UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Ingegneria Elettronica e delle Telecomunicazioni

Elaborato finale

Analisi delle prestazioni di diverse kernel nel CBIR basato su SVM

Supervisori
Bruzzone Lorenzo
Begüm Demir

Laureando
Quiri Francesco
154746

Anno accademico 2014/2015

Indice

Elenco delle figure

Capitolo 1

Introduzione generale

Negli ultimi anni con lo sviluppo delle tecnologie satellitari sono state rese disponibili una grandissima quantità di immagini telerilevate, quindi il recupero delle immagini dai grandi database in base ai bisogni dei fruitori di questi dati, si è rivelato uno degli hot topic della comunità del telerilevamento.

Per far fronte a questi nuovi bisogni sono stati introdotti i sistemi CBIR, dall'acronimo inglese di Content-Based Image Retrieval system, che sono un'insieme di tecniche finalizzate alla ricerca e al recupero di immagini digitali, basate sull'utilizzo degli attributi visuali dell'immagine stessa, piuttosto che dei classici metadata. I sistemi CBIR si suddividono in due moduli, il primo è finalizzato all'estrazione delle features dalle immagini, mentre il secondo si concentra sulla ricerca e sul recupero delle immagini maggiormente simili alla query image. Nel particolare questo lavoro si focalizza su un approccio basato sulla Bag-of-visual-words e sulle Support Vector Machine (SVM), col fine ultimo di studiare il comportamento di SVM in funzione dei quattro seguenti parametri:

1. Variazione di funzioni kernel, che sono HIK, generalized HIK e χ^2 kernel;
2. Diversa lunghezza dei vettori di features;
3. Diversa grandezza del training-set;
4. Numero di immagini recuperate.

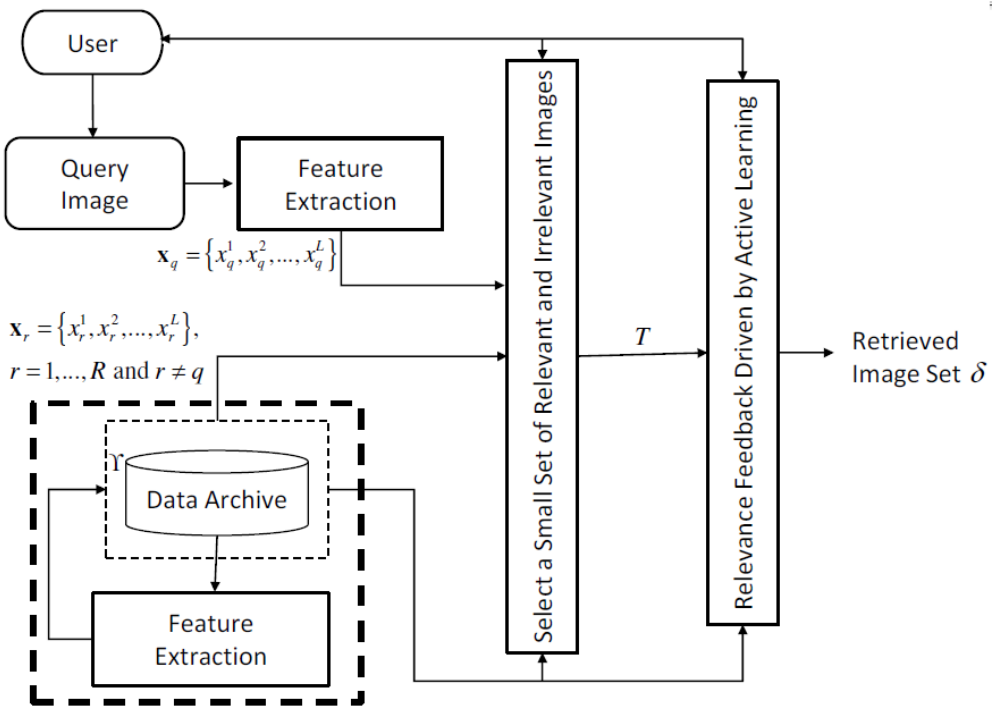
Il presente scritto si divide in tre parti principali, nella prima parte ci si occupa della preparazione del database, più precisamente sul metodo utilizzato per l'estrazione degli attributi delle immagini, creando una Bag-Of-Visual-Words contenente le informazioni raccolte. Poi si descrive il metodo SVM introducendo le funzioni kernel utilizzate durante i test, mentre nella parte finale vengono illustrati e commentati i risultati ottenuti.

Capitolo 2

Content-Based Images Retrieval

2.1 Estrazione delle features e BOVW

Come già menzionato i sistemi CBIR si suddividono sostanzialmente in due parti [1], una dedicata alla preparazione delle immagini e quindi all'estrazione delle features dalle stesse, la seconda si concentra sul recupero delle immagini desiderate dal database.



Lo scopo ultimo dell'estrazione delle features è creare la BOVW, cioè la bag-of-visual-words, ovvero il database delle immagini viene riscritto sottoforma di istogrammi associati ai vettori SIFT features[2, 3]. Si ricorda che l'algoritmo SIFT estrae il contenuto informativo dell'immagine, attraverso la raccolta di descrittori robusti al cambiamento di scala, di traslazione e di rotazione. Grazie a queste caratteristiche le SIFT features sono ritenute migliori confronto ad altre features locali [4] e si sono guadagnate anche una certa visibilità nel contesto delle immagini telerilevate, per i buoni risultati ottenuti[5]. Prima di tutto per creare la BOVW si procede con l'estrazione delle SIFT features. Questo processo avviene in due fasi, nel primo step vengono localizzati i punti riconoscibili anche da prospettive diverse, dopodiché vengono allegati dei descrittori SIFT ai punti selzionati, per un maggiore approfondimento si rimanda a [6]. Una volta terminato questo processo le features raccolte vengono clusterizzate attraverso l'algoritmo k-means e infine ad ogni immagine si associa un istogramma, il quale rappresenta il numero di volte che una determinata SIFT features si presenta nell'immagine analizzata. L'insieme di tutti questi istogrammi forma la BOVW, la quale rappresenta lo stato dell'arte in molti problemi di image retrieving.

2.2 Recupero immagini

La ricerca delle immagini avviene per mezzo di SVM [7], che viene utilizzata per eseguire una classificazione automatica delle immagini presenti nel database e quindi per il recupero di quest'ultime in base ai bisogni del fruitore del database. Si ricorda che la classificazione è un processo per mezzo del quale le immagini vengono assegnate a una categoria piuttosto di un'altra da un algoritmo di apprendimento automatico. Si è optata per la tecnica SVM perché riesce a risolvere molto bene anche problemi di classificazione non lineari.

2.2.1 Support Vector Machine

SVM è una tecnica di apprendimento supervisionato che estrapola un modello da un training-set noto. La classificazione eseguita da SVM ha lo scopo di separare gli esempi di input, massimizzando il margine tra le due categorie mappate nell'iperpiano delle features. Di seguito si osserva una raffigurazione dell'iperpiano ottimo, ovvero quello maggiormente distante dai sample delle classi di input:

Per costruire un classificatore SVM bisogna minimizzare la norma del vettore dei pesi w, obbedendo al vincolo che ogni elemento di training risieda dalla parte "giusta" della superficie di separazione, ovvero dalla parte della classe di appartenenza. Quindi dato un training-set formato da $\{(xi, yi)\}_{i=1}^{m}$, in cui $x_i \in X \subseteq \Re^n$ è il vettore di input e $y_i \in$ {-1,1} è la classe, irrilevante o rilevante, di appartenenza del vettore associato. L'equazione finale risulta essere:

$$y_i(w \cdot x_i + b) \geq 1$$

Tutti gli elementi di training che soddisfano il vincolo di cui sopra vengono chiamati support vector, i quali definiscono l'iperpiano di separazione. Nel caso di due classi non perfettamente separabili bisogna introdurre il cosiddetto soft margin, ovvero nel processo di ottimizzazione si introduce una variabile che tiene conto degli elementi "ambigui ".

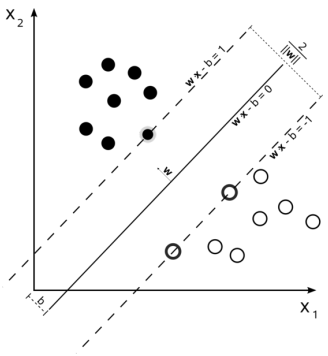


Figura 2.1: Rappresentazione dell'iperpiano ottimo

Quindi si risolve il problema:

$$min_{w,b,\xi} \frac{1}{2}||w||^2 + C \cdot \sum_{i=1}^{m} \xi_i$$

il cui vincolo associato è

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \qquad \forall i = 1, ..., m$$

$$\xi_i \geq 0 \qquad \forall i = 1, ..., m$$

La scelta del parametro C viene chiamata selezione del modello, tale scelta è una decisione delicata, in quanto ha effetto direttamente sull'errore di predizione dell'algoritmo di classificazione stesso, infatti il parametro rappresenta la traduzione matematica del compromesso tra la massimizzazione del margine e la classificazione del training-set senza errori [8].

Il problema di ottimizzazione si riconduce alla risoluzione di:

$$max_\alpha = \sum_{i=1}^{m} \alpha - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j x_i x_j$$

però si ricorda che le features vengono mappate nell'iperspazio, quindi si introduce la funzione di mapping Φ. Perciò il nuovo problema si traduce come:

$$max_\alpha = \sum_{i=1}^{m} \alpha - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\Phi(x_i) \cdot (\Phi(x_j))$$

con vincoli $0 \leq \alpha_i \leq C$ e $\sum_{i=1}^{m} \alpha_i \cdot y_i = 0 \qquad \forall i = 1, ..., m$. Si nota che α_i rappresenta il moltiplicatore di lagrange per l'i-esimo vincolo. Dato che i samples di training mappati sono relazionati da un prodotto, si può introdurre la funzione kernel, t.c.:

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

Questo passaggio viene chiamato kernel trick e permette di definire il metodo SVM anche per problemi non lineari.

Dopo la risoluzione del problema duale si ottiene $w = \sum_{i=1}^{m} y_i \alpha_i \Phi(x_i)$ e la funzione di decisione è nella forma

$$\hat{y} = f(x) = sgn(\sum_{i=1}^{m} y_i \alpha_i k(x_i, x) + b)$$

Dopo il processo di training SVM è pronta per la classificazione vera e propria. Per approfondimenti si rimanda a [9, 10, 11]. Di conseguenza si può intuire che le caratteristiche del training-set e il metodo di scelta del medesimo hanno un impatto significativo sulle performance dell'intero processo di classificazione, non solo, pure la scelta di una funzione kernel piuttosto di un'altra ha un effetto decisivo sui risultati finali. Infatti solitamente nei problemi di image retriving la funzione kernel più utilizzata è RBF, ma recentemente si è dimostrato che per features raccolte in istogrammi (BOVW) è meglio utilizzare funzioni diverse, ad esempio la funzione HIK[12].

Nel presente lavoro si sono sperimentate le seguenti funzioni kernel:

1. **Chi-square kernel**

   La chi-square kernel deriva dalla teoria della distribuzione Chi-square, la versione utilizzata è:

   $$k(x, y) = \sum_{i=1}^{n} 2 \frac{x_i \cdot y_i}{(x_i + y_i)}$$

2. **Histogram Intersection Kernel**

   La Histogram intersection kernel [13], conosciuta anche come min kernel è scritta come:

   $$k(x, y) = \sum_{i=1}^{n} min(x_i, y_i)$$

3. **Generalized Histogram Intersection Kernel**

   Si utilizza pure Generalized HIK introdotta in [14], che è una kernel basata su HIK:

   $$k(x, y) = \sum_{i=1}^{n} min(|x_i|^{\alpha}, |y_i|^{\alpha})$$

Capitolo 3

Esperimenti

3.1 Strumenti utilizzati

Prima di illustrare i risultati raccolti si menziona che durante gli esperimenti si è utilizzato del codice esterno, per quanto riguarda la prima parte, quindi per l'estrazione delle features, si è usato l'algoritmo SIFT proveniente da [15]. Invece per la sezione del recupero delle immagini, quindi per tutta la parte riguardante SVM, si è optato per la libreria LIBSVM [16], perchè permette di usare funzioni kernel personalizzate o comunque diverse da quelle standard. Per la selezione degli iper-parametri di SVM si è utilizzato la tecnica della cross-validation. L'accuratezza della classificazione è stata studiata variando i seguenti parametri:

1. Le funzioni kernel, descritte nel capitolo precedente.
2. La lunghezza delle features, che assume i seguenti valori: 150, 250, 350, 450, 550, 650, 750.
3. Il numero di immagini recuperate dal dataset sono pari a 10, 15, 20.
4. La lunghezza del training-set, indicato come numero di immagini selezionate per categoria e sono: 5(solo nel caso con il numero di immagini recuperate pari a 20), 10, 15, 20.

Il data-set usato è formato da immagini suddivise in 21 categorie e sono tutte ortofotografie provenienti da varie zone degli USA, concesse dal USGS National Map [17]. Le immagini hanno una risoluzione spaziale di 30 cm e una dimensione di 256 x 256 pixel ed ognuna di esse è descritta nello spazio dei colori RGB. Per renderle utilizzabili dall'algoritmo di estrazione delle features si è utilizzata la seguente trasformazione in scala di grigi [12, 5]:

$$0{,}299 \cdot R + 0{,}587 \cdot G + 0{,}144 \cdot B$$

Infine si ricorda che il set disponibile è formato da 2100 immagini suddivise su 21 categorie, mentre durate gli esperimenti si sono utilizzate 1050 immagini sempre suddivise su tutte le categorie disponibili. Nel seguito sono elencate tutte le categorie seguite da alcuni esempi.

Figura 3.1: Alcuni esempi per categoria

3.2 Raccolta dati

Tutti i dati collezionati sono stati raccolti sotto forma di istogrammi per un totale di 163 grafici. L'asse delle ordinate di tutti i grafici rappresentano l'accuratezza raggiunta in funzione dei parametri, mentre l'ascissa varia di caso in caso in base al parametro che si vuole evidenziare. Nel seguito saranno visualizzati dei grafici di esempio.

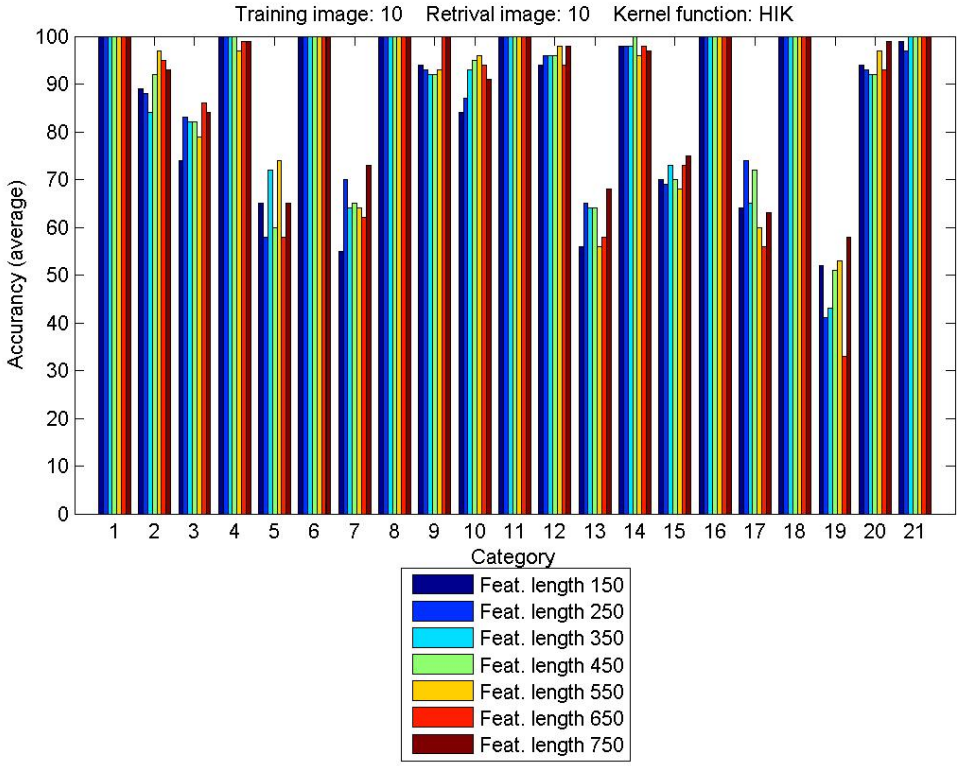


Figura 3.2: 1.agricultural, 2.airplane, 3.baseballdiamond, 4.beach, 5.buildings, 6.chaparral, 7.denseresidential, 8.forest, 9.freeway, 10.golfcourse, 11.harbor, 12.intersection, 13.mediumresidential, 14.mobilehomepark, 15.overpass, 16.parkinglot, 17.river, 18.runway, 19.sparseresidential, 20.storagetanks, 21.tenniscourt

Il grafico appena proposto mostra come varia l'accuratezza finale in funzione della lunghezza delle features ed è chiaro che in media il risultato finale migliora all'aumentare del parametro evidenziato. Dai dati traspare anche che sia HIK e sia χ^2 hanno un rendimento molto simile, mentre, da come si può osservare dal grafico sottostante, generalized HIK ha un comportamento molto instabile al variare della lunghezza delle features.

Si evidenzia che le classi che raggiungono i valori minori utilizzando le altre funzioni kernel, peggiorano radicalmente utilizzando generalized HIK, questo comportamento è causato dalla difficoltà di riconoscere gli elementi di queste categorie rispetto alle altre,

mentre si osserva che le classi più semplici ottengono valori alti, anche con la lunghezza di features settata al minimo. Questo è un risultato importante, perchè ci suggerisce che per le classi meno impegnative si può ottenere la stessa accuratezza settando al minimo i parametri studiati, diminuendo così l'onere computazionale necessario.

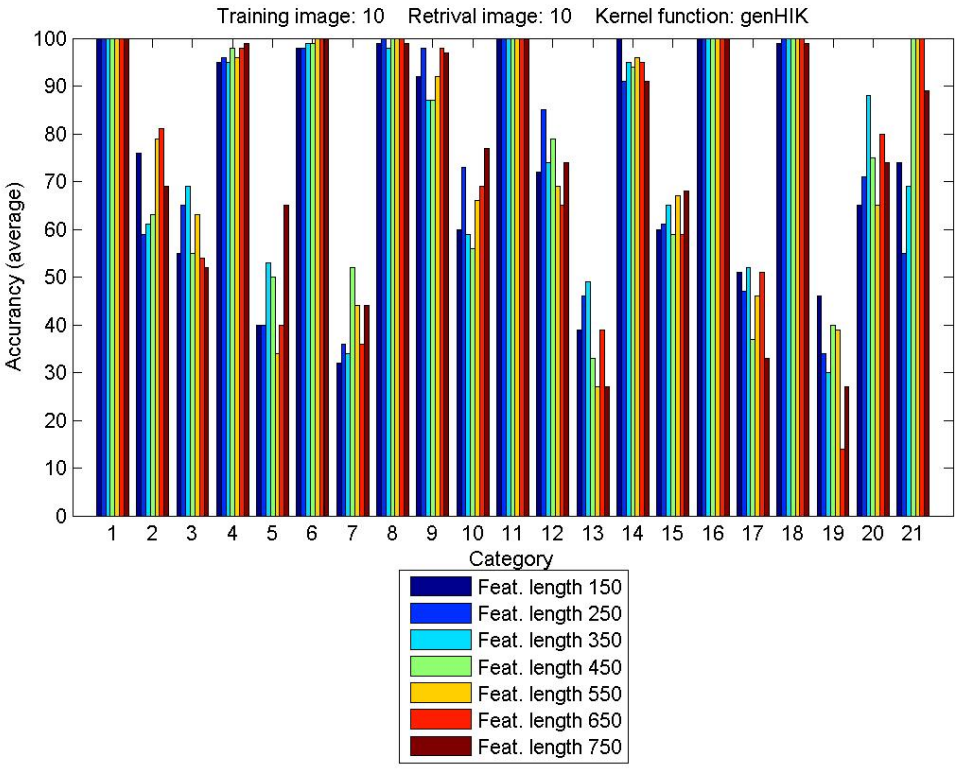


Figura 3.3: 1.agricultural, 2.airplane, 3.baseballdiamond, 4.beach, 5.buildings, 6.chaparral, 7.denseresidential, 8.forest, 9.freeway, 10.golfcourse, 11.harbor, 12.intersection, 13.mediumresidential, 14.mobilehomepark, 15.overpass, 16.parkinglot, 17.river, 18.runway, 19.sparseresidential, 20.storagetanks, 21.tenniscourt

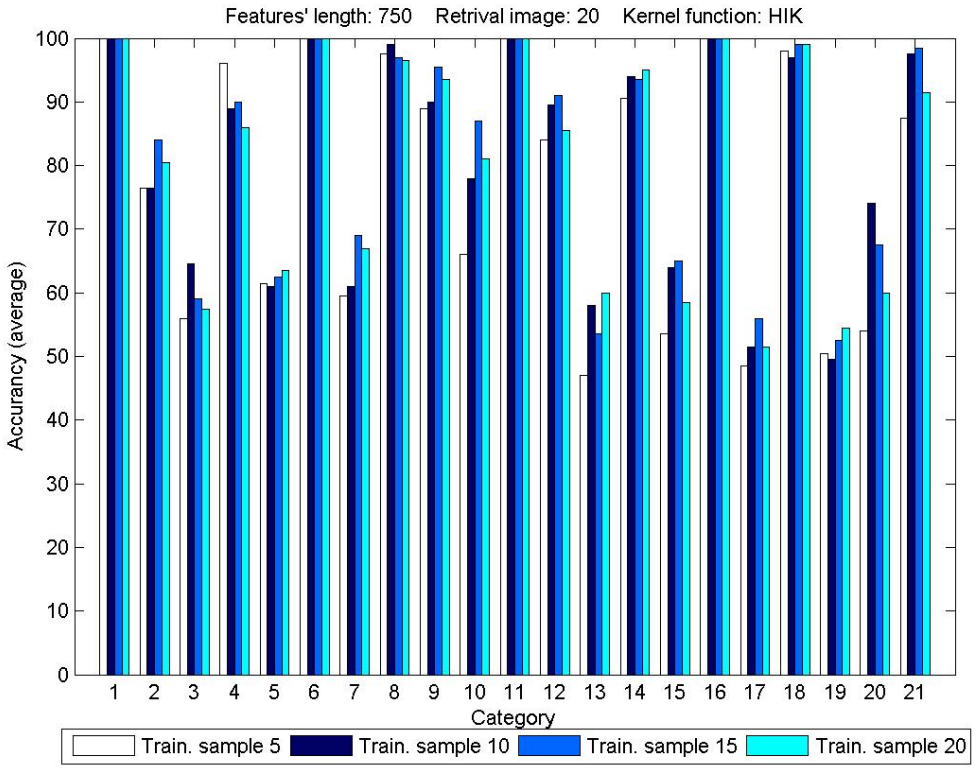


Figura 3.4: 1.agricultural, 2.airplane, 3.baseballdiamond, 4.beach, 5.buildings, 6.chaparral, 7.denseresidential, 8.forest, 9.freeway, 10.golfcourse, 11.harbor, 12.intersection, 13.mediumresidential, 14.mobilehomepark, 15.overpass, 16.parkinglot, 17.river, 18.runway, 19.sparseresidential, 20.storagetanks, 21.tenniscourt

In questa tavola si osserva come varia l'accuratezza di predizione in funzione della larghezza del training set e anche in questo caso, come nei grafici precedenti, emerge la differenza tra le classi semplici da riconoscere e quelle più impegnative. Dallo studio dei grafici di questa tipologia, si nota che in media i risultati migliori si ottengono regolando la grandezza del training set a 10 o a 15 immagini per categoria, però è palese che per le classi più semplici da classificare bastano anche 5 immagini per categoria per ottenere comunque ottimi risultati. Inoltre, guardando il grafico sottostante, si evidenzia che generalized HIK ottiene risultati molto variabili, totalizzando anche in questo caso i risultati peggiori.

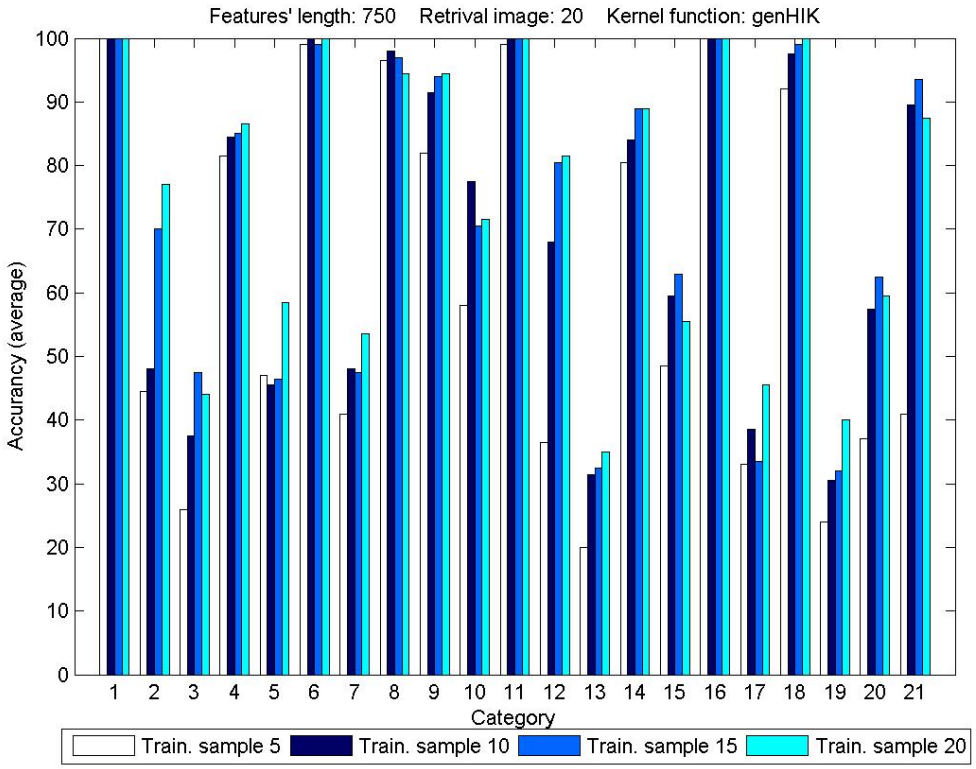


Figura 3.5: 1.agricultural, 2.airplane, 3.baseballdiamond, 4.beach, 5.buildings, 6.chaparral, 7.denseresidential, 8.forest, 9.freeway, 10.golfcourse, 11.harbor, 12.intersection, 13.mediumresidential, 14.mobilehomepark, 15.overpass, 16.parkinglot, 17.river, 18.runway, 19.sparseresidential, 20.storagetanks, 21.tenniscourt

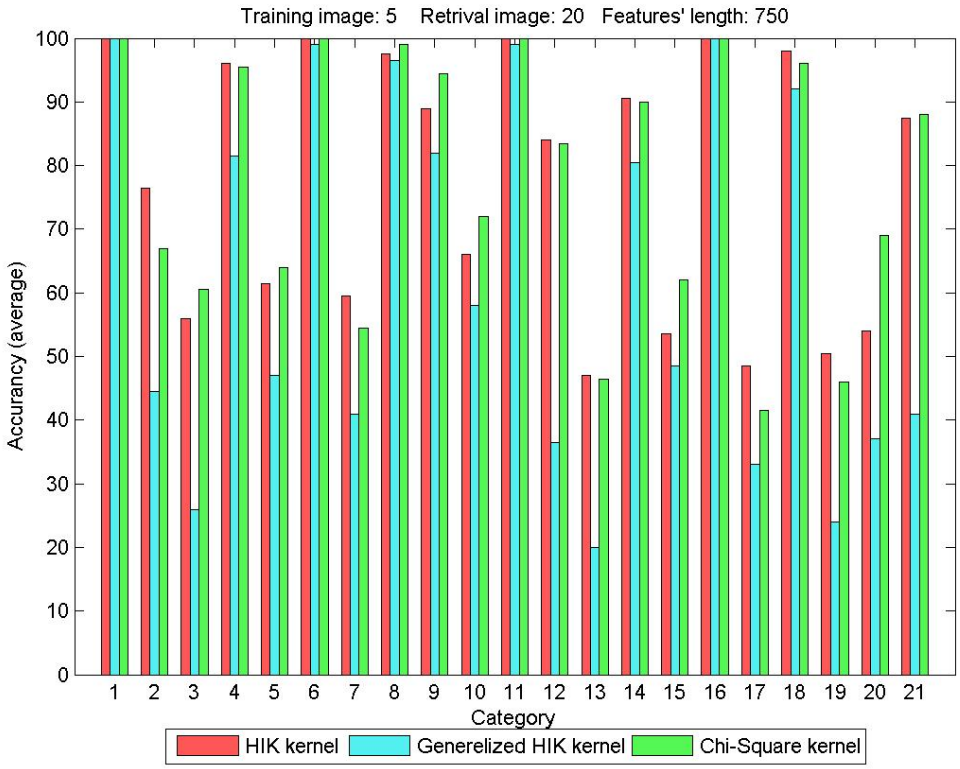


Figura 3.6: 1.agricultural, 2.airplane, 3.baseballdiamond, 4.beach, 5.buildings, 6.chaparral, 7.denseresidential, 8.forest, 9.freeway, 10.golfcourse, 11.harbor, 12.intersection, 13.mediumresidential, 14.mobilehomepark, 15.overpass, 16.parkinglot, 17.river, 18.runway, 19.sparseresidential, 20.storagetanks, 21.tenniscourt

In quest'ultimo grafico si confrontano le kernel e proprio in quest'ultima tipologia di grafici, si quantifica quanto generalized HIK sia inadatto a processare dati raccolti in BOVW. Infatti questa kernel ottiene risultati uguali, o comunque leggermente inferiori alle altre kernel solo nelle classi più semplici, mentre nelle classi impegnative l'inferiorità è netta. Questo trend è riconoscibile in tutti i dati raccolti, dimostrando così l'inadeguatezza di generalized HIK in questo tipo di compiti.

Capitolo 4

Conclusioni

Dopo aver analizzato tutti i grafici risulta chiaro che generalized HIK raggiunge i risultati peggiori su quasi tutte le categorie o al massimo simili a HIK e a χ^2, che d'altro canto tendono ad avere risultati molto affini al variare dei parametri utilizzati, quindi si può sostenere che la funzione kernel generalized HIK non è adatta a lavorare sulle BOVW. Questo probabilmente è dovuta alla presenza degli iper-parametri presenti in questa funzione. Comunque sia anche in presenza di questi problemi da parte di generalized HIK si sono individuate delle classi più forti rispetto ad altre, cioè quelle che hanno collezionato i risultati migliori e sono agricultural, beach, chaparral, forest, freeway, harbor, mobilehomepark, parkinglot e runway. Di conseguenza per riconoscere gli elementi di queste categorie forti basta settare al minimo tutte le variabili prese in considerazione negli esperimenti, riducendo notevolmente il tempo di individuazione e di recupero di queste immagini. I risultati, oltre che dalla scelta della funzione kernel, sono direttamente influenzati dall'estrazione delle features, quindi una scelta diversa sui parametri di estrazione, potrebbe avere ripercussioni positive sul riconoscimento delle classi impegnative. Inoltre non bisogna dimenticare che anche la scelta del training-set ha un forte impatto sui risultati, quindi lavorando su questi fattori si potrebbero raggiungere buoni risultati anche con le categorie difficili da classificare.

Bibliografia

[1] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008.

[2] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[3] Stephen O'Hara and Bruce A. Draper. Introduction to the bag of features paradigm for image classification and retrieval. *CoRR*, abs/1101.3354, 2011.

[4] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, Oct 2005.

[5] Yi Yang and S. Newsam. Geographic image retrieval using local invariant features. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(2):818–832, Feb 2013.

[6] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[8] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, pages 1–22. Adaptive computation and machine learning. MIT Press, 2002.

[9] Ethem Alpaydin. *Introduction to Machine Learning.* The MIT Press, 2nd edition, 2010.

[10] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, June 1998.

[11] S.R. Gunn. Support vector machines for classification and regression. Technical report, Dept. of Electronics and Computer Science, University of Southampton, 1998. Address: Southampton, U.K.

[12] B. Demir and L. Bruzzone. A Novel Active Learning Method in Relevance Feedback for Content-Based Remote Sensing Image Retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 53:2323–2334, May 2015.

[13] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages III–513–16 vol.2, Sept 2003.

[14] S. Boughorbel, J.-P. Tarel, and Nozha Boujemaa. Generalized histogram intersection kernel for image recognition. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–161–4, Sept 2005.

[15] A. Vedaldi. An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD, 2007.

[16] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[17] Uc merced land use dataset. http://vision.ucmerced.edu/datasets/landuse.html. Accessed: 2015-07-08.