Identificazione del modello di fotocamera

A. De Crecchio, U. Giordano, R. Pane, C. Pulcrano
Progetto ESM - gruppo 12
(a.a. 2019-2020)

1 - Introduzione

L'obiettivo di questo progetto è l'identificazione del modello di fotocamera che ha acquisito una data fotografia.
Tale problema risulta essere di particolare interesse poiché necessario in molteplici aree di applicazione, come ad esempio l'Analisi Forense, la rivendicazione della proprietà intellettuale, la Stegoanalisi o comunque in tutte quelle attività in cui risulta utile risalire all'origine di un'immagine.
Per raggiungere tale obiettivo si è scelto di procedere basandosi sulle tracce distintive, caratteristiche di ogni particolare modello, lasciate dalle elaborazioni interne di ogni fotocamera. [1][3].
Infatti, nelle fotocamere digitali, l'immagine in uscita è ottenuta applicando una serie di sofisticati algoritmi, che vanno quindi a caratterizzare fortemente il modello di fotocamera.
Tali descrittori, nel modello applicato, vengono estratte localmente, sulla base di Matrici di co-occorrenza, e poi utilizzate per formare un Classificatore Lineare.[3]

2 - Approccio implementato

L'approccio che abbiamo utilizzato è quello del "Perfect Knowledge", ispirato al lavoro presentato in [3].
Ovvero si considera un numero finito di device, nel nostro caso 4, di cui si ha una conoscenza completa (Tabella 1). In particolare, per ogni modello, si avrà un numero di immagini di formazione abbastanza grande da poter effettuare stime molto affidabili delle caratteristiche di interesse. Sarà quindi possibile utilizzare un Classificatore Multiclasse e calcolare una Matrice di Confusione Completa [3].
Per calcolare le caratteristiche locali basate sulle co-occorrenze, come proposto in [2][3], si procede con i seguenti passaggi:
- calcolo dei residui attraverso filtraggio passa-alto;
- quantizzazione e troncamento dei residui;
- calcolo dell'istogramma delle co-occorrenze.

Tali passaggi sono stati applicati in due esperimenti base paralleli: una volta sulla sola componente di Luminanza, ed un'altra sulle singole bande dello Spazio RGB.

Figura 1.

Tabella 1.

Proprietario	Cellulare	Sensore	MP	Risoluzione
Alessandro	Samsung S8	Samsung ISOCELL S5K3H1	12MP	4032x3024
Cesare	OnePlus 5T	Sony IMX398 Exmor RS	20MP	4608x3456
Raffaele	Sony Xperia Z5	Sony IMX 300 Exmor RS	23MP	5520x4140
Umberto	Xiaomi Mi 9	Sony IMX586 Exmor RS	48MP	3264x1836

Scendendo nel dettaglio, per entrambi si è utilizzato un filtro Derivata Seconda lungo le due direzioni, verticale (I_v) ed orizzontale (I_h). Per la simmetria del problema la maschera orizzontale sarà proprio la trasposta di quella verticale.
Ad esse è stata poi applicata una quantizzazione ed un troncamento, con la seguente formula:

$$Q_k = min(max(round(I_k) + 2, 0), 4)$$

dove $k \in \{h, v\}$.
A Q_v e Q_h è poi applicato un filtro orizzontale [0,1,5,25,125] che formerà C_v e C_h. Da quest'ultimi si creano i rispettivi Istogrammi, che concatenandosi andranno a formare il Vettore di Feature.
Nel caso RGB gli Istogrammi saranno 6, due per ogni banda di colore.
Ognuno di questi Vettori di Feature, estratto da ogni immagine, sarà utile a creare il cosiddetto Feature Space. Attraverso la Principal Component Analysys si effettua quindi una riduzione dimensionale di tale spazio attraverso la ricerca di un nuovo sistema di riferimento che massimizzi la varianza delle variabili rappresentate lungo gli assi. La varianza totale delle variabili viene così suddivisa in un numero di variabili uguali a quello di partenza, ma il cui numero può essere facilmente ridotto. Un classificatore verrà quindi poi addestrato a settare gli Iperpiani Separatori in tale spazio, così da definire le regioni di decisione per ogni classe.

3 - Dataset

Per addestrare e testare il Classificatore si è creato un Dataset di 800 foto, 200 fotografie per ogni device. Ognuna di queste foto è stata catturata con soggetti ed illuminazioni differenti, ma sempre mantenendo le impostazioni di scatto nella loro forma più trasparente possibile: quindi eliminando qualsiasi tipo di enhancement o zoom.
Nel nostro esperimento si è inoltre aggiunta una pre-fase di cropping per ogni immagine, antecedente all'estrazione di feature, così da ridurre l'elaborazione ad un quadrato centrato di dimensioni 512x512.
Per ogni fotocamera è poi stata effettuata una suddivisione del Dataset: il 75% è stato utilizzato per la fase di Training, il restante 25% per quella di Test.

Come già specificato i vari set subiscono entrambi quindi una fase di cropping e una di estrazione delle Feature. Ciò che li differenzia sono quindi le elaborazioni successive.
Al Dataset di Training si è applicata la PCA, con una riduzione del 50% delle Componenti Principali. In uscita si avranno quindi due matrici: la T dei Coefficienti della PCA per la data Matrice di Feature Space e quella Xtrain_pca che è la rappresentazione del Feature Space nello Spazio delle Componenti Principali calcolate. In aggiunta vi è anche una terza matrice che rappresenta la media stimata di ogni singola variabile nel Feature Space.
Una volta fatto questo si pone la matrice Xtrain_pca ed una di Etichette in ingresso al nostro Classificatore Lineare che, settando gli Iperpiani Separatori, ci restituirà quindi un modello multiclasse completamente addestrato.
Fatto ciò si passa al Dataset di Test. Si moltiplica la Matrice di Coefficienti della PCA per la differenza tra il Feature Space di Test e il vettore delle Medie. In formule:

$$Xtest\_pca = (features\_space\_test - average) * T$$

La matrice ottenuta ed il modello addestrato nella fase di training saranno poi gli strumenti necessari per predire le etichette, e quindi la classe di appartenenza, delle immagini di test.
Tale predizione è stata effettuata attraverso la funzione Matlab *predict*, che utilizza un modello di classificazione probabilistico basato sull'applicazione del Teorema di Bayes, il cosiddetto "naïve Bayes".
Il risultato saranno quindi proprio le associazioni foto-device, che era nostro obiettivo ricercare.

Tabella 2. Risultati ottenuti per i due esperimenti base: Luminanza ed RGB, con Crop a 512.

Feature	Samsung S8	OnePlus 5T	Sony Xperia Z5	Xiaomi Mi 9	Accuracy
Lum_2nd Deriv	100%	94%	100%	100%	98.5%
RGB_2nd Deriv	100%	96%	100%	98%	98.5%

Tabella 3. Esperimenti sulle derivate di ordine superiore al 2.

Feature	Samsung S8	OnePlus 5T	Sony Xperia Z5	Xiaomi Mi 9	Accuracy
Lum_2	100%	94%	100%	100%	98.5%
Lum_3	88%	72%	94%	88%	85.5%
Lum_4	100%	84%	96%	94%	93.5%
Lum_5	94%	82%	94%	98%	92%
Lum_6	88%	92%	92%	98%	92.5%
RGB_2	100%	96%	100%	98%	98.5%
RGB_3	92%	82%	92%	78%	86%
RGB_4	92%	84%	98%	98%	93%
RGB_5	90%	82%	98%	94%	91%
RGB_6	90%	78%	96%	90%	88.5%

4 - Risultati sperimentali

Prima di parlare dei risultati ottenuti, è necessario fare qualche breve considerazione sui device utilizzati. Su 4 telefonini, 3 montano sensori Sony; il restante invece ne monta uno Samsung.
Da notare inoltre che sia il Samsung S8 che il Sony Xperia Z5 utilizzano sensori proprietari, ovvero prodotti dalla medesima azienda che ha progettato gli smartphone. I restanti due invece no.
Come si vedrà, questi aspetti, all'apparenza secondari, possono invece giustificare alcuni risultati ottenuti.
Per definire la precisione del modello abbiamo usato la misura dell'accuratezza come quantitativo percentile delle previsioni esatte.

Partendo dai due esperimenti base, come mostrato in Tabella 2, abbiamo ottenuto lo stesso ottimo punteggio del 98,5% di previsioni esatte, molto in linea con i risultati ottenuti in [3].
Ciò comprova l'affidabilità di tale soluzione, chiaramente riscontrabile anche dalla evidente localizzazione delle varie classi mostrate in Figura 2, che indubbiamente facilita il lavoro al classificatore.
A questo punto si è cercato di testare i limiti di tale soluzione, procedendo verso tre direzioni differenti: riduzione dell'area di crop, compressione delle immagini di dataset, aumento di grado del filtro passa alto derivata.

Iniziando da quest'ultimo, ed ispirati dalle soluzioni proposte in [3] con la derivata terza, abbiamo proceduto alla sperimentazione fino al sesto ordine.
Il motivo che ci ha spinto in questa direzione è sorto nel constatare il ritardo che si crea nelle differenze di grado dispari di segnali discreti. Siamo partiti, quindi, prima nel ricercare una conferma nelle derivate pari di ordine superiore al 2, e poi nella 5^a.
Risulta evidente, dalla Tabella 3, una robustezza generale del metodo, con risultati sempre tutto sommato più che positivi, inficiati solamente da un naturale decadimento al salire dell'ordine. È inoltre visibile come le più modeste performance della derivata terza, rispetto agli ordini adiacenti, siano un caso isolato, dovuto probabilmente alla maggiore sensibilità al ritardo degli ordini più bassi.

Ai fini applicativi risultano probabilmente molto più interessanti gli esperimenti fatti in situazioni di compressione. In un mondo sempre più Informatizzato la compressione è diventata una prassi consolidata e necessaria per alleggerire le dimensioni sempre crescenti dei file. Si potrebbe addirittura osare nel dire che risulta ormai praticamente quasi impossibile, al di fuori di contesti molto specifici, trovare materiali a cui non è stata effettuata nessuna sorta di compressione.

Tabella 4. Nella prima parte della tabella ci sono gli esperimenti con le varie combinazioni di compressione tra i test: both_comp indica che è stato compresso sia il dataset di training che quello di test; train_comp solo quella di training e test_comp solo quella di test. La seconda parte della tabella indica invece i vari esperimenti con Crop più decisi., rispettivamente a 256 ed a 128.

Feature	Samsung S8	OnePlus 5T	Sony Xperia Z5	Xiaomi Mi 9	Accuracy
Lum_both_comp	88%	64%	88%	48%	72%
RGB_both_comp	88%	70%	94%	64%	79%
Lum_test_comp	8%	40%	0%	38%	21.5%
RGB_test_comp	60%	32%	0%	22%	28.5%
Lum_train_comp	2%	64%	92%	2%	40%
RGB_train_comp	2%	58%	88%	4%	38%
Lum256	96%	86%	98%	92%	93%
Lum128	84%	74%	98%	74%	82.5%
RGB256	94%	94%	100%	92%	95%
RGB128	94%	72%	98%	80%	86%

Il test da noi effettuato, ed ispirato da ciò che è stato fatto in [3], è una compressione di tipo JPEG con fattore 0.7.
I risultati, mostrati in Tabella 4, sono estremamente in linea con quelli riscontrati in [3], e mostrano come percentuali accettabili siano solo riscontrabili quando sia il set di training che quello di test sono soggetti alla medesima compressione. Qualsiasi soluzione ibrida inficia totalmente le performance, rendendo il modello inutilizzabile.
Un tale comportamento può essere facilmente spiegato per le caratteristiche basate sulla co-occorrenza, perché analizzano i micro-pattern nel residuo dell'immagine, che sono certamente alterati dalla compressione e dal ricampionamento [3]. Tale alterazione è facilmente visibile nella Figura 2 in cui le modifiche dovute al processo di compressione hanno creato uno spazio più confuso, dove è molto più complesso effettuare una distinzione netta delle classi.

In ultima fase si è testato come reagisse il modello a dei crop più invasivi.
Tale idea è nata dal fatto che tale processo ridurrebbe in maniera significativa il carico di lavoro del calcolatore durante le varie fasi dell'elaborazione. È inoltre giustificata dalla crescente dimensionalità e densità di pixel delle immagini catturate dai device moderni, nonché dalla digitalizzazione di molte delle impostazioni di scatto (un esempio è come sulle macchine fotografiche professionali lo zoom sia spesso analogico, mentre tale soluzione è praticamente impossibile da trovare adottata sugli smartphone).
In particolare quest'ultimi renderebbero, in linea teorica, le immagini scattate da telefonino molto più dense di caratteristiche distintive.
I risultati ottenuti sono molto positivi. Con un crop centrato di dimensioni 256x256 si è avuta una riduzione totale di accuratezza rispettivamente del 5.5% per la luminanza e del 3.5% per l'RGB, rispetto a ciò che si è ottenuto nell'esperimento base. Si resta quindi sempre ampiamente al di sopra del 90% di previsioni esatte, a fronte di una riduzione di 4 volte il numero di pixel da elaborare.
Aumentando ancora si è provato con un quadrato centrato 128x128. In questo caso le prestazioni sono scese di molto, restando tuttavia piuttosto buone, come si vede in Tabella 4.
In questo caso la riduzione è di addirittura 16 volte il numero di pixel rispetto al crop originale.
È interessante notare come i device con sensore proprietario siano particolarmente robusti a crop molto forti, probabilmente a causa di un maggior interlacciamento tra gli algoritmi interni di più basso livello, che ne permette quindi una maggiore caratterizzazione a livello di feature. In particolare, il Sony Xperia Z5 ha mantenuto una percentuale praticamente quasi perfetta di predizioni esatte.

5 – Conclusioni

Tali esperimenti hanno confermato, quindi, gli ottimi risultati del modello proposto in [3], nonché le problematiche relative alla compressione asincrona. Si è inoltre messo in evidenza come le

immagini catturate dagli smartphone di nuova generazione siano particolarmente ricche di feature utili nel processo qui usato, così da permettere il raggiungimento di ottimi risultati anche con crop molto decisi.
Tale possibilità è sicuramente un'opportunità in più sia in termini computazionali, sia in presenza di situazioni in cui è disponibile solo una sezione dell'immagine da elaborare.

Riferimenti bibliografici

[1] M. Kirchner and M. Gloe, "Forensic camera model identification," *Handbook of digital forensics of multimedia data and devices*, pp. 231–259, 2015.
[2] J. Fridrich, and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 868–882, 2012.
[3] F. Marra, G. Poggi, C. Sansone and L. Verdoliva, "A study of co-occurrence based local features for camera model identification," *Multimedia Tools and Applications*, vol.76, pp.4765-4781, Feb. 2017.

Figura 2. Grafici relativi alle classificazioni dei vari modelli. loo e roo rappresentano i due esperimenti base, in cui le classi sono visibilmente discernibili. Come si può notare nella figura lcc la presenza di compressione ha introdotto invece un gran quantitativo di confusione, e quindi di incertezza, tra le varie zone, giustificando I peggiori risultati del modello. Nell'immagine l256 viene invece mostrata la versione con crop a 256: l'incertezza introdotta è minima, quindi le varie classi sono ancora tutte chiaramente discernibili.