

Università degli Studi di Bari

Dipartimento di Informatica



**Tesi di Laurea Triennale in Informatica e Tecnologie per la
Produzione del Software**

Progettazione sistema SPIR

Autore:

Antonio Ricciardi

Relatore:

Giovanni Dimauro

Correlatori:

Rosalia Maglietta

Carla Cherubini

Anno Accademico 2022/2023

Indice

1	Introduzione	3
1.1	Stato dell'arte	3
1.1.1	Elaborazione delle immagini	3
1.1.2	Feature Extraction	4
1.1.3	Object Segmentation	4
1.1.4	User Interface	4
1.1.5	SPIR	4
1.2	Obiettivi	4
2	Metodi	5
2.1	Porting del sistema SPIR	6
2.1.1	Riscrittura dell'algoritmo originale (MATLAB) in Python .	6
2.1.2	Utilizzo di Python come linguaggio per backend di SPIR .	6
2.1.3	Estrazione della maschera e della pinna del delfino . . .	6
2.1.4	Estrazione delle feature e salvataggio del dataset	6
2.1.5	Match basato sulle feature estratte da SIFT	6
2.1.6	Architettura del sistema	6
2.2	Approccio basato su OpenCV per l'estrazione delle maschere . .	6
2.2.1	Miglioramenti apportati al processo di ritaglio	6
2.2.2	Parallelismo e tempi di esecuzione ridotti	6

2.2.3	Riduzione dello spazio occupato dal dataset di feature estratte	6
2.2.4	Adozione di gRPC per il trasferimento dei dati	6
2.3	Approccio con deep learning per l'estrazione delle maschere . .	6
2.3.1	Implementazione di una rete neurale basata su U-Net .	6
2.3.2	Utilizzo delle maschere per il ritaglio della pinna	6
2.4	Interfaccia utente e interazione	6
2.4.1	Creazione di un'interfaccia per l'interazione con il backend Python	6
2.4.2	Utilizzo di Flutter per sviluppare un'interfaccia multiplatforma	6
3	Risultati sperimentali	7
3.1	Valutazione dei risultati	7
3.1.1	Confronto tra i diversi approcci implementati	7
3.1.2	Analisi delle prestazioni e dei risultati ottenuti	7
3.2	Limitazioni e problematiche riscontrate con approccio Deep Learning	7
4	Conclusioni	8
4.1	Riepilogo degli obiettivi raggiunti	8
4.2	Possibili sviluppi futuri	8
5	Codice sorgente	9
5.1	Implementazione dell'algoritmo SPIR	9
5.2	Implementazione della rete neurale U-Net	9
5.3	Codice per l'estrazione delle maschere con OpenCV	9
5.4	Codice per l'interfaccia utente in Flutter	9

Capitolo 1

Introduzione

Fondamentale implementare la fotoid -

1.1 Stato dell'arte

1.1.1 Elaborazione delle immagini

La manipolazione e l'elaborazione delle immagini svolgono un ruolo fondamentale in numerosi settori, tra cui la visione artificiale, la grafica computazionale e l'analisi di immagini biomediche. In quest'ambito, gli operatori morfologici e logici giocano un ruolo chiave nella modifica e nell'estrazione di informazioni significative da immagini digitali.

Gli operatori morfologici sono basati sulla teoria della morfologia matematica e vengono utilizzati per la manipolazione di forme geometriche all'interno di un'immagine.

Essi consentono di eseguire operazioni come l'erosione, la dilatazione, l'apertura e la chiusura, che possono essere utilizzate per rimuovere rumore, riempire buchi o separare oggetti connessi.

L'erosione riduce le regioni di colore o intensità nell'immagine, mentre la dilatazione le amplia. L'apertura è una combinazione di erosione seguita da dilata-

zione e viene utilizzata per rimuovere piccoli oggetti o dettagli indesiderati. La chiusura, al contrario, consiste in una dilatazione seguita da erosione ed è utile per riempire buchi o connettere regioni separate. Parallelamente agli operatori morfologici, gli operatori logici sono ampiamente utilizzati per combinare e confrontare immagini. Gli operatori logici fondamentali includono l'AND, l'OR e il NOT, che consentono di creare maschere o combinare informazioni da diverse immagini.

Ad esempio, l'operatore AND può essere utilizzato per identificare le regioni comuni tra due immagini, mentre l'operatore OR può essere impiegato per unire due immagini sovrapposte in una sola immagine risultante.

L'operatore NOT, invece, inverte i valori di pixel di un'immagine, consentendo di creare effetti speciali o evidenziare determinate regioni.

1.1.2 Feature Extraction

1.1.3 Object Segmentation

1.1.4 User Interface

1.1.5 SPIR

1.2 Obiettivi

Capitolo 2

Metodi

2.1 Porting del sistema SPIR

2.1.1 Riscrittura dell'algoritmo originale (MATLAB) in Python

2.1.2 Utilizzo di Python come linguaggio per backend di SPIR

2.1.3 Estrazione della maschera e della pinna del delfino

2.1.4 Estrazione delle feature e salvataggio del dataset

2.1.5 Match basato sulle feature estratte da SIFT

2.1.6 Architettura del sistema

2.2 Approccio basato su OpenCV per l'estrazione delle maschere

2.2.1 Miglioramenti apportati al processo di ritaglio

2.2.2 Parallelismo e tempi di esecuzione ridotti

2.2.3 Riduzione dello spazio occupato dal dataset di feature estratte

2.2.4 Adozione di gRPC per il trasferimento dei dati

2.3 Approccio con deep learning per l'estrazione delle maschere

2.3.1 Implementazione di una rete neurale basata su U-Net

6

2.3.2 Utilizzo delle maschere per il ritaglio della pinna

2.4 Interfaccia utente e interazione

2.4.1 Creazione di un'interfaccia per l'interazione con il backend Python

Capitolo 3

Risultati sperimentali

3.1 Valutazione dei risultati

3.1.1 Confronto tra i diversi approcci implementati

3.1.2 Analisi delle prestazioni e dei risultati ottenuti

3.2 Limitazioni e problematiche riscontrate con approccio Deep Learning

Capitolo 4

Conclusioni

4.1 Riepilogo degli obiettivi raggiunti

4.2 Possibili sviluppi futuri

Capitolo 5

Codice sorgente

- 5.1 Implementazione dell'algoritmo SPIR**
- 5.2 Implementazione della rete neurale U-Net**
- 5.3 Codice per l'estrazione delle maschere con OpenCV**
- 5.4 Codice per l'interfaccia utente in Flutter**