Hybrid Clustering of Shared Images on Social Networks for Digital Forensics

Samuele Evangelisti

samuele.evangelisti@studio.unibo.it

Laurea Magistrale in Informatica

a.a. 2019/2020

Introduzione

In ambito digitale, in particolare quando si parla di crimini informatici, le immagini catturate e condivise dagli utenti dei social network sui propri profili possono acquistare importanza. Si possono presentare diversi scenari:

- *Smartphone verification*: si vuole determinare se una precisa immagine sia stata catturata da un preciso smartphone
- *Smartphone identification*: si vuole determinare quale, in un certo insieme di smartphone, abbia catturato una precisa immagine

Occorre avere a disposizione gli smartphone da testare

SPN

A causa di imperfezioni nel processo produttivo le camere degli smartphone presentano il **Sensor Pattern Noise** (SPN) che funge da impronta digitale della camera.

Il SPN può essere calcolato come media dei **Residual Noise** (RN) delle immagini catturate da una stessa camera.

Il RN viene estratto come differenza tra l'immagine originale e l'immagine ripulita dal "rumore di fondo"

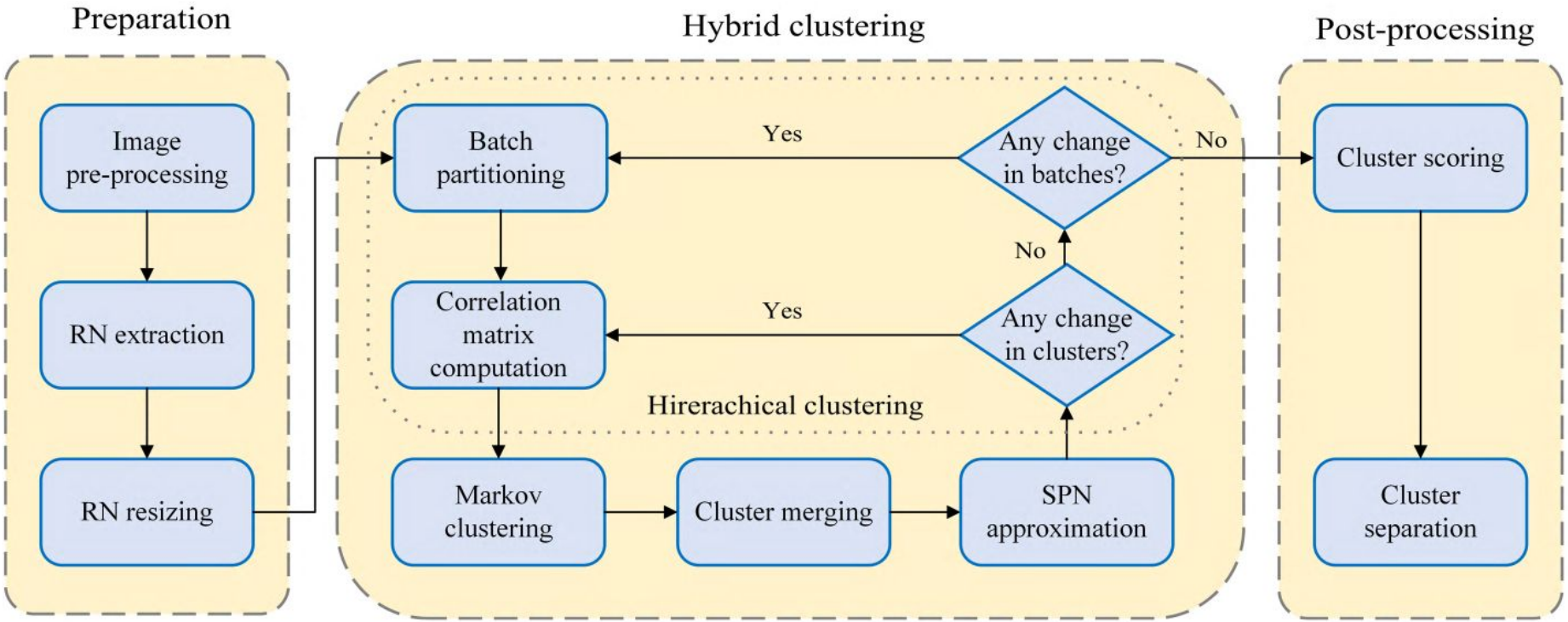
HMC

Si vuole quindi proporre di seguito un algoritmo in grado di fare clustering di immagini basandosi sul SPN.

L'algoritmo in questione tiene conto di diversi aspetti:

- Il numero di smartphone che hanno fisicamente acquisito le immagini è sconosciuto
- La RAM può essere un limite di cui tener conto, rendendo impossibile caricare tutto il dataset contemporaneamente
- Per estrarre il RN può acquisire importanza il corretto orientamento dell'immagine

HMC


Preparation
Image pre-processing
RN extraction
RN resizing
Hybrid clustering
Batch partitioning
Yes
Any change in batches?
No
Correlation matrix computation
Yes
Any change in clusters?
No
Hirerachical clustering
Markov clustering
Cluster merging
SPN approximation
Post-processing
Cluster scoring
Cluster separation

Preparazione

Vengono rimosse le immagini scure o saturate

- Scure: il 70% dei pixel ha intensità minore di 50
- Saturate: il 70% dei pixel ha intensità maggiore di 250

Le immagini vengono orientate tutte allo stesso modo utilizzando i metadati. Se questi non sono presenti, perchè rimossi dal social network sul quale sono caricate, si orienta l'immagine in accordo con il dataset.

Infine le immagini vengono poste in bianco e nero

Preparazione

Nel calcolo del RN e del SPN è chiaro che influiscono fortemente il filtro utilizzato e il numero di RN a disposizione

$$RN = I - d(I)$$

$$SPN = \frac{1}{n}\sum_{j=1}^{n} RN_j$$

Clustering ibrido

Avendo a disposizione una RAM limitata non è possibile caricare tutti i RN e calcolare la **correlation matrix** completa. Risulta quindi necessario partizionare i RN in batch in base alla RAM a disposizione

$$N \text{ (numero di RN)}$$
$$q \text{ (dimensione di un batch)}$$
$$t = \left\lceil \frac{N}{q} \right\rceil \text{ (numero di batch)}$$
$$B = \{b_1, b_2, ..., b_t\} \text{ (batch)}$$

Clustering ibrido

Per ogni batch viene costruita la correlation matrix. Ogni elemento A(i, j) della matrice è la **Normalized Cross Correlation similarity** (NCC) tra i due SPN fi e fj

$$f_i = [x_1, ..., x_l] \text{ (SPN)}$$
$$f_j = [y_1, ..., y_l] \text{ (SPN)}$$

$\overline{f_i}$, $\overline{f_j}$ (medie dei due vettori SPN)

$$\mathcal{A}(f_i, f_j) = \frac{\sum_{n=1}^{l}(x_n - \overline{f_i})(y_n - \overline{f_j})}{\sqrt{\sum_{n=1}^{l}(x_n - \overline{f_i})^2 \sum_{n=1}^{l}(y_n - \overline{f_j})^2}}$$

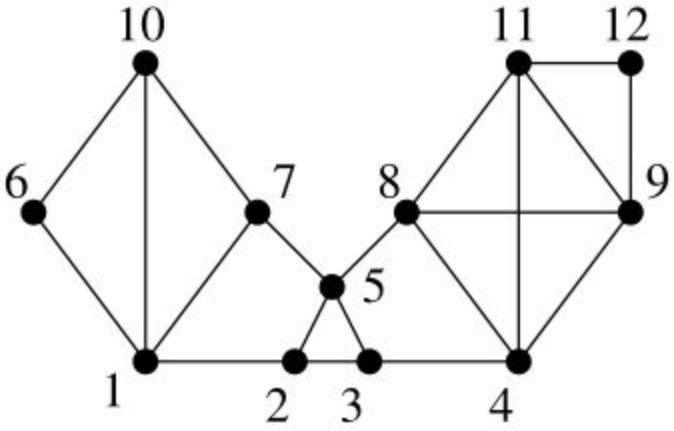
Clustering ibrido

Lo scopo dell'algoritmo è quello di interpretare ogni batch come un grafo sul quale effettuare un cammino aleatorio (Markov).

Al termine dell'algoritmo la matrice di Markov associata al grafo, in stato di convergenza, rappresenta i cluster ottenuti.

Il metodo completo e le basi matematiche sono riportate in [2]

Clustering ibrido - Markov



Clustering ibrido - Markov

$$\begin{pmatrix}
0.200 & 0.250 & -- & -- & -- & 0.333 & 0.250 & -- & -- & 0.250 & -- & -- \\
0.200 & 0.250 & 0.250 & -- & 0.200 & -- & -- & -- & -- & -- & -- & -- \\
-- & 0.250 & 0.250 & 0.200 & 0.200 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & 0.250 & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & -- \\
-- & 0.250 & 0.250 & -- & 0.200 & -- & 0.250 & 0.200 & -- & -- & -- & -- \\
0.200 & -- & -- & -- & -- & 0.333 & -- & -- & -- & 0.250 & -- & -- \\
0.200 & -- & -- & -- & 0.200 & -- & 0.250 & -- & -- & 0.250 & -- & -- \\
-- & -- & -- & 0.200 & 0.200 & -- & -- & 0.200 & 0.200 & -- & 0.200 & -- \\
-- & -- & -- & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & 0.333 \\
0.200 & -- & -- & -- & -- & 0.333 & 0.250 & -- & -- & 0.250 & -- & -- \\
-- & -- & -- & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & 0.333 \\
-- & -- & -- & -- & -- & -- & -- & -- & 0.200 & -- & 0.200 & 0.333
\end{pmatrix}$$

Clustering ibrido - Markov

$$
\begin{pmatrix}
0.380 & 0.087 & 0.027 & -- & 0.077 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\
0.047 & 0.347 & 0.210 & 0.017 & 0.150 & 0.019 & 0.066 & 0.011 & -- & 0.012 & -- & -- \\
0.014 & 0.210 & 0.347 & 0.055 & 0.150 & -- & 0.016 & 0.046 & 0.009 & -- & 0.009 & -- \\
-- & 0.027 & 0.087 & 0.302 & 0.062 & -- & -- & 0.184 & 0.143 & -- & 0.143 & 0.083 \\
0.058 & 0.210 & 0.210 & 0.055 & 0.406 & -- & 0.083 & 0.046 & 0.009 & 0.019 & 0.009 & -- \\
0.142 & 0.017 & -- & -- & -- & 0.295 & 0.083 & -- & -- & 0.184 & -- & -- \\
0.113 & 0.069 & 0.017 & -- & 0.062 & 0.097 & 0.333 & 0.011 & -- & 0.147 & -- & -- \\
-- & 0.017 & 0.069 & 0.175 & 0.049 & -- & 0.016 & 0.287 & 0.143 & -- & 0.143 & 0.083 \\
-- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\
0.246 & 0.017 & -- & -- & 0.019 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\
-- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\
-- & -- & -- & 0.044 & -- & -- & -- & 0.046 & 0.120 & -- & 0.120 & 0.278
\end{pmatrix}
$$

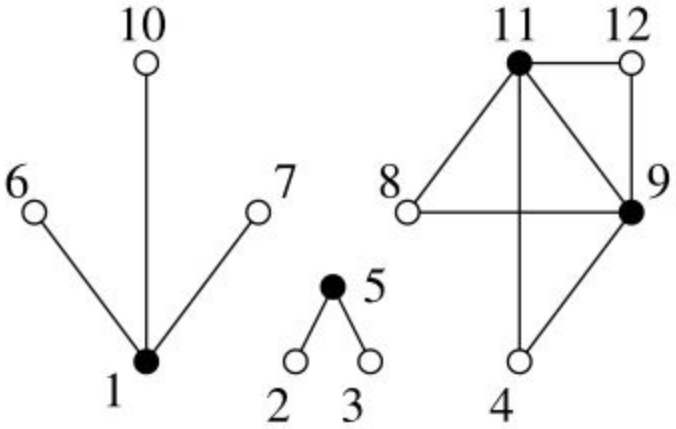
Clustering ibrido - Markov

$$\begin{pmatrix}
0.448 & 0.080 & 0.023 & 0.000 & 0.068 & 0.426 & 0.359 & 0.000 & 0.000 & 0.432 & 0.000 & -- \\
0.018 & 0.285 & 0.228 & 0.007 & 0.176 & 0.006 & 0.033 & 0.005 & 0.000 & 0.007 & 0.000 & 0.000 \\
0.005 & 0.223 & 0.290 & 0.022 & 0.173 & 0.000 & 0.010 & 0.017 & 0.003 & 0.001 & 0.003 & 0.001 \\
0.000 & 0.018 & 0.059 & 0.222 & 0.040 & 0.000 & 0.001 & 0.187 & 0.139 & 0.000 & 0.139 & 0.099 \\
0.027 & 0.312 & 0.314 & 0.028 & 0.439 & 0.005 & 0.054 & 0.022 & 0.003 & 0.010 & 0.003 & 0.001 \\
0.116 & 0.007 & 0.001 & 0.000 & 0.004 & 0.157 & 0.085 & 0.000 & -- & 0.131 & -- & -- \\
0.096 & 0.040 & 0.013 & 0.000 & 0.037 & 0.083 & 0.197 & 0.001 & 0.000 & 0.104 & 0.000 & 0.000 \\
0.000 & 0.012 & 0.042 & 0.172 & 0.029 & 0.000 & 0.002 & 0.198 & 0.133 & 0.000 & 0.133 & 0.096 \\
0.000 & 0.001 & 0.015 & 0.256 & 0.009 & -- & 0.000 & 0.266 & 0.326 & 0.000 & 0.326 & 0.346 \\
0.290 & 0.021 & 0.002 & 0.000 & 0.017 & 0.323 & 0.260 & 0.000 & 0.000 & 0.316 & 0.000 & -- \\
0.000 & 0.001 & 0.015 & 0.256 & 0.009 & -- & 0.000 & 0.266 & 0.326 & 0.000 & 0.326 & 0.346 \\
-- & 0.000 & 0.001 & 0.037 & 0.000 & -- & 0.000 & 0.039 & 0.069 & -- & 0.069 & 0.112
\end{pmatrix}$$

Clustering ibrido - Markov

$$
\begin{pmatrix}
1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}
$$

Clustering ibrido - Markov



Clustering ibrido

Per prima cosa è necessario costruire il grafo, quindi bisogna costruire la matrice associata. Le colonne della matrice di Markov sono normalizzate

$\mathcal{A}$ (matrice di similarità)
$\mathcal{A} = \mathcal{A}+\mathcal{I}$ (aggiunta dei loop nei nodi)
$\mathcal{D}$ (matrice di grado di $\mathcal{A}$)
$\mathcal{M} = \mathcal{A}\mathcal{D}^{-1}$ (matrice di Markov)

Clustering ibrido

La matrice di Markov presenta alcune proprietà.

Vedendo i vertici del grafo come stati, il valore riportato in M(i, j) è la probabilità di arrivare allo stato j dallo stato i.

$$\mathcal{M} = [\rho(i,j)] \in \mathbb{R}^{n \times n}$$
$$0 \leq \rho(i,j) \leq 1$$

$$\sum_{i=1}^{n} \rho(i,j) = 1$$

Clustering ibrido

Vengono definiti tre operatori:

- L'**espansione** permette di simulare il cammino aleatorio
- L'**inflazione** permette di normalizzare nuovamente le colonne
- Il **pruning** permette di ridurre il numero di valori non nulli

$$\mathcal{M}_{exp} = \mathcal{M}^e$$

$$\mathcal{M}_{inf}(i,j) = \frac{\mathcal{M}_{exp}(i,j)^\eta}{\sum_{k=1}^{n} \mathcal{M}_{exp}(k,j)^\eta}$$

$$\mathcal{M}_{pru}(i,j) = \begin{cases} 0, & \mathcal{M}_{inf}(i,j) < \zeta \\ \mathcal{M}_{inf}(i,j), & \text{otherwise} \end{cases}$$

Clustering ibrido

Per stabilire la convergenza della matrice viene considerato il **caos globale** della matrice di Markov partendo dal caos di ogni singola colonna

$$\mathcal{C}_j = \frac{\max\limits_{i=1,2,\ldots,n} \mathcal{M}_{pru}(i,j)}{\sum_{i=1}^{n} \mathcal{M}_{pru}(i,j)}$$

$$\mathcal{G} = \max_{j=1,2,\ldots,n} \mathcal{C}_j$$

Clustering ibrido

Algorithm 1 Markov Clustering Algorithm

input : Pairwise correlation matrix, $\mathcal{A}$
output: Probabilities matrix, $\mathcal{M}$
- expansion parameter: e
- inflation parameter: η
- global chaos: $\mathcal{G}$
- prune parameter: ς
- threshold for global chaos: ξ
- add self-loops to the graph $\mathcal{A}$, $\mathcal{A} = \mathcal{A} + \mathcal{I}$
- create the diagonal degree matrix of $\mathcal{A}$, $\mathcal{D}$
- create Markov matrix, $\mathcal{M} = \mathcal{A}\mathcal{D}^{-1}$

while $\mathcal{G} > \xi$ **do**
- expansion on $\mathcal{M}$, based on (5)
- inflation on $\mathcal{M}_{exp}$, based on (6)
- pruning on $\mathcal{M}_{inf}$, based on (7)
- update $\mathcal{G}$ based on (8) and (9)
- $\mathcal{M} = \mathcal{M}_{pru}$

end
return $\mathcal{M}$

Clustering ibrido

Ad ogni iterazione dello Hierarchical clustering, il Markov clustering viene effettuato su ogni batch.

I cluster ottenuti vengono ora uniti per formare cluster più grandi.

Per unire i cluster si utilizza la tecnica del **nearest neighbor** alle colonne della matrice di probabilità in stato di convergenza

Clustering ibrido

Nella matrice ottenuta come risultato consideriamo una **colonna non-sparsa** se il numero di valori non-zero è inferiore a 20.

Per ogni colonna non sparsa, appartenente al cluster *ci*, si cerca il cluster *cj* più prossimo controllando il valore di probabilità più alto all'interno della colonna.

I due cluster *ci* e *cj* vengono selezionati per il **merging**

Clustering ibrido

Per aumentare la precisione del merging dei cluster viene utilizzato un valore di soglia

$\mathcal{A}(f_i, f_j) > \mathcal{T}$ (condizione di merge)
τ (valore di soglia minimo)
$n_{c_i},\ n_{c_j}$ (numero di RN nei cluster)
$\mu_{c_i},\ \mu_{c_j}$ (media dei valori di similarità dei RN nei cluster)
ψ (fattore di scala predefinito)

$$\mathcal{T} = \max\left(\tau, \frac{\psi\sqrt{n_{c_i} n_{c_j} \mu_{c_i}^2 \mu_{c_j}^2}}{\sqrt{[(n_{c_i}-1)\mu_{c_i}^2 + 1][(n_{c_j}-1)\mu_{c_j}^2 + 1]}}\right)$$

Clustering ibrido

Algorithm 2 Proposed Hybrid Clustering Algorithm

input : pre-processed RNs
output: list of clusters, C
- number of RNs, N
- scaling factor, ψ in (10)
- minimum threshold, τ in (10)
- size of batches, q
- clustering initialization, $C_{old} = \{\}$
- considering a set of single clusters corresponding to the RNs, $C_{new} = \{c_1, c_2, \ldots, c_N\}$
- initializing a set of camera fingerprints with the RNs corresponding to the clusters, $F = \{f_1, f_2, \ldots, f_N\}$
- partitioning initialization, $B_{old} = \{\}$
- $t = \lceil \frac{N}{q} \rceil$
- randomly partition C_{new} into t batches with size q, $B_{new} = \{b_1, b_2, \ldots, b_t\}$

Clustering ibrido

while $|B_{new}| \neq |B_{old}|$ **do**
  for $k = 1 : t$ **do**
    while $|C_{new}| \neq |C_{old}|$ **do**
      - compute correlation matrix $\mathcal{A}$ by (3)
      - apply Markov clustering to $\mathcal{A}$ and generate the probability matrix $\mathcal{M}$ by Algorithm 1
      - put non-sparse column's indices in the list $\mathcal{L}$
      for $i = 1 : |\mathcal{L}|$ **do**
        - find the nearest cluster c_j to the cluster c_i from the list $\mathcal{L}$
        - compute the adaptive threshold $\mathcal{T}$ by (10)
        if $\mathcal{A}(f_i, f_j) > \mathcal{T}$ **then**
          - merge clusters c_i and c_j
        else
          - continue
        end
      end

Clustering ibrido

- put the obtained clusters in C_{new}
- update the camera fingerprints in F for the merged clusters by (2)
- $C_{old} = C_{new}$

end

end

- consider all the obtained clusters from batches as a new cluster C_{new}
- $B_{old} = B_{new}$
- $N = |C_{new}|$
- update t, $t = \lceil \frac{N}{q} \rceil$
- partition the clusters in C_{new} into t batches with size q, and form B_{new}

end

- $C = C_{new}$

return C

Post-processo

Al termine del processo si punta ad ottenere cluster di dimensioni ragionevoli contenenti RN che condividono le stesse caratteristiche.

A causa della natura del SPN però saranno presenti anche cluster di piccole dimensioni molto specifici.

Per aumentare la precisione dello strumento proposto i cluster di piccole dimensioni vengono rimossi

Post-processo

Per poter distinguere i cluster in base alla dimensione viene calcolato **ζ**i.

Se **ζ**i ≤ 1 il cluster viene rimosso

$$\zeta_i = \frac{|c_i| \cdot |C|}{N}$$

Sperimentazione

Per effettuare una sperimentazione dell'algoritmo è necessario definire alcuni punti:

- VISION dataset: riferimento per i metodi basati su SPN
- HMC dataset: formato ridotto del VISION dataset su cui applicare l'algoritmo
- Misure sperimentali: per definire la qualità del metodo

VISION dataset

VISION è un dataset di immagini e video creato per testare gli strumenti che operano su SPN.

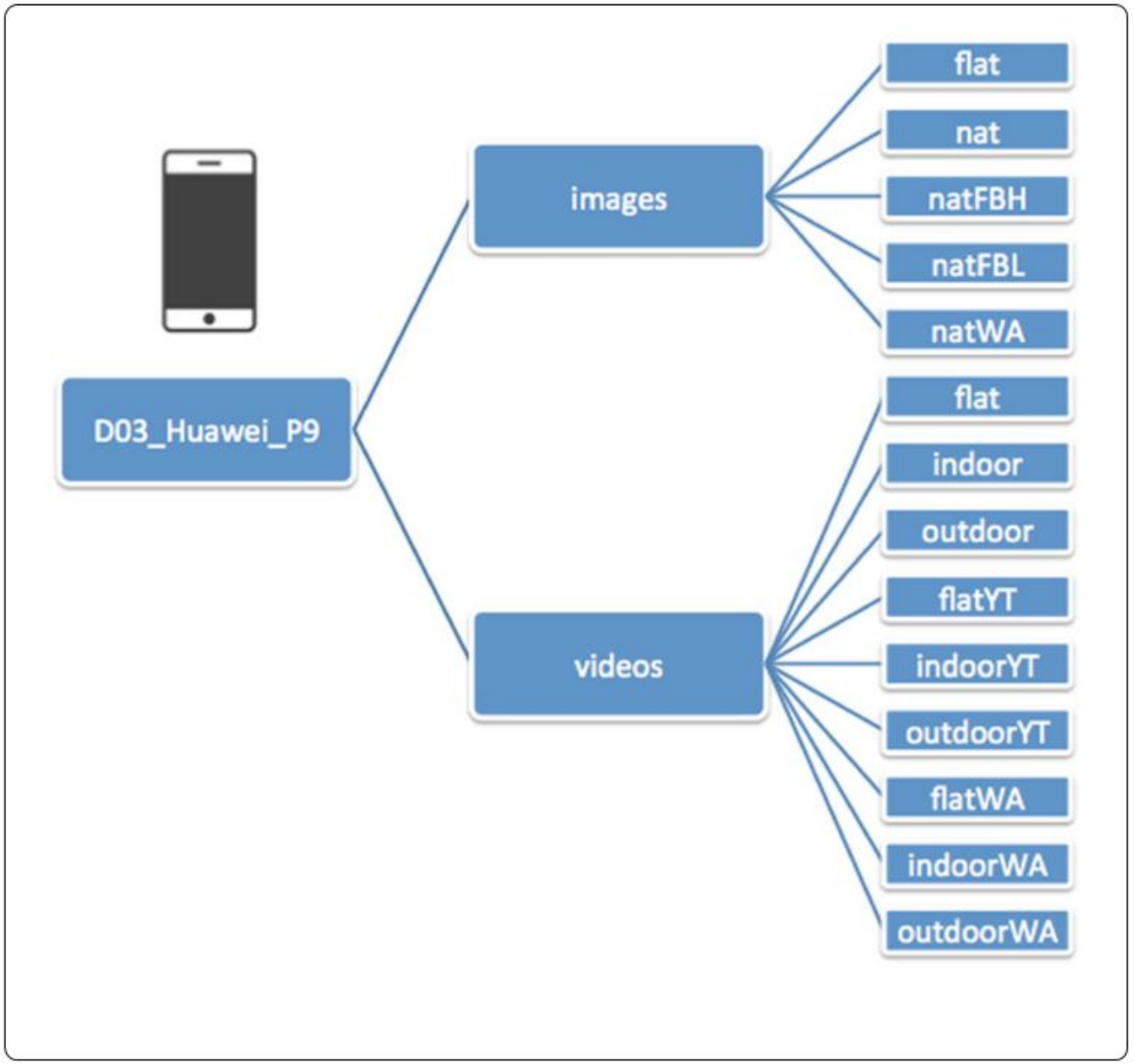
- **Immagini**
  - *Flat*: immagini in landscape rappresentanti superfici piane (muri, cielo, …)
  - *Nat*: immagini generiche, condivise su Facebook e WhatsApp
- **Video**
  - *Flat*: video in landscape di superfici piane (muri, cielo, …)
  - *Indoor*: video di interni (uffici, negozi, …)
  - *Outdoor*: video all'aperto (giardini, …)

VISION dataset

Composizione:

- **35 smartphone e tablet** appartenenti a **11 brand**: Apple, Asus, Huawei, Lenovo, LG electronics, Microsoft, OnePlus, Samsung, Sony, Wiko, and Xiaomi
- 11,732 immagini native, 7565 delle quali condivise su Facebook, in alta e bassa risoluzione, e su WhatsApp, risultando un totale di **34,427 immagini**
- 648 video nativi, 622 dei quali condivisi su YouTube alla risoluzione più alta disponibile, e 644 condivisi su WhatsApp, risultando un totale di **1914 video**

VISION dataset



HMC dataset

Operando su immagini è stato considerato solo il dataset di immagini. Vengono considerate solo le immagini *Nat*.

Il dataset ottenuto Ɗ viene diviso in due:

- Ɗ1: modelli di smartphone uguali
- Ɗ2: modelli di smartphone completamente diversi

Vengono escluse da Ɗ1 e Ɗ2 le immagini:

- Scure (70% dei pixel con intensità inferiore a 50)
- Saturate (70% dei pixel con intensità superiore a 250)

HMC dataset

Il dataset finale è quindi:

- $\mathcal{D}1$
  - 2250 immagini
  - 11 smartphone
  - 5 modelli
- $\mathcal{D}2$
  - 5230 immagini
  - 24 smartphone
  - modelli completamente differenti
- $\mathcal{D} = \mathcal{D}1 \cup \mathcal{D}2$
  - 7480 immagini

HMC dataset

Dataset D1:

ID	Brand	Model	Original resolution	#images
S_1	Apple	iPhone 4S	3264×2448	178
S_2	Apple	iPhone 4S	3264×2448	200
S_3	Apple	iPhone 5	3264×2448	203
S_4	Apple	iPhone 5	3264×2448	223
S_5	Apple	iPhone 5c	3264×2448	201
S_6	Apple	iPhone 5c	3264×2448	206
S_7	Apple	iPhone 5c	3264×2448	333
S_8	Apple	iPhone 6	3264×2448	129
S_9	Apple	iPhone 6	3264×2448	227
S_{10}	Samsung	Galaxy S III Mini GT-I8190	2560×1920	150
S_{11}	Samsung	Galaxy S III Mini GT-I8190N	2560×1920	200

HMC dataset

Dataset D2:

ID	Brand	Model	Original resolution	#images
S_1	Apple	iPad2	960 × 720	170
S_2	Apple	iPad mini G	2592 × 1936	157
S_3	Apple	iPhone 4	2592 × 1936	217
S_4	Apple	iPhone 6 Plus	3264 × 2448	256
S_5	Asus	Zenfone	3264 × 1836	208
S_6	Huawei	Ascend G6-U10	3264 × 2448	153
S_7	Huawei	Honor 5C	4160 × 3120	271
S_8	Huawei	P8 GRA-L09	4160 × 2336	265
S_9	Huawei	P9 EVA-L09	3968 × 2976	237
S_{10}	Huawei	P9 Lite VNS-L31	4160 × 3120	234
S_{11}	Lenovo	P70-A	4784 × 2704	216
S_{12}	LG	D290	3264 × 2448	224
S_{13}	Microsoft	Lumia 640 LTE	3264 × 2448	180
S_{14}	OnePlus	A3000	4640 × 3480	284
S_{15}	OnePlus	A3003	4640 × 3480	236
S_{16}	Samsung	Galaxy S3 GT-I9300	3264 × 2448	207
S_{17}	Samsung	Galaxy S4 Mini GT-I9195	3264 × 1836	208
S_{18}	Samsung	Galaxy S5 SM-G900F	5312 × 2988	254
S_{19}	Samsung	Galaxy Tab 3 GT-P5210	2048 × 1536	166
S_{20}	Samsung	Galaxy Tab A SM-T555	2592 × 1944	154
S_{21}	Samsung	Galaxy Trend Plus GT-S7580	2560 × 1920	163
S_{22}	Sony	Xperia Z1 Compact D5503	5248 × 3936	216
S_{23}	Wiko	Ridge 4G	3264 × 2448	249
S_{24}	Xiaomi	Redmi Note 3	4608 × 2592	305

HMC dataset

Per ognuno dei due dataset Ɗ1 e Ɗ2 vengono considerate sia le immagini native sia le immagini condivise. Ognuno dei quattro dataset finali contiene 7480 immagini

$$\mathcal{D}^N = \mathcal{D}_1^N \cup \mathcal{D}_2^N \text{ (native)}$$
$$\mathcal{D}^W = \mathcal{D}_1^W \cup \mathcal{D}_2^W \text{ (WhatsApp)}$$
$$\mathcal{D}^{FH} = \mathcal{D}_1^{FH} \cup \mathcal{D}_2^{FH} \text{ (Facebook alta risoluzione)}$$
$$\mathcal{D}^{FL} = \mathcal{D}_1^{FL} \cup \mathcal{D}_2^{FL} \text{ (Facebook bassa risoluzione)}$$

Misure sperimentali

Consideriamo i cluster reali, i cluster ottenuti come risultato e due campioni presi dal dataset

$$T = \{t_1, t_2, ..., t_g\} \text{ (cluster reali)}$$
$$C = \{c_1, c_2, ..., c_h\} \text{ (cluster ottenuti)}$$
$$D = \{d_1, d_2, ..., d_N\} \text{ (dataset)}$$
$$d_i, d_j \text{ (due campioni)}$$

Misure sperimentali

Otteniamo quindi quattro insiemi basati sull'aderenza dei cluster ottenuti ai cluster reali

True Positive

$$TP = \{(d_i, d_j) : t_i = t_j \wedge c_i = c_j\}$$

False Negative

$$FN = \{(d_i, d_j) : t_i = t_j \wedge c_i \neq c_j\}$$

False Positive

$$FP = \{(d_i, d_j) : t_i \neq t_j \wedge c_i = c_j\}$$

True Negative

$$TN = \{(d_i, d_j) : t_i \neq t_j \wedge c_i \neq c_j\}$$

Misure sperimentali

Sulla base degli insiemi precedenti vengono definite le seguenti misure

Precision rate

$$\mathcal{P} = \frac{|TP|}{|TP| + |FP|}$$

Recall rate o true positive rate

$$\mathcal{R} = \frac{|TP|}{|TP| + |FN|}$$

F1-measure

$$\mathcal{F} = 2 \cdot \frac{\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}}$$

Rand index

$$RI = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|}$$

Misure sperimentali

Adjusted rand index

$$ARI = \frac{RI - \overline{RI}}{1 - \overline{RI}}$$

Purity

$$Purity = \frac{\sum_{i=1}^{|C|} \frac{|\widehat{c_i}|}{|c_i|}}{|C|}$$

False positive rate

$$FPR = \frac{|FP|}{|FP| + |TN|}$$

Ratio del numero di cluster

$$\mathcal{N} = \frac{n_o}{n_g}$$

Misure sperimentali

Può essere difficile ottenere i risultati migliori in ognuna delle misure, quindi ci concentreremo su valori alti per **precisione rate** e **purity**, su valori bassi per **false positive rate** e su un valore accurato per il **ratio del numero di cluster**

Risultati

Analizzando i risultati ottenuti è possibile fare considerazioni su vari aspetti. In particolare:

- Resizing e cropping dei RN
- Valori dei parametri dell'algoritmo
- Qualità generale del metodo proposto
- Confronto con altri metodi basati su SPN

Resizing

E’ necessario portare le immagini tutte alla stessa risoluzione. Dai risultati sperimentali si è scelto il **resizing** ad una definizione di **1024x1024**

	Resizing							Cropping*						
Size	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
1280 × 1024	**0.997**	0.758	0.861	0.858	**0.997**	**0.000**	**35/35**	——	——	——	——	——	——	– /35
1024 × 1024	**0.997**	**0.765**	**0.866**	**0.863**	0.996	**0.000**	**35/35**	——	——	——	——	——	——	– /35
960 × 720	0.986	0.725	0.835	0.831	0.992	**0.000**	34/35	0.693	0.614	0.651	0.641	0.966	0.007	31/35
512 × 512	0.953	0.440	0.602	0.595	0.964	**0.000**	48/35	0.676	0.497	0.572	0.561	0.962	0.007	37/35
256 × 256	0.508	0.027	0.051	0.048	0.596	**0.000**	280/35	0.654	0.303	0.411	0.401	0.882	0.004	60/35
128 × 128	0.031	0.146	0.050	0.004	0.176	0.132	26/35	0.487	0.138	0.212	0.203	0.598	0.004	159/35

Parametri

Sono stati definiti quattro dataset campione, ognuno dei quali prendendo **100 immagini random** dai **35 modelli** di smartphone, per un totale di **3500 immagini**

$$\mathcal{D}_0^N \subseteq \mathcal{D}^N$$
$$\mathcal{D}_0^W \subseteq \mathcal{D}^W$$
$$\mathcal{D}_0^{FH} \subseteq \mathcal{D}^{FH}$$
$$\mathcal{D}_0^{FL} \subseteq \mathcal{D}^{FL}$$

Parametri

Usando i quattro dataset campione, i parametri sono stati impostati empiricamente in modo da ottenere la miglior qualità nei cluster

Notation	Value	Description
q	1000	batch size for partitioning dataset
e	2	expansion parameter in (5)
η	1	inflation parameter in (6)
ς	0.005	prune parameter in (7)
$\mathcal{G}$	2	initial value of global chaos in (9) and Algorithm 1
ξ	0.3	threshold for global chaos in (9)
ψ	0.15	scaling factor in (10) for $\mathcal{D}_1^{\mathrm{N}}$,$\mathcal{D}_2^{\mathrm{N}}$ and $\mathcal{D}^{\mathrm{N}}$
	0.09	scaling factor in (10) for $\mathcal{D}_1^{\mathrm{W}}$,$\mathcal{D}_2^{\mathrm{W}}$ and $\mathcal{D}^{\mathrm{W}}$
	0.07	scaling factor in (10) for $\mathcal{D}_1^{\mathrm{FH}}$,$\mathcal{D}_2^{\mathrm{FH}}$ and $\mathcal{D}^{\mathrm{FH}}$
	0.03	scaling factor in (10) for $\mathcal{D}_1^{\mathrm{FL}}$,$\mathcal{D}_2^{\mathrm{FL}}$ and $\mathcal{D}^{\mathrm{FL}}$
τ	0.004	minimum threshold in (10) for adaptive threshold

Risultati finali

L'algoritmo partiziona casualmente i RN in batch.
Eseguendo più volte l'algoritmo i risultati possono variare.
Quindi per ogni dataset l'algoritmo è stato eseguito 10 volte; i valori riportati sono le medie dei valori ottenuti con le 10 esecuzioni

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	ARI	$Purity$	FPR	$\mathcal{N}$
$\mathcal{D}^{\mathrm{N}}$	0.992	0.720	0.834	0.830	0.994	0.000	37/35
D^{W}	0.964	0.600	0.733	0.727	0.975	0.000	33/35
$\mathcal{D}^{\mathrm{FH}}$	0.962	0.610	0.746	0.740	0.975	0.000	33/35
$\mathcal{D}^{\mathrm{FL}}$	0.750	0.513	0.609	0.599	0.847	0.005	33/35

Risultati finali

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
$\mathcal{D}_1^{\mathrm{N}}$	1.000	0.826	0.905	0.896	1.000	0.000	11/11
$\mathcal{D}_1^{\mathrm{W}}$	0.975	0.775	0.863	0.850	0.993	0.002	13/11
$\mathcal{D}_1^{\mathrm{FH}}$	0.994	0.720	0.835	0.821	0.994	0.001	12/11
$\mathcal{D}_1^{\mathrm{FL}}$	0.866	0.601	0.705	0.680	0.914	0.009	9/11

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
$\mathcal{D}_2^{\mathrm{N}}$	0.992	0.672	0.801	0.794	0.992	0.000	27/24
$\mathcal{D}_2^{\mathrm{W}}$	0.970	0.610	0.750	0.741	0.972	0.000	24/24
$\mathcal{D}_2^{\mathrm{FH}}$	0.958	0.627	0.758	0.749	0.966	0.001	25/24
$\mathcal{D}_2^{\mathrm{FL}}$	0.798	0.543	0.647	0.634	0.876	0.006	29/24

Confronto con altri algoritmi

L'algoritmo proposto, HMC, è stato confrontato con altri tre algoritmi di clustering di immagini basati su SPN. Questi sono **Correlation Clustering** (CC), **Fast Clustering** (FC) e **Hierarchical Clustering** (HC).

Tutti gli algoritmi sono stati testati sui quattro dataset ĐN, ĐW, ĐFH e ĐFL analizzando i risultati sia per quanto riguarda la **qualità dei cluster** sia per quanto riguarda il **tempo di esecuzione**

Qualità dei cluster

In alto ƉN, in basso ƉW

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
HMC	**0.992**	0.720	0.834	0.830	**0.994**	**0.000**	**37/35**
CC	0.987	0.863	**0.921**	**0.919**	0.915	**0.000**	46/35
FC	0.952	0.759	0.845	0.841	0.982	0.001	63/35
HC	0.205	**0.949**	0.338	0.304	0.828	0.112	23/35

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
HMC	**0.964**	0.600	0.733	0.727	**0.975**	**0.000**	**33/35**
CC	0.952	0.787	**0.862**	**0.858**	0.856	0.001	56/35
FC	0.919	0.722	0.809	0.804	0.974	0.001	65/35
HC	0.245	**0.863**	0.382	0.352	0.813	0.081	32/35

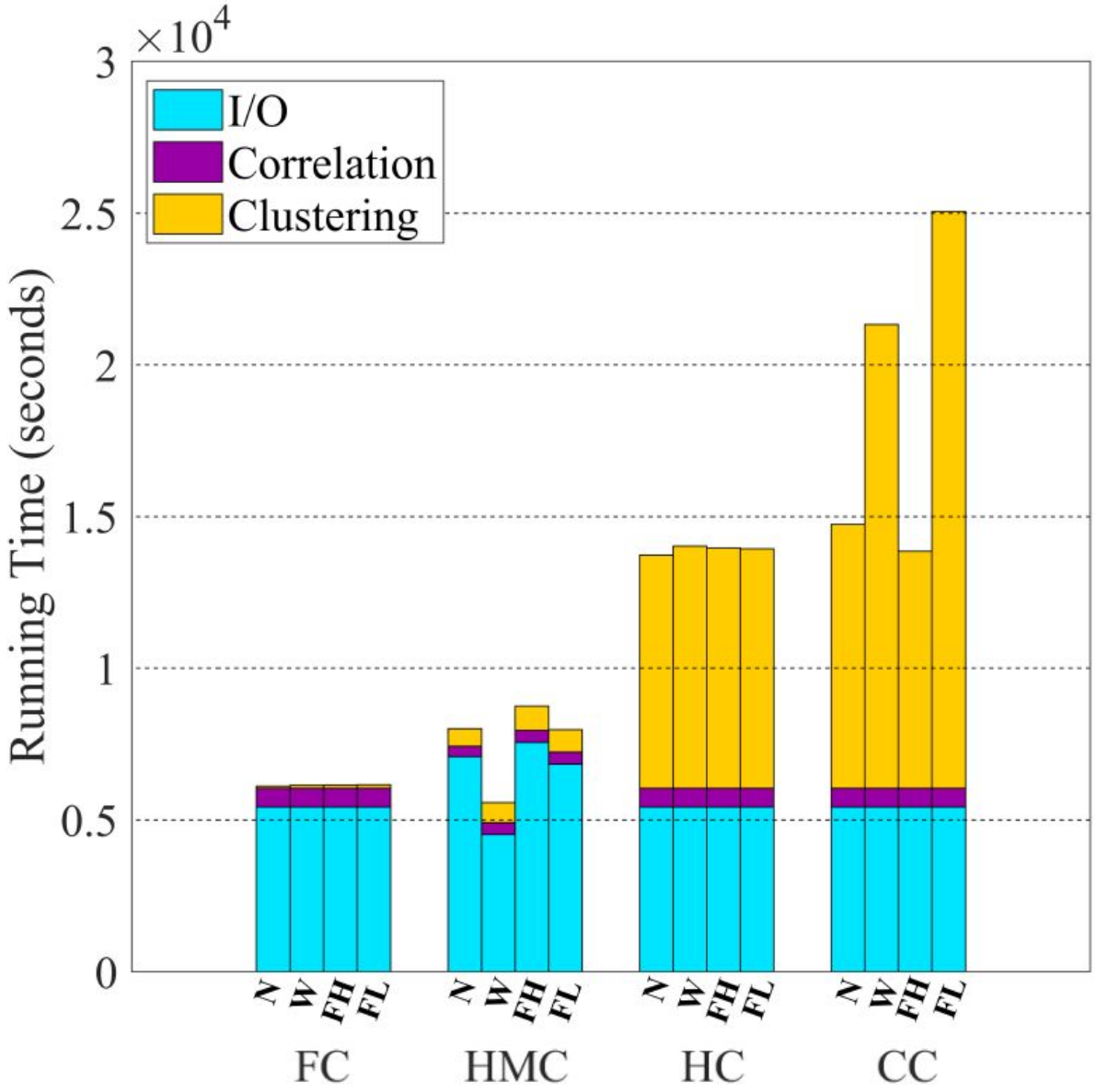
Qualità dei cluster

In alto ƊFH, in basso ƊFL

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
HMC	**0.962**	0.610	0.746	0.740	**0.975**	**0.000**	**33/35**
CC	0.955	0.793	**0.866**	**0.863**	0.841	0.001	58/35
FC	0.913	0.758	0.828	0.824	0.974	0.002	64/35
HC	0.475	**0.823**	0.602	0.587	0.793	0.027	30/35

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
HMC	**0.750**	0.513	0.609	0.599	0.847	**0.005**	**33/35**
CC	0.712	0.632	0.669	0.660	0.776	0.007	42/35
FC	0.665	0.690	**0.717**	**0.680**	**0.915**	0.011	52/35
HC	0.031	**0.999**	0.061	0.003	0.520	0.941	2/35

Tempo di esecuzione



Conclusioni

E’ stato presentato un metodo per il clustering di immagini condivise sui social network basato su SPN. Il metodo si propone di:

- Operare senza la conoscenza a priori del numero di cluster da ottenere
- Effettuare un ridimensionamento dei RN per ottenere risultati migliori nei cluster
- Risolvere il problema del limite della RAM partizionando i RN in batch; quindi favorendo la scalabilità
- Fondere iterativamente i cluster ottenuti adattando un valore di soglia per ottenere un’ottima qualità dei cluster

Riferimenti

[1] Rahimeh Rouhi, Flavio Bertini, Danilo Montesi, Xufeng Lin, Yijun Quan, and Chang-Tsun Li, Hybrid Clustering of Shared Images on Social Networks for Digital Forensics. IEEE Access 2019

[2] Stijn Van Dongen, Graph Clustering Via a Discrete Uncoupling Process. SIAM J. Matrix Anal. Appl. 2008

[3] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya & Alessandro Piva, VISION: a video and image dataset for source identification. EURASIP Journal on Information Security 2017