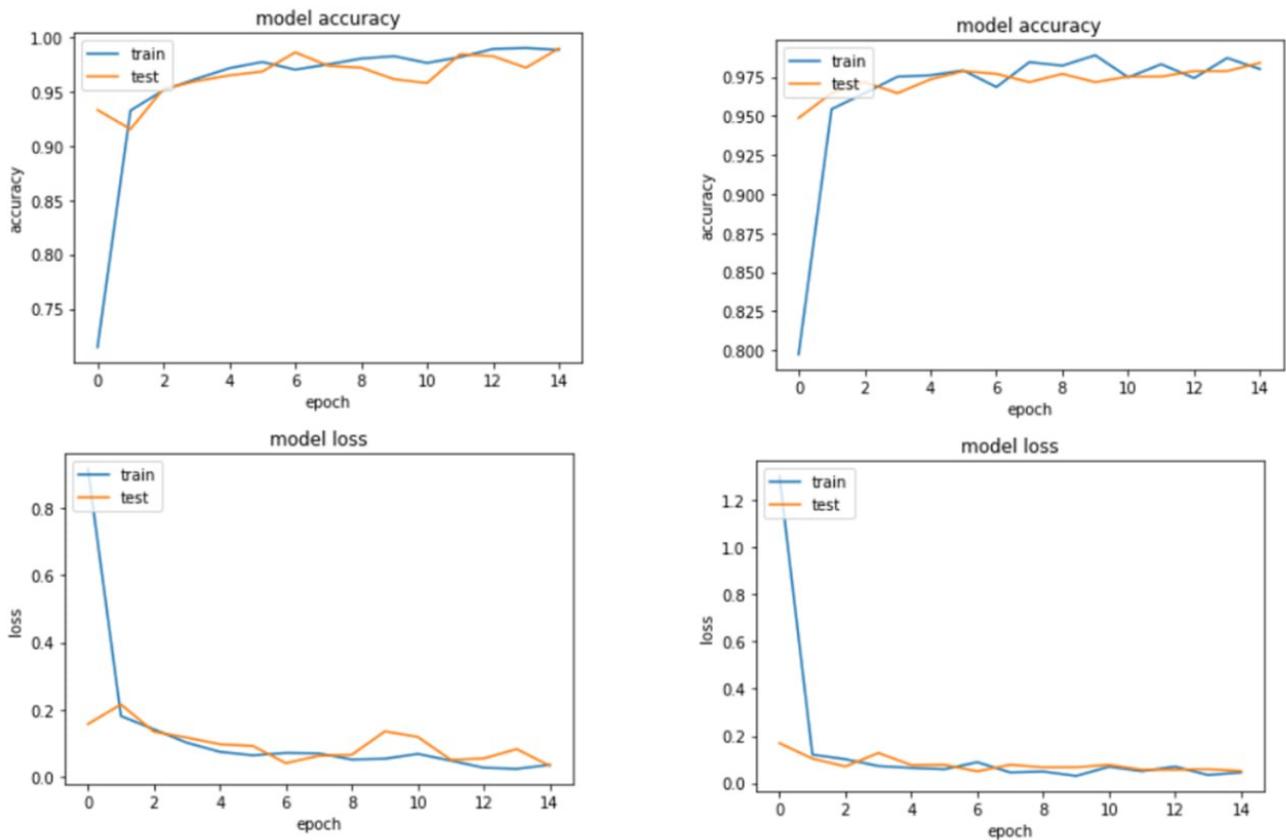


Figure 3.11: Valutazione addestramento

### Chapter 3 Metodologie utilizzata

Qui di seguito invece si illustra un insieme di dati di esempio che sono stati classificati dalla rete nella fase di test:

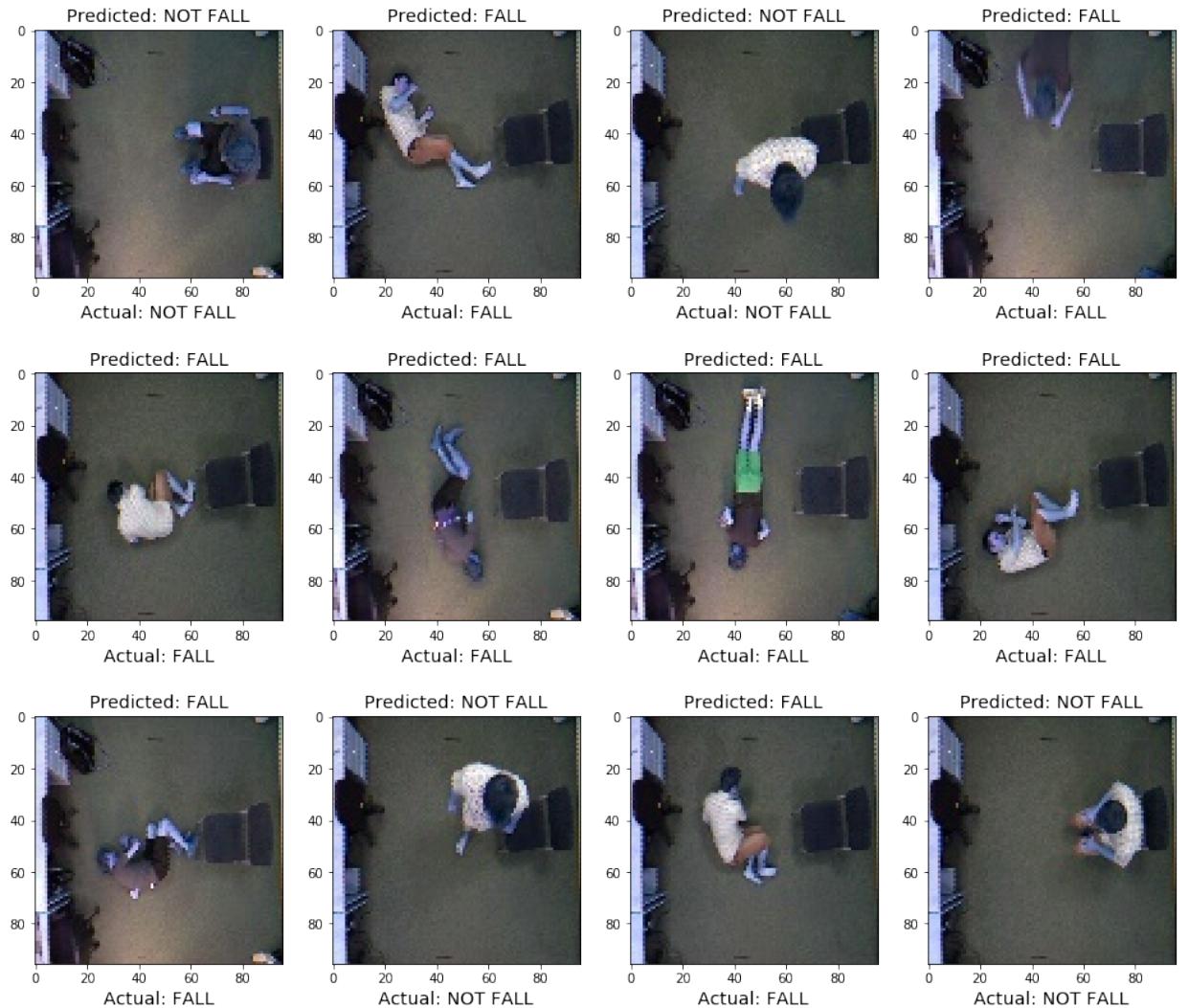


Figure 3.12: Predizioni della rete

Per quanto riguarda il modello basato su MHI, è stato riscontrato un valore del 97% in accuracy su un piccolo set di dati di test. La rete riceve in ingresso un batch di 32 immagini che processa e restituisce in output un valore numerico binario che indica se nella sequenza ricevuta vi è una caduta o meno.

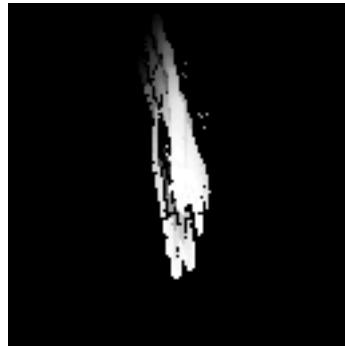


Figure 3.13: Processamento immagine in MHI

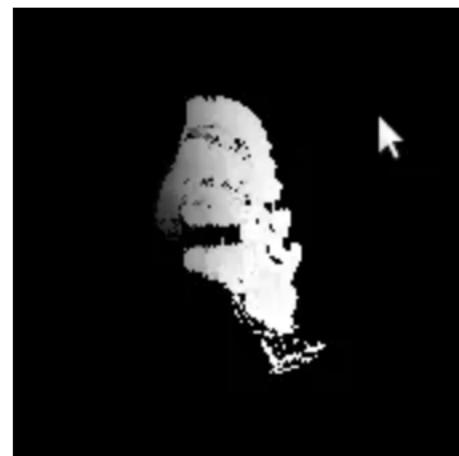
Il processamento in batch delle immagini è stato reso disponibile grazie all'utilizzo delle utility di PyTorch chiamate DataLoader e ImageFolder. Un DataLoader in Python è una classe helper della libreria PyTorch che fornisce un modo conveniente per caricare e gestire i set di dati per l'addestramento e il test dei modelli di deep learning. Funge da wrapper attorno a un oggetto del set di dati, consentendo di raggruppare, mescolare e parallelizzare facilmente il processo di caricamento. Il DataLoader fornisce anche molte altre funzionalità come il multi-threading e il batch automatico, che lo rendono uno strumento utile per velocizzare il caricamento e l'elaborazione dei dati. ImageFolder è una classe di set di dati nella libreria PyTorch per caricare un set di dati di immagini da una struttura di directory che segue una particolare convenzione, con una cartella per classe. La struttura del folder deve essere nota, *root* è la directory radice contenente tutte le classi, *fall* e *not fall* sono sottodirectory contenenti rispettivamente immagini di cadute e non cadute. La classe ImageFolder restituisce quindi una tupla dell'immagine e la relativa etichetta (l'indice della cartella

### *Chapter 3 Metodologie utilizzata*

della classe). Ciò è utile per caricare i set di dati di classificazione delle immagini, in cui si desidera assegnare un’etichetta a ciascuna immagine in base alla cartella in cui si trova. La classe DataLoader può quindi essere utilizzata per raggruppare e mescolare facilmente i dati da un set di dati di tipo ImageFolder.

### 3.4 Fase di training e test

Qui di seguito vengono presentati due esempi di motion history image. Il primo esempio riguarda il caso di una sequenza di caduta, mentre l'altro caso riguarda una sequenza di immagini che non soddisfano i requisiti di caduta:



Fall



Not Fall

Figure 3.14: Predizioni della FDNet

### 3.4.1 PushBullet

Pushbullet [29] è un'applicazione multi-piattaforma che facilita la condivisione rapida e semplice di informazioni tra i dispositivi. È possibile inviare file, link, note, immagini e altro ancora da un dispositivo all'altro semplicemente utilizzando l'app. Inoltre, l'app supporta il mirroring delle notifiche, il che significa che è possibile visualizzare le notifiche del proprio smartphone sul computer e rispondere alle notifiche di messaggistica direttamente.

E' possibile utilizzare Pushbullet con un Raspberry Pi e Python. Per farlo, è necessario installare il client API di Pushbullet, disponibile come pacchetto chiamato *pushbullet.py*. Questo pacchetto fornisce un'interfaccia semplice e facile da usare per interagire con l'API Pushbullet e inviare push da un Raspberry Pi utilizzando il codice Python.

Una volta installato il pacchetto "pushbullet.py", sono stati eseguiti i seguenti passi per iniziare a usare Pushbullet su Raspberry Pi:

1. Registrazione di un account Pushbullet e ottenimento di una chiave API
2. Installazione del pacchetto *pushbullet.py* con pip
3. Creazione di un client API Pushbullet utilizzando la propria chiave API
4. Utilizzo del client per inviare push nel momento in cui viene rilevata una caduta

### 3.4 Fase di training e test

Di seguito un'illustrazione di una segnalazione aperta dalla Raspberry Pi appena rilevata la caduta:

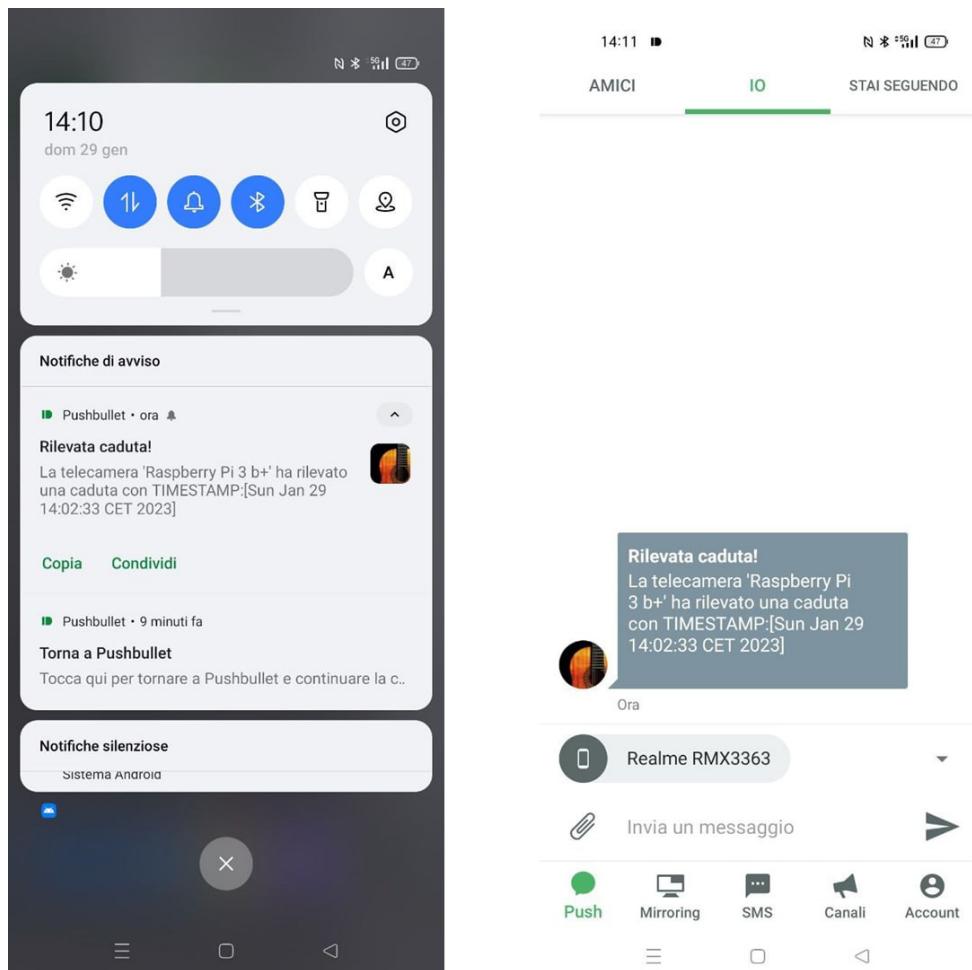


Figure 3.15: Segnalazione caduta da PushBullet



# **Chapter 4**

## **Risultati ottenuti**

I risultati ottenuti sono stati valutati attraverso l'utilizzo delle principali metriche: accuracy, precision, recall, f1 score.

### Accuracy

Nel machine learning, l'accuracy è una metrica comunemente usata per valutare le prestazioni di un modello. È il numero di previsioni corrette fatte dal modello diviso per il numero totale di previsioni. In altre parole, è la frazione di istanze nel set di test che sono state classificate correttamente dal modello.

L'accuracy può essere una buona metrica da utilizzare quando le classi del dataset sono bilanciate, cioè hanno approssimativamente lo stesso numero di istanze. Nei casi in cui le classi sono sbilanciate, tale valore può essere fuorviante, in quanto potrebbe dare una falsa sensazione di buone prestazioni. In questi casi, altre metriche come precision, recall e F1-score potrebbero essere più appropriate.

Vale la pena notare che l'accuracy non è sempre la metrica migliore da utilizzare, poiché potrebbe non cogliere altri aspetti importanti delle prestazioni di un modello, come la sua capacità di fare previsioni corrette per classi rare o la sua capacità di identificare modelli sottili nei dati. In questi casi, è importante scegliere una metrica di valutazione appropriata, adatta al problema che si sta cercando di risolvere.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Figure 4.1: Formula Accuracy

## Precision

La precision è una metrica comunemente utilizzata per valutare le prestazioni di un modello, soprattutto nei compiti di classificazione. La precision è definita come il numero di predizioni vere positive fatte dal modello diviso per il numero totale di predizioni positive fatte dal modello. Un vero positivo è una previsione in cui il modello identifica correttamente una classe positiva e una previsione positiva è una previsione che il modello ha assegnato alla classe positiva.

La precisione è una metrica utile quando l'obiettivo del modello è identificare istanze di una certa classe e si vogliono evitare previsioni falsi positivi. Ad esempio, in un compito di diagnosi medica, la precisione sarebbe una metrica importante, in quanto catturerebbe la proporzione di pazienti che hanno la malattia e sono correttamente identificati come positivi dal modello.

La precisione può essere combinata per ottenere una valutazione più completa delle prestazioni del modello. La metrica F1 score è una metrica comunemente utilizzata in combinazione con la precisione ed è particolarmente utile nei casi in cui le classi del set di dati sono sbilanciate.

$$Precision = \frac{TP}{TP + FP}$$

Figure 4.2: Formula Precisione

### Recall

La recall (nota anche come sensibilità) è una metrica comunemente utilizzata per valutare le prestazioni di un modello, soprattutto nei compiti di classificazione. La recall è definita come il numero di predizioni vere e positive fatte dal modello diviso per il numero totale di istanze nella classe positiva. Un vero positivo è una previsione in cui il modello identifica correttamente una classe positiva.

Tale metrica risulta utile quando l'obiettivo del modello è quello di identificare il maggior numero possibile di istanze di una certa classe e si vogliono evitare le previsioni false negative. Per esempio, in un'attività di rilevamento delle frodi, la recall sarebbe una metrica importante, in quanto catturerebbe la percentuale di transazioni fraudolente che sono correttamente identificate come tali dal modello.

Anche qui possiamo combinare questo valore con altre metriche come la precisione per ottenere una valutazione più completa delle prestazioni del modello.

$$Recall = \frac{TP}{TP + FN}$$

Figure 4.3: Formula Recall

## F1 score

L'F1-score (noto anche come F-score o F-measure) è una metrica comunemente utilizzata in machine learning. La metrica rappresenta la media armonica fra precisione e recall e fornisce un singolo valore che riassume l'equilibrio tra precisione e recall.

La metrica f1 score è particolarmente utile nei casi in cui le classi del dataset sono sbilanciate, in quanto fornisce una valutazione più completa delle prestazioni del modello rispetto all'utilizzo della sola accuratezza, precisione o recall. Un F1 score elevato indica che il modello ha un buon equilibrio tra precisione e recall e che sta facendo un numero elevato di previsioni vere positive e un numero ridotto di previsioni false positive.

$$F1 \text{ score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Figure 4.4: Formula f1 score

### Matrice di confusione

La matrice di confusione è uno strumento utilizzato in statistica e machine learning per valutare la performance di un modello di classificazione. Essa rappresenta graficamente le interazioni tra le previsioni del modello e i veri valori delle classi, permettendo di calcolare importanti metriche come la precisione, l'accuratezza e il recall. La matrice di confusione aiuta a identificare eventuali problemi di classificazione, come la confusione tra classi simili o la tendenza del modello a classificare tutte le osservazioni in una sola classe.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 4.5: Matrice di confusione

## 4.1 Valutazione modelli

Di seguito vengono riportate le metrice e la matrice di confusione del modello fallnet:

Table 4.1: Metriche del modello

Modello	Accuracy	Precision	Recall	F1_score
Fallnet	96.44%	84.18%	73.68%	78.58%
FDNet	93.17%	93.13%	100%	96.44%

Di seguito viene mostrata la matrice di confusione per la Fallnet

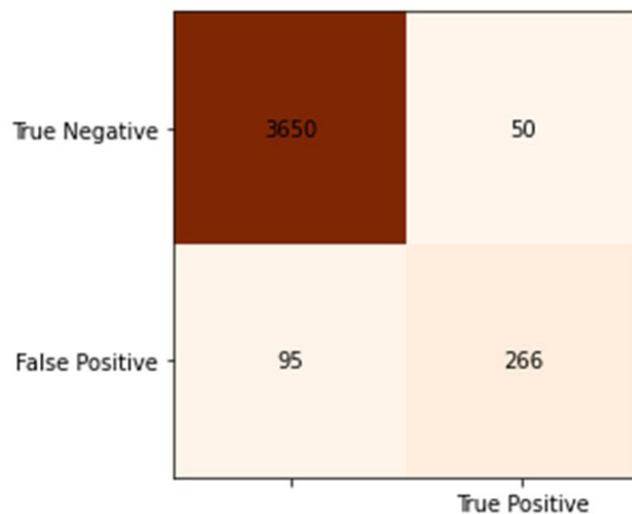


Figure 4.6: Matrice di confusione fallnet

Qui di seguito invece viene mostrata la matrice di confusione della rete basata su MHI. Il test è stato effettuato su pochi frame di test:

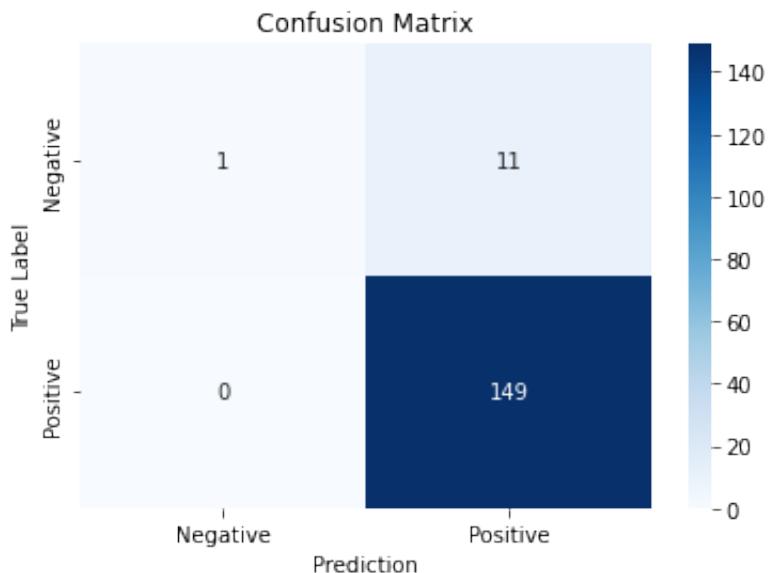


Figure 4.7: Matrice di confusione FDNet

E' positivo far notare che per entrambi i modelli il numero di falsi negativi è molto più basso dei falsi positivi. Questo sta a significare che, presupponendo che vi sia un errore di rilevazione, vi sarà comunque prevalenza di falsi allarmi rispetti alle cadute vere non segnalate. Da far notare che nell'approccio frame-by-frame è molto difficile evitare di avere errori di tutti e due i tipi perchè la valutazione viene ripetuta n volte per ogni frame del video. Questo vuol dire che qualora dovesse apparire un falso negativo nella valutazione, non vuol dire necessariamente che il modello non è stato in grado di rilevare la caduta, piuttosto potrebbe rilevarla nei frame immediatamente successivi. Discorso diverso viene fatto per la rete basata su MHI perchè elabora i frame in batch. In questo caso la presenza di un falso negativo potrebbe indicare la reale possibilità che la rete non sia stata in grado di rilevare la caduta. Qui la valutazione infatti viene presa considerando un insieme di frame, e quindi i frame relativi alla caduta potrebbero

#### *4.1 Valutazione modelli*

non ripetersi più per le valutazioni successive e la caduta potrebbe mai essere rilevata. Fortunatamente tale scenario non è stato incontrato durante la fase di testing del modello. Difatti il numero di falsi negativi per la rete MHI è pari a 0.

### Inferenza su video personale

Per concludere, è stato eseguito un test di inferenza tramite il modello fallnet su un video personale. Nelle immagini di seguito di mostra come il modello abbia imparato a distinguere lo scenario di una caduta da uno di activity day living, come lo star seduti:



Figure 4.8: Scenario nella norma



Figure 4.9: Scenario di caduta



# **Chapter 5**

## **Conclusioni**

In conclusione, possiamo dire che i modelli basati sulla visione possono essere utilizzati per rilevare le cadute. Questi si integrano in sistemi che utilizzano telecamere e tecnologie di elaborazione delle immagini per rilevare e analizzare eventi di caduta. Sistemi di questo tipo possono essere utilizzati in molti contesti, come ad esempio nelle case di cura per anziani o nei luoghi di lavoro per monitorare la sicurezza dei dipendenti. I sistemi basati sulla visione utilizzano algoritmi per riconoscere pattern e movimenti specifici associati alle cadute, e possono inviare allarme in caso di emergenza. Questi sistemi possono essere integrati con altre tecnologie di allarme per fornire una soluzione completa per la prevenzione delle cadute.

I vantaggi dei metodi basati sulla visione per la rilevazione delle cadute includono:

Precisione: questi sistemi utilizzano algoritmi sofisticati per rilevare con precisione le cadute e distinguerle da altri movimenti.

Tempo di reazione rapido: in caso di caduta, il sistema invia immediatamente un allarme per garantire un intervento tempestivo.

Personalizzabilità: i sistemi possono essere personalizzati per soddisfare le esigenze specifiche di ogni contesto, come ad esempio la configurazione della telecamera e la soglia di rilevamento della caduta.

Facilità d'uso: molti sistemi basati sulla visione sono progettati

per essere facili da usare e installare.

Tra gli svantaggi dei metodi basati sulla visione per la rilevazione delle cadute, si possono includere:

Costo: i sistemi basati sulla visione possono essere costosi rispetto ad altri metodi di rilevazione delle cadute.

Requisiti di illuminazione: per funzionare correttamente, questi sistemi richiedono una buona illuminazione, il che può essere un problema in alcuni contesti.

Interferenze: altri elementi presenti nell'area monitorata, come ad esempio animali domestici o oggetti in movimento, possono causare falsi allarmi.

Privacy: la presenza di telecamere può essere percepita come invasiva da alcune persone e può sollevare preoccupazioni relative alla privacy.

In generale, i sistemi basati sulla visione per la rilevazione delle cadute possono essere una soluzione efficace per migliorare la sicurezza in molte situazioni, ma è importante valutare attentamente i pro e i contro prima di scegliere questo tipo di sistema.

## 5.1 Sommario

In questa tesi sono stati esplorati due metodi basati sulla visione per la rilevazione delle cadute. Il primo metodo si basa su una rete neurale convoluzionale addestrata su varie sequenze di caduta. Il task di fall detection è stato definito come un problema di classificazione binaria, in cui ad ogni frame del video, viene assegnata una label. Il vantaggio di questo metodo è che l'elaborazione è molto veloce e i risultati sono subito disponibili. Lo svantaggio principale è che, essendo l'elaborazione frame-by-frame, il modello non concepisce la "storia" del movimento di una persona e ciò potrebbe comportare la presenza di numerosi falsi positivi (ad esempio la rete può generare un allarme nel caso una persona si adagia lentamente sul pavimento in posizione sdraiata). Il secondo approccio è basato sulla motion history image, una tecnica che estrae il contenuto informativo da un insieme di frame. Quindi con questa tecnica siamo in grado di concepire la storia del movimento di una persona, e concepire sistemi più resistenti al rumore (adagiarsi lentamente sul pavimento non è più condizione sufficiente per generare un falso positivo, in quanto il modello MHI capisce che la quantità di moto in questo caso non è riconducibile a quella di una caduta). Lo svantaggio principale dell'elaborazione immagini in batch con MHI, è che i risultati non sono disponibili immediatamente, in quanto il modello ha bisogno di raccogliere un insieme di frame prima di poter elaborare e sviluppare una decisione. Entrambi i modelli sono stati convertiti in TensorFlow Lite, una piattaforma più leggera e flessibile a supporto di dispositivi hardware con limitata capacità computazionale. In questa tesi è stata utilizzata una Raspberry Pi 3 per il porting del modello su dispositivo mobile. Il test dei modelli ha risuitato una accuracy robusta e buoni valori di metriche, che fanno ben sperare in prospettiva di una futura applicazione nel campo. La ricerca in

*Chapter 5 Conclusioni*

questo ambito sta procedendo in avanti e si spera si svilupperanno sistemi sempre più affidabili.

## 5.2 Limiti dei modelli

Attualmente entrambi i modelli sono stati addestrati mediante l'utilizzo di soli 2 dataset. L'aumento della disponibilità dei dati può sicuramente portare a capacità prestazionali del sistema migliori. E' importante infatti che il sistema percepisca e comprenda che ci sono numerosi modi di caduta, e spesso risulta complesso racchiudere tutte le possibilità in pochi dataset. Ad esempio, si potrebbe considerare un set di dati aggiuntivo in cui si rappresenta una serie di cadute "vere" e una serie di cadute "false". Da qui il modello estende la propria capacità predittiva anche per quanto riguarda la possibilità di scindere il caso di una caduta accidentale da una volontaria. Da ricordare che questo obiettivo può difficilmente essere raggiunto mediante modelli che lavorano frame-by-frame, in quanto è necessario avere a disposizione la conoscenza del movimento e della quantità di moto che varia fra un frame e l'altro. In questo senso, sarà importante sviluppare modelli che supportano una buona velocità di esecuzione, mantenendo comunque valori di precisione a livelli soddisfacenti.

### **5.3 Sviluppi futuri**

Questa ricerca ha dimostrato che è possibile un accurato rilevamento delle cadute tramite videocamere installate su dispositivi mobili. Il test del modello finale è stato eseguito su una Raspberry pi 3 b+.

È possibile eseguire ulteriori ricerche sul tema, soprattutto per quanto riguarda le possibili integrazioni con altre tecnologie. Ad esempio per ridurre il numero di falsi allarmi e falsi negativi si potrebbero aggiungere tecnologie intelligenti basate sull'audio [30].

Il rilevamento delle cadute tramite audio è un metodo per rilevare le cadute attraverso l'analisi del suono. Ciò si ottiene solitamente utilizzando sensori e microfoni posizionati in una stanza o sulla persona per acquisire dati audio, che vengono poi elaborati utilizzando algoritmi per identificare il suono di una caduta. L'obiettivo è rilevare il verificarsi di una caduta il più rapidamente possibile e allertare i custodi o il personale medico per fornire assistenza. I caricamenti e l'apprendimento basati sul cloud per l'evoluzione del modello scaricherebbero tutta l'elaborazione audio nel cloud. I campioni verrebbero caricati sul cloud TPU ed elaborati per aggiornare le unità distribuite esistenti. È possibile eseguire un'ulteriore elaborazione del segnale sui segnali acquisiti per filtrare qualsiasi rumore bianco ambientale o rumore di scoppio che i sensori potrebbero captare da fonti meccaniche o elettriche circostanti.

Oltre a ciò, possiamo sicuramente dire che i sistemi di fall detection basati sulla visione diventeranno probabilmente ancora più avanzati e sofisticati, sfruttando i progressi della computer vision e del machine learning per migliorare la precisione e l'affidabilità. Alcuni possibili sviluppi futuri potrebbero includere un maggiore utilizzo di telecamere 3D e sensori di profondità: queste telecamere e sensori possono fornire informazioni più complete e

### *5.3 Sviluppi futuri*

precise sull'ambiente e sui movimenti della persona, migliorando il rilevamento delle cadute.



## Bibliography

- [1] World Health Organization. Global report on falls prevention in older age: Geneva, switzerland. 2007.
- [2] R. Rodrigues. Facts and figures on healthy ageing and long-term care; european centre for social welfare policy and research: Viena, austria,. 2012.
- [3] M.; Selmes J. Brunete, A.; Selmes. Can smart homes extend people with alzheimer's disease stay at home? j. enabling technol. 2017.
- [4] M. Mubashir. A survey on fall detection: Principles and approaches. neurocomputing. 2013.
- [5] F. Bagalà. Evaluation of accelerometer-based fall detection algorithms on real-world falls. 2021.
- [6] C. Wang. Development of a fall detecting system for the elderly residents. 2008.
- [7] U. Lindemann. Evaluation of a fall detector based on accelerometers: A pilot study. 2005.
- [8] M.J. Mathie. Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement. 2014.
- [9] S.J. Bianchi, F.; Redmond. Barometric pressure and triaxial accelerometry-based falls event detection. 2010.

## *Bibliography*

- [10] R. Ghasemzadeh, H.; Jafari. A body sensor network with electromyogram and inertial sensors: Multimodal interpretation of muscular activities. 2010.
- [11] M.; Bonatesta F. Abbate, S.; Avvenuti. A smartphone-based fall detection system. 2012.
- [12] M. Aihua. Highly portable, sensor-based system for human fall monitoring. sensors. 2017.
- [13] K.; Herrmann M. Albert, M.V.; Kording. Fall classification by machine learning using mobile phones. 2012.
- [14] Z. Zhang. A survey on vision-based fall detection. 2015.
- [15] W. Chen. Fall detection based on key points of human-skeleton using openpose. 2020.
- [16] Brian; Zhen-Peng. Fall detection based on body part tracking using a depth camera. 2014.
- [17] Ahad; Rahman. Motion history image: its variants and applications. 2012.
- [18] Albawendi; Suad. Video based fall detection with enhanced motion history images. 2016.
- [19] Huang; Zhanyuan. Video-based fall detection for seniors with human pose estimation. 2018.
- [20] Pang; Bo; Erik Nijkamp; Ying Nian. Deep learning with tensorflow: A review. 2020.
- [21] Bisong; Ekaba. Google colaboratory. building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners. 2019.
- [22] Nath; Omkar. Review on raspberry pi 3b+ and its scope. 2020.

## Bibliography

- [23] David Robert. Tensorflow lite micro: Embedded machine learning for tinyml systems. 2021.
- [24] Szegedy Christian. Going deeper with convolutions. 2015.
- [25] Murray; Naila; Florent. Generalized max pooling. 2014.
- [26] <http://fenix.univ.rzeszow.pl/mkepski/ds/uf.html>.
- [27] J. Dubois I. Charfi, J. Mitéran. Fall detection dataset (fdd): Optimised spatio-temporal descriptors for real-time fall detection. 2013.
- [28] Sandler; Mark. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018.
- [29] <https://www.pushbullet.com/>.
- [30] Geertsema; Evelien E. Automated remote fall detection using impact features from video and audio. 2019.