



Calcolo della distanza di oggetti tramite reti neurali

MATTEO GIRI

Introduzione

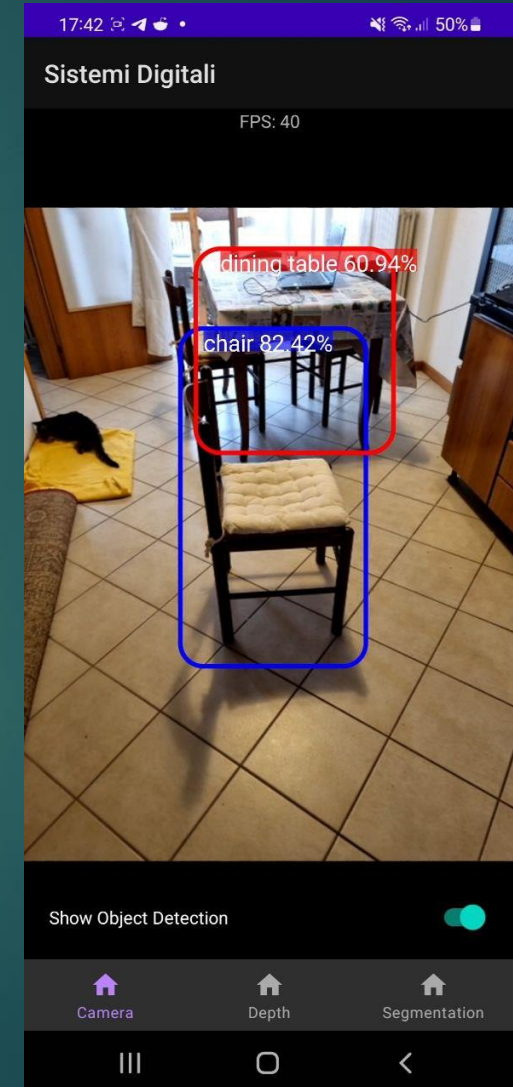
- ▶ L'idea generale del progetto è quella di realizzare un prototipo di applicazione per persone con difficoltà visive che faccia uso di reti neurali per approssimare la distanza di un oggetto dalla fotocamera dello smartphone, e nel caso di segnalarne la troppa vicinanza con un segnale audio
- ▶ Inoltre, l'applicazione permette di visualizzare mappe di profondità e maschere di segmentazione semantica del feed della fotocamera in tempo reale, e permette l'invio di comandi vocali per modificare il comportamento dell'app

Tecnologie utilizzate

Object Detection

Alla base dell'approssimazione della distanza di un oggetto c'è la tecnologia della Object Detection

- ▶ **Object Detection:** rilevamento di istanze di un set conosciuto di oggetti con informazioni sulla posizione
- ▶ **SSD MobileNet v1.1:** rete Single Shot Multibox Detector pre-addestrata sul dataset Coco che prende in input un'immagine e fornisce in output dei vettori che indicano posizione, classe, punteggio e numero di oggetti riconosciuti.



Tecnologie utilizzate

Depth Estimation

Per la creazione di mappe di profondità è stata utilizzata la tecnica della Depth Estimation

- ▶ **Depth estimation:** misurazione della distanza di ogni singolo pixel dalla fotocamera
- ▶ **MiDas:** modello pre-addestrato su 10 dataset diversi che prende in input un'immagine e fornisce in uscita la corrispondente relative inverse depth map



Tecnologie utilizzate

Semantic Scene Segmentation

Per la creazione di maschere di segmentazione è stata usata la tecnica della Semantic Scene Segmentation

- ▶ **Scene Segmentation:** divisione della scena nei suoi vari componenti, assegnando ogni pixel a un oggetto di una certa classe
- ▶ **DeepLab v3.1:** modello di image segmentation pre-addestrato sul dataset Coco che prende in input un'immagine e ritorna le maschere relative agli oggetti riconosciuti

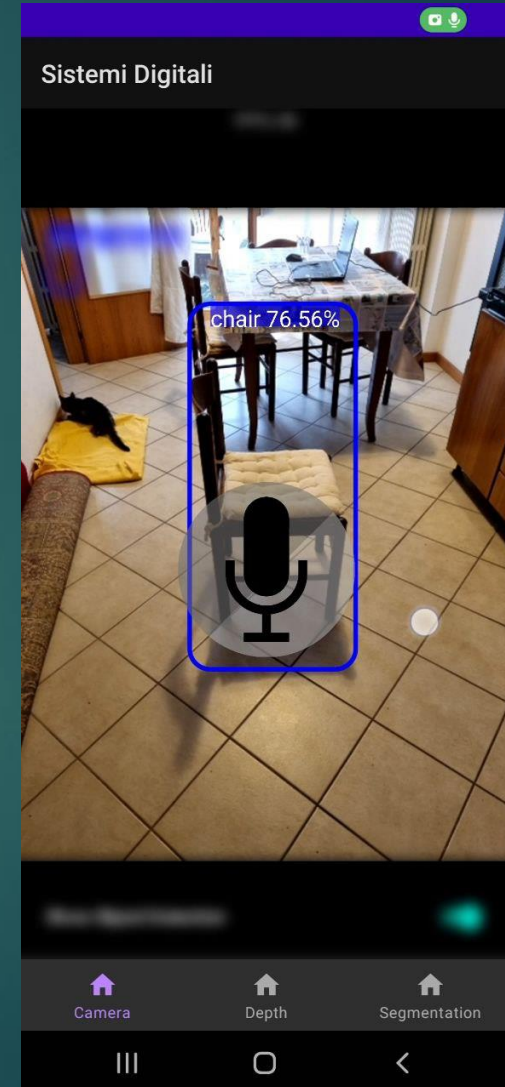


Tecnologie utilizzate

Speech Recognition

Per impartire comandi vocali all'applicazione è stata utilizzata la Speech Recognition

- ▶ **Speech Recognition:** capacità di un programma di processare le parole di un umano trasformandole in testo scritto (Speech-To-Text)
- ▶ **Speech Recognizer:** API Android che permette di accedere al servizio di riconoscimento vocale



Fasi del funzionamento

- ▶ Acquisizione di immagini dal live feed della fotocamera
- ▶ Processamento delle immagini con algoritmi di machine learning
- ▶ Calcolo della distanza degli oggetti dalla fotocamera
- ▶ Attivazione o disattivazione del suono relativo alla troppa vicinanza

- ▶ Trasformazione di messaggi vocali impartiti tramite pressione sullo schermo in comandi testuali che cambiano i parametri dell'applicazione

Aquisizione di immagini

ImageReader.OnImageAvailableListener



onImageAvailable

CameraCaptureSession



Frame



Camera Live Feed

Processamento delle immagini

Una volta acquisito, il frame viene convertito in bitmap per poter essere passato ai modelli di ML

In base alla schermata in cui ci troviamo la bitmap verrà passata a un diverso modello che lo processerà di conseguenza

- ▶ Se ci troviamo nella schermata della Depth Estimation o della Scene Segmentation, il frame viene passato al rispettivo modello e il risultato ottenuto viene mostrato nel layout corrispondente
- ▶ Se ci troviamo nella schermata del Live Feed e Object Detection il risultato viene passato a un tracker che disegna sopra il feed della fotocamera dei rettangoli in base alla posizione degli oggetti rilevati

L'effettivo processamento viene eseguito all'interno di coroutine Kotlin

Approssimazione della distanza

Dopo che un frame è stato processato dal modello di Object Detection, ogni oggetto individuato viene passato a una funzione per calcolarne la distanza

L'approssimazione della distanza viene calcolata confrontando l'area del rettangolo che rappresenta la posizione dell'oggetto rilevato con l'area della scena. Più l'area del rettangolo è grande e più l'oggetto viene considerato vicino alla fotocamera

```
private fun calculateDistance(location: RectF): Float{  
    val maxArea = 640 * 480  
    val area = location.width() * location.height()  
    val ratio: Float = if (area/maxArea <= 1) area/maxArea else 1F  
    val distance = 1F - ratio  
    return distance  
}
```

Attivazione del suono

- ▶ Il suono che indica la troppa vicinanza di un oggetto alla fotocamera viene attivato se tra tutti gli oggetti individuati dalla Object Detection ce ne è almeno uno la cui distanza calcolata è minore di un certo valore di soglia
- ▶ E' realizzato da una coroutine Kotlin che ripete un beep con una frequenza che varia in base a quanto è vicino l'oggetto. Più l'oggetto è vicino e più il beep sarà frequente.

Speech recognition

Tramite una pressione prolungata sullo schermo è possibile attivare il servizio di Speech Recognition che inizia ad ascoltare dal microfono del telefono le frasi che vengono dette

Ogni volta che viene riconosciuta una frase, questa viene trasformata in testo che viene passato a una funzione che controlla se quel testo corrisponde a un comando specifico per l'applicazione

```
fun executeCommand(data: String){
    if (data.contains("disattiva allarme", ignoreCase = true)){
        soundVolume = false
    }
    else if (data.contains("attiva allarme", ignoreCase = true)){
        soundVolume = true
    }
    else if (data.contains("aumenta volume", ignoreCase = true)){
        volume += 0.1f
        alarmSound.setVolume(volume, volume)
    }
    else if (data.contains("diminuisce volume", ignoreCase = true)){
        volume -= 0.1f
        alarmSound.setVolume(volume, volume)
    }
    else if (data.contains("aumenta distanza", ignoreCase = true)){
        MINIMUM_DISTANCE_TRIGGER += 0.05f
    }
    else if (data.contains("diminuisce distanza", ignoreCase = true)){
        MINIMUM_DISTANCE_TRIGGER -= 0.05f
    }
}
```

Video di esempio

