**Esperimenti modulo face detection**

Vengono di seguito presentati i risultati relativi alle performance del modulo di face detection forgeries su coppie di facce.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Numero** | **Train** | **Test** | **ACC** | **BIC** | **CCV** | **LCH** |
| **1** | DSO-1 | DSO-1 | 0.75 | 0.75 | 0.72 | **0.78** |
| **2** | DSI-1 | DSI-1 | 0.78 | 0.79 | 0.77 | **0.82** |
| **3** | DSO-1 | DSI-1 | 0.56 | **0.57** | 0.53 | 0.53 |
| **4** | DSI-1 | DSO-1 | 0.58 | 0.55 | 0.51 | **0.59** |

Nella tabella soprastante sono riportati i valori di Accuray nella classificazione utilizzando un solo descrittore alla volta. Sono quindi considerati solamente i due modelli dati dalle due coppie formate da GGE e IIC con il descrittore considerato.

Per ciascuno dei casi di test presenti nella tabella sottostante vengono invece riportati:

* Il dataset su cui è stato fatto il training
* Il dataset su cui è stato effettuato il test
* Il valore di accuratezza (**Accuracy**) raggiunta nella classificazione
* Il valore dell’*area sottesa alla curva ROC* (**AUC**)
* Il valore di accuracy espresso in F1-Score (**F-Score**)

I risultati portati in precedenza erano dati da una classificazione con votazione a maggioranza secca, in cui ciascun classificatore (degli otto totali) votava con 0 o 1 (a seconda della predizione). Gli *scores* qui considerati sono invece il risultato della somma di *predizioni soft* in [0, 1], in cui il singolo valore indica la confidenza della classificazione.

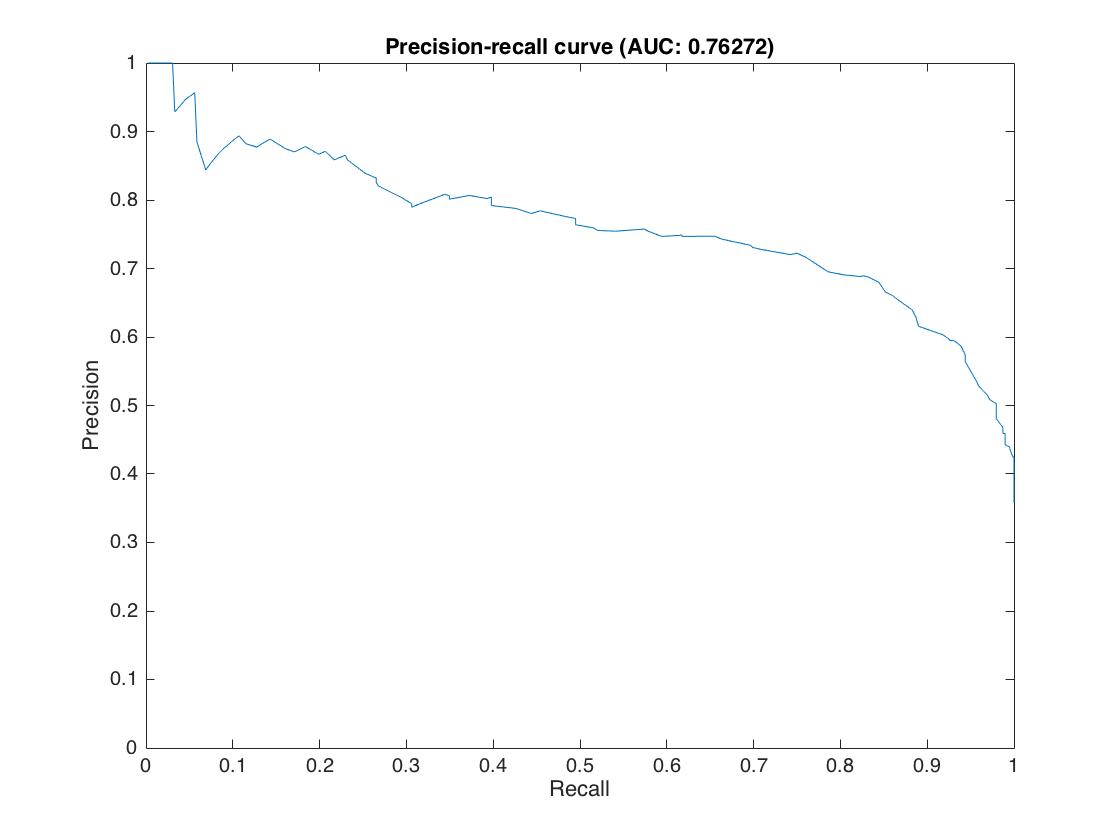
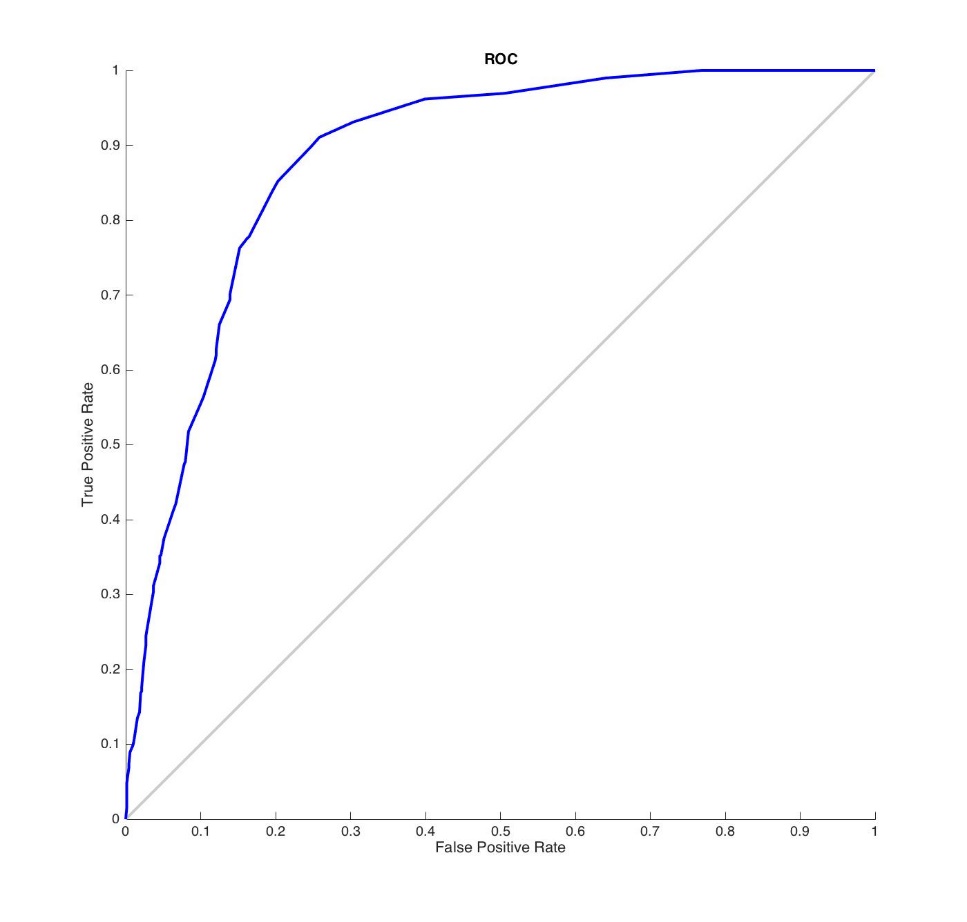
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numero** | **Train** | **Test** | **Accuracy** | **AUC** | **F-Score** |
| **1** | DSO-1 | DSO-1 | 0.82 | 0.88 | 0.77 |
| **2** | DSI-1 | DSI-1 | 0.87 | 0.92 | 0.87 |
| **3** | DSO-1 | DSI-1 | 0.58 | 0.58 | 0.62 |
| **4** | DSI-1 | DSO-1 | 0.62 | 0.59 | 0.53 |

**Nota**: *nei precedenti risultati, il valore di accuratezza nel caso di test numero 3 era di 0.43. Nel caso di soglia con valori soft questo valore si alza di molto.*

Per ciascuno dei casi di test vengono di seguito riportate le curve ROC e Precision/Recall.

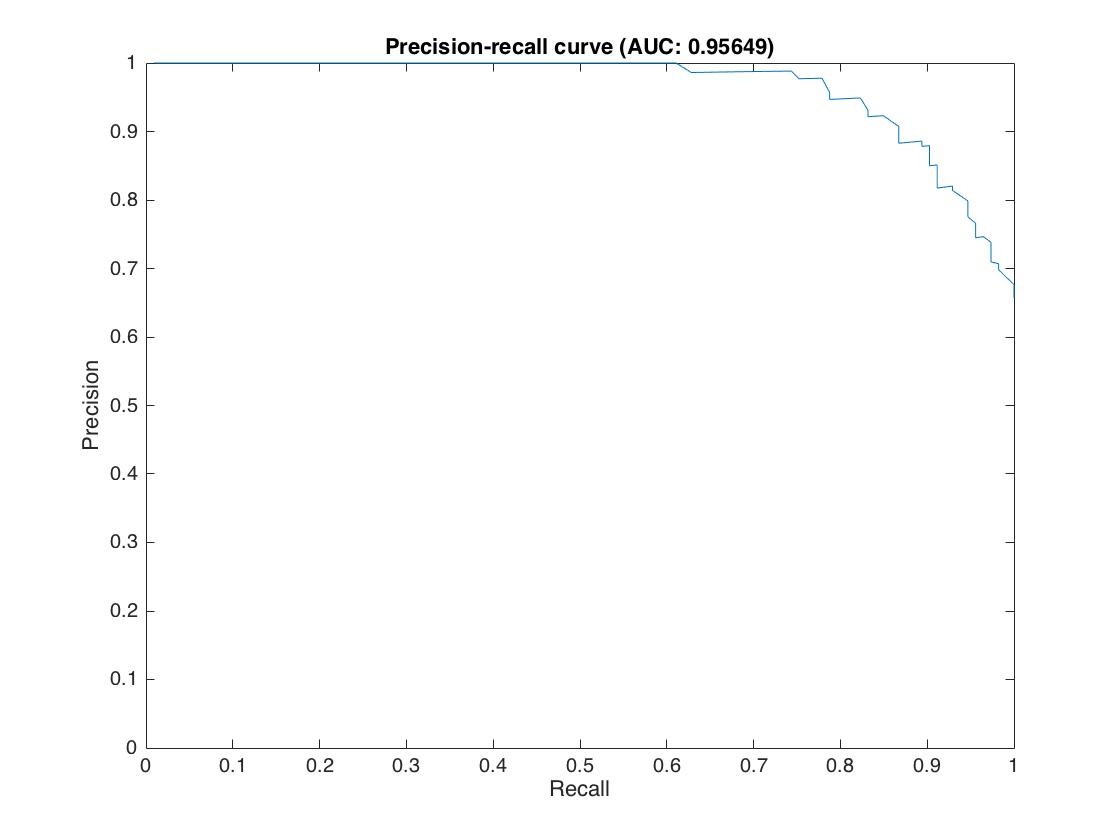
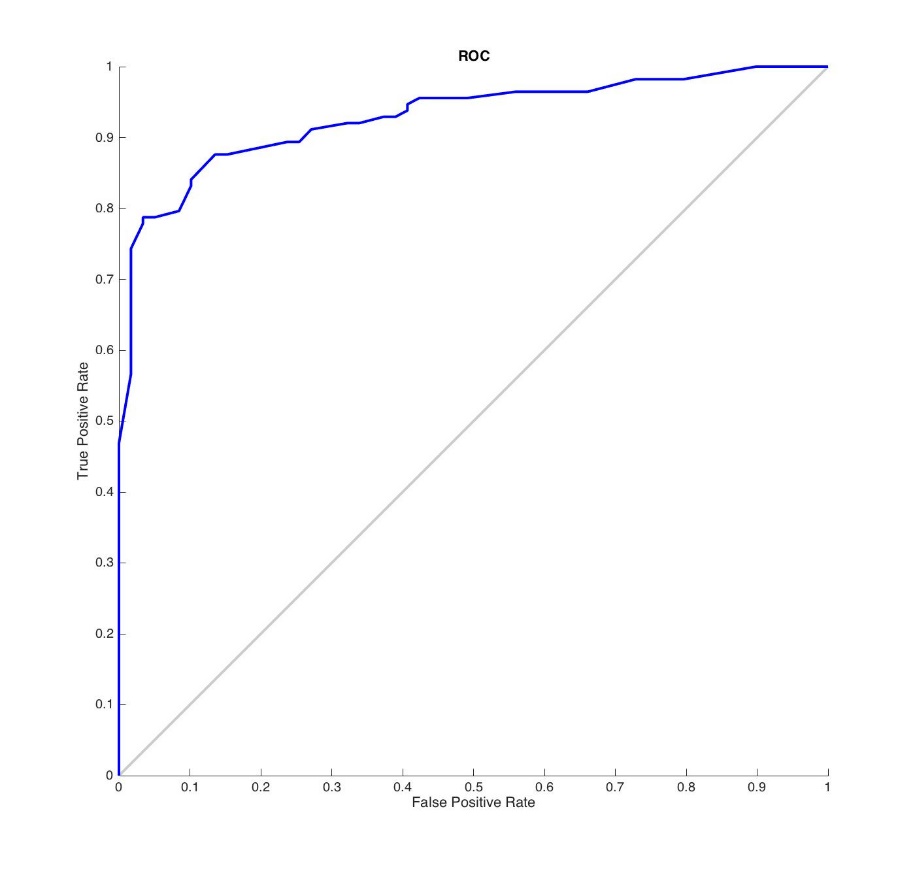
**Caso numero 1**

Train DSO-1, test DSO-1 - crossvalidazione



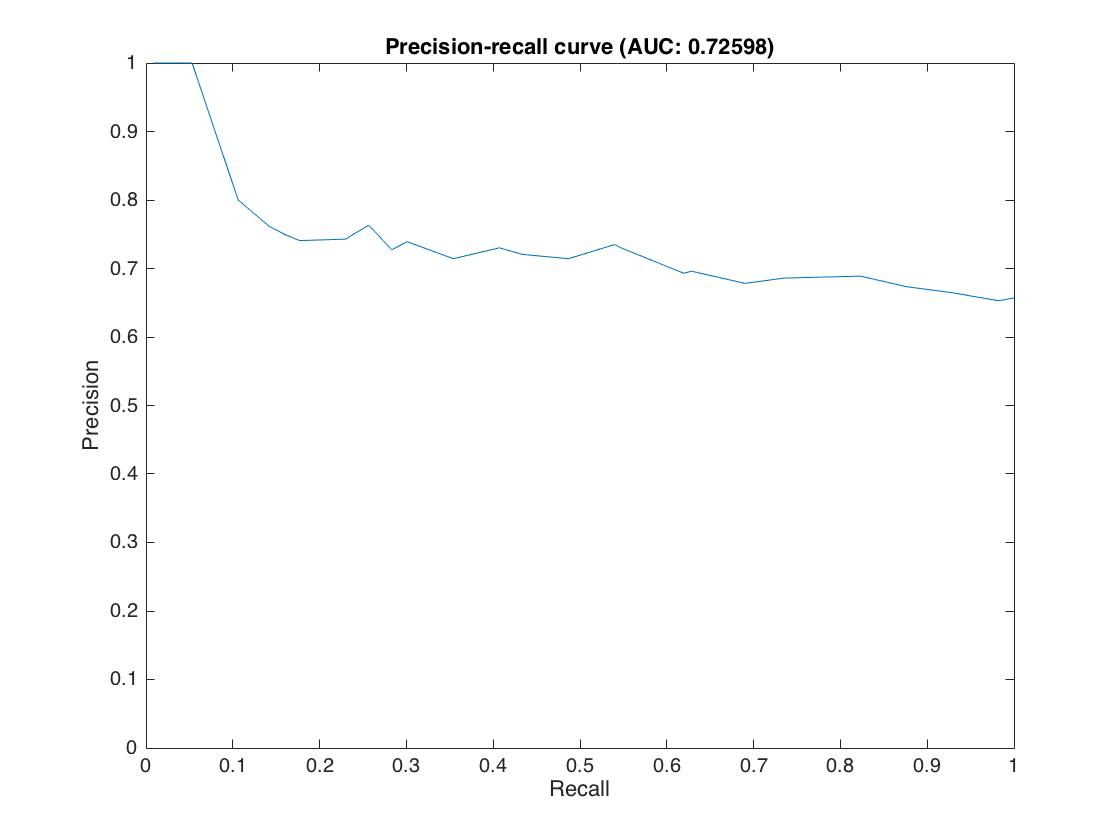
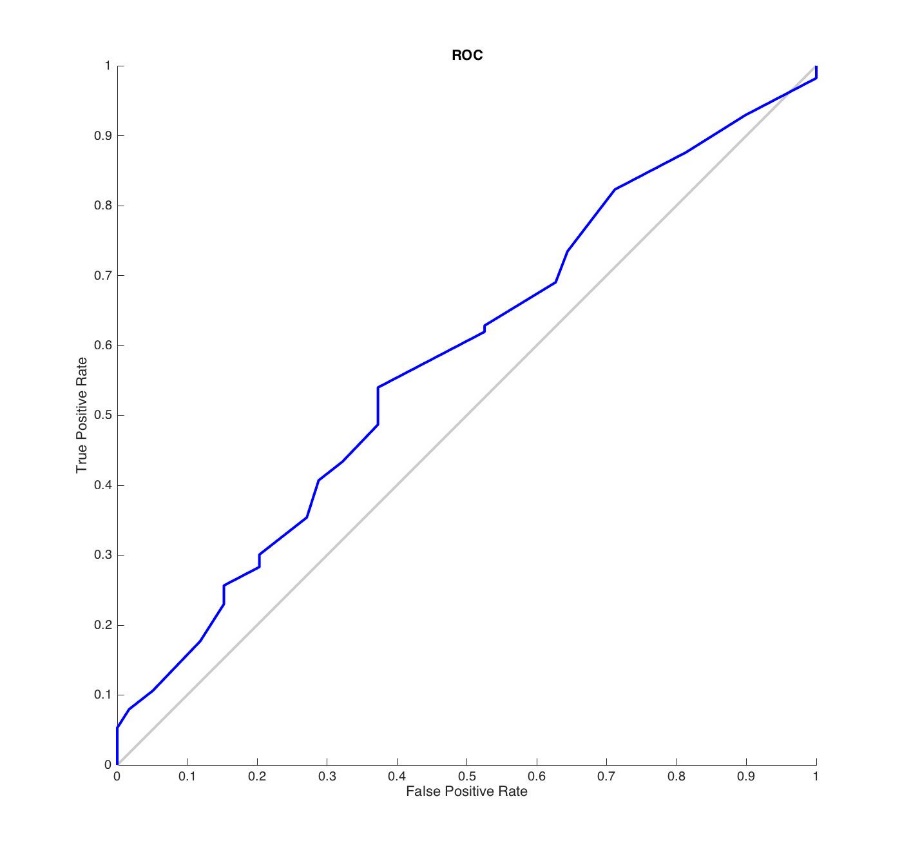
**Caso numero 2**

Train DSI-1, test DSI-1 – crossvalidazione



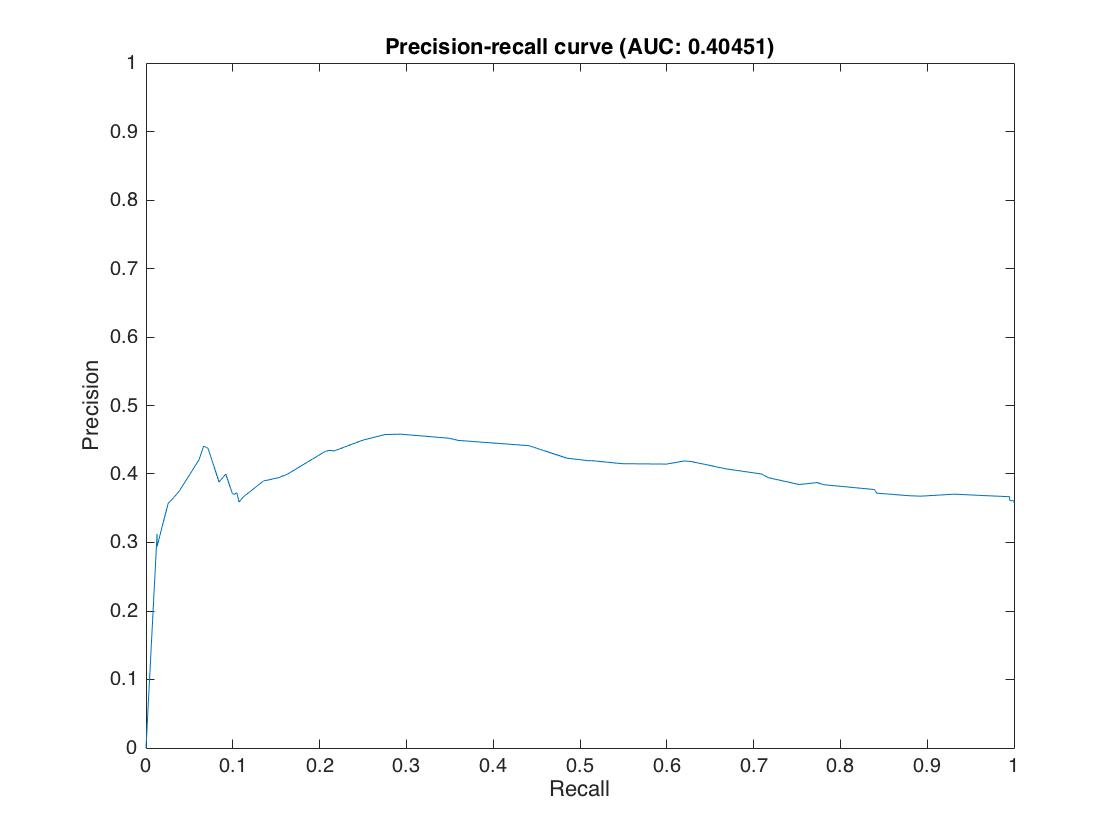
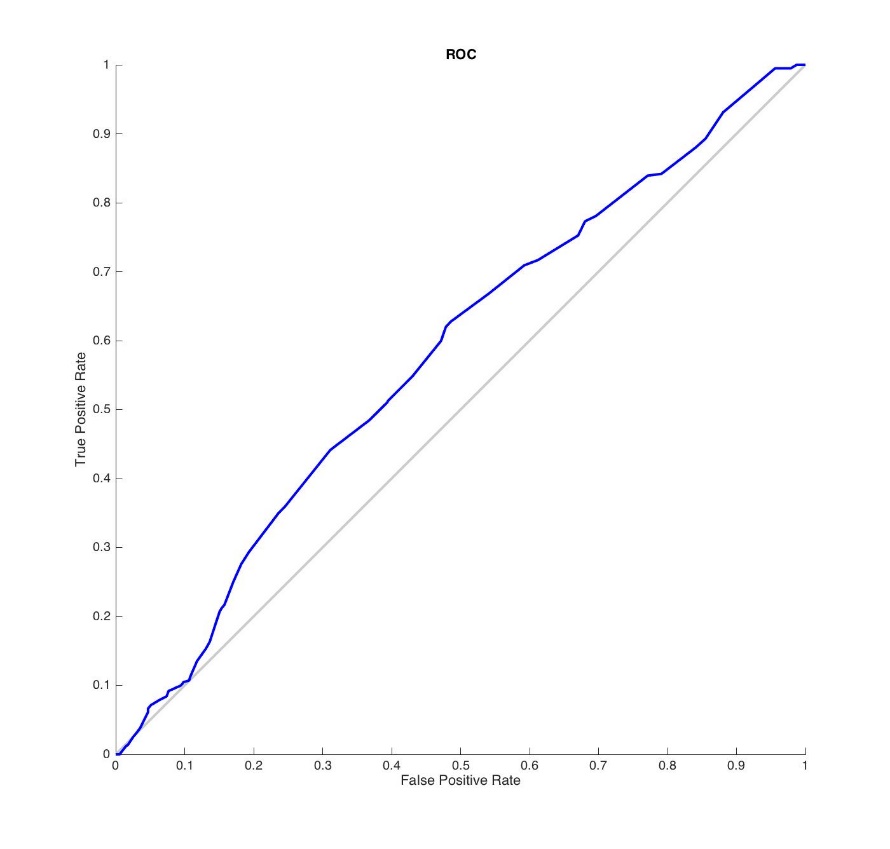
**Caso numero 3**

Train DSO-1, test DSI-1

****

**Caso numero 4**

Train DSI-1, test DSO-1



**Esperimenti modulo regional detection**

Vengono di seguito riportati gli esperimenti relativi al metodo di detection regionale basato sulla segmentazione in bande dell’immagine.

Per ciascuno dei casi in esame:

1. Viene calcolata la detection map
2. La classificazione viene valutata per singolo pixel dell’immagine: a partire dalla detection map si ricavano i pixel corrispondenti alla regione in cui è presente il tampering (data da una maschera groundtruth) e si considera i rispettivi valori della mappa. Al termine dell’operazione si avranno così due insiemi di pixel (e rispettivi scores), uno per pixel classificati come positivi ed uno per quelli negativi.
3. Si ripete il punto 2 per ciascuna delle immagini del dataset di riferimento, concatenando gli insiemi di pixel ottenuti.
4. Si procede alla valutazione delle performance sugli insiemi di pixel finali.

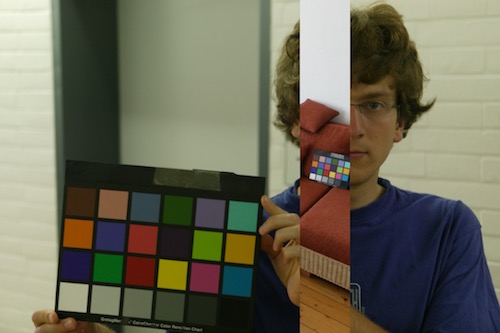
Valutazione su 25 immagini del dataset.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Numero** | **Train** | **RC** | **ACC** | **AUC** | **F-Score** | **Th** |
| **1** | SplicedCC | Global | 0.57 | 0.57 | 0.29 | 0.66 |
| **2** | SplicedCC | Median | 0.54 | 0.53 | 0.26 | 0.46 |
| **3** | - | Global | 0.52 | 0.40 | 0.27 | 0.56 |
| **4** | - | Median | 0.49 | 0.32 | 0.25 | 0.32 |
| **5** | SplicedDSO | Global | 0.61 | 0.63 | 0.32 | 0.78 |
| **6** | SplicedDSO | Median | 0.53 | 0.50 | 0.27 | 0.60 |

**Addestramento**

Per la fase di training ho generato due diversi dataset con la tecnica che avevamo concordato all’ultimo ricevimento. In entrambi i casi, i dataset sono a loro volta suddivisi per tipo di splicing, orizzontale e verticale. L’addestramento è distinto per direzione, vengono ciò creati due modelli, uno per bande orizzontali ed uno per le verticali.

Il primo dataset, *SplicedCC*, utilizza le immagini del dataset di ColorChecker (tutte vere) e ne crea di false con bande orizzontali e verticali (la banda viene estratta da un’altra immagine casuale tra le originali dello stesso dataset).



Il secondo dataset, *SplicedDSO*, utilizza le immagini non alterate del DSO-1 e ne crea di false con bande orizzontali e verticali (la banda viene estratta da un’altra immagine casuale tra le originali dello stesso dataset).



Nella fase di training, le immagini vengono segmentate in bande non sovrapposte, in fase di test le bande si sovrappongono del 25% del loro spessore (in questo modo ciascun pixel viene valutato 8 volte in totale, 4 per la scansione in orizzontale e 4 per quella in verticale).

**Alcune considerazioni**

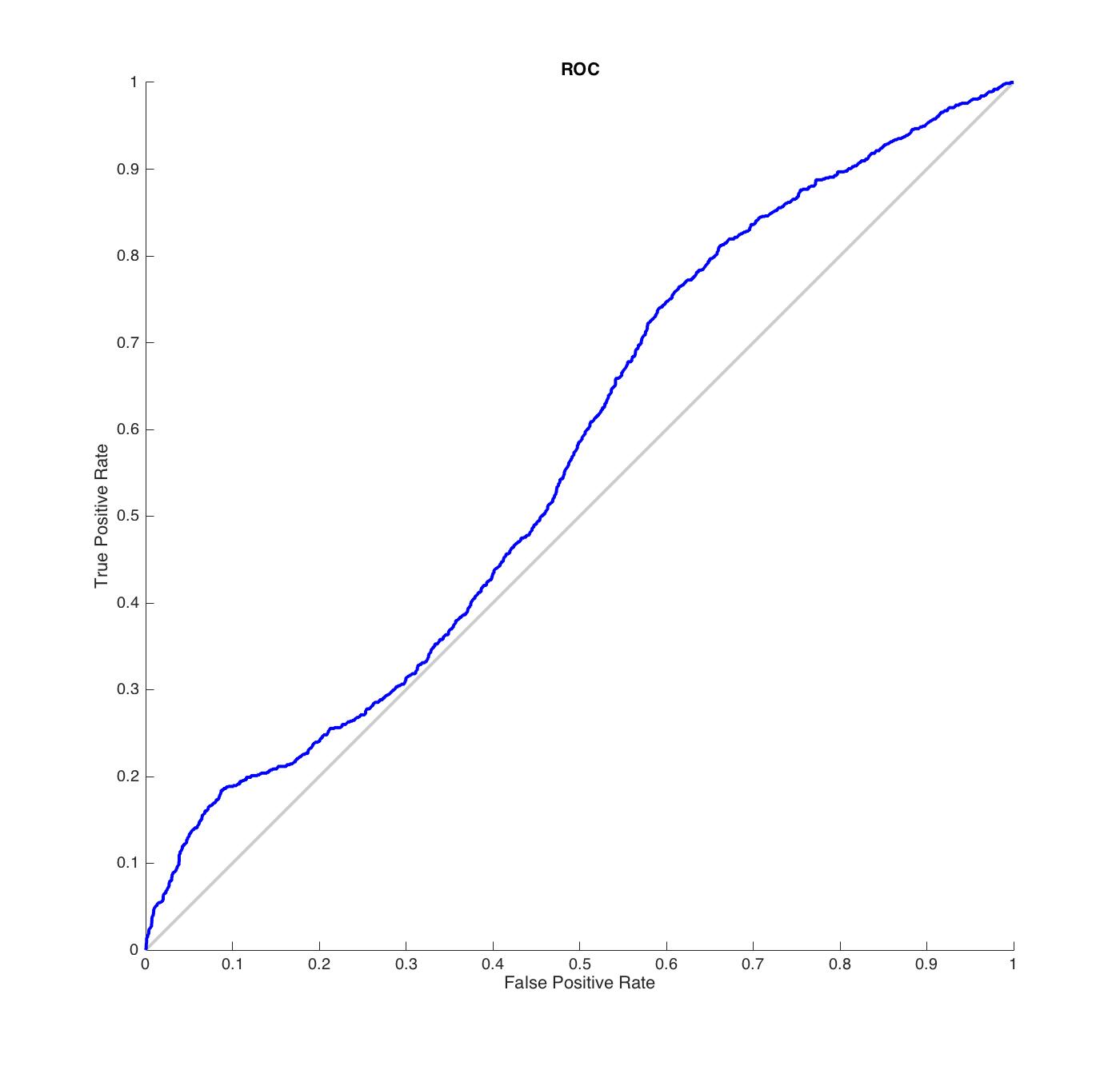
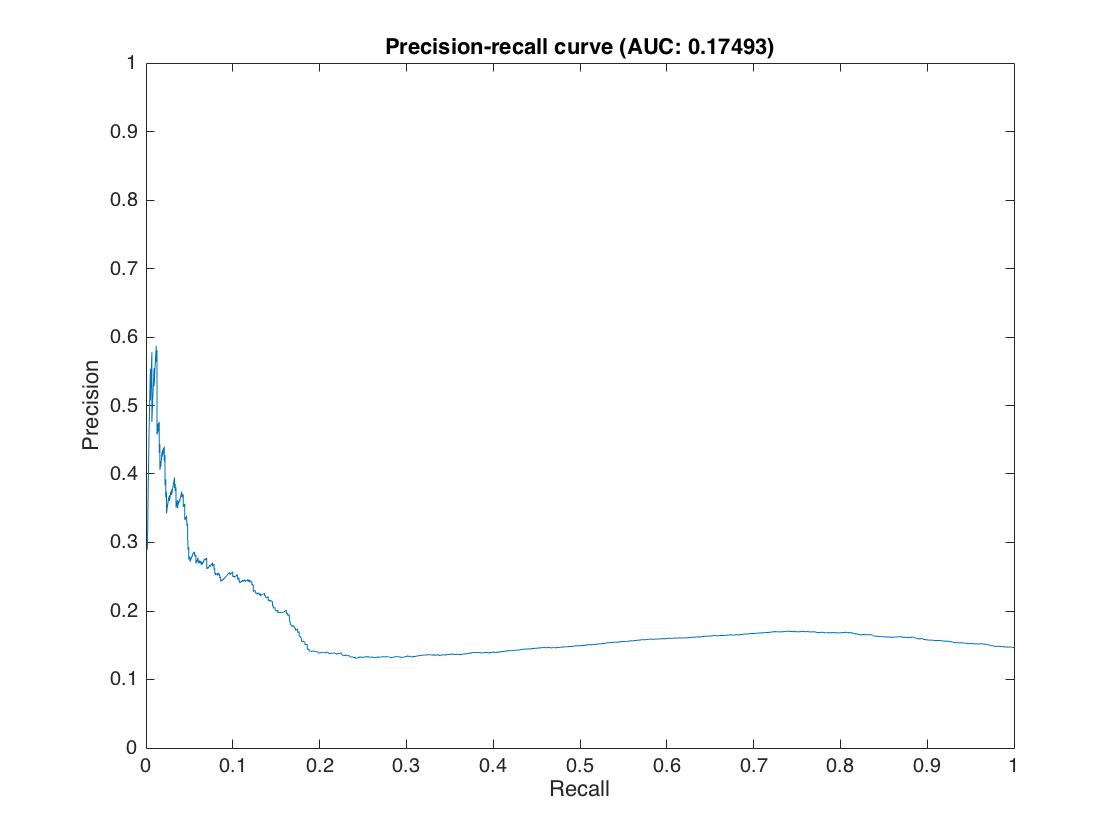
In tutti i casi emerge che è più conveniente utilizzare come reference color quello calcolato sull’intera immagine anziché quello mediano tra tutte le bande di una stessa direzione.

Il risultato migliore si ottiene utilizzando il modello addestrato sullo SplicedDSO con reference color globale. Questo era un risultato atteso poiché il test è effettuato su immagini dello stesso tipo (tutte però diverse) e, per come è composto il dataset, con generalmente le stesse condizioni di luce.

**Caso 1 – Metodo con addestramento SVM, global reference color**

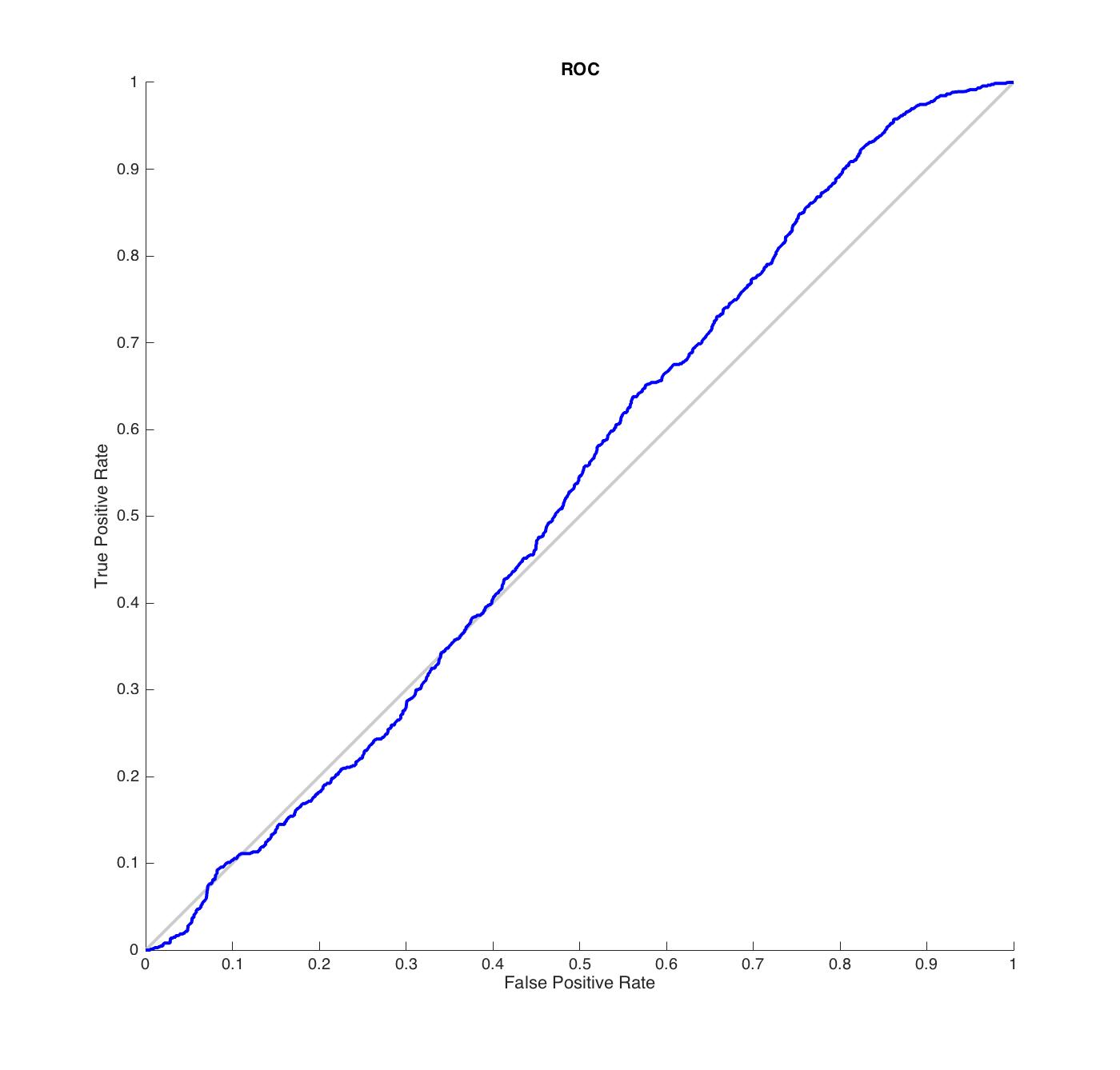
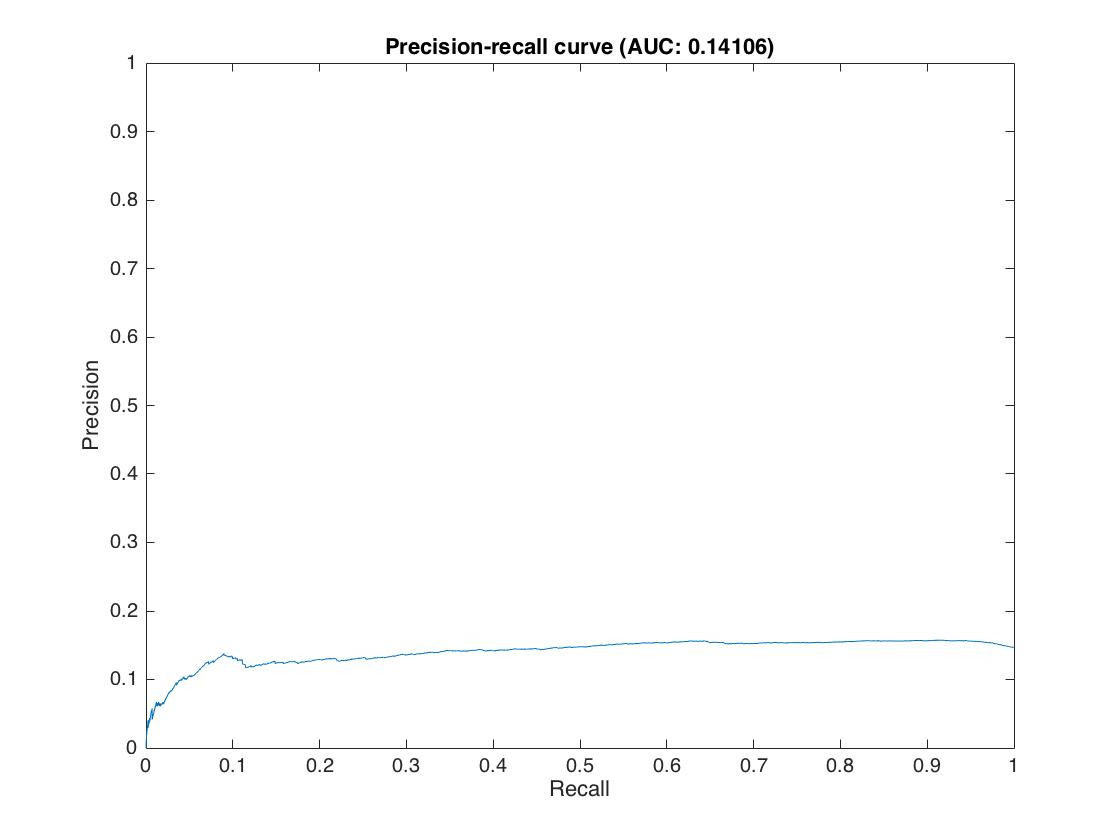
In questo caso, data un’immagine da analizzare, per ciascuna banda (orizzontale e verticale) questa è caratterizzata da un feature vector

dove è il valore della distanza tra il reference color di banda (valutato con l’algoritmo i-esimo) e il reference color globale valutato su tutta l’immagine (sempre tramite l’i-esimo algoritmo).

****

**Caso 2 – Metodo con addestramento SVM, median reference color**

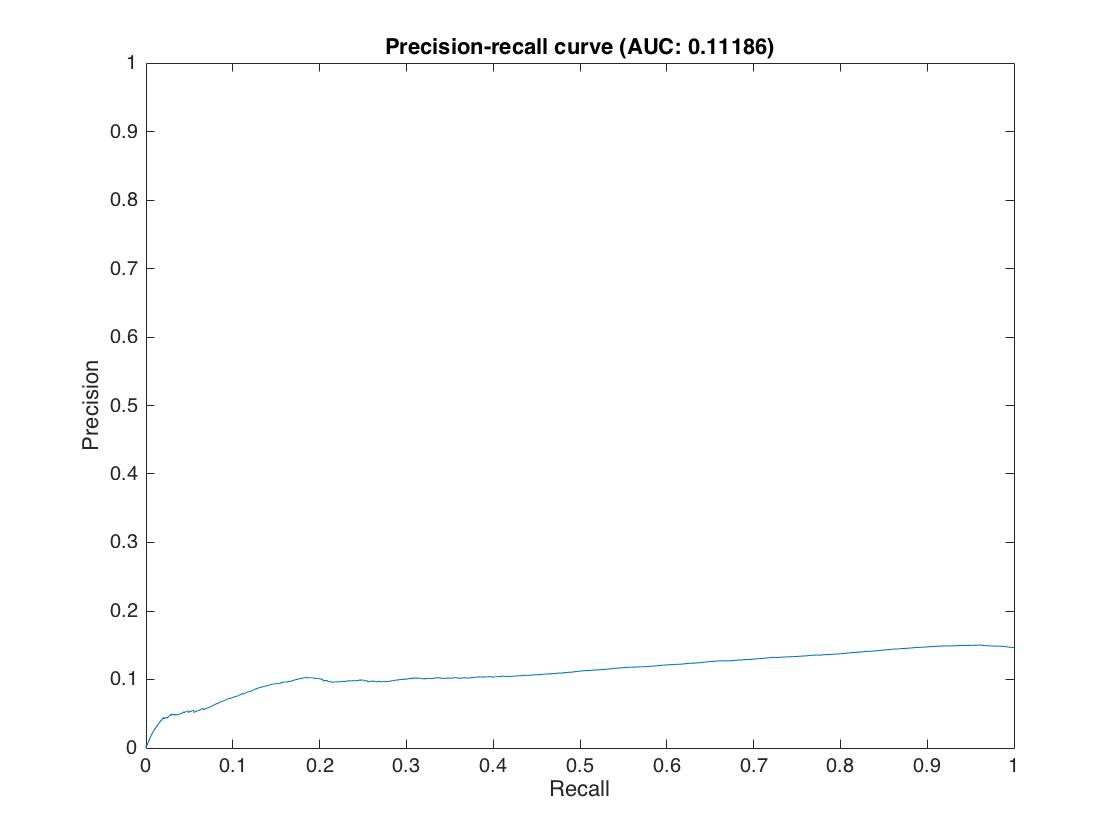
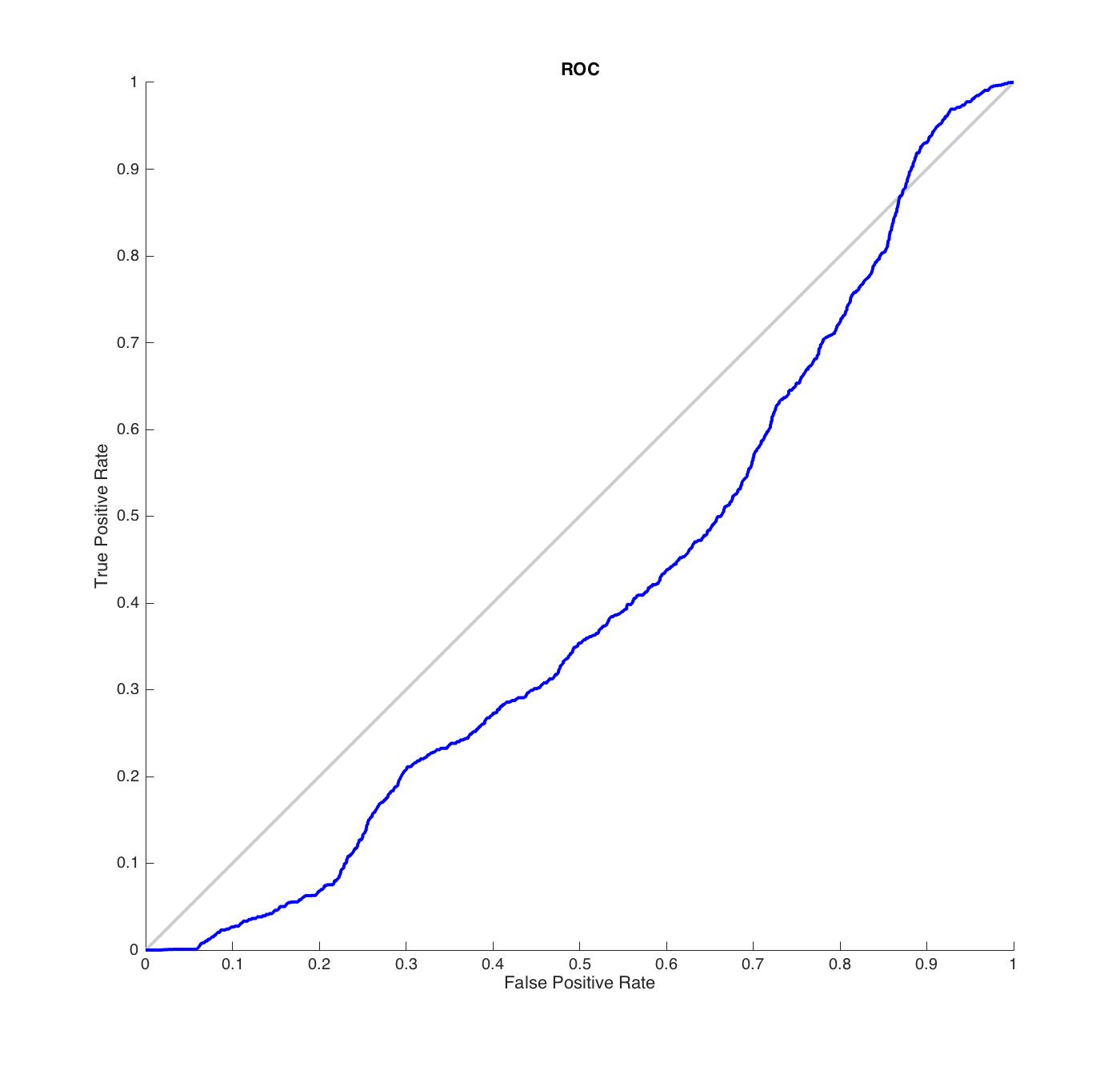
Come nel caso precedente, ma come reference color per il calcolo delle distanze viene considerato il valore della mediana tra tutti i reference color di banda per quella data direzione.

****

**Caso 3 – Metodo senza addestramento, global reference color**

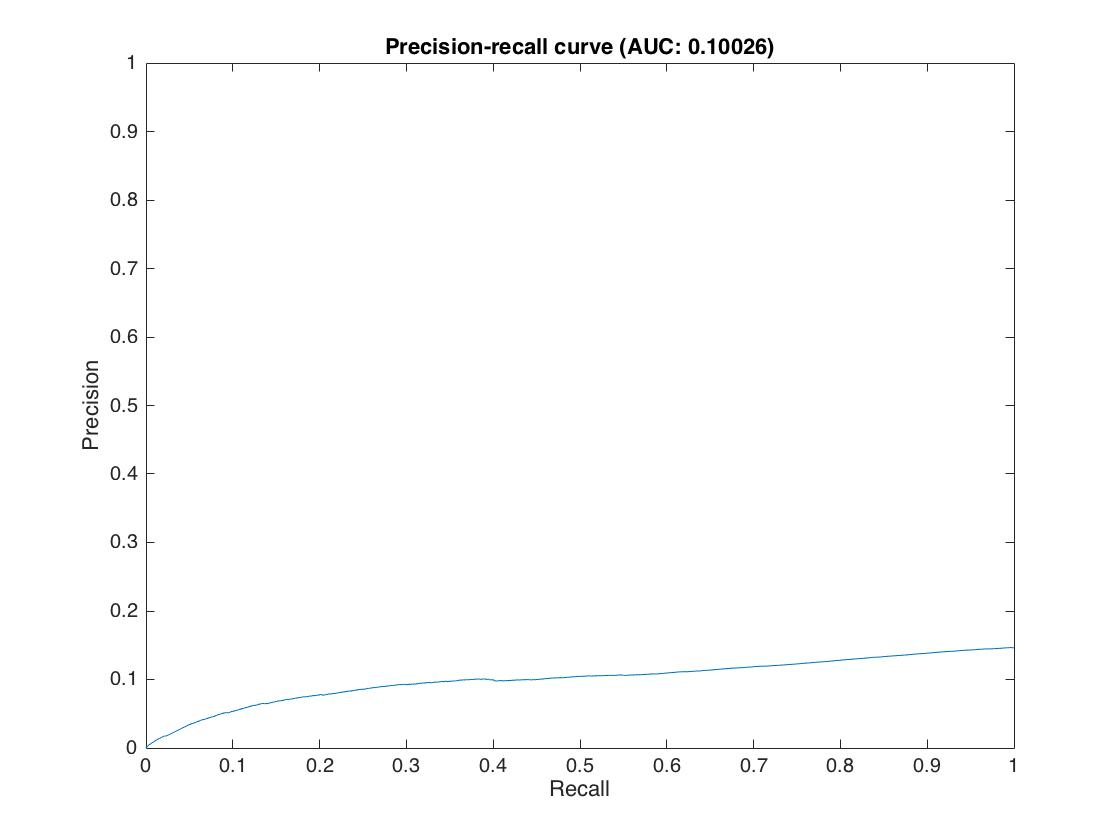
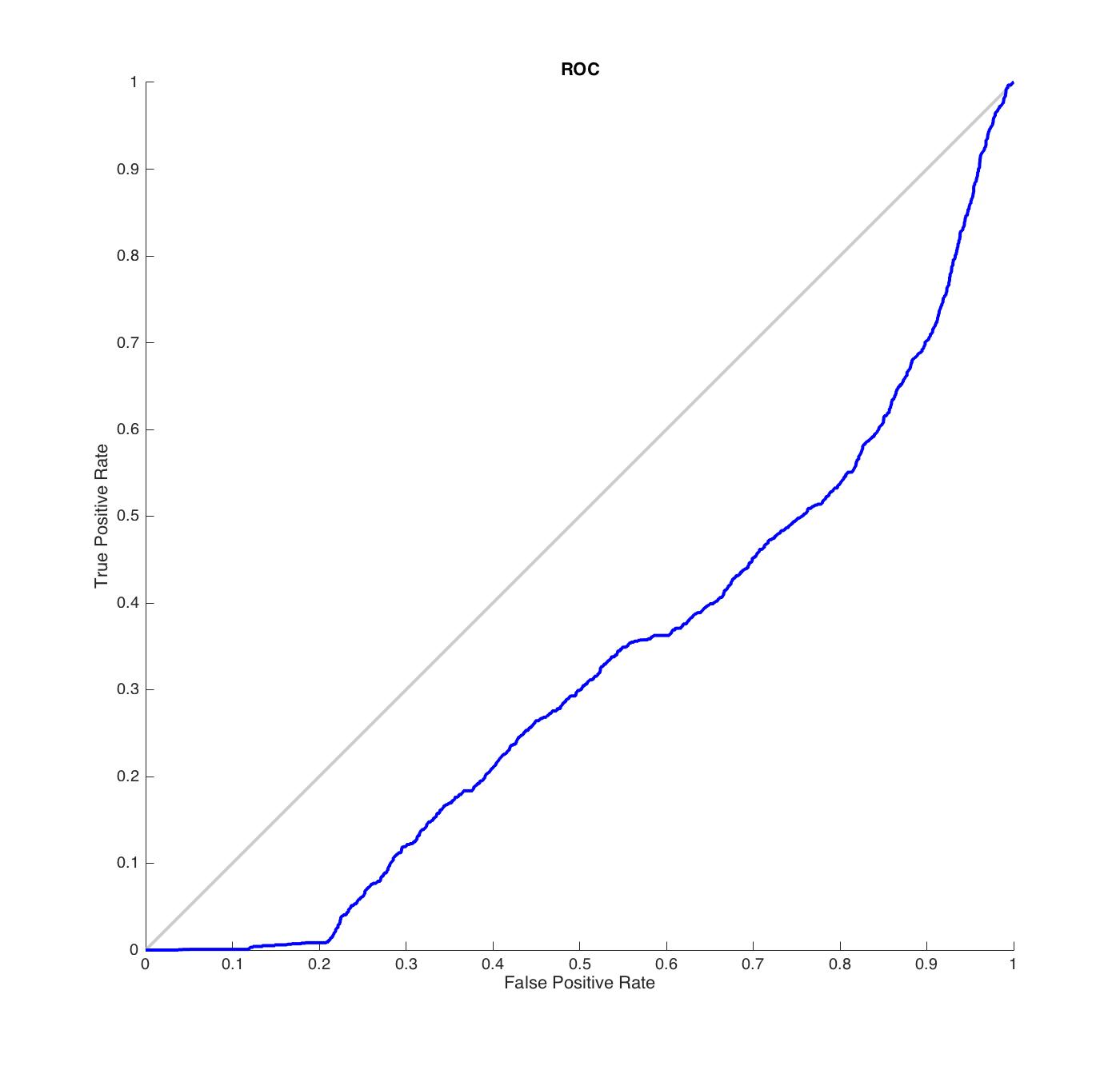
In questo caso, data un’immagine da analizzare, per ciascuna banda (orizzontale e verticale) viene calcolato un vettore di 5 distanze tra il reference color (colore dell’illuminante RGB) ed il colore di riferimento per quella direzione (illuminante stimato globalmente per l’intera immagine).

Sia il vettore di distanze calcolato, Per ciascun pixel

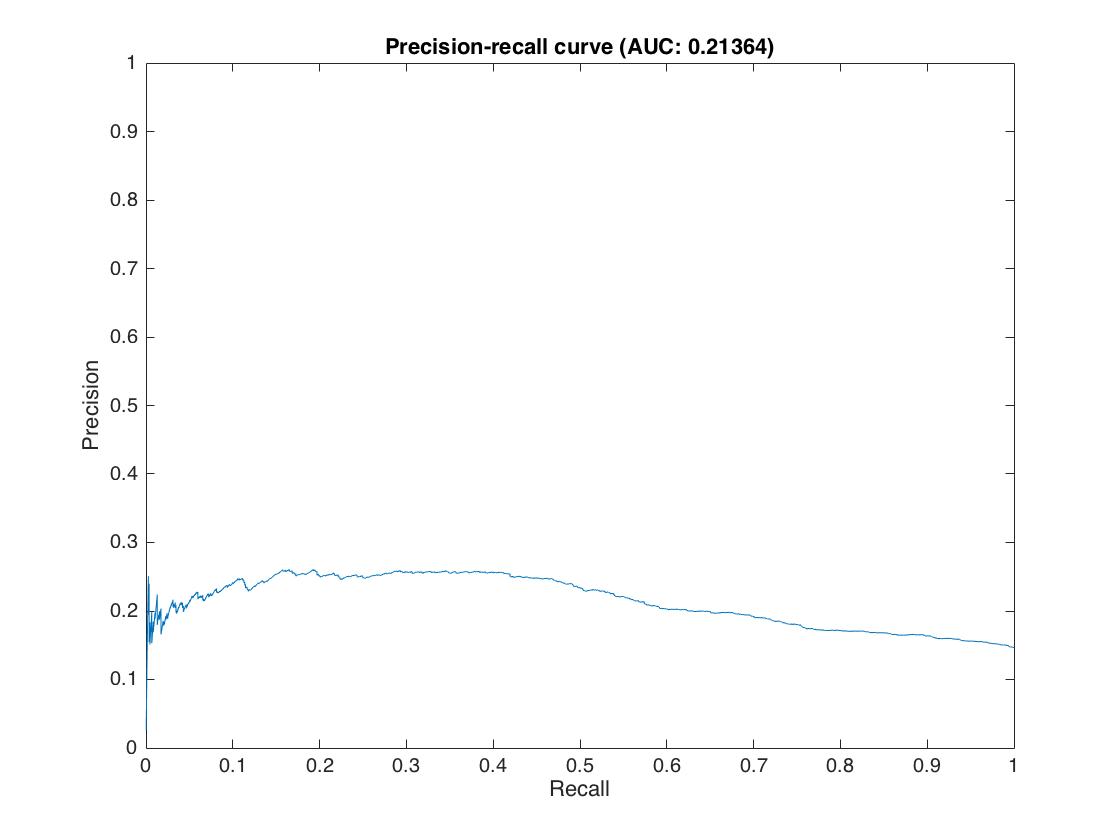
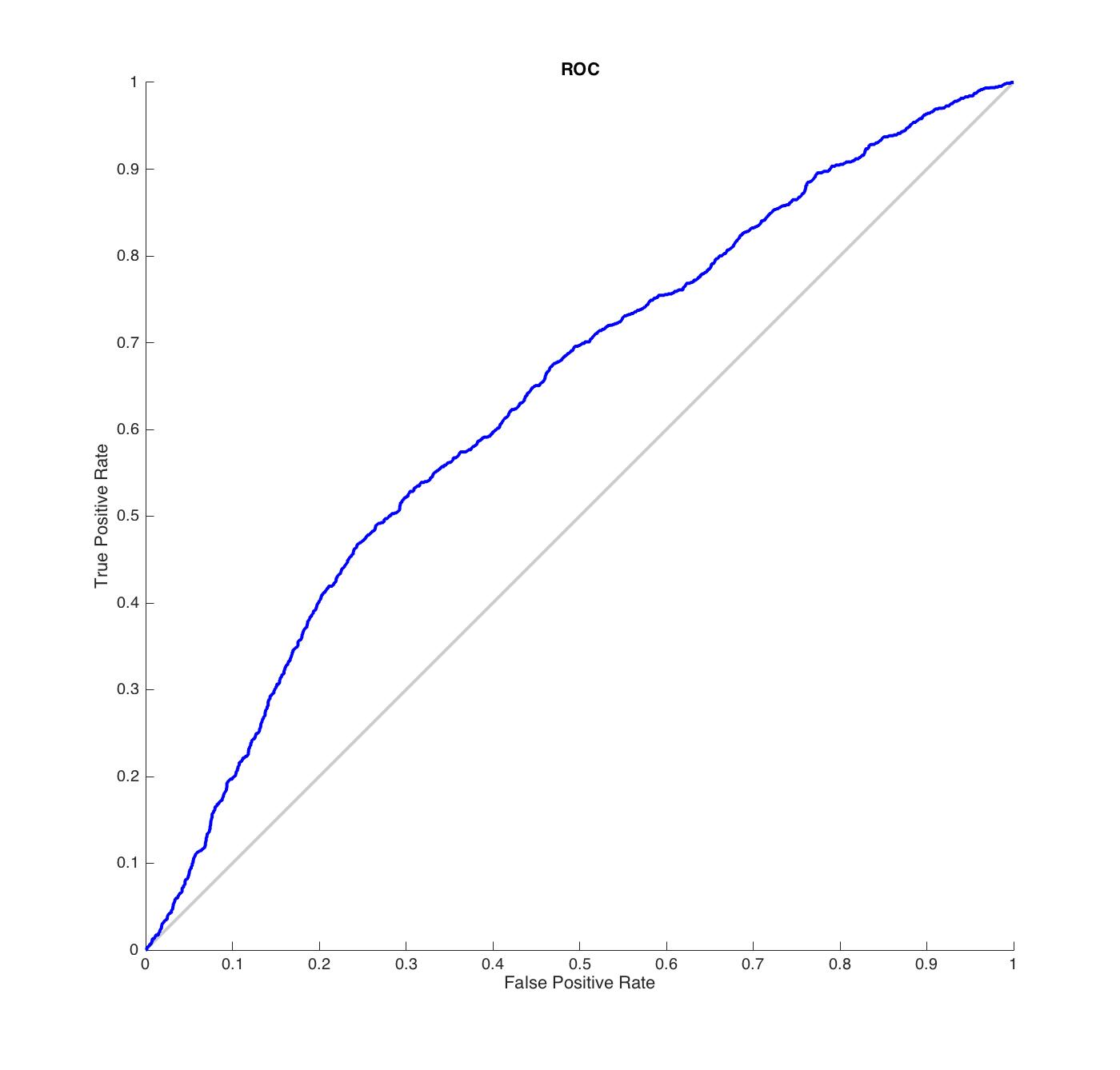


**Caso 4 – Metodo senza addestramento, median reference color**

Come nel caso precedente, ma come reference color per il calcolo delle distanze viene considerato il valore della mediana tra tutti i reference color di banda per quella data direzione.



**Caso 5 – Metodo con addestramento SVM, global reference color**



**Caso 6 – Metodo con addestramento SVM, median reference color**

