Università degli studi di Cagliari

Facoltà di Scienze - Dipartimento di Matematica e Informatica

Progetto di Computer Vision

EgoCart: a Benchmark Dataset for Large-Scale Indoor Image-Based Localization in Retail Stores

Giorgia Campanile
Stefano R. Usai

1 Introduzione

Il progetto consiste nell'affrontare un problema di localizzazione basato su immagini, ovvero costruire un algoritmo che, data un'immagine acquisita in uno spazio noto, permetta di inferire la posizione dalla quale l'immagine è stata scattata. In particolare, noi ci occuperemo di *indoor localization*, ovvero localizzazione all'interno di un edificio. Inoltre, considereremo il problema della classificazione 2D, invece del più generale problema di localizzazione 3D.
Il tema affrontato viene proposto dai ricercatori dell'Università di Catania [1], i quali utilizzano metodi classici basati sul recupero di immagini e metodi emergenti basati sulla regressione. Si tratta di rilevare la posizione di un utente all'interno di un SuperMarket.
Questo problema viene proposto anche come progetto di esame agli studenti del corso di Machine Learning tenuto nell' A.A. 2017/2018 dal Prof. Giovanni Maria Farinella all'interno del corso di studi in Informatica Magistrale dell'Università di Catania [2].
Noi affronteremo questo problema mediante l'utilizzo del BOVW (*Bag Of Visual Words*) assieme alla classificazione mediante i classificatori KNN e SVM, che illustreremo in dettaglio nella sezione successiva.

2 Strumenti utilizzati

In questa sezione verranno presentati gli strumenti da noi utilizzati per la realizzazione del progetto, a partire da quelli forniti dall'Università di Catania assieme alla proposta di progetto, fino a quelli valutati e scelti da noi in seguito allo studio del problema.

2.1 Rappresentazione delle immagini

Le immagini sono state rappresentate tramite modello BOVW (*Bag Of Visual Words*), che consiste nel trattare le feature dell'immagine come fossero delle parole. In breve, un'immagine può essere trattata come un documento, e le feature estratte dall'immagine possono essere considerate come "visual words". Il funzionamento del modello BOVW è illustrato nella figura 1 [3].

- Estrazione dei Key-point

- Individuazione dei centroidi
  Dopo aver portato i vettori in uno spazio unidimensionale [1xN], ciascun vettore verrà identificato da un centroide.

- Clustering
  Per ogni vettore, si verifica a quale centroide sia più vicino.

- Creazione del vocabolario delle Visual Word
  Il vettore viene trasformato in parola (Visual Word). Per ogni punto viene estratto un identificatore numerico (Code Word).

- Creazione dell'istogramma
  Viene creato un istogramma in base alle occorrenze delle Visual Word (in base quindi alle occorrenze dei Code Word), per ciascuna immagine.

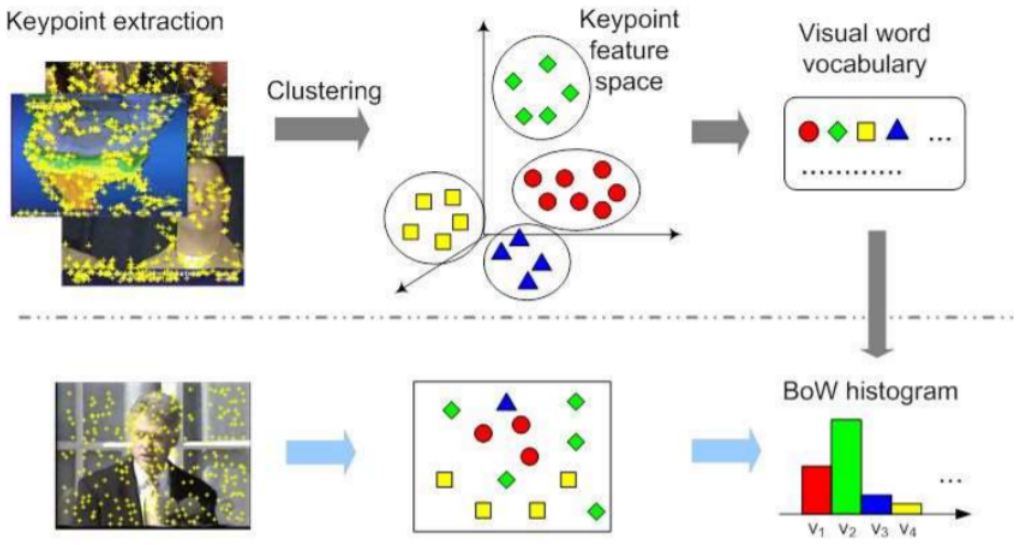

Figura 1: Modello BOVW

2.2 Descrittori

I descrittori vengono usati in diversi contesti e con diversi fini: principalmente, per eseguire il confronto tra punti caratteristici, per generare la map-

pa di disparità nella visione stereoscopica o per fornire una rappresentazione compatta di una porzione dell'immagine (grazie alla quale preserva gran parte dell'informazione). Per il nostro progetto ci siamo focalizzati su due particolari descrittori, e sono i seguenti:

- SURF
  L'algoritmo *Speeded Up Robust Feature* (in sigla SURF) è un rilevatore robusto di caratteristiche locali di un'immagine, presentato da Herbert Bay nel 2006, che può essere usato nell'ambito del riconoscimento di oggetti e ricostruzione 3D in Computer Vision. È parzialmente ispirato al descrittore *scale-invariant feature transform* (SIFT). La versione standard di SURF è diverse volte più veloce di SIFT e, come dichiarano i suoi autori, più robusta. SURF si basa sulle risposte della wavelet di Haar 2D e fa un uso efficiente di immagini integrali. Per rilevare punti di interesse (keypoint), SURF usa un'approssimazione intera del determinante del rilevatore blob di Hessian, che può essere calcolato con 3 operazioni intere usando un'immagine integrale precalcolata. Il suo descrittore di feature è basato sulla somma della risposta della wavelet Haar intorno al punto di interesse, il quale anch'esso può essere calcolato con l'aiuto di un'immagine integrale. I descrittori SURF possono essere usati per localizzare e riconoscere oggetti, persone o volti, per creare scene 3D, per tracciare oggetti e per estrarre punti di interesse. Un'applicazione dell'algoritmo è brevettata negli Stati Uniti d'America [4].

- MSER
  Maximally stable extremal regions (in sigla MSER), che significa "Regioni estremali stabili massimamente", è un metodo usato per il riconoscimento di regioni all'interno di immagini. Fu proposta da Matas et al. per trovare corrispondenze tra elementi da due immagini con due differenti punti di vista. Questo metodo (estrarre un numero comprensivo di elementi da immagini corrispondenti) ha condotto a un miglior matching e ha portato alla creazione di algoritmi migliori di riconoscimento di oggetti [5].

2.3 Dataset

Il dataset utilizzato è stato costruito utilizzando frame estratti da nove diversi video acquisiti in un negozio al dettaglio con un'estensione di 782 m2. I

video sono stati acquisiti con due diverse zed-camera montate su un carrello con i loro assi focali paralleli al pavimento del negozio. Ogni video è stato sottocampionato a 3 fps. Il dataset è costituito complessivamente da 19.531 coppie di immagini RGB e depth. Per i nostri esperimenti abbiamo utilizzato le due suddivisioni del dataset fornite dagli autori [1] [2] .

- La prima suddivisione del dataset (originale) è così strutturata:
  - 13360 immagini per il training set
  - 6171 immagini per il test set
- La seconda suddivisione del dataset (ridotto) è così strutturata:
  - 10259 immagini per il training set
  - 6171 immagini per il test set
  - 3101 immagini per il validation set

  Per questa suddivisione abbiamo utilizzato il training set come addestramento e il validation set come test set, in quanto il test set fornito non riportava le etichette delle classi di appartenenza.

È importante specificare che tutte le immagini sono state trasformate da RGB a scala grigi per poterci lavorare ed estrarre le feature.

2.4 Classificatori

I classificatori utilizzati sono i seguenti:

- K- means clustering
  L'algoritmo K-means è un algoritmo di analisi dei gruppi partizionale che permette di suddividere un insieme di oggetti in K gruppi sulla base dei loro attributi. È una variante dell'algoritmo di aspettativa-massimizzazione (EM) il cui obiettivo è determinare i K gruppi di dati generati da distribuzioni gaussiane. Si assume che gli attributi degli oggetti possano essere rappresentati come vettori, e che quindi formino uno spazio vettoriale [6].

- SVM - *Support Vector Machine*
  Il *Support Vector Machine* (SVM) è un algoritmo di apprendimento automatico supervisionato che può essere utilizzato sia per scopi di classificazione che di regressione. L'algoritmo SVM ottiene la massima efficacia nei problemi di classificazione binari. L'SVM è basato sull'idea di trovare un iperpiano che divida al meglio un set di dati in due classi [7].

- K-NN - *K-Nearest Neighbor*
  Il *K-Nearest Neighbor* (K-NN) è un algoritmo utilizzato nel riconoscimento di pattern per la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini a quello considerato. Un oggetto è classificato in base alla maggioranza dei voti dei suoi K vicini. K è un intero positivo tipicamente non molto grande. Se K=1 allora l'oggetto viene assegnato alla classe del suo vicino. In un contesto binario in cui sono presenti esclusivamente due classi è opportuno scegliere K dispari per evitare di ritrovarsi in situazioni di parità. Questo metodo può essere utilizzato per la tecnica di regressione assegnando all'oggetto la media dei valori dei K oggetti suoi vicini [8].

3 Scelta delle configurazioni ed esperimenti

Abbiamo impostato e provato diverse configurazioni al fine di trovare il modello che portasse ai risultati migliori. Le configurazioni si distinguono per dataset, classificatori, descrittori, parametri (quali dimensione del dizionario delle feature) e metriche considerate. Tutti gli esperimenti sono stati realizzati su MATLAB [9]. Vediamo ora le varie ipotesi e valutazioni che ci hanno permesso di scegliere le configurazioni finali e i parametri utilizzati nella pratica.

3.1 SVM

Abbiamo utilizzato un classificatore multiclasse utilizzando il framework di codici di uscita di correzione degli errori (ECOC) con classificatori binari di supporto vettoriale (SVM). La funzione *trainImageCategoryClassfier* utilizza il BOVW restituito dall'oggetto *bagOfFeatures* per codificare le immagini nell'istogramma dalle feature, che equivalgono alle nostre parole in questo caso.

L'istogramma delle parole viene quindi utilizzato come campione per addestrare il classificatore. Per quanto riguarda i parametri, abbiamo utilizzato quelli di default già pre-impostati da MATLAB.

3.2 K-NN

I modelli di classificatori K-NN sono stati parametrizzati inizialmente in base alla grandezza del BOVW provando manualmente le distanze classiche, come quella euclidea o del coseno, e scegliendo senza un vero criterio la costante K. Dopo i primi esperimenti, valutando i risultati del classificatore K-NN, abbiamo utilizzato la funzione fornita dall'ambiente di sviluppo MATLAB chiamata *fitcknn*. [10]
Questa itera vari tipi di distanza valutando la funzione obiettivo (figura 2) e valutando, di conseguenza, quale sia il K migliore da utilizzare e quale distanza dia i migliori risultati al fine della classificazione.

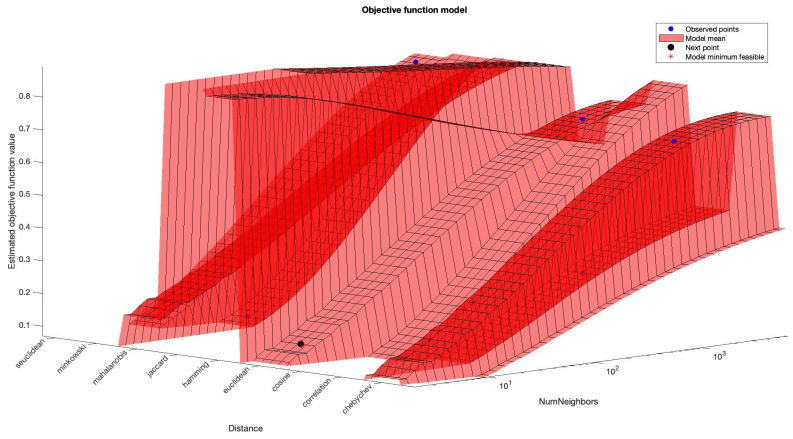

Figura 2: Funzione Obiettivo

3.3 Dizionario

Per quanto riguarda la creazione del dizionario, abbiamo utilizzato la funzione di estrazione delle features *BOVW*. Le feature vengono poi classificate tramite una clusterizzazione effettuata con l'algoritmo K-means. Questi cluster vengono costruiti in base a tutte le feature delle immagini e, in base

alla distanza (distribuzione Gaussiana), vengono assegnate al cluster più vicino. Abbiamo inizialmente considerato le prime 500. Una volta ottenuti i dizionari, gli esperimenti sono stati eseguiti utilizzando i classificatori SVM e KNN, come classificatori di immagini. Abbiamo successivamente esaminato le matrici di confusione risultanti e, in base ad una valutazione di queste ultime, abbiamo preso in considerazione il caso di aumentare o diminuire le parole nel nostro dizionario.
La creazione del dizionario è stata effettuata, per diversi modelli, con il seguente numero di cluster: 100, 500, 1000 o 5000.

Abbiamo notato che, in base al numero di parole, si potesse ottenere un buon riscontro nei risultati dati dai classificatori delle immagini. Considerando un dizionario più grande infatti, ad esempio con 1000 parole, la classificazione delle immagini ha portato più errori sulle classi aventi più punti in comune con le altre classi (le feature considerate non sono abbastanza espressive per descrivere correttamente il target). Infatti, come verrà spiegato più avanti, ci sono classi che hanno descrittori in comune con tutte le altre classi del set. Questo apporta troppo rumore nella classificazione, cioè non risulta possibile discriminare una classe dall'altra. Al contrario, un dizionario troppo piccolo (ad es. 100), non ci ha permesso di ottenere una corretta classificazione in quanto le feature a disposizione non erano in grado di rappresentare correttamente le immagini fornite, data la poca flessibilità del modello.

Queste parole sono classificate tramite K-Means, e il loro numero è di fatto il numero di cluster utilizzato dall'algoritmo. Una volta ottenute le feature da ogni immagine, per ottenere il dizionario vengono selezionate le più rappresentative in modo da poterle selezionare come parola nel nostro dizionario. Specifichiamo inoltre di aver considerato sempre il parametro *Strong Feature* a 1, ovvero abbiamo sempre utilizzato il 100% delle feature più rappresentative per ogni label nell'input dell'*imds*.

3.4 SURF

Per quanto riguarda il descrittore SURF, il quale ricordiamo che ci permette di ottenere i punti caratteristici delle immagini, lo abbiamo utilizzato con due differenti modalità. Quando si imposta PointSelection su 'Detector', i punti

caratteristici vengono selezionati utilizzando un rilevatore di caratteristiche robuste accelerate (SURF).

- Grid
  Consiste nel sovrapporre una griglia di una certa dimensione all'immagine e considerare come keypoint le intersezioni di tale griglia. I keypoint sono i punti selezionati da cui verranno estratte le feature. Naturalmente, è importante una buona parametrizzazione: più le intersezioni della griglia sono distanti (e quindi più i blocchi sono grandi), meno memoria verrà utilizzata. Abbiamo utilizzato le seguenti impostazioni dei parametri di MATLAB: *GridStep* a [32 32], *BlockWidth* a [64 96] per la raccolta dei punti di feature. Nella figura 3 si può osservare il concetto di grid step.

Figura 3: Illustrazione del grid step

- Detector
  Vengono rilevati i punti di interesse. In particolare, SURF utilizza filtri quadrati come un'approssimazione dello Smoothing Gaussiano.

3.5 MSER

Per quanto riguarda MSER, abbiamo utilizzato i parametri di default di MATLAB.

3.6 Elenco delle configurazioni utilizzate

Segue l'elenco di tutte le configurazioni da noi utilizzate per gli esperimenti.

- Dataset originale, dizionario 100, SURF detector, SVM

- Dataset originale, dizionario 500, SURF detector, SVM
- Dataset originale, dizionario 1000, SURF detector, SVM
- Dataset originale, dizionario 5000, SURF detector, SVM
- Dataset originale, dizionario 100, SURF Grid, SVM
- Dataset originale, dizionario 500, SURF Grid, SVM
- Dataset originale, dizionario 1000, SURF Grid, SVM
- Dataset originale, dizionario 5000, SURF Grid, SVM
- Dataset originale, dizionario 100, MSER, SVM
- Dataset originale, dizionario 500, MSER, SVM
- Dataset originale, dizionario 1000, MSER, SVM
- Dataset originale, dizionario 5000, MSER, SVM
- Dataset originale, dizionario 100, SURF detector, KNN
- Dataset originale, dizionario 500, SURF detector, KNN
- Dataset originale, dizionario 1000, SURF detector, KNN
- Dataset originale, dizionario 5000, SURF detector, KNN
- Dataset originale, dizionario 100, SURF Grid, KNN
- Dataset originale, dizionario 500, SURF Grid, KNN
- Dataset originale, dizionario 1000, SURF Grid, KNN
- Dataset originale, dizionario 5000, SURF Grid, KNN
- Dataset originale, dizionario 100, MSER, KNN
- Dataset originale, dizionario 500, MSER, KNN
- Dataset originale, dizionario 1000, MSER, KNN
- Dataset originale, dizionario 5000, MSER, KNN

- Dataset ridotto, dizionario 100, SURF detector, SVM
- Dataset ridotto, dizionario 500, SURF detector, SVM
- Dataset ridotto, dizionario 1000, SURF detector, SVM
- Dataset ridotto, dizionario 5000, SURF detector, SVM
- Dataset ridotto, dizionario 100, SURF Grid, SVM
- Dataset ridotto, dizionario 500, SURF Grid, SVM
- Dataset ridotto, dizionario 1000, SURF Grid, SVM
- Dataset ridotto, dizionario 5000, SURF Grid, SVM
- Dataset ridotto, dizionario 100, MSER, SVM
- Dataset ridotto, dizionario 500, MSER, SVM
- Dataset ridotto, dizionario 1000, MSER, SVM
- Dataset ridotto, dizionario 5000, MSER, SVM
- Dataset ridotto, dizionario 100, SURF detector, KNN
- Dataset ridotto, dizionario 500, SURF detector, KNN
- Dataset ridotto, dizionario 1000, SURF detector, KNN
- Dataset ridotto, dizionario 5000, SURF detector, KNN
- Dataset ridotto, dizionario 100, SURF Grid, KNN
- Dataset ridotto, dizionario 500, SURF Grid, KNN
- Dataset ridotto, dizionario 1000, SURF Grid, KNN
- Dataset ridotto, dizionario 5000, SURF Grid, KNN
- Dataset ridotto, dizionario 100, MSER, KNN
- Dataset ridotto, dizionario 500, MSER, KNN
- Dataset ridotto, dizionario 1000, MSER, KNN
- Dataset ridotto, dizionario 5000, MSER, KNN

4 Analisi dei risultati

Il modello più forte e soddisfacente tra quelli sperimentati è quello dato dalla configurazione seguente: dataset originale, Vocabulary Size 500, feature estratte con SURF Detector, classificatore KNN. Con questo modello abbiamo ottenuto il 95.37% di accuracy.
La matrice di confusione relativa ai risultati ottenuti mediante questo modello è riportata in figura 4.

162	0	0	0	0	0	0	0	0	0	3	0	0	0	1	7
0	121	0	0	0	0	0	0	0	0	6	0	0	0	4	0
0	0	152	0	0	0	0	0	0	0	17	0	0	4	4	0
0	0	0	111	0	0	0	0	0	0	0	0	0	0	6	0
0	0	0	0	169	0	0	0	0	0	2	0	0	0	3	0
0	0	0	0	0	208	0	0	0	0	9	0	0	0	0	0
0	0	0	0	0	0	182	0	0	0	2	0	0	0	17	0
0	0	0	0	0	0	0	186	0	0	0	0	0	0	4	0
0	0	0	0	0	0	0	0	223	5	1	0	0	0	6	0
0	2	0	0	0	0	0	0	0	719	13	8	0	3	6	0
0	0	0	3	2	0	0	2	3	8	623	12	3	0	1	1
0	0	0	0	0	0	0	0	0	0	0	403	5	0	1	0
3	10	0	0	0	0	0	0	0	0	0	7	430	5	1	0
0	1	1	0	0	0	0	1	0	0	0	0	0	776	9	0
2	1	1	0	2	13	0	2	7	19	1	7	0	13	1057	5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	363

Figura 4: Matrice di confusione Dataset originale, 500, SURF detector, KNN

Nella tabella 1, esportata da MATLAB, sono riportati i parametri utilizzati per la creazione del modello KNN risultanti dalla funzione *fitcknn* [10].

Iter	Eval result	Objective	Objective runtime	BestSoFar obser.	BestSoFar estim.	NumNeighbors	Distance
1	*Best*	0.6881	34.363	0.6881	0.6881	2171	*euclidean*
2	*Accept*	0.87058	40.503	0.6881	0.7011	2	*hamming*
3	*Best*	0.042814	27.825	0.042814	0.043394	20	*spearman*
4	*Accept*	0.76272	43.316	0.042814	0.11409	4852	*seuclidean*
5	*Best*	0.038398	27.955	0.038398	0.04065	17	*spearman*
6	*Best*	0.02253	26.33	0.02253	0.022586	2	*cityblock*
7	*Accept*	0.085105	29.391	0.02253	0.022575	2	*chebychev*
8	*Accept*	0.023129	22.716	0.02253	0.022566	2	*correlation*
9	*Best*	0.021108	22.64	0.021108	0.021139	1	*cosine*
10	*Accept*	0.8003	1723.8	0.021108	0.021145	6666	*mahalanobis*
11	*Accept*	0.021108	23.213	0.021108	0.021141	1	*minkowski*
12	*Accept*	0.90487	47.541	0.021108	0.021147	1	*jaccard*
13	*Accept*	0.80015	30.328	0.021108	0.021151	6643	*cosine*
14	*Accept*	0.089895	23.862	0.021108	0.021147	53	*minkowski*
15	*Accept*	0.29454	23.649	0.021108	0.021147	393	*cityblock*
16	*Accept*	0.28653	23.446	0.021108	0.021146	373	*correlation*
17	*Accept*	0.68503	24.9	0.021108	0.021144	1097	*chebychev*
18	*Best*	0.020734	26.835	0.020734	0.020759	1	*spearman*
19	*Accept*	0.75045	33.302	0.020734	0.020782	6664	*spearman*
20	*Accept*	0.023054	23.506	0.020734	0.020778	5	*minkowski*
21	*Accept*	0.80015	30.616	0.020734	0.020783	6661	*minkowski*
22	*Accept*	0.03256	22.97	0.020734	0.020783	10	*correlation*
23	*Accept*	0.020883	23.324	0.020734	0.020782	1	*cityblock*
24	*Accept*	0.02515	23.358	0.020734	0.020782	6	*cosine*
25	*Accept*	0.021707	27.486	0.020734	0.020878	3	*spearman*
26	*Best*	0.021332	25.529	0.020734	0.020876	1	*seuclidean*
27	*Accept*	0.024701	25.341	0.020734	0.020877	6	*seuclidean*
28	*Accept*	0.14064	1824.5	0.020734	0.020868	1	*mahalanobis*
29	*Accept*	0.021108	23.83	0.020734	0.020849	1	*euclidean*
30	*Accept*	0.8003	55.052	0.020734	0.020863	6636	*jaccard*

Tabella 1: Tabella funzione *fitcknn*

Per quanto riguarda gli esperimenti effettuati con la dimensione del dizionario impostata a 5000, possiamo trarre l'unica conclusione che essi risultino eccessivamente pesanti in termini di computazione, tanto che non sono mai giunti al termine.
Qui di seguito riportiamo le matrici di confusione di alcuni tra tutti gli altri esperimenti effettuati, i cui risultati non sono però soddisfacenti.

0,9480	0	0	0	0	0,0058	0	0,0058	0,0058	0	0	0	0,0116	0	0,0058	0,0173
0,0076	0,9389	0	0,0076	0	0	0	0,0076	0,0076	0	0	0	0,0153	0,0076	0	0,0076
0,0113	0,0113	0,8023	0,0395	0	0,0056	0,0056	0,0113	0,0056	0	0,0282	0	0,0056	0,0339	0,0395	0
0,0256	0,0256	0,0085	0,8632	0,0513	0	0,0171	0	0	0	0,0085	0	0	0	0	0
0,0287	0,0057	0,0172	0,0402	0,8161	0,0115	0,0115	0,0057	0,0057	0,0057	0	0,0172	0	0,0057	0,0230	0,0057
0,0046	0,0046	0	0	0	0,9355	0	0,0046	0	0,0046	0,0092	0	0	0	0,0369	0
0	0	0	0	0,0100	0	0,9851	0	0	0	0,0050	0	0	0	0	0
0	0	0,0053	0	0	0,0053	0	0,8789	0,0053	0,0263	0,0105	0,0053	0,0211	0,0211	0,0211	0
0,0043	0	0	0,0170	0	0,0085	0,0340	0,0043	0,9021	0,0085	0,0043	0,0128	0,0043	0	0	0
0,0027	0,0173	0,0186	0,0040	0,0226	0,0027	0,0040	0,0133	0,0386	0,6178	0,0333	0,0945	0,0479	0,0213	0,0559	0,0053
0,0122	0,0046	0,0182	0,0289	0,0091	0,0258	0,0076	0,0152	0,0046	0,0182	0,6793	0,0395	0,0623	0,0091	0,0350	0,0304
0	0	0,0049	0	0	0	0,0024	0,0098	0,0049	0,0147	0,0489	0,8778	0,0171	0	0,0098	0,0098
0,0088	0,0022	0,0044	0,0132	0,0022	0,0022	0,0088	0,0351	0,0461	0,0219	0,0373	0,0263	0,7632	0,0110	0,0154	0,0022
0,0190	0,0190	0,0051	0,0457	0,0165	0,0241	0,0216	0,0305	0,0228	0,0254	0,0152	0,0063	0,0178	0,6802	0,0330	0,0178
0,0106	0,0177	0,0142	0,0212	0,0310	0,0310	0,0159	0,0265	0,0221	0,0398	0,0195	0,0292	0,0363	0,0487	0,6177	0,0186
0,0495	0,0165	0,0275	0,0027	0,0110	0,0137	0,0082	0,0247	0,0027	0,0055	0,0137	0,0027	0,0055	0,0110	0,0055	0,7995

Figura 5: Matrice di confusione Dataset originale, 100, MSER, SVM

0,9827	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,0173
0	0,9924	0,0076	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0,9831	0,0113	0	0	0	0	0	0	0,0056	0	0	0	0	0
0	0	0	0,9915	0	0	0,0085	0	0	0	0	0	0	0	0	0
0	0	0	0	0,9885	0	0	0	0	0	0,0057	0	0	0	0,0057	0
0	0	0	0	0	0,9908	0	0	0	0	0,0092	0	0	0	0	0
0	0	0	0	0	0	1,0000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0,9947	0	0	0,0053	0	0	0	0	0
0	0	0	0	0	0	0	0	0,9957	0	0	0	0	0	0,0043	0
0	0	0,0027	0,0053	0,0013	0	0,0080	0	0,0040	0,9108	0,0133	0,0173	0,0120	0,0107	0,0133	0,0013
0,0061	0,0046	0,0152	0,0061	0,0106	0,0015	0,0137	0,0046	0,0106	0,0152	0,8708	0,0228	0,0076	0,0015	0,0091	0
0	0	0	0	0	0	0,0024	0	0	0	0,0122	0,9731	0,0122	0	0	0
0	0	0	0	0	0	0,0022	0,0022	0	0	0	0,0110	0,9803	0,0044	0	0
0,0063	0,0038	0	0,0038	0,0025	0,0051	0	0,0013	0,0013	0,0038	0,0063	0	0,0076	0,9327	0,0178	0,0076
0,0106	0,0044	0,0204	0,0044	0,0062	0,0115	0,0186	0,0071	0,0088	0,0310	0,0018	0,0062	0,0053	0,0434	0,8044	0,0159
0,0110	0	0	0	0	0	0	0	0	0	0	0,0027	0,0027	0	0,0027	0,9808

Figura 6: Matrice di confusione Dataset originale, 500, MSER, SVM

0,9827	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,0173
0	0,9847	0,0153	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0,9944	0	0	0	0	0	0	0	0,0057	0	0	0	0	0
0	0	0	0,9744	0,0171	0	0	0	0,0085	0	0	0	0	0	0	0
0	0	0	0,0057	0,9770	0	0	0	0	0,0057	0,0058	0	0	0,0057	0,0057	0
0	0	0	0	0	0,9908	0	0	0	0,0046	0,0092	0	0	0	0	0
0	0	0	0	0	0	0,9950	0	0	0	0,0050	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0,0263	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0,9915	0,0085	0,0085	0	0	0	0	0
0,0040	0	0	0	0	0	0	0	0,0040	0,9521	0,0133	0,0160	0,0039	0	0,0067	0
0,0091	0,0030	0,0152	0,0046	0,0046	0,0030	0,0061	0,0076	0,0106	0,0152	0,8982	0,0106	0,0076	0,0015	0,0015	0,0015
0	0	0	0	0	0	0	0	0	0,0025	0,0073	0,9829	0,0073	0	0	0
0,0066	0,0066	0	0	0	0	0,0022	0	0,0066	0,0022	0	0,0219	0,9430	0,0022	0,0088	0
0	0,0013	0,0076	0,0013	0,0025	0	0	0	0	0	0	0	0,0038	0,9721	0,0038	0,0076
0,0115	0,0071	0,0044	0,0062	0,0071	0,0124	0,0177	0,0124	0,0080	0,0239	0,0089	0,0053	0,0018	0,0531	0,8150	0,0133
0,0192	0	0	0	0	0	0	0	0	0	0	0	0	0	0,0027	0,9780

Figura 7: Matrice di confusione Dataset originale, 500, SURF detector, SVM

0,9075	0,0058	0,0116	0,0116	0,0116	0	0	0	0,0058	0,0058	0,0058	0	0	0,0116	0	0,0231
0,0153	0,9008	0,0153	0,0076	0,0076	0	0,0076	0	0	0	0,0076	0	0	0,0305	0,0076	0
0,0113	0,0169	0,8757	0,0226	0	0,0226	0,0056	0,0113	0	0,0056	0,0113	0	0	0,0113	0,0056	0
0,0171	0	0,0085	0,9145	0,0171	0	0,0256	0,0085	0	0	0,0085	0	0	0	0	0
0,0057	0,0230	0	0,0230	0,8678	0	0,0057	0,0115	0	0,0172	0,0057	0,0115	0,0057	0,0115	0	0,0115
0,0046	0	0	0	0	0,9078	0,0046	0,0046	0	0	0,0138	0,0092	0,0276	0,0046	0,0230	0
0	0	0	0	0	0	0,9900	0,0050	0	0	0,0050	0	0	0	0	0
0	0	0	0	0	0	0	0,9789	0	0,0053	0,0053	0	0,0053	0	0	0,0053
0	0	0	0	0	0	0	0	0,9745	0,0085	0,0128	0,0043	0	0	0	0
0,0053	0,0027	0,0040	0,0067	0,0053	0,0000	0,0067	0,0080	0,0692	0,8083	0,0213	0,0133	0,0146	0,0067	0,0226	0,0053
0,0137	0,0274	0,0243	0,0122	0,0152	0,0091	0,0076	0,0152	0,0076	0,0091	0,7796	0,0228	0,0258	0,0122	0,0106	0,0076
0	0,0073	0	0	0,0024	0	0	0	0	0,0049	0,0098	0,9707	0,0049	0	0	0
0,0022	0,0022	0,0066	0,0066	0,0022	0,0044	0,0022	0,0088	0,0088	0,0197	0,0110	0,0241	0,8816	0,0044	0,0044	0,0110
0,0114	0,0292	0,0076	0,0203	0,0254	0,0140	0,0051	0,0203	0,0063	0,0127	0,0190	0,0013	0,0165	0,7741	0,0190	0,0178
0,0115	0,0124	0,0186	0,0088	0,0097	0,0195	0,0389	0,0451	0,0310	0,0434	0,0257	0,0257	0,0124	0,0584	0,6133	0,0257
0,0275	0,0027	0,0027	0,0055	0	0	0	0	0,0027	0,0165	0,0055	0	0,0082	0,0110	0,0110	0,9066

Figura 8: Matrice di confusione Dataset originale, 500, SURF grid, SVM

0,7654	0,0370	0,0741	0,0123	0	0,0247	0,0247	0,0123	0,0247	0	0,0123	0	0,0123	0	0	0
0,0690	0,6207	0,1724	0,0230	0	0	0,0230	0,0230	0,0230	0	0,0230	0	0,0115	0	0,0115	0
0	0	0,9565	0,0435	0	0	0	0	0	0	0	0	0	0	0	0
0,0465	0,0116	0,1047	0,7326	0	0	0,0349	0	0,0116	0	0,0233	0,0116	0	0,0116	0,0116	0
0	0,0328	0,1257	0,0383	0,6230	0,0273	0,0109	0,0273	0,0055	0	0,0164	0,0055	0,0164	0,0164	0,0437	0,0109
0	0,0583	0,0291	0,0097	0	0,8058	0,0097	0,0000	0,0388	0	0	0,0097	0	0	0,0291	0,0097
0,0729	0,0104	0,2083	0,0417	0,0208	0,0313	0,4063	0,0417	0,0521	0,0104	0,0104	0	0,0208	0,0208	0,0313	0,0208
0,0125	0,0500	0,1375	0,0250	0,0750	0,0125	0,0250	0,4750	0,0500	0,0125	0,0250	0,0250	0,0125	0	0,0500	0,0125
0	0,0130	0,0390	0	0	0	0	0	0,9221	0,0130	0	0	0	0	0,0130	0
0,0022	0,0086	0,0538	0,0086	0,0065	0,0129	0,0043	0,0022	0,0215	0,6753	0,0086	0,0495	0,0430	0,0043	0,0753	0,0237
0,0199	0,0199	0,0861	0,0232	0,0033	0,0563	0,0265	0,0099	0,0099	0,0563	0,4371	0,0430	0,0861	0,0298	0,0762	0,0166
0,0037	0,0037	0,0296	0,0037	0	0	0	0	0,0148	0,0519	0	0,8630	0,0111	0,0037	0,0148	0
0,0067	0,0067	0,0600	0,0200	0,0133	0,0067	0	0,0067	0,0067	0,0200	0,0200	0,0267	0,7600	0	0,0400	0,0067
0,0476	0,0863	0,1012	0,1339	0,0268	0,0417	0,0536	0,0268	0,0417	0,0268	0,0714	0,0149	0,0506	0,1637	0,0685	0,0446
0,0145	0,0271	0,0723	0,0090	0,0072	0,0271	0,0181	0,0253	0,0344	0,0289	0,0307	0,0163	0,0344	0,0072	0,6329	0,0145
0,0269	0,0215	0,0108	0,0108	0	0,0054	0,0108	0,0054	0,0054	0	0,0054	0	0	0,0054	0,0108	0,8817

Figura 9: Matrice di confusione Dataset ridotto, 100, MSER, SVM

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0,9651	0	0	0	0	0	0	0	0	0,0116	0,0116	0	0,0116
0,0055	0	0,0328	0,0055	0,9235	0,0055	0,0055	0,0055	0	0	0,0109	0	0	0	0,0055	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0,0104	0	0	0,0208	0,9375	0,0104	0	0	0	0,0104	0	0	0	0,0104
0	0	0	0	0	0	0	0,9750	0	0	0,0250	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0,0022	0,0022	0,0043	0	0,0043	0,0065	0,0022	0,0043	0,0065	0,9333	0,0086	0,0022	0,0086	0,0022	0,0129	0
0,0066	0,0099	0,0132	0,0099	0,0033	0,0397	0,0033	0	0	0,0099	0,8245	0,0132	0,0066	0,0331	0,0199	0,0066
0	0	0,0074	0	0	0,0037	0	0	0	0,0259	0	0,9630	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0,0067	0,9867	0,0067	0	0
0,0208	0,0149	0,0238	0,0268	0	0,0208	0,0060	0,0089	0,0060	0,0030	0,0119	0,0089	0,0060	0,8155	0,0208	0,0060
0,0090	0,0163	0,0090	0,0127	0,0108	0,0506	0,0199	0,0145	0,0108	0,0235	0,0072	0,0072	0,0127	0,0181	0,7667	0,0108
0,0108	0	0,0054	0,0054	0	0	0	0	0	0	0	0	0	0,0054	0,0054	0,9677

Figura 10: Matrice di confusione Dataset ridotto, 500, MSER, , SVM

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0,0233	0,9767	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0,0164	0,0055	0,9672	0	0	0	0	0	0,0055	0	0	0	0	0,0055
0	0	0	0	0	0,9903	0,0097	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0,0129	0	0	0,0043	0	0	0,0065	0,9548	0,0022	0,0086	0,0022	0,0086	0	0
0,0066	0,0033	0,0762	0,0132	0	0,0166	0	0,0033	0,0033	0,0166	0,8344	0	0,0066	0,0099	0,0099	0
0	0	0,0148	0,0037	0	0	0	0	0	0,0259	0,0000	0,9519	0,0037	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0,0030	0,0536	0,0089	0,0060	0,0089	0,0030	0,0030	0,0030	0	0	0	0	0,9077	0	0,0030
0,0072	0,0054	0,0524	0,0127	0,0054	0,0289	0,0054	0,0145	0,0036	0,0054	0,0018	0,0036	0,0036	0,0235	0,8174	0,0090
0	0	0,0054	0	0	0	0	0	0	0	0	0	0	0	0	0,9946

Figura 11: Matrice di confusione Dataset ridotto, 500, SURF detector, SVM

0,8889	0	0,0864	0,0123	0	0	0	0,0123	0	0	0	0	0	0	0	0
0	0,9080	0,0805	0	0	0	0	0	0	0,0115	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0,9070	0	0	0	0	0	0	0	0	0	0	0	0
0,0109	0,0219	0,1585	0,0164	0,6885	0	0,0164	0,0164	0	0,0055	0,0109	0,0109	0,0055	0,0109	0,0055	0,0219
0	0,0097	0,1650	0,0097	0	0,7670	0	0,0291	0	0	0	0	0	0	0	0,0194
0	0	0,1458	0	0	0	0,8438	0	0	0	0	0	0	0	0,0104	0
0	0,0500	0,1375	0	0	0	0	0,8000	0	0	0,0125	0	0	0	0	0
0,0130	0	0,0130	0	0	0	0	0	0,9740	0	0	0	0	0	0	0
0,0065	0,0151	0,0495	0,0065	0,0043	0,0065	0,0086	0,0043	0,0366	0,7677	0,0086	0,0366	0,0151	0,0022	0,0215	0,0108
0,0232	0,0199	0,1093	0,0066	0,0033	0,0199	0,0066	0	0,0033	0,0199	0,7020	0,0132	0,0232	0,0132	0,0331	0,0033
0,0037	0,0037	0,0037	0	0,0037	0	0	0,0037	0	0,0296	0	0,9259	0,0074	0,0074	0,0074	0,0037
0	0	0,0133	0	0	0	0	0,0133	0,0067	0	0,0067	0	0,9600	0	0	0
0,0298	0,0387	0,1488	0,0179	0,0238	0,0060	0,0327	0,0298	0,0119	0,0149	0,0238	0,0208	0,0179	0,5327	0,0268	0,0238
0,0199	0,0145	0,0687	0,0090	0,0036	0,0072	0,0181	0,0072	0,0072	0,0362	0,0434	0,0199	0,0181	0,0163	0,6926	0,0181
0	0,0054	0,0161	0,0054	0,0054	0,0054	0	0	0	0	0,0108	0	0	0,0054	0,0054	0,9409

Figura 12: Matrice di confusione Dataset ridotto, 500, SURF grid, SVM

161	0	0	0	0	0	0	0	0	0	4	0	0	0	1	7
0	121	0	0	0	0	0	0	0	0	6	0	0	1	2	1
1	0	152	0	0	0	0	1	0	0	18	0	0	1	4	0
0	0	0	110	0	0	0	0	0	0	1	0	0	0	6	0
0	0	0	0	168	0	0	0	0	0	2	0	0	0	4	0
0	0	0	0	0	208	0	0	0	0	9	0	0	0	0	0
0	0	0	0	0	0	171	0	0	0	5	0	0	0	25	0
0	0	0	0	0	0	0	185	0	0	0	0	0	0	5	0
0	0	0	0	0	0	1	0	218	7	0	1	0	0	8	0
1	1	0	0	1	0	0	0	2	677	20	10	4	5	29	1
0	0	2	2	4	1	0	3	2	14	609	13	1	2	5	0
0	0	0	0	0	0	0	0	0	5	2	393	9	0	0	0
1	0	0	0	0	0	2	1	1	2	0	7	428	6	8	0
0	2	0	0	5	7	0	3	0	0	8	0	6	732	25	0
3	1	1	0	0	10	1	1	5	19	5	7	4	22	1050	1
3	0	0	2	1	0	0	0	0	0	0	0	0	1	1	356

Figura 13: Matrice di confusione Dataset originale, 100, MSER, KNN

164	0	0	0	0	0	0	0	0	0	1	0	0	0	1	7
0	121	0	0	0	0	0	0	0	0	6	0	0	1	3	0
1	1	153	0	1	0	1	0	0	0	17	0	0	0	3	0
0	0	0	111	0	0	0	0	0	0	0	0	0	0	6	0
0	0	0	0	170	0	0	0	0	0	2	0	0	0	2	0
0	0	0	0	0	209	0	0	0	0	7	0	0	0	1	0
0	0	0	0	0	0	177	0	0	0	4	0	0	0	20	0
0	0	0	0	1	0	0	187	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	223	3	0	0	0	0	9	0
1	0	0	1	6	0	3	0	0	681	16	8	1	9	25	0
0	1	0	2	5	2	4	3	2	6	622	6	1	0	4	0
0	0	0	0	0	0	0	0	0	0	1	402	6	0	0	0
0	1	2	0	0	0	0	0	0	2	0	5	440	3	2	1
0	0	2	0	10	1	0	2	0	1	2	0	1	748	21	0
4	4	1	1	1	6	4	0	1	6	12	8	0	15	1067	0
3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	360

Figura 14: Matrice di confusione Dataset originale, 500, MSER, KNN

165	0	0	1	0	0	0	0	0	0	2	0	0	0	1	4
0	122	1	0	0	0	0	0	0	0	5	0	0	0	3	0
0	3	147	2	1	0	0	0	0	0	18	0	0	0	6	0
0	0	1	109	1	0	0	0	0	0	0	0	0	0	6	0
0	0	0	0	169	1	0	0	0	0	1	0	0	0	3	0
0	0	0	0	0	206	0	0	0	0	11	0	0	0	0	0
0	0	0	0	0	0	181	0	0	0	4	0	0	0	16	0
0	9	0	0	0	0	0	174	0	0	0	0	0	0	7	0
0	0	0	0	0	0	0	0	223	9	2	0	0	0	1	0
1	0	0	1	1	0	0	0	1	704	14	12	1	1	15	0
0	0	0	2	3	0	3	2	6	9	618	11	0	0	4	0
0	0	0	0	0	0	0	0	0	0	0	404	5	0	0	0
1	2	0	0	0	0	0	0	0	1	0	8	427	1	10	6
0	1	0	0	4	7	1	3	1	1	1	0	1	723	42	3
4	0	0	1	1	3	2	2	7	30	2	10	1	11	1053	3
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	363

Figura 15: Matrice di confusione Dataset originale, 500, SURF Grid, KNN

57	0	0	1	1	3	1	0	1	0	8	0	1	1	4	3
1	66	5	2	0	1	2	0	0	0	1	0	0	2	6	1
0	0	40	0	0	0	1	1	0	1	2	0	1	0	0	0
0	0	2	70	3	0	2	0	0	0	3	1	0	0	5	0
0	1	0	0	175	0	0	0	0	0	4	0	0	0	3	0
0	0	0	0	1	90	1	0	0	0	9	0	0	0	2	0
0	1	0	0	0	1	87	0	0	0	2	0	0	0	5	0
0	1	0	0	2	1	0	73	0	0	1	0	0	1	1	0
0	0	0	0	0	0	0	0	74	1	1	0	0	0	1	0
0	0	1	0	0	1	2	0	1	357	14	48	16	4	12	9
2	5	0	3	2	2	1	1	5	7	258	4	2	1	9	0
0	0	1	0	0	0	0	0	0	3	0	252	4	3	5	2
0	0	0	0	0	0	0	0	0	1	0	2	147	0	0	0
1	1	2	3	3	1	0	1	0	1	15	0	2	286	18	2
1	0	1	2	6	4	13	5	1	2	5	3	3	9	492	6
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	184

Figura 16: Matrice di confusione Dataset ridotto, 100, MSER, KNN

56	0	0	1	0	4	3	2	0	0	2	0	0	6	4	3
2	66	1	2	4	3	1	3	0	0	2	0	0	0	3	0
0	0	41	0	0	0	0	0	0	0	5	0	0	0	0	0
0	0	0	79	2	0	0	1	0	0	1	0	0	0	3	0
0	0	1	0	175	0	0	0	0	0	3	0	0	0	4	0
0	0	0	0	2	86	2	0	0	0	9	0	0	2	2	0
0	0	0	0	1	4	79	1	1	0	4	0	1	0	4	1
0	0	0	0	0	1	0	78	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	76	0	0	0	0	0	1	0
1	3	0	0	2	1	1	1	1	395	11	11	2	6	29	1
0	2	1	2	1	1	0	4	3	2	273	8	0	2	3	0
0	1	0	0	0	0	0	0	0	1	2	262	1	0	3	0
0	0	0	0	0	0	0	0	0	0	0	3	147	0	0	0
4	1	1	2	1	10	3	2	0	0	6	0	2	293	11	0
2	6	2	1	6	15	31	4	2	1	6	2	1	8	463	3
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	184

Figura 17: Matrice di confusione Dataset ridotto, 500, MSER, KNN

69	0	0	0	1	0	0	0	0	0	4	0	0	0	2	5
0	71	1	4	0	0	3	1	0	0	4	0	0	0	3	0
0	0	39	0	0	0	1	0	0	0	5	0	0	0	1	0
0	0	0	77	0	2	1	0	0	0	0	0	0	0	6	0
0	0	0	0	178	0	0	0	0	0	3	0	0	0	2	0
0	0	0	0	0	91	0	0	0	0	12	0	0	0	0	0
0	0	0	0	0	0	93	0	0	0	3	0	0	0	0	0
0	0	0	0	1	0	0	77	0	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	75	0	0	0	0	0	2	0
0	0	0	1	0	0	1	0	0	421	6	21	4	2	7	2
0	2	1	2	0	1	0	4	4	3	278	5	0	1	1	0
0	0	1	0	0	0	0	0	0	7	1	257	1	1	2	0
0	0	0	0	0	0	0	0	0	0	0	2	148	0	0	0
0	0	3	3	1	1	0	2	0	0	3	1	1	316	4	1
0	3	0	1	1	2	29	1	0	1	2	3	0	4	497	9
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	186

Figura 18: Matrice di confusione Dataset ridotto, 500, SURF detector, KNN

29	3	2	8	2	7	9	1	1	2	7	0	0	1	5	4
3	26	4	0	10	11	12	1	0	1	5	0	1	1	10	2
0	1	27	3	2	1	6	0	0	0	4	0	0	0	2	0
2	1	3	42	5	8	7	2	1	0	6	1	0	5	3	0
1	0	0	5	136	5	5	1	0	5	6	0	1	3	11	4
2	0	2	6	0	69	8	1	1	1	8	0	0	1	4	0
3	0	1	1	3	4	69	1	1	1	3	1	0	1	6	1
1	1	1	4	4	4	15	40	0	0	3	2	0	0	3	2
0	0	0	1	0	0	2	2	68	0	2	0	0	1	0	1
4	5	4	3	12	5	4	4	2	283	28	15	33	11	37	15
1	4	1	3	7	6	2	1	5	8	220	11	3	9	17	4
0	1	0	1	1	1	0	0	0	7	7	229	10	3	8	2
0	0	0	0	0	0	1	0	0	2	0	4	138	2	2	1
5	6	5	15	15	18	12	7	2	14	31	5	5	163	26	7
2	7	2	5	19	11	34	9	6	13	25	2	4	11	392	11
2	1	1	0	5	0	1	0	0	2	2	0	3	0	7	162

Figura 19: Matrice di confusione Dataset ridotto, 500, SURF Grid, KNN

Nelle tabelle 2 e 3 riportiamo i valori di Accuracy, Precision e Recall per gli esperimenti eseguiti con il dizionario impostato a 500 sul dataset originale, confrontando i tre metodi di estrazione delle feature. Abbiamo analizzato questi modelli più nel dettaglio in quanto da noi ritenuti più soddisfacenti degli altri modelli con parametri impostati diversamente.

Features (Vocab 500)	**Accuracy**	**Recall**	**Precision**
SURF Detection	0.9537	0.9581	0.9488
SURF Grid	0.9379	0.9442	0.9363
MSER	0.9456	0.9505	0.9451

Tabella 2: Accuracy, Precision e Recall per il modello KNN con i tre tipi di estrazione di feature a confronto

Features (Vocab 500)	**Accuracy**	**Recall**	**Precision**
SURF Detection	0.9035	0.9645	0.9645
SURF Grid	0.878	0.8782	0.8773
MSER	0.9243	0.9608	0.9604

Tabella 3: Accuracy, Precision e Recall per il modello SVM con i tre tipi di estrazione di feature a confronto

Per analizzare ed aiutarci a comprendere i *failur cases*, abbiamo visualizzato la posizione spaziale di ciascuna immagine all'interno del Supermarket. Per fare ciò abbiamo stampato dei pallini colorati, uno per ciascuna immagine, in base alle coordinate spaziali associate ad esse. Possiamo osservare ciò nelle seguenti immagini:

- In figura 20, ciascuna delle sedici classi viene rappresentata con un colore diverso, tenendo conto delle etichette corrette originali.
- In figura 21, ciascuna delle sedici classi viene rappresentata con un colore diverso in base alle predizioni effettuate dal nostro modello.
- In figura 22, ciascuna delle sedici classi viene rappresentata con un colore diverso, tenendo conto delle etichette corrette originali, con in aggiunta le predizioni errate marcate con dei pallini rossi.

- In figura 23, osserviamo le classi suddivise per colore, con in aggiunta le predizioni errate stampate con il colore della classe predetta. Questa immagine in particolare ci ha permesso di analizzare le principali classi in cui la classificazione fallisce e le motivazioni. Le classi 11 e 15, le quali rappresentano le principali zone di transito tra un reparto e l'altro del SuperMarket (le due zone sel SuperMarket che si sviluppano in orizzonatale ed intersecano tutti gli altri reparti), soffrono la presenza di molti errori proprio perché semanticamente risultano molto vicine a tutte le altre classi. Viceversa vale per le altre classi: la maggior parte di errori corrispondono ad una classificazione errata in cui vengono riconosciute le classi 11 e 15.

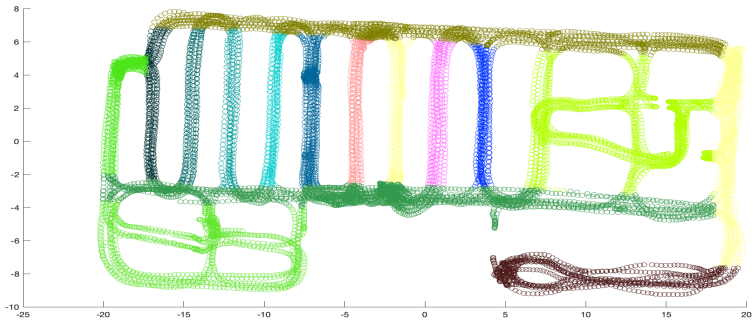

Figura 20: Classi suddivise per colore, etichette corrette

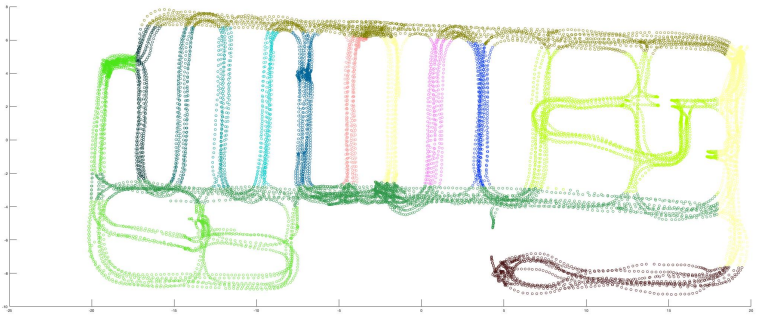

Figura 21: Predizioni del nostro modello, classi suddivise per colore

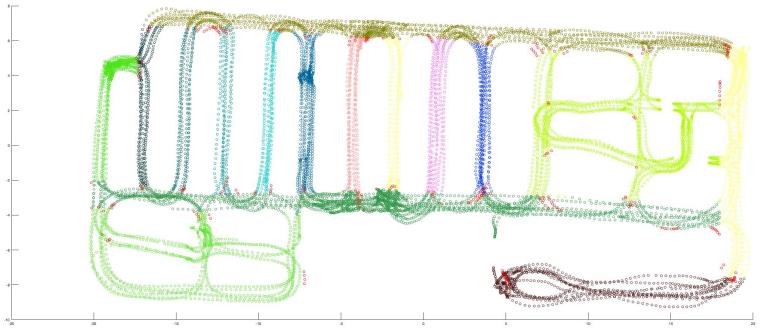

Figura 22: Classi suddivise per colore, errori del nostro modello in aggiunta

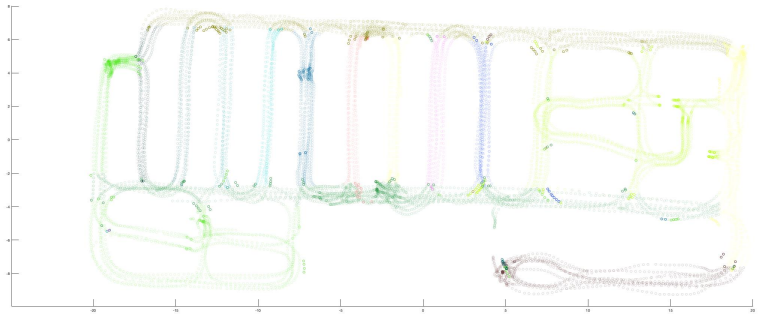

Figura 23: Classi con errori in evidenza

5 Conclusioni

Abbiamo realizzato un progetto che tratta il problema di *indoor localization*. Abbiamo utilizzato tutti gli strumenti forniti dall'Università di Catania ed i concetti appresi dal corso di Computer Vision dell'Università di Cagliari, tenuto dal professor Giovanni Puglisi. Siamo giunti alla conclusione che il miglior modello, tra quelli da noi realizzati, è il seguente (come si può dimostrare dai risultati presentati nella sezione precedente): dataset originale, Vocabulary Size 500, feature estratte con SURF Detector, classificatore KNN, ottenendno il 95.37% di Accuracy. Siamo consapevoli che uno studio più approfondito dei concetti e delle strategie di Computer Vision e Machine Learning potrebbe portare a dei risultati ancora più soddisfacenti ed accurati, o meglio ancora, ottimali. Un possibile sviluppo futuro, infatti, potrebbe essere l'analisi delle immagini in dettaglio in termini di feature, ovvero cercare di capire quali feature in particolare permettano il verificarsi di tali errori, tramite l'aiuto di un feature match più dettagliato tra le immagini.

Riferimenti bibliografici

[1] E. Spera, A. Furnari, S. Battiato, and G. M. Farinella, "Egocart: A benchmark dataset for large-scale indoor image-based localization in retail stores." `https://iplab.dmi.unict.it/EgocentricShoppingCartLocalization/`.

[2] "Machine learning challenge 2018." `https://iplab.dmi.unict.it/MLC2018/`.

[3] G. Puglisi, "Materiale didattico corso di computer vision, università di cagliari, dipartimento di matematica e informatica."

[4] Wikipedia, "Speeded up robust feature." `https://it.wikipedia.org/wiki/Speeded_Up_Robust_Feature`.

[5] Wikipedia, "Maximally stable extremal regions." `https://it.wikipedia.org/wiki/Maximally_stable_extremal_regions`.

[6] Wikipedia, "K-means." `https://it.wikipedia.org/wiki/K-means`.

[7] Wikipedia, "Macchine a vettori di supporto." https://it.wikipedia.org/wiki/Macchine_a_vettori_di_supporto.

[8] Wikipedia, "K-nearest neighbors." https://it.wikipedia.org/wiki/K-nearest_neighbors.

[9] MathWorks, "Matlab." https://it.mathworks.com/.

[10] MathWorks, "fitcknn." https://it.mathworks.com/help/stats/fitcknn.html#bt6cr9l-4.