



Road Damage Detector

Web app per il rilevamento di strade
accidentate





Indice dei contenuti

01

Introduzione

02

FURPS+

03

Workflow

**Tecnologie
utilizzate**

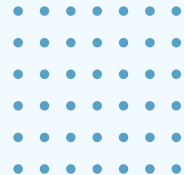
04

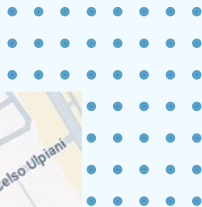
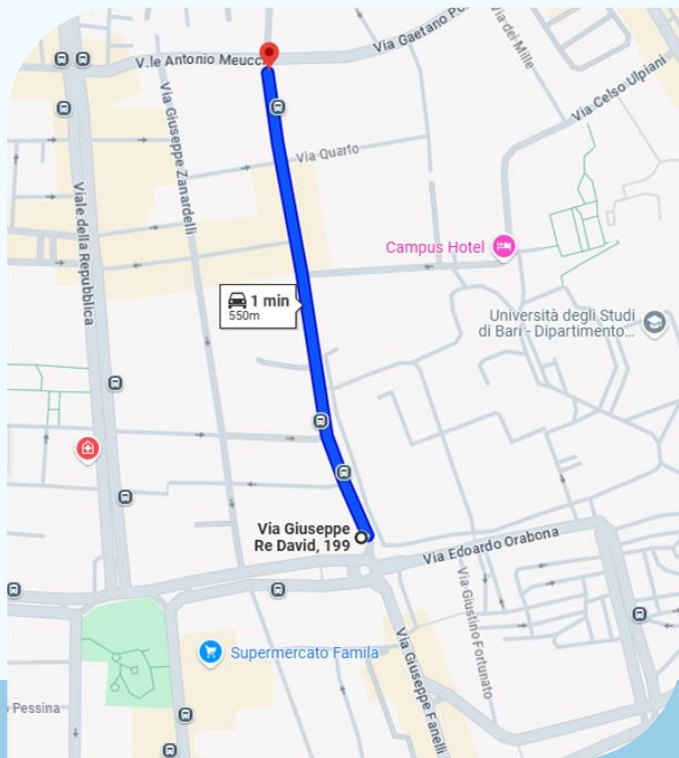
Architettura

05

**Analisi dei
risultati**

06





01

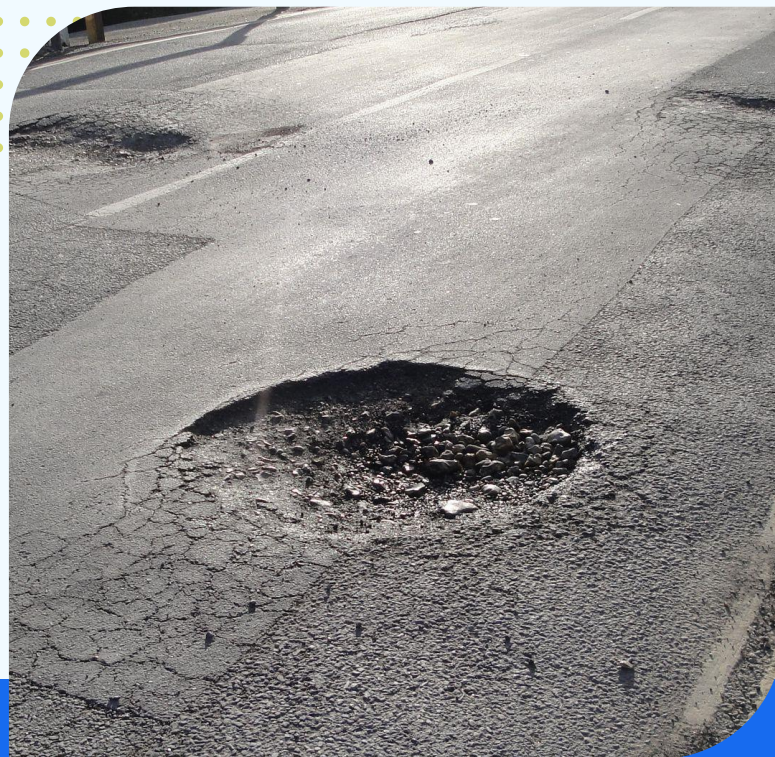
Introduzione



Perchè?

Problema:

- Necessità di far fronte a un problema comune
- Incidenti causati dalle buche stradali
- Importanza delle condizioni stradali per la sicurezza e manutenzione.



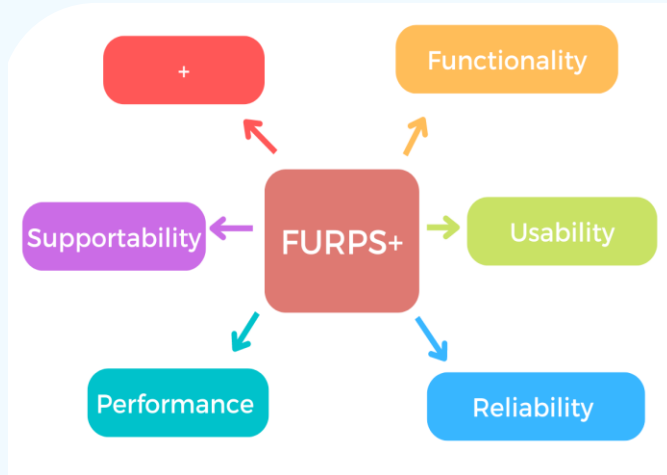
La nostra soluzione

Soluzione proposta:

- Uso della visione artificiale per rilevamenti automatizzati.
- Migliorare la qualità della navigazione
- Fornire uno strumento pratico a enti pubblici e privati



02

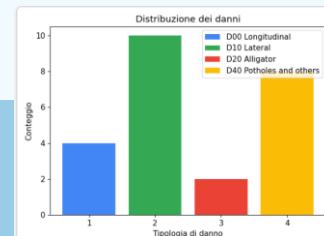
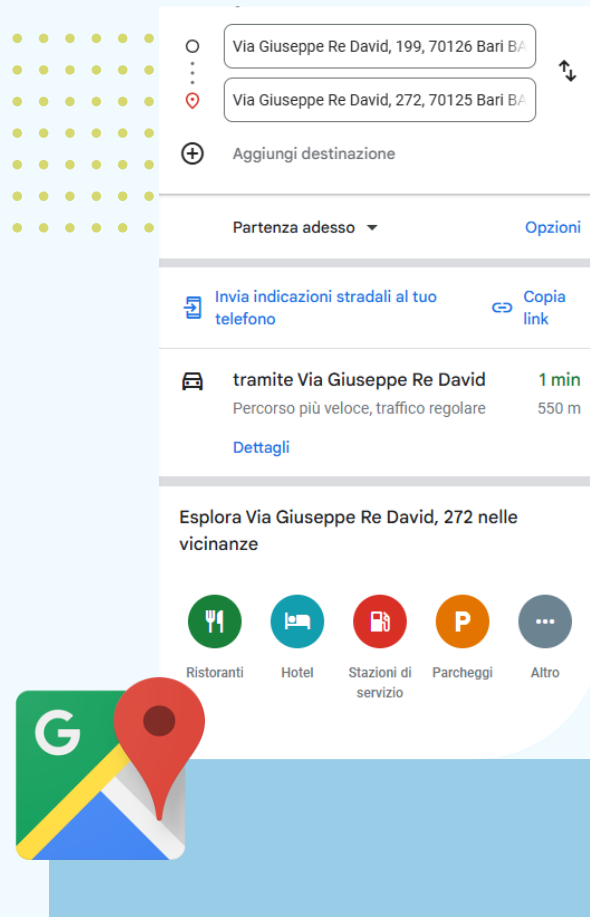


FURPS+

Requisiti Funzionali

Requisiti Funzionali:

- Operare sul percorso inserito dall'utente
- Ottenimento automatico delle immagini
- Rilevamento danni da immagini
- Classificazione per tipo di danno
- Integrazione con API di Navigazione
- Specifica del luogo e del periodo di ciascuna immagine
- Reportistica e Esportazione Dati



Requisiti Non Funzionali

URPS :

- Usability
- Reliability
- Performance
- Supportability



Requisiti Non Funzionali

Altri fattori + :

- Scalabilità
- Logging e Monitoraggio
- Compatibilità Multi-Dispositivo





03

WORKFLOW

Workflow: fase I

- Analisi del problema
- Analisi delle soluzioni proposte
- Nostro approccio

Inserire partenza e destinazione

Partenza:

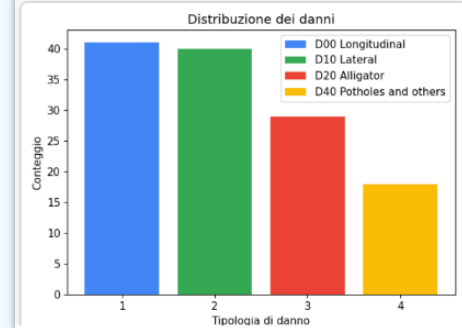
Destinazione:

Ottieni immagini

Webapp Road
Damage Detector



Danni totali: 128

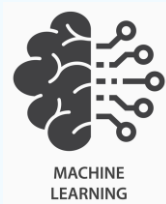


Workflow: fasi 2-3



Determinazione dei requisiti

Framework FURPS+



Selezione del modello

Analisi leaderboard della
Crowdsensing-based
Road Damage Detection
Challenge (CRDDC2022)

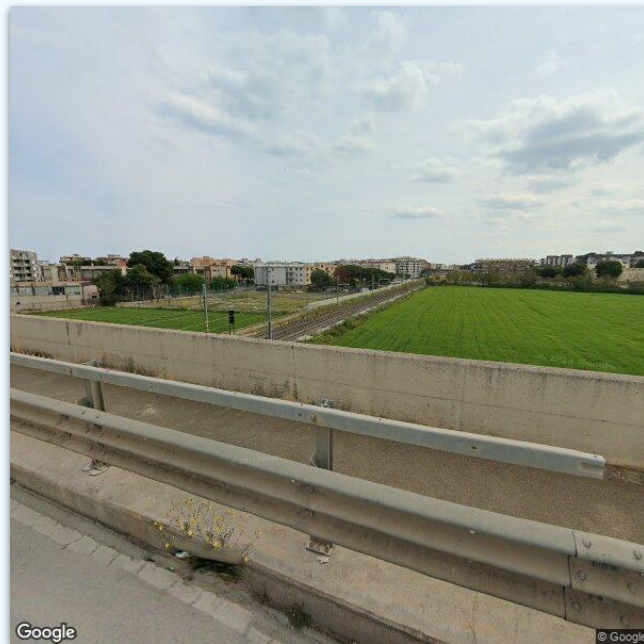
Rank	Team Name	Average Score
1	ShiYu_SeaView	0.716
2	DongjunJeong	0.674
3	MDPT	0.663
4	SGG-RS-Group	0.652
5	IRCV-URV	0.651
6	IMSC	0.649
7	NJUPT	0.640
8	TUT	0.636
9	MILA	0.629
10	SIAI	0.545
11	kubapok	0.530



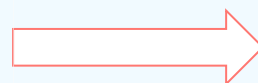
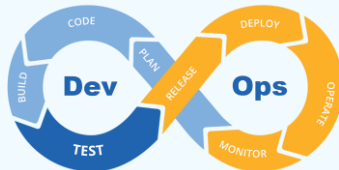
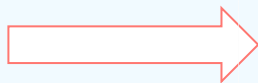
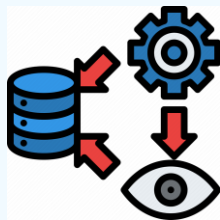
Workflow: fase 4

Scelta delle tecnologie

- Flask per collegare Front-end e Back-end
- API OSRM per ottenere percorso
- API Google Cloud Platform per ottenere le immagini



Workflow: fasi 5-7



Scelta dell'architettura

Utilizzo del pattern MVC

Sviluppo DevOps

- CI/CD
- Collaborazione

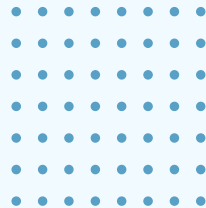
Fase di test

App testata su 3 casi d'uso





Google Cloud Platform



04



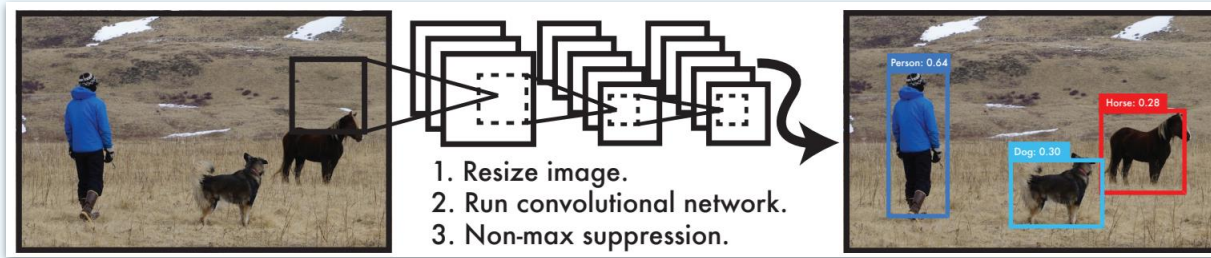
TECNOLOGIE UTILIZZATE





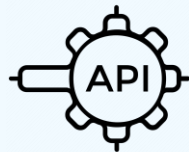
MACHINE
LEARNING

YOLOv7x_640



- Task: Object Detection
- Architettura: Convolutional Neural Network
- Transfer learning





API Esterne

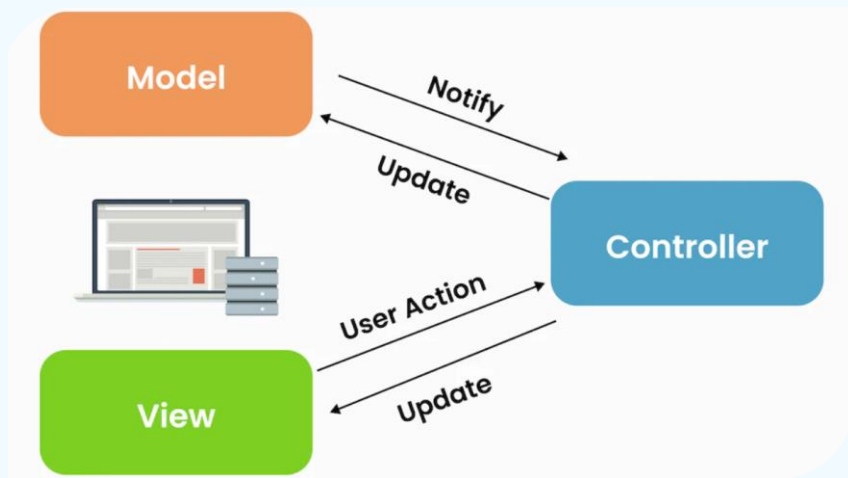
Servizio	Utilizzo
Maps Javascript API	Suggerimenti nella scrittura
Geocoding API	Coordinate partenza e destinazione
Open Source Routing Machine (OSRM)	Calcolo del percorso
Street View Static API	Ottenimento immagini



Flask

- Esposizione di endpoint lato server a disposizione del client
- Motore di templating Jinja2

05



ARCHITETTURA

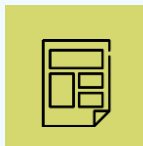
PATTERN MVC

PATTERN MVC



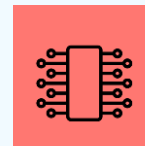
MODEL

Rappresenta la logica dei dati e delle informazioni. Gestisce l'interazione con il database o le API per recuperare e salvare i dati



VIEW

È responsabile di come i dati vengono mostrati all'utente, utilizzando template HTML. Presenta i risultati in modo chiaro e intuitivo.

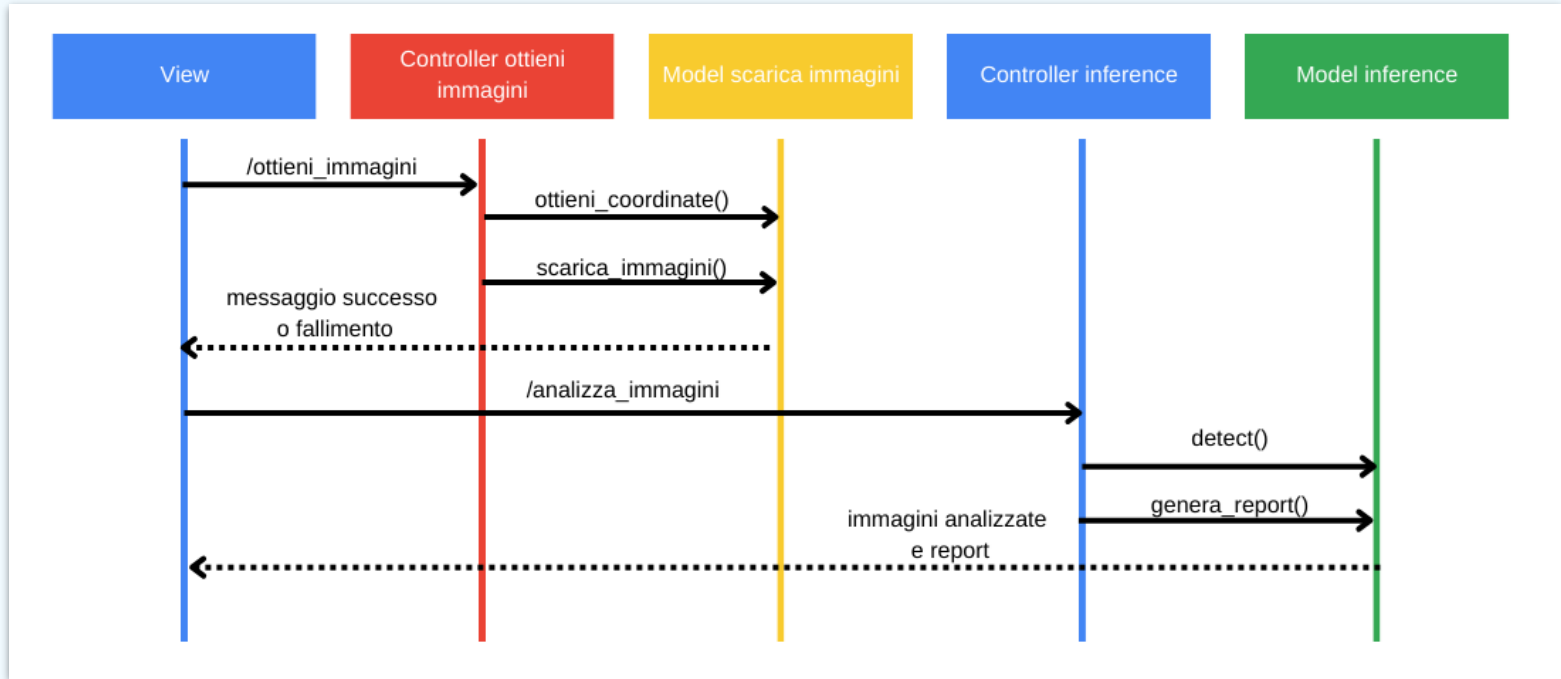


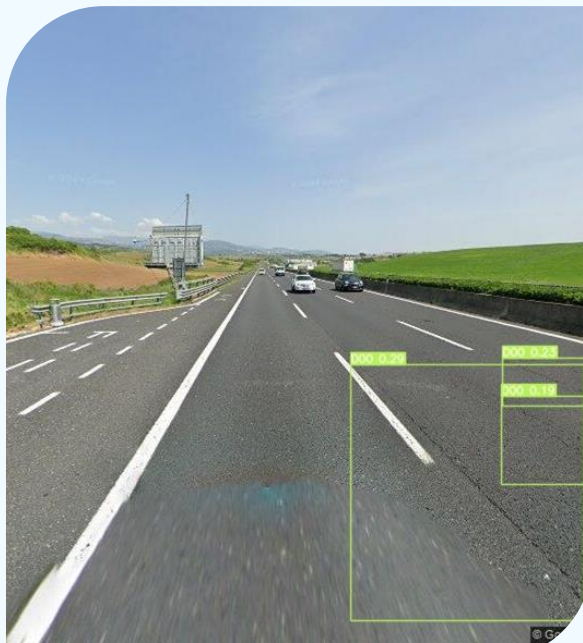
CONTROLLER

Coordina l'interazione tra Model e View. Gestisce le richieste degli utenti, elabora i dati e invia i risultati alla View.



SEQUENCE DIAGRAM





06

ANALISI DEI RISULTATI



OUTPUT DELL'APPLICAZIONE

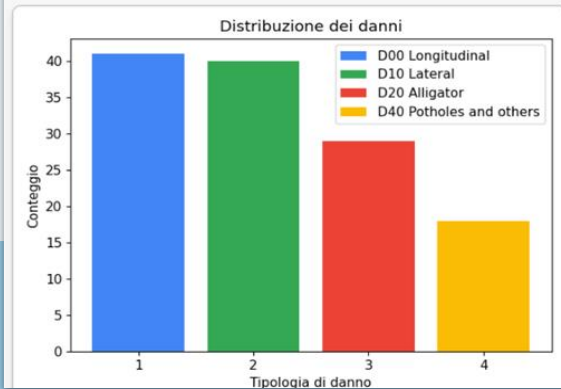
Inserire partenza e destinazione

Partenza:

Destinazione:

Ottieni immagini

Danni totali: 128

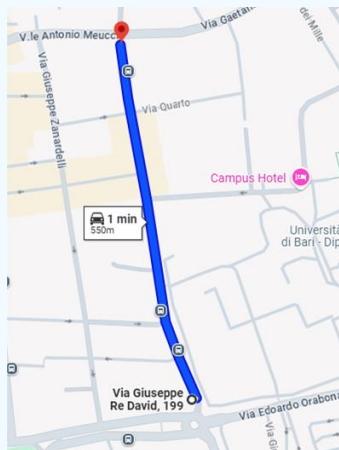


da: Via Giovanni Amendola, 189, 70126 Bari
a: Via Edoardo Orabona, 4, 70125 Bari

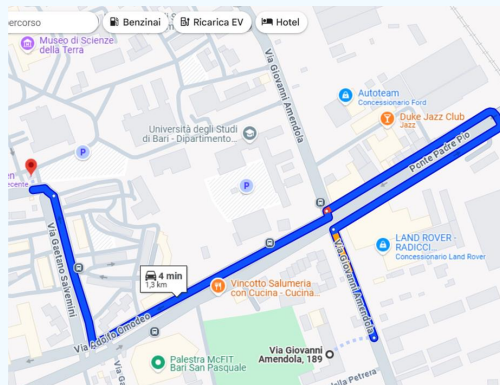
Stradella Petrera 2c, Bari 05-2023



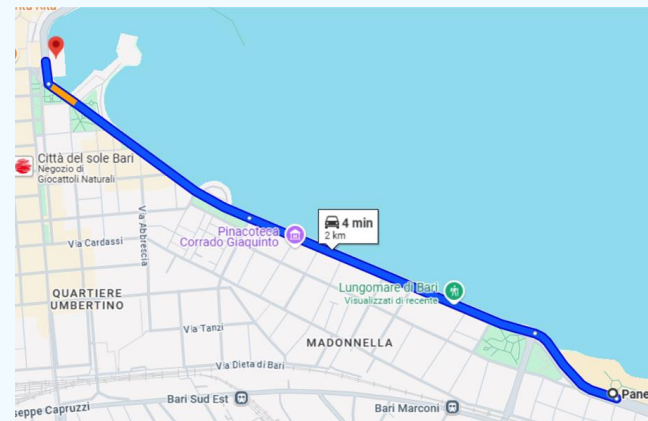
CASI DI TEST



Partenza: via Re David 199
Destinazione: via Re David 272



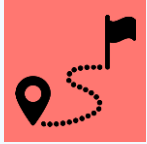
Partenza: via Amendola 189
Destinazione: via Orabona 4



Partenza: Corso Trieste
Destinazione: Piazza IV Novembre

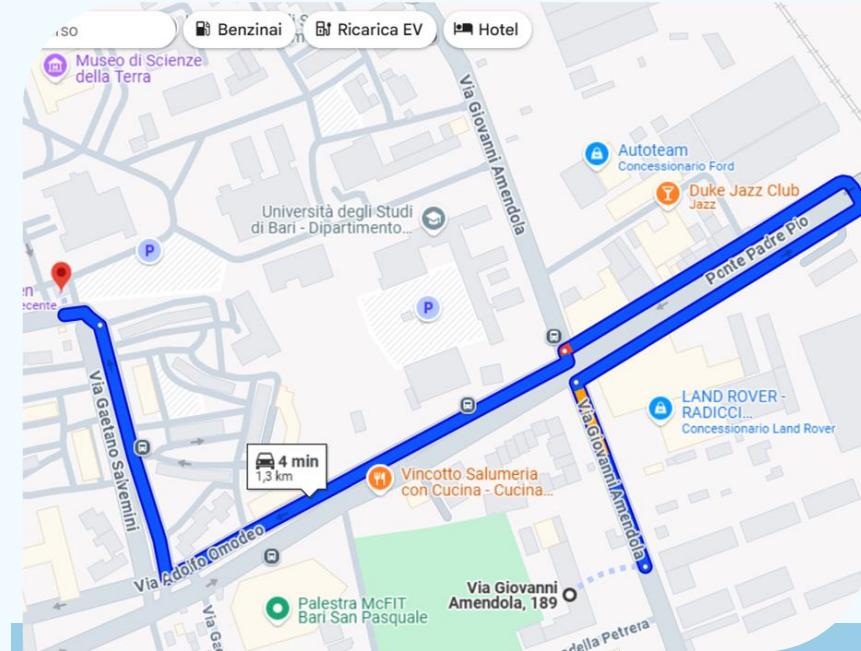


CASO DI TEST

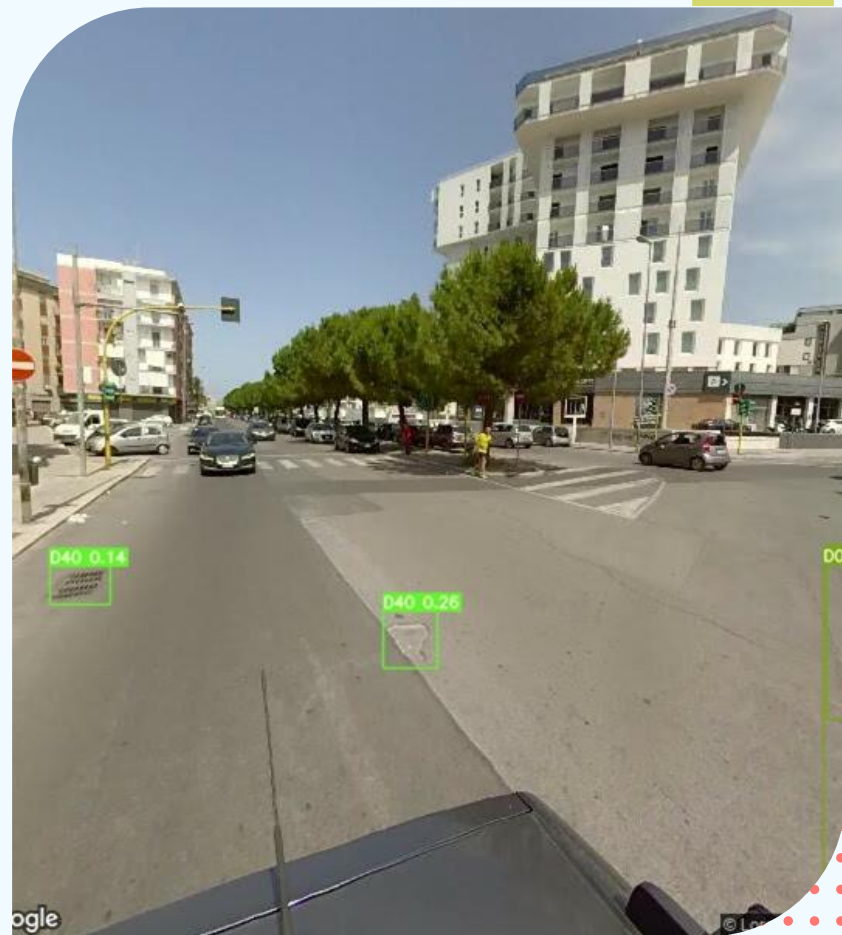
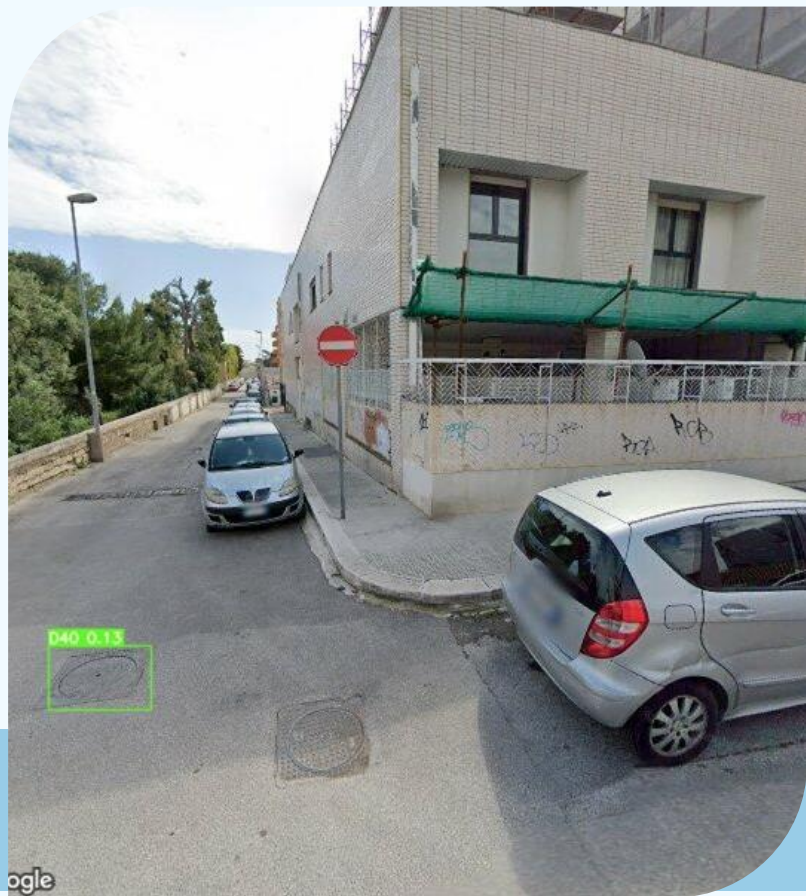


SCELTA DEL PERCORSO

- Lunghezza di circa 1.5 km, ideale per un caso di test urbano.
- Il percorso include diversi tipi di vie, utili per simulare scenari complessi.
- Prossimità al politecnico di Bari.

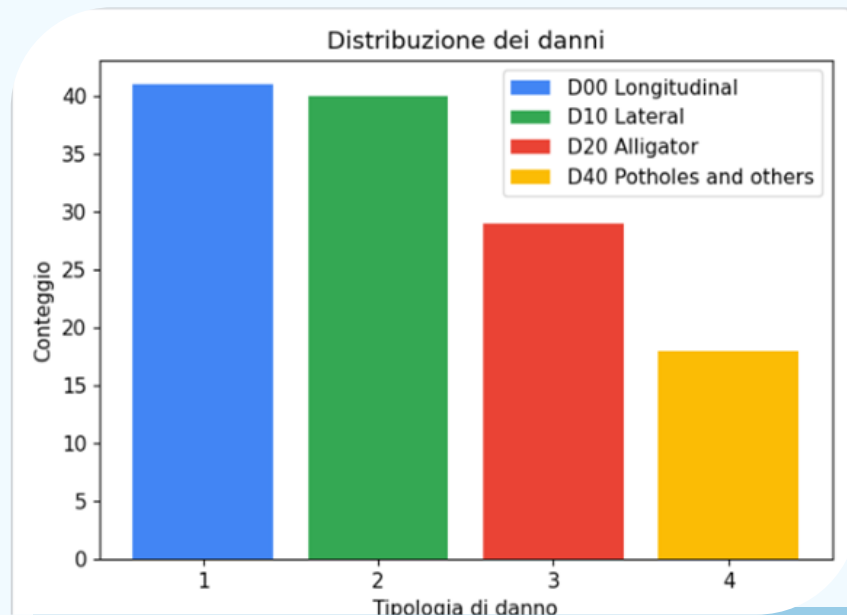






DANNI RISCONTRATI

- Numero di danni consistente, circa 130 danni rilevati.
- Per il percorso scelto i danni longitudinali sono la maggior parte.



CONCLUSIONI



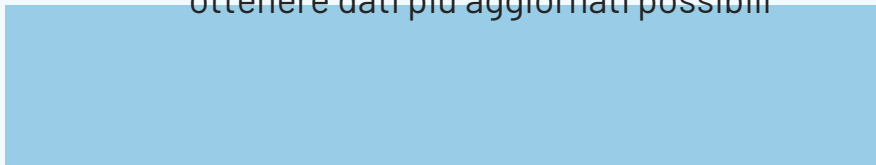
OBIETTIVI RAGGIUNTI

- Analisi automatica di immagini utilizzando un modello di AI per il rilevamento.
- Implementazione di un'interfaccia user-friendly per la gestione dei dati e dei risultati.
- Validazione del sistema su percorsi reali.



PUNTI DI FORZA

- Elevata precisione nel riconoscimento delle categorie di oggetti.
- Struttura modulare del backend Flask, che garantisce flessibilità e scalabilità.
- Utilizzo di microservizi di GCP per ottenere dati più aggiornati possibili





GRAZIE PER L'ATTENZIONE!

Giuseppe Farano, Vito Guida, Antonio Colacicco

