

# UrbanSecurity

Progetto di Ingegneria della Conoscenza  
A.A. 2021/2022

## Componenti del gruppo:

- *Salvatore Napoli*

Matricola: 706483

Mail: s.napoli3@studenti.uniba.it

## Link del repository:

<https://github.com/napolisalvatore/ICON>

## Documentazione

### Introduzione

Il progetto consiste in un'applicazione chiamata *UrbanSecurity*, una tecnologia di rilevamento della violenza in tempo reale dallo stream di una telecamera ma che può essere facilmente integrata con qualsiasi sistema di sicurezza. La sua funzione principale è garantire la sicurezza pubblica attraverso la sorveglianza visiva della folla, quindi qualsiasi attività violenta genera un allarme che automaticamente allerta le autorità.

Attraverso l'applicazione di tecniche di elaborazione delle immagini e apprendimento automatico, i sistemi di sorveglianza intelligenti sono in grado di estrarre e interpretare le informazioni dai filmati CCTV più velocemente e in modo molto più efficiente di qualsiasi osservatore umano. Inoltre, tutti i componenti lavorano insieme in un ambiente flessibile in modo da poter essere personalizzati per ottenere uno scopo specifico che può anche cambiare nel tempo.

Le feature che sono state inserite sono:

- *Rilevamento di persone e riconoscimento scene di violenza*: il sistema prende in input lo stream di una telecamera di sorveglianza, lo elabora e restituisce se vi sono presenti scene di violenza;
- *Predizione di violenza*: il sistema cerca di predire, attraverso vari fattori, i posti in cui si potrebbero verificare scene di violenza;
- *Ricerca in un grafo*: dato un punto di partenza (una stazione di polizia o la posizione esatta di una volante), il sistema calcola il percorso migliore per raggiungere la posizione della telecamera che ha lanciato l'allarme;
- *Interrogare una base di conoscenza*: il sistema permette di interrogare la KB per ottenere informazioni (ad esempio: elenco scene di violenza rilevate da una telecamera, elenco delle date in cui si è verificata violenza, persone coinvolte in una scena di violenza, ...).

- **Rilevamento di persone e riconoscimento scene di violenza**

Una rete neurale convoluzionale è stata addestrata per il rilevamento della violenza in tempo reale.

Per il riconoscimento di persone nel video è stato usato InceptionV3, un modello di riconoscimento delle immagini ampiamente utilizzato che ha dimostrato di raggiungere una precisione superiore al 78,1% sul set di dati.

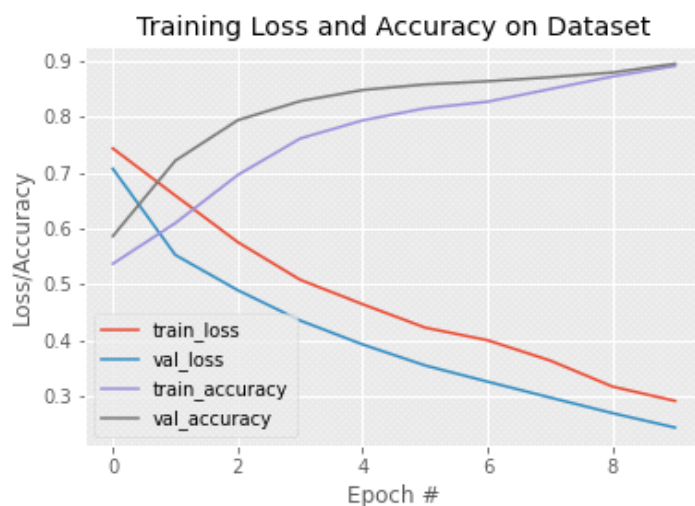
Per l'addestramento è stato utilizzato un set di dati contenente 1600 video suddiviso in due folder: nella folder Fight sono presenti 800 video in cui vi sono scene di violenza, nella folder NonFight, invece, gli altri 800 con scene di non violenza.

Il training della rete è stato effettuato in 10 epoche. Di seguito i risultati ottenuti dal nostro modello:

```
[INFO] evaluating network...
```

	precision	recall	f1-score	support
NonViolence	0.86	0.93	0.89	360
Violence	0.92	0.86	0.89	360
accuracy			0.89	720
macro avg	0.89	0.89	0.89	720
weighted avg	0.89	0.89	0.89	720

Guardando i valori qui presenti, si può dedurre di aver ottenuto buoni risultati, nonostante le limitazioni dovute a Google Colab in quanto non è stato possibile effettuare l'addestramento della rete neurale con più di 10 epoche.



Dopo aver creato e addestrato il modello utilizzando l'apprendimento supervisionato, il sistema è in grado di elaborare in tempo reale il filmato e di restituire in output "State: Normal" se non vi è presente violenza nel video e "State: Violence" se, invece, vi è presente violenza nel video e con quale probabilità.

In caso di violenza il sistema dovrà lanciare un allarme.

Dopo un'attenta e approfondita analisi è stato scelto che l'allarme sarà lanciato solo se la probabilità che vi sia violenza supera il 70%.



NB: saranno forniti due file InceptionV3.ipynb e InceptionV3.py , è consigliabile usare .ipynb con Google Colab. Per testare il sistema è sufficiente fornire un qualsiasi video in input.

- **Ricerca in un grafo**

Si è scelto di implementare questa funzionalità affinché un'ipotetica volante possa orientarsi, attraverso l'uso di una mappa, al raggiungimento della posizione esatta della telecamera che ha lanciato l'allarme.

Per la creazione della mappa, è stata selezionata la città di Bari.

Le telecamere sono state posizionate in diversi punti della città sia nelle zone del centro ma anche in zone periferiche e sono locate in piazze strade ma anche in vicoli.

Di seguito la posizione esatta di ciascuna telecamera:

- Via Sparano da Bari, 97, 70121 Bari BA
- Via Abate Giacinto Gimma, 70122 Bari BA
- Via Alessandro Manzoni, 70122 Bari BA
- Via Benedetto Cairoli, 2, 70122 Bari BA

- Via Nicolò Putignani, 70121 Bari BA
- Via Dante Alighieri, 24, 70121 Bari BA
- Via Andrea da Bari, 59, 70122 Bari BA
- Corso Benedetto Croce, 70125 Bari BA
- Corso Sidney Sonnino, 70121 Bari BA
- Corso A. de Gasperi, 429, 70125 Bari BA
- Via Bruno Buozzi, 3-13, 70132 Bari BA
- Viale Biagio Accolti Gil, 22, 70132 Zona Industriale BA
- V.le Francesco de Blasio, 70132 Zona Industriale BA
- Via Giambattista Bonazzi, 70122 Bari BA
- Molo S. Nicola, 1, 70121 Bari BA
- Piazza Aldo Moro, 13, 70123 Bari BA
- Lungomare Armando Perotti, 70126 Bari BA
- Piazza del Redentore, 230c, 70123 Bari BA
- Piazza Giulio Cesare, 70122 Bari BA
- Via della Resistenza, 154, 70125 Bari BA
- Piazza Umberto I, 1, 70121 Bari BA
- Piazza Mercantile, 70122 Bari BA
- Strada Casamassimi, 11, 70122 Bari BA
- Piazza del Ferrarese, 10, 70122 Bari BA
- Piazza S. Nicola, 13, 70122 Bari BA
- Piazza Armando Diaz, 70121 Bari BA

Per la struttura del grafo, è stato creato appositamente un file csv contenente le informazioni riguardanti i luoghi e le vie di Bari.

	A	B	C	D	E	F	G
1	Name, []number, X, Y						
2	Corso Sidney Sonnino (41.119167, 16.882222)						
3	Via Sparano, (41.122500, 16.869444)						
4	Via Calefati,(41.123611, 16.863889)						
5	Via Abate Gimma,(41.124444, 16.864722)						
6	Via Manzoni(41.121389, 16.861389)						
7	Via Nicolò Piccinni(41.125278, 16.867222)						
8	Via Putignani(41.123333, 16.871667)						
9	via Dante[3](41.121944, 16.871667)						
10	Via Andrea da Bari(41.124167, 16.868611)						
11	Corso Benedetto Croce(41.111944, 16.871944)						
12	Corso Alcide de Gasperi(41.092500, 16.870000)						
13	Via Bruno Buozzi[6](41.118611, 16.842778)						
14	Viale Biagio Accolti Gil[5](41.119722, 16.811111)						
15	Viale Francesco de Blasio[7](41.115556, 16.810556)						
16	Via Napoli[8](41.126667, 16.863056)						
17	Molo San Nicola(41.125556, 16.874167)						
18	piazza aldo moro, (41.118333, 16.870000)						
19	gramsci,(41.118333, 16.890000)						
20	redentore,(41.118889, 16.856389)						
21	giulio cesare,(41.112222, 16.862778)						
22	parco 2 giugno,(41.102500, 16.874722)						
23	piazza unmberto I,(41.120556, 16.869167)						
24	mercantile,(41.128056, 16.871944)						
25	largo albicocca,(41.127222, 16.867778)						
26	piazza ferrarese,(41.127222, 16.871667)						
27	piazza san nicola,(41.130278, 16.869444)						
28	piazza armando diaz(41.122500, 16.878056)						
Posizione telecamere							

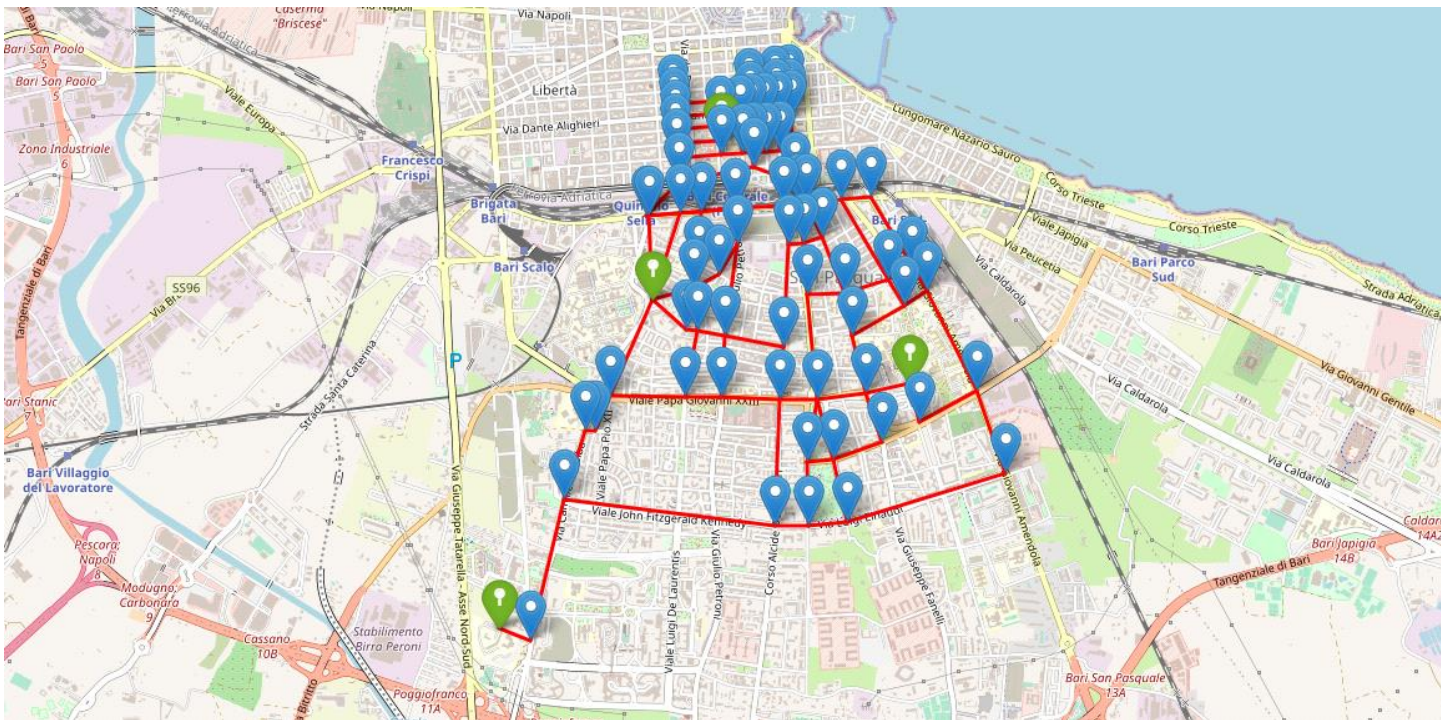
È stato utilizzato [Google Maps](https://www.google.com/maps) per ottenere le coordinate dei luoghi dove sono state posizionate le telecamere.



Per ogni riga del dataset, sono state salvate:

- nome della strada (**Name**);
- le coordinate (latitudine Y e longitudine X).

Attraverso l'utilizzo della libreria **folium**, è stata applicata la parte di visualizzazione grafica sul servizio di [OpenStreetMap](#).



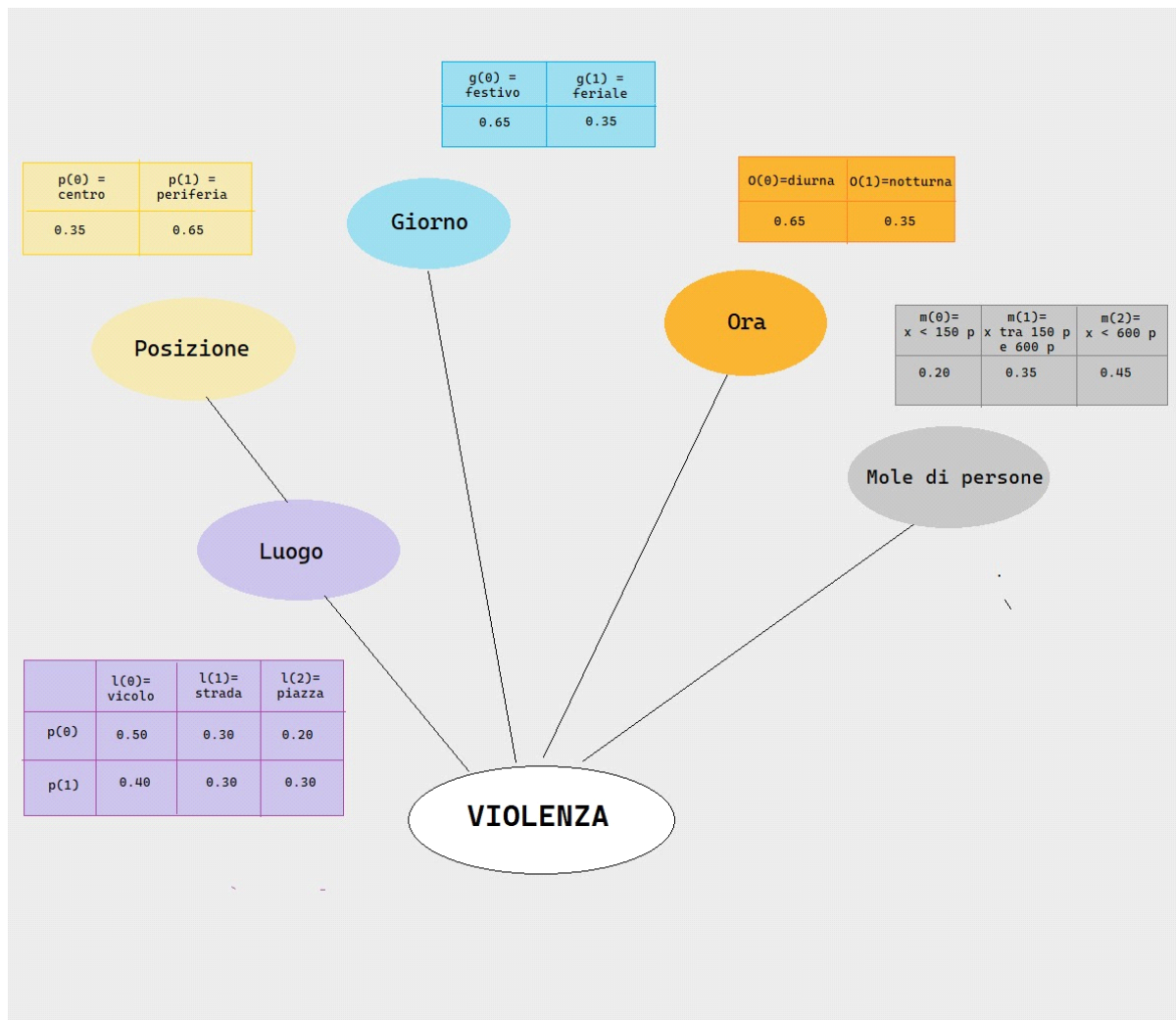
Per la ricerca su grafo, è stato implementato l'algoritmo **A\*** con funzione euristica calcolata mediante la *distanza euclidea* (in linea d'aria tra due punti).

Il risultato dell'algoritmo viene poi visualizzato sulla mappa tracciando in rosso il percorso e indicando con un'icona rossa il punto di partenza, in verde l'obiettivo e in blu i nodi intermedi.





La rete bayesiana costruita è la seguente:



La visualizzazione testuale completa delle tabelle è presente nel file retebayesiana.txt

Come si evince dall'immagine, la probabilità che si verifichi violenza può essere( [Bassa], [Media], [Alta] ) ed è influenzata dalle seguenti features:

- *mole di persone presenti in quell'istante* [ $x < 150$  persone,  $150 \text{ persone} \leq x \leq 600$  persone,  $x > 600$  persone] ;

- *orario* [notturno, diurno] ;
- *giorno*[festivo, feriale] ;
- luogo [vicolo, strada, piazza]
  - influenzato, a sua volta, anche dalla *posizione* [centro, periferia].

A chi utilizzerà il sistema è richiesto di indicare i valori di uno o più di questi fattori e il sistema calcolerà la predizione di violenza in base alle probabilità.

Contenuto del package *bayesianNetwork*:

- *bayesianNetwork*: file che contiene la creazione della Belief Network e la funzione per fare inferenza.

Librerie utilizzate:

- [pgmpy](#)

- **Interrogare la base di conoscenza**

Un'altra funzionalità effettuata dal sistema permette di interrogare la base di conoscenza al fine di ottenere varie informazioni tra cui: luoghi osservati, date in cui è stata rilevata violenza, numero di persone coinvolte nelle scene di violenza.

Per far ciò è stata usata la libreria *pythoLog* che permette di popolare una base di conoscenza ed effettuare query su di essa.

Le query:

- Permettono di verificare se una clausola è conseguenza logica della base di conoscenza (in questo caso, il sistema risponderà “Yes”, altrimenti “No”);
- Possono comprendere variabili in modo che la risposta riporti tutti i valori che possono essere assunti dalla/e variabile/i.

*Esempio:* la query “ $osserva(T, L)$ ” restituirà tutte le coppie ove  $T$  è una telecamera e  $L$  è il luogo osservato.

Gli assiomi utilizzati nella KB sono i seguenti (è riportato solo un esempio con relativa spiegazione) :

- `osserva(telecamera1, marciapiede_viaZanardelli)`  
`osserva(telecamera,luogo)`
- `data(telecamera1, 12/02/2018)`  
`data(telecamera,data_violenza)`
- `num(12/02/2018,2)`  
`numero persone coinvolte(data, persone)`
- `Violenza(T,L,D,P):-osserva(T,L),data(T,D),num(D,P)`  
`Violenza(IdTelecamera,Luogo>Data,Persone coinvolte):-`  
`osserva(IdTelecamera,Luogo),data(IdTelecamera>Data),`  
`NumPersoneCoinvolte(Data,nPersone)`

Le informazioni salvate nella KB sono dei dati casuali utili per testare la KB.

Contenuto del package *Prolog*:

- *prolog*: file che contiene la creazione della KB e la funzione per interrogarla.

Librerie utilizzate:

- [pytholog](#)