

LOCALIZZAZIONE DI SCALE IN PLANIMETRIE

Candidato:

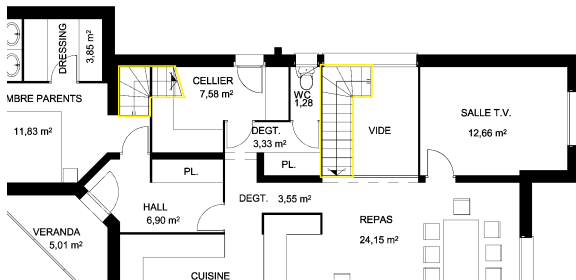
Manuel Salamino

Relatore:

Prof. Simone Marinai

Obiettivo

Realizzare un programma che, data una planimetria, restituisce le scale presenti.

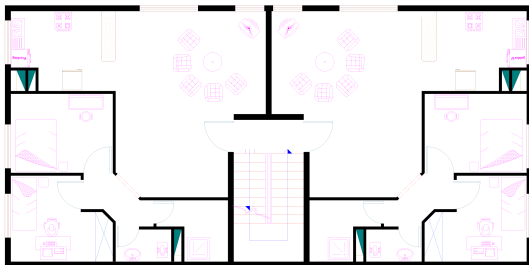


Passi Principali



Elaborazione dell'immagine

Trasformazione in scale di grigio



Esempio di planimetria su cui viene eseguito l'algoritmo

Elaborazione dell'immagine

Trasformazione in scale di grigio

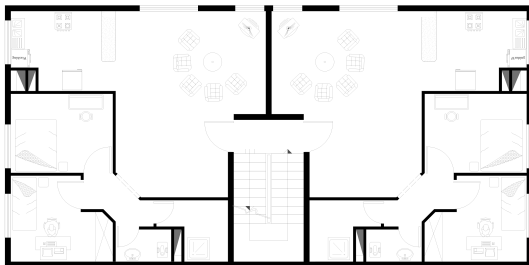


Immagine trasformata in scale di grigio

Elaborazione dell'immagine

Binarizzazione

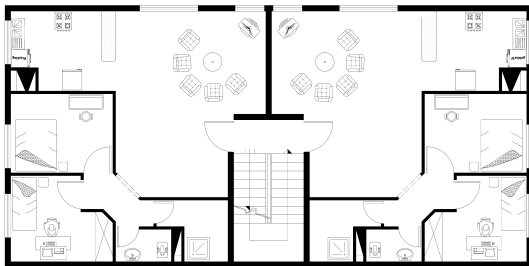
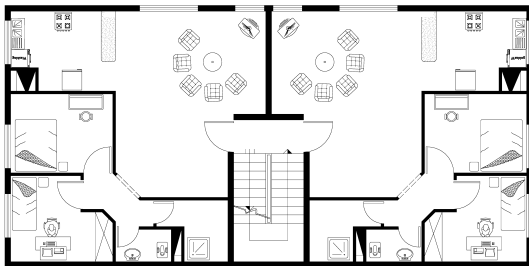


Immagine binarizzata

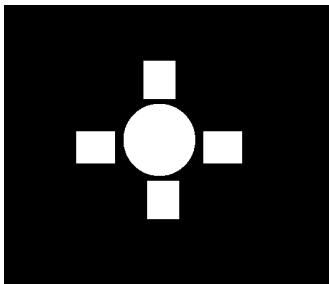
Elaborazione dell'immagine

Dilatazione morfologica del nero

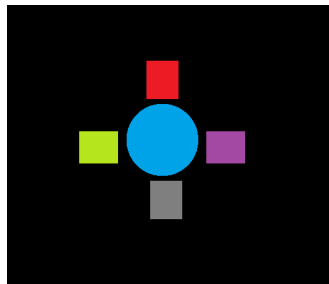


Dilatazione morfologica del nero, al fine di recuperare pixel nelle linee interrotte

Estrazione delle Componenti Connesse

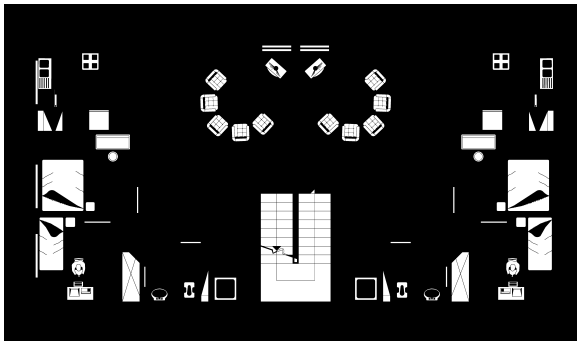


Immagine



Componenti connesse
bianche

Estrazione delle Componenti Connesse



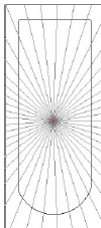
Componenti connesse trovate nel grafo, escludendo quelle di dimensione troppo grande e quelle di dimensione troppo piccola

Grafo di Adiacenza

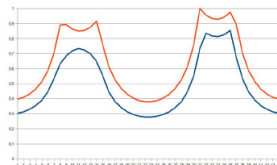
Nodi: Componenti connesse che hanno altre componenti connesse ad una distanza $\leq 6px$



Componente
connessa



Sezionamento



Descrittore

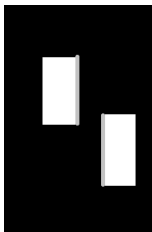
Grafo di Adiacenza

Archi: tra componenti connesse, solo se:

- distanza ≤ 6 px
- intersezione tra le componenti connesse rappresentate dai nodi ≥ 0.4

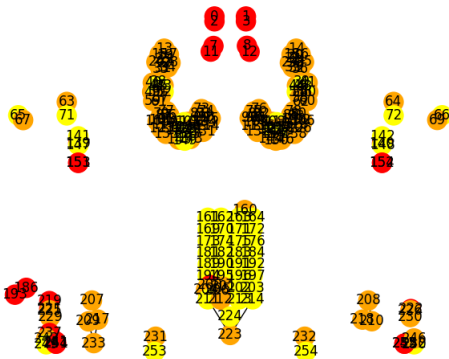


Sarà creato arco



Non sarà creato arco

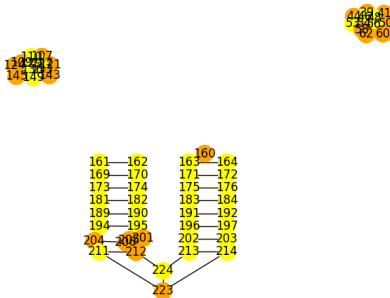
Grafo di Adiacenza



Grafo estratto dalla planimetria

Riduzione Grafo di Adiacenza

Elimino le componenti connesse del grafo con numero di nodi < 5



Grafo ridotto

Riduzione Grafo di Adiacenza

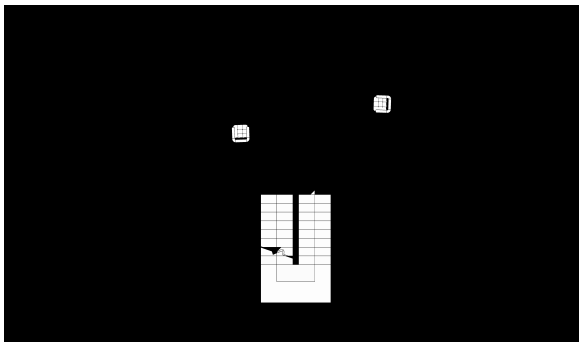
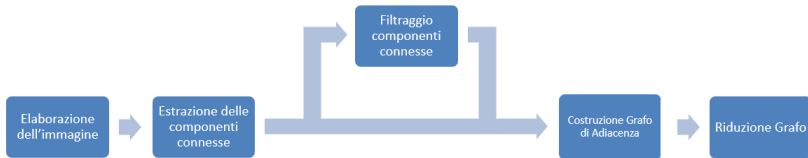


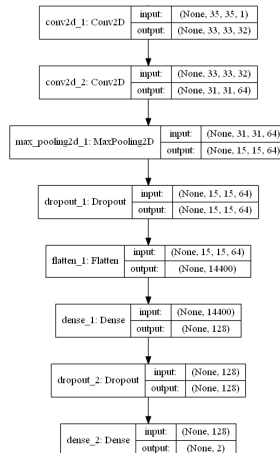
Immagine corrispondente al grafo ridotto

Passi Principali



Rete Neurale

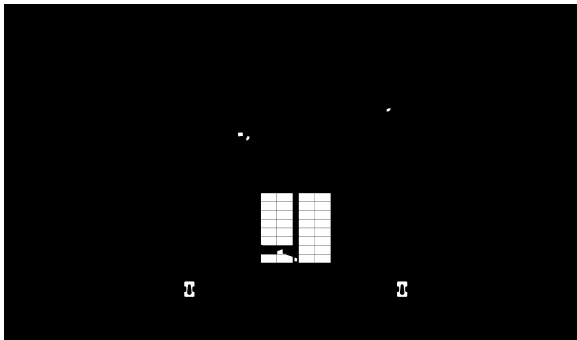
Utilizzata per fare una **predizione** sulla classe di appartenenza di una componente connessa.



Architettura

Rete Neurale

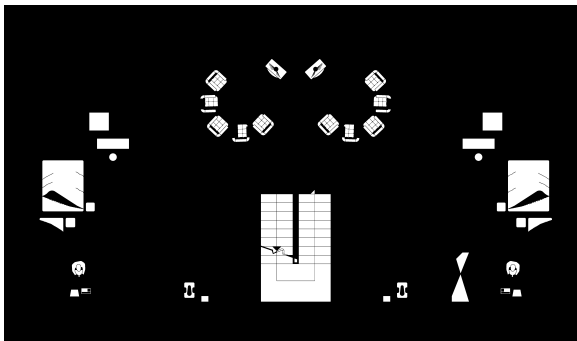
Esempio di funzionamento



Componenti connesse dell'immagine che la rete ha predetto essere di classe "Scalino"

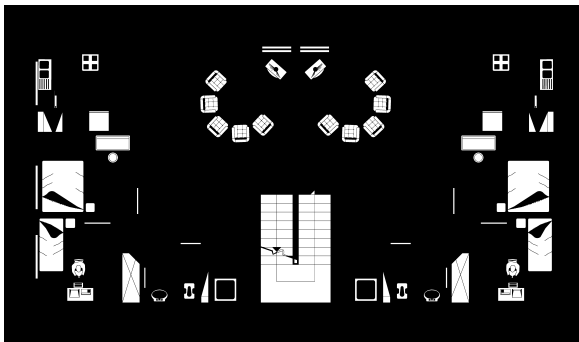
Rete Neurale

Esempio di funzionamento



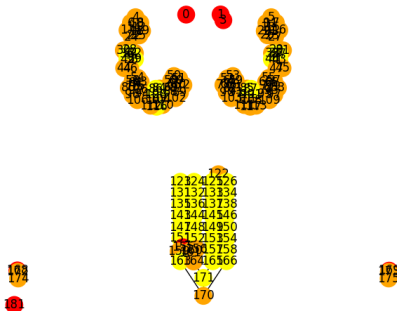
Componenti connesse dell'immagine a cui la rete ha attribuito
probabilità di essere "Scalino" > 0.3

Estrazione delle Componenti Connesse



Grafo di adiacenza

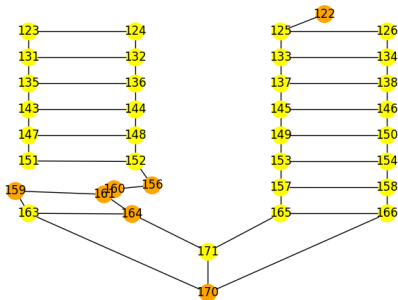
Dopo Rete Neurale



Grafo di adiacenza ottenuto dalle componenti con probabilità di essere scalino > 0.3

Riduzione Grafo di Adiacenza

Dopo Rete Neurale



Grafo ridotto dopo l'utilizzo della rete neurale

Riduzione Grafo di Adiacenza

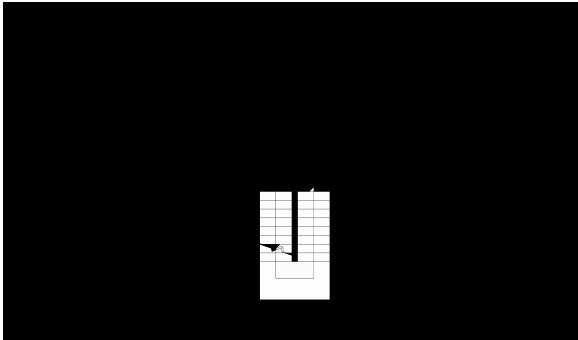


Immagine corrispondente al grafo ridotto

Riduzione Grafo di Adiacenza

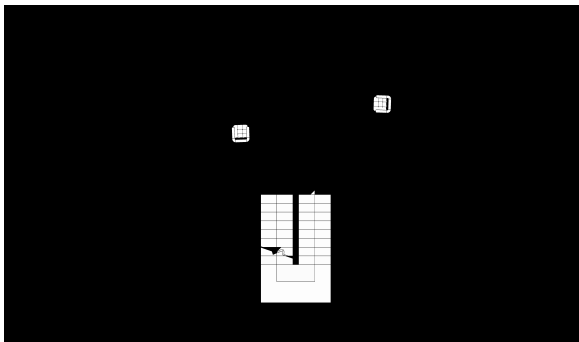


Immagine corrispondente al grafo ridotto nella modalità senza Rete Neurale

Dataset Utilizzati

Dataset_1	
Num planimetrie	87
Num scale presenti	36
Dim planimetrie	1500x2000

Dataset_2	
Num planimetrie	31
Num scale presenti	32
Dim planimetrie	5000x7000

Differenza fondamentale tra i Dataset è che nelle planimetrie del Dataset_2 sono presenti molti più oggetti.

Metriche

Per valutare il risultato ottenuto uso il **Coefficiente di Jaccard**.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Imposto *Jaccard_Threshold*: tutte le immagini risultanti che hanno *Coefficiente di Jaccard* $>$ *Jaccard_Threshold* sono definite come *Trovate giuste*.

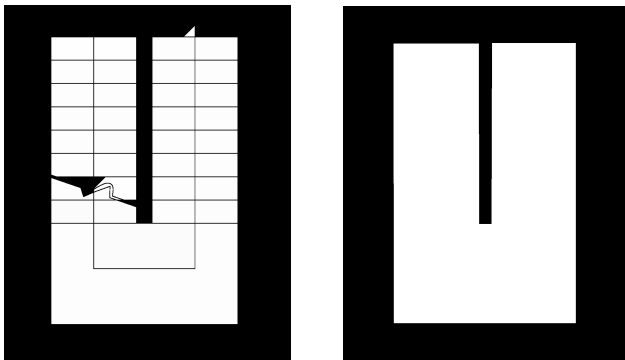
Per valutare l'algoritmo utilizzo i valori di *Precision* e *Recall* al variare della soglia *Jaccard_Threshold*.

$$Precision = \frac{\#trovati_giusti}{\#trovati}$$

$$Recall = \frac{\#trovati_giusti}{\#giusti}$$



Test del risultato



Si confronta il risultato ottenuto con la scala etichettata:
coefficiente di Jaccard = 0.85

Risultati Dataset_1

Senza Rete Neurale

#Trovati = 42

#Giusti = 36

Jaccard_Threshold	#Trovati_giusti	Precision	Recall
0.6	35	0.83	0.97
0.65	33	0.79	0.92
0.7	31	0.74	0.86
0.75	29	0.69	0.81
0.8	22	0.52	0.61

Con Rete Neurale

#Trovati = 40

#Giusti = 36

Jaccard_Threshold	#Trovati_giusti	Precision	Recall
0.6	34	0.85	0.94
0.65	33	0.83	0.92
0.7	27	0.68	0.75
0.75	23	0.58	0.64
0.8	13	0.33	0.36

Risultati Dataset_2

Senza Rete Neurale

#Trovati = 41

#Giusti = 32

Jaccard_Threshold	#Trovati_giusti	Precision	Recall
0.6	30	0.73	0.94
0.65	28	0.68	0.875
0.7	28	0.68	0.875
0.75	27	0.65	0.84
0.8	26	0.63	0.81

Con Rete Neurale

#Trovati = 33

#Giusti = 32

Jaccard_Threshold	#Trovati_giusti	Precision	Recall
0.6	26	0.78	0.81
0.65	23	0.70	0.72
0.7	23	0.70	0.72
0.75	23	0.70	0.72
0.8	22	0.67	0.69

Conclusioni

- è stata effettuata la rappresentazione delle immagini mediante un grafo
- è stata utilizzata una rete neurale per il filtraggio delle componenti connesse non idonee ad essere parte di una scala
- il programma nella sua forma finale evidenzia che la sua versione con Rete Neurale migliora la *Precision*, ma non della *Recall*, ed ha performance migliori se eseguita su planimetrie con numero di oggetti maggiori.

Grazie a tutti per l'attenzione.

