

The Invisible Man

Eleonora Cicalla

Michele Marcucci

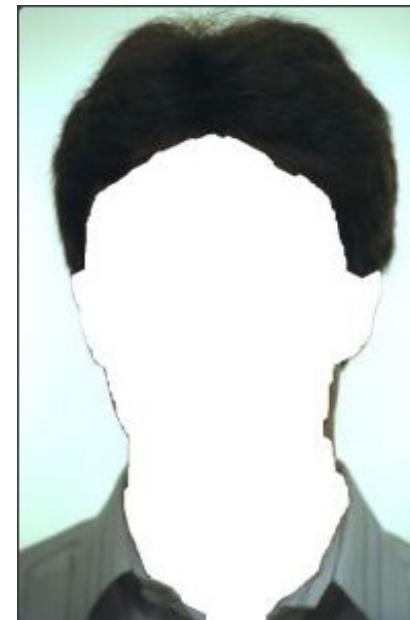
Mattia Napoli

Indice

- ▶ Obbiettivo del Progetto
- ▶ Analisi dei Dati
- ▶ Assunzioni
- ▶ Pipeline
- ▶ Preprocessing e Postprocessing
- ▶ Risultati
- ▶ Analisi Critica dei Risultati
- ▶ Contributi

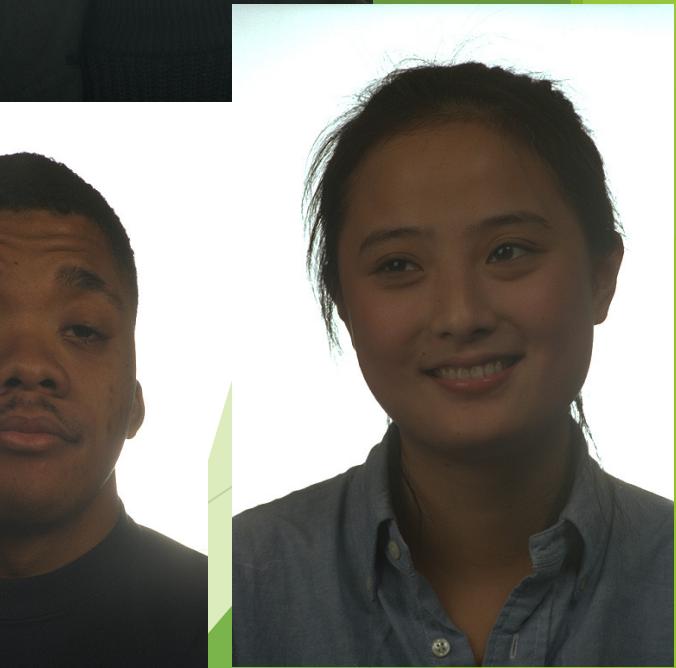
Obiettivo del Progetto

- ▶ Dato un video in input, sia live da webcam o su file, produrre un video dove al posto dei pixel di pelle viene sostituita un'immagine di background acquisita precedentemente



Analisi dei Dati

- ▶ Un primo approccio potrebbe essere quello di individuare manualmente dei range per il colore ma e' subito evidente che data la varieta' di tipologie di pelle e' una strada non percorribile.
- ▶ L'alternativa piu' strutturata e' quella di utilizzare un classificatore, un algoritmo in grado di catalogare automaticamente i dati in delle classi.
- ▶ Ai fini di questo progetto il classificatore in questione doveva soddisfare almeno i seguenti requisiti:
 - ▶ Rapidita' di predizione
 - ▶ Miss rate minimo
 - ▶ Accuratezza ragionevole (>85%)
 - ▶ Rapidita' di training (secondaria ma non troppo vista la necessita' di testare piu' combinazioni)



Dataset

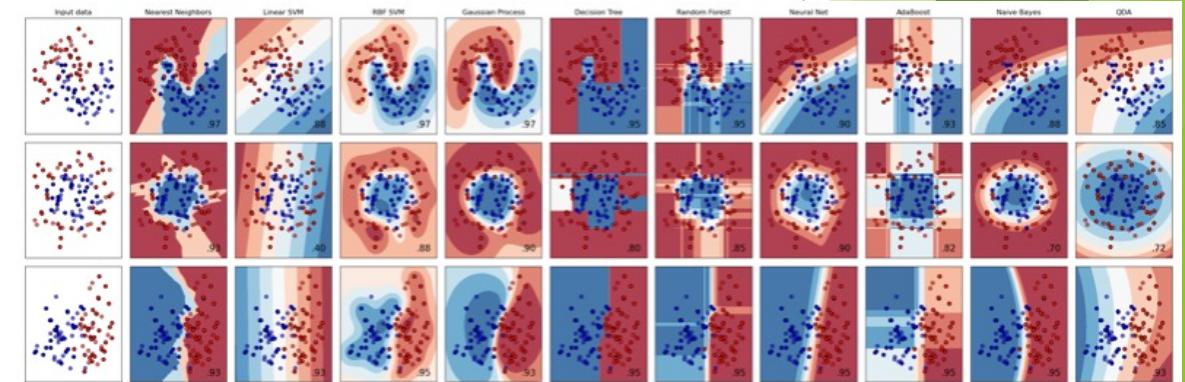
- ▶ I dataset utilizzati sono due, SFA e VDM, che risultano tra i più precisi e forniscono non solo dati di training ma anche immagini e maschere di ground truth utilizzate per testare l'efficacia dei classificatori generati.
- ▶ Dalle prove effettuate sembrerebbe che il dataset SFA sia più preciso del VDM perché, nel progetto, è il dataset sul quale viene fatto il training del classificatore, tuttavia risulta difficile stabilire a priori quale sia migliore e se ciò è valido in tutte le situazioni. Per completezza sono disponibili entrambi.

Table 2. Some of the most used datasets per skin detection

Name	Ref	Images	Ground truth	Download	Year
Compaq	(Jones & Rehg, 2002)	4675	Semi-supervised	ask to the authors	2002
TDSD	(Zhu et al., 2004)	555	Imprecise	http://lbmedia.ece.ucsb.edu/research/skin/skin.htm	2004
UChile	(Ruiz-Del-Solar & Verschae, 2004)	103	Medium Precision	http://agami.die.uchile.cl/skindiff/	2004
ECU	(Phung et al., 2005)	4000	Precise	http://www.uow.edu.au/~phung/download.html (currently not available)	2005
Schmugge	(Schmugge et al., 2007)	845	Precise {3 classes}	https://www.researchgate.net/publication/257620282_skin_image_Data_set_with_ground_truth	2007
Feeval	(Stöttinger et al., 2009)	8991	Low quality, imprecise	http://www.feeval.org/Datasets/Skin_Colors.html	2009
MCG	(Huang, Xia, Zhang, & Lin, 2011)	1000	Imprecise	http://mcg.ict.ac.cn/result_data_02mcg_skin.html (ask to authors)	2011
Pratheepan	(Tan et al., 2012)	78	Precise	http://web.fsktm.um.edu.my/~cschan/downloads_skin_dataset.html	2012
VDM	(Sanmiguel & Suja, 2013)	285	Precise	http://www-vou.eps.uam.es/publications/SkinDetDM/	2013
SFA	(Casati et al., 2013)	1118	Medium Precision	http://www1.sel.eesc.usp.br/sfa/	2013
HGR	(Kawulok, Kawulok, Nalepa, et al., 2014)	1558	Precise	http://sun.aei.polsl.pl/~mkawulok/gestures/	2014
SDD	(Mahmoodi et al., 2015)	21000	Precise	Not available	2015

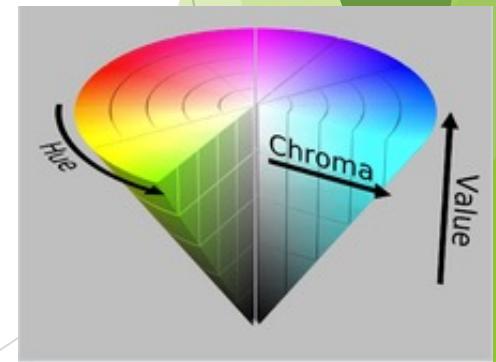
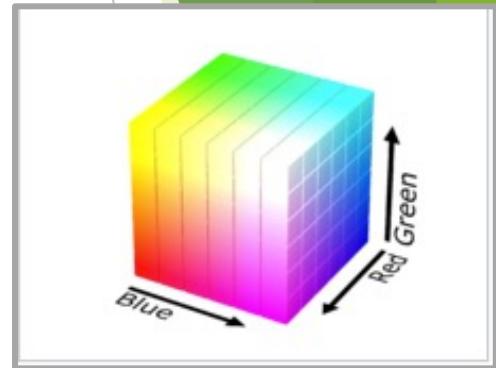
Algoritmo di Classificazione

- ▶ Scikit-learn fornisce algoritmi di classificazione con approcci differenti. Pertanto si trattava semplicemente di trovare il piu' adeguato al nostro problema. Tuttavia si e' realizzato che l'algoritmo influenza notevolmente i tempi di training e di predizione.
- ▶ Inizialmente usavamo il KNN ma sorta l'esigenza di testare velocemente nuove feature e dataset i tempi di computazione erano insostenibili.
- ▶ Dopo diversi tentativi abbiamo scelto il Baesiano che dava tempi di training di pochi secondi e predizioni quasi immediate. Naturalmente l'accuratezza rispetto al KNN era diminuita ma non al punto da non permetterci, tramite scelte accurate di features e processing, di produrre una pipeline efficace per classificare la pelle.



Features

- ▶ Trovato il dataset, il passo successivo e' l'individuazione delle feature da estrarre, sulle quali il classificatore effettuera' le sue scelte.
- ▶ Trattandosi di pelle la prima feature che viene in mente e' il colore. Tuttavia, dal momento che viene codificato attraverso uno spazio colore come l' RGB, HSV o LAB era necessario decidere quali fossero i canali piu' adatti da utilizzare.
- ▶ Altre feature possibili prese in considerazione sono state indicatori di texture di varia natura come LBP ed Haralick (che non e' stato poi possibile tradurre efficacemente in codice). Anche di questi difficile dire a priori l'utilita' anche se speravamo che permettessero in qualche modo di aiutare a discriminare la pelle dal resto.



Testing

- ▶ Vista la difficolta' di stabilire a priori quale classificatore e feature utilizzare la conseguenza piu' immediata e' stata quella di costruire un piccolo "framework" di testing delle varie combinazioni per stabilire quelle che rispettassero i requisiti.
- ▶ Selezionato un dataset e un classificatore si procede all'estrazione di tutte le features indicate, ne si creano tutte le diverse possibili combinazioni. Sulla base di queste si esegue il training di un classificatore e lo si testa sulla ground truth del dataset.
- ▶ Questo permette facilmente di soddisfare i requisiti di tempo. Gli altri due requisiti invece venivano soddisfatti semplicemente selezionando la combinazione di feature che dava i risultati migliori.

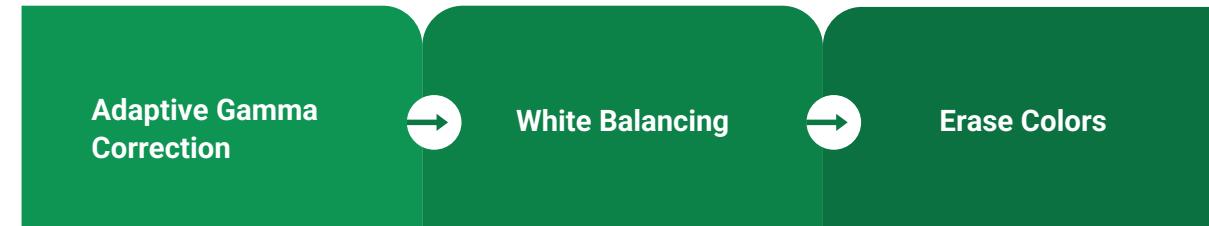
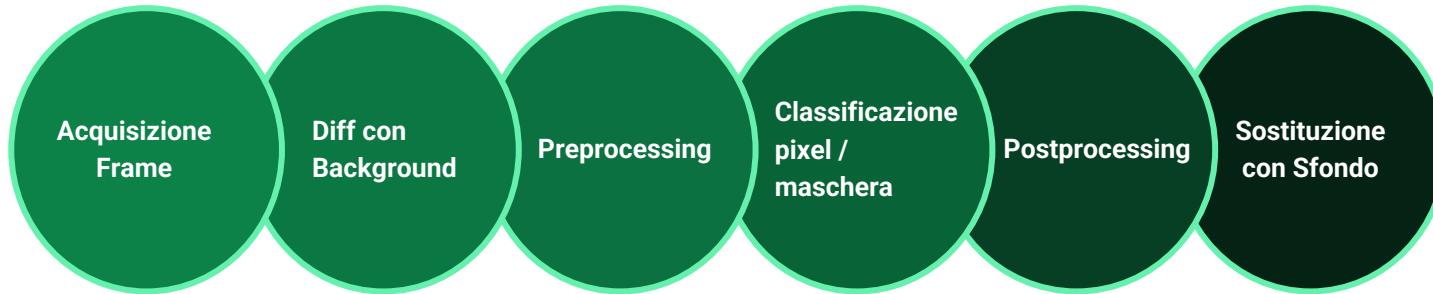
```
(env) mattia@Gecko SkinDetection % python src/tester.py
[16:58:34] Loading features and labels from cache..                                cache.py:27
[16:58:38] Testing feature combinations on                                         TESTER
[16:58:38] Pipeline(steps=[('standardscaler', StandardScaler()),                  tester.py:157
[16:58:38]     ('gaussiannb', GaussianNB())])
[16:58:38] Using <adv> dataset..                                              tester.py:158
[16:58:38] Total number of pixels: 2171436                                         tester.py:159
[16:58:38] skin/total ratio: 0.38                                              tester.py:160
[16:58:38] Dataset made of 2171436 samples                                         tester.py:161
[16:58:38] Threshold for analysis set at: 90.0 %
[16:58:38] Got 5 features and 31 combinations                                     tester.py:162
[16:58:50] Fetched 18 test and gt images from SFA                           tester.py:38
[16:58:52] Fetched 25 test and gt images from VDM                           tester.py:47
[16:58:52] Pre/Post Processing Setup
[16:58:53] Full Processing Suite                                         tester.py:169
-----
[16:58:53] SFA GTs Report
[16:58:53] Minimal mean miss rate found with:
[16:58:53] Accuracy: 0.90852. Features: ('G', 'H')
[16:58:53] accuracy => 0.9285 | 0.8772 | 0.9707
[16:58:53] precision => 0.8342 | 0.6302 | 0.9962
[16:58:53] miss rate => 0.0642 | 0.0019 | 0.2119
[16:58:53] Max mean accuracy found with:
[16:58:53] Accuracy: 0.93368. Features: ('G', 'H', 'CIEA')
[16:58:53] accuracy => 0.9326 | 0.8946 | 0.9747
[16:58:53] precision => 0.8814 | 0.6986 | 0.9979
[16:58:53] miss rate => 0.1111 | 0.0044 | 0.3016
[16:58:53] Max mean precision found with:
[16:58:53] Accuracy: 0.91333. Features: ('G', 'Cr')
[16:58:53] accuracy => 0.8441 | 0.6613 | 0.9867
[16:58:53] precision => 0.9592 | 0.8126 | 1.0000
[16:58:53] miss rate => 0.4670 | 0.0147 | 0.9374
[16:58:53] Top-5 other features and scores:
[16:58:53] Accuracy: 0.94019. Features: ('G', 'Cr', 'H', 'CIEA')
[16:58:53] accuracy => 0.9236 | 0.8648 | 0.9786
[16:58:53] precision => 0.9196 | 0.7628 | 0.9992
[16:58:53] miss rate => 0.1812 | 0.0065 | 0.4317
[16:58:53] Accuracy: 0.93678. Features: ('G', 'Cr', 'H', 'CIEA', 'CIEB')
[16:58:53] accuracy => 0.9095 | 0.8438 | 0.9801
[16:58:53] precision => 0.9362 | 0.7754 | 0.9994
[16:58:53] miss rate => 0.2420 | 0.0067 | 0.5416
[16:58:53] Accuracy: 0.93659. Features: ('G', 'Cr', 'H')
[16:58:53] accuracy => 0.9279 | 0.8684 | 0.9787
[16:58:53] precision => 0.8950 | 0.7520 | 0.9989
[16:58:53] miss rate => 0.1416 | 0.0032 | 0.3778
[16:58:53] Accuracy: 0.93494. Features: ('G', 'H', 'CIEA', 'CIEB')
[16:58:53] accuracy => 0.9183 | 0.8627 | 0.9787
[16:58:53] precision => 0.9215 | 0.7786 | 0.9994
[16:58:53] miss rate => 0.2003 | 0.0046 | 0.4679
```

Statistiche

- ▶ La presenza di immagini e maschere di ground truth permette l'estrazione di una serie di parametri fondamentali per definire un classificatore, tra questi ad esempio accuratezza, precisione, miss rate, sensibilità'
- ▶ Abbiamo scelto di focalizzarci su Accuratezza, Precisione e Miss Rate con la logica che il nostro classificatore dovesse essere abbastanza accurato cercando di mantenere la percentuale di pelle mancata il più bassa possibile (nonostante la perdita di precisione)

```
SFA GTs Report
[-----] Minimal mean miss rate found with:
Accuracy: 0.90852. Features: ('G', 'H')
accuracy => 0.9285 | 0.8772 | 0.9707
precision => 0.8342 | 0.6302 | 0.9962
miss rate => 0.0642 | 0.0019 | 0.2119
[-----] Max mean accuracy found with:
Accuracy: 0.93368. Features: ('G', 'H', 'CIEA')
accuracy => 0.9326 | 0.8946 | 0.9747
precision => 0.8814 | 0.6986 | 0.9979
miss rate => 0.1111 | 0.0044 | 0.3016
[-----] Max mean precision found with:
Accuracy: 0.91333. Features: ('G', 'Cr')
accuracy => 0.8441 | 0.6613 | 0.9867
precision => 0.9592 | 0.8126 | 1.0000
[miss rate => 0.4670 | 0.0147 | 0.9374]
```

Pipeline



Assunzioni

- ▶ Sfondo statico (niente cambi di inquadratura)
- ▶ Soggetti a distanza di almeno 2/3 metri dal dispositivo di acquisizione e non troppo lontani (oltre i 6/7 metri)
- ▶ Qualita' del video maggiore di 480p
- ▶ Se live acquisire lo sfondo prima di testarlo, se su video nei primi frame ci deve essere solo sfondo

Differenza con il Background

- ▶ In molti test si e' notato che la maschera fatta attraverso il classificatore rileva anche una parte di pixel "falsi positivi" nello sfondo.
- ▶ Questo e' causato da ombre, rumore, artefatti o alterazioni dei pixel che la fase di preprocessing non riesce ad eliminare.
- ▶ Si e' quindi aggiunta una funzione che fa una grossolana individuazione dei pixel "nuovi" rispetto allo sfondo tramite una differenza con il background (che viene poi sogliata e sottoposta a dilate). Il risultato e' una "premaschera", che scontorna tutto il foreground.



Senza premaschera

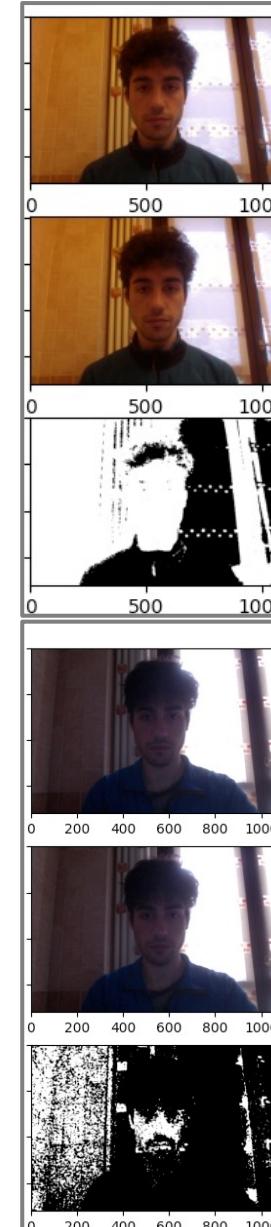


Con premaschera

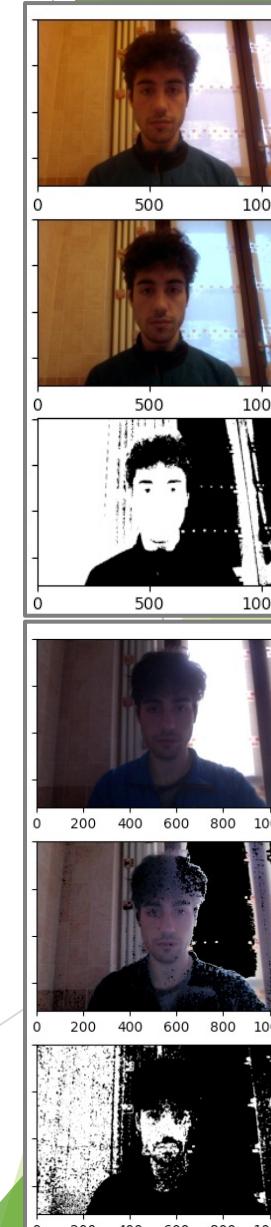
Preprocessing

- ▶ Adaptive Gamma Correction:
 - ▶ Implementata seguendo le indicazioni di un paper, classifica le immagini in “chiare” o “scure” per poi applicare una specifica correzione
- ▶ White Balance:
 - ▶ Implementa il bilanciamento del bianco
- ▶ Erase Colors:
 - ▶ Il classificatore ha la tendenza a catalogare come pelle colori troppo accesi e saturi come il bianco, il rosso, il giallo e l’arancio; questa funzione mitiga gli errori andando ad eliminare preventivamente certi range di colore

No preprocessing

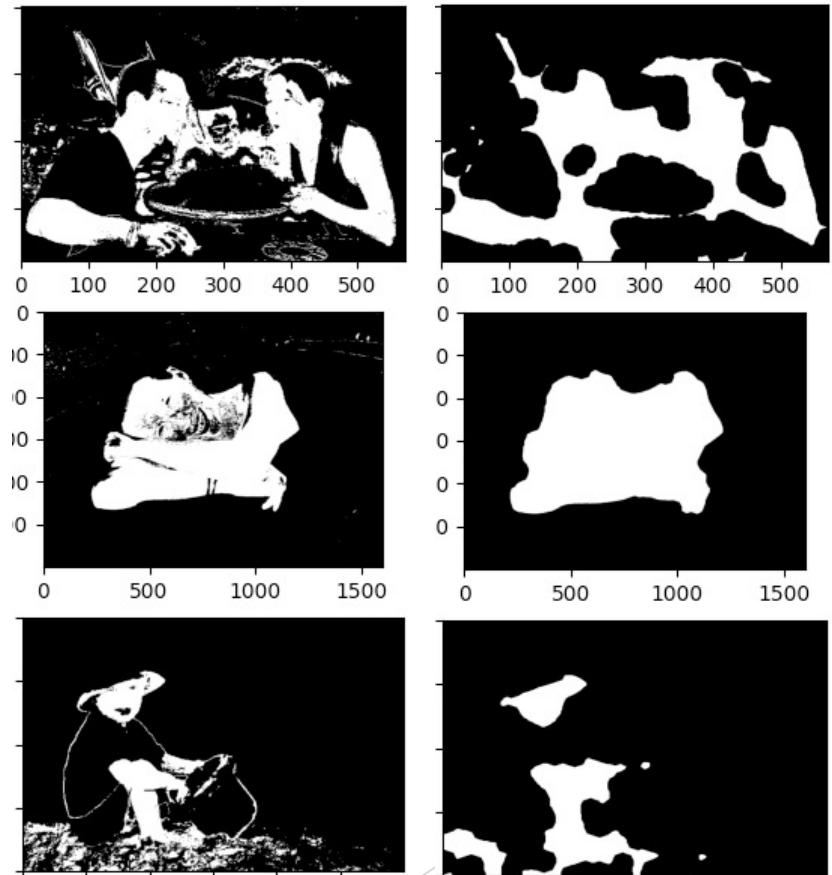


Preprocessing



Postprocessing

- ▶ Remove Mask Noise
 - ▶ Operazione di opening molto leggera per rimuovere eventuali pixel di rumore lasciati dal classificatore
- ▶ Adjust Mask:
 - ▶ Operazione di closing aggressiva per rimuovere i buchi formati da occhi, sopracciglia etc..
- ▶ Remove Contour:
 - ▶ Semplice dilate di poche iterazioni sulla maschera di pelle per ovviare alla formazione del bordo in seguito sostituzione con il background



Scelta Dataset e Features

- ▶ Per la scelta definitiva si e' usato il classificatore Baesiano sul dataset SFA, dopo aver filtrato le combinazioni di feature che davano uno score >90% le si e' testate sulla ground truth fornita nel dataset
 - ▶ Tra i risultati si e' cercata la combinazione che desse:
 - ▶ Minimo Miss Rate: Green, Hue
 - ▶ Massima Accuracy: Green, Hue, CIEA
 - ▶ Massima Precisione: Green, Cr
 - ▶ Alla fine si e' optato per quella che dava massima Accuracy poiche' era la piu' bilanciata le alternative sacrificavano troppo miss rate o accuratezza

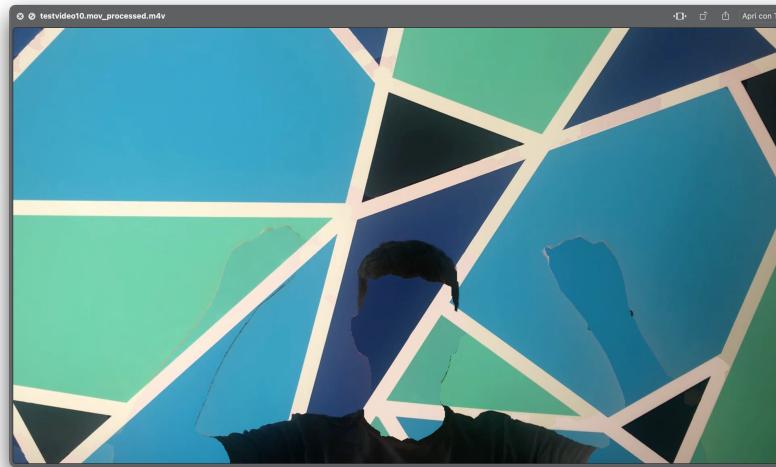
```
(env) mattia@Gecko SkinDetection % python src/tester.py
[10:07:05] Loading features and labels from cache.. cache.py:27
TESTER
[10:07:07] Testing feature combinations on tester.py:157
    Pipeline(steps=[('standardscaler', StandardScaler()),
                     ('gaussiannb', GaussianNB())])
        Using <adv> dataset..
        Total number of pixels: 2171436
        skin/total ratio: 0.38
        Dataset made of 2171436 samples
        Threshold for analysis set at: 90.0 %
        Got 5 features and 31 combinations
[10:07:17] Fetched 18 test and gt images from SFA
[10:07:19] Fetched 25 test and gt images from VDM
Pre/Post Processing Setup
Full Processing Suite tester.py:169

----- SFA GTs Report -----
Minimal mean miss rate found with:
Score: 0.90852. Features: ('G', 'H')
accuracy => 0.9285 | 0.8772 | 0.9707
precision => 0.8342 | 0.6302 | 0.9962
miss rate => 0.0642 | 0.0019 | 0.2119
Max mean accuracy found with:
Score: 0.93368. Features: ('G', 'H', 'CIEA')
accuracy => 0.9326 | 0.8946 | 0.9747
precision => 0.8814 | 0.6986 | 0.9979
miss rate => 0.1111 | 0.0044 | 0.3016
Max mean precision found with:
Score: 0.91333. Features: ('G', 'Cr')
accuracy => 0.8441 | 0.6613 | 0.9867
precision => 0.9592 | 0.8126 | 1.0000
miss rate => 0.4670 | 0.0147 | 0.9374
Top-5 other features and scores:
Score: 0.94019. Features: ('G', 'Cr', 'H', 'CIEA')
accuracy => 0.9236 | 0.8648 | 0.9786
precision => 0.9196 | 0.7628 | 0.9992
miss rate => 0.1812 | 0.0065 | 0.4317
Score: 0.93678. Features: ('G', 'Cr', 'H', 'CIEA', 'CIEB')
accuracy => 0.9095 | 0.8438 | 0.9801
precision => 0.9362 | 0.7754 | 0.9994
miss rate => 0.2420 | 0.0067 | 0.5416
Score: 0.93659. Features: ('G', 'Cr', 'H')
accuracy => 0.9279 | 0.8684 | 0.9787
precision => 0.8950 | 0.7520 | 0.9989
miss rate => 0.1416 | 0.0032 | 0.3778
Score: 0.93494. Features: ('G', 'H', 'CIEA', 'CIEB')
accuracy => 0.9183 | 0.8627 | 0.9787
precision => 0.9215 | 0.7706 | 0.9994
miss rate => 0.2003 | 0.0046 | 0.4679
```

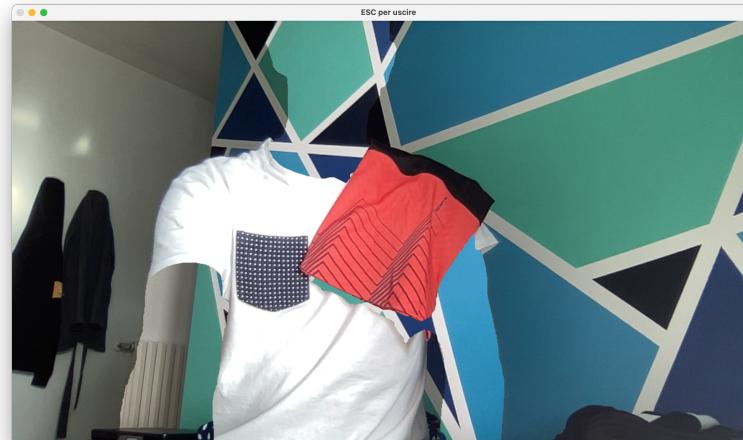
Risultati



Immagine rumorosa



Corretto funzionamento



Non prende il rosso

Risultati



Abbondante illuminazione



Ombra su pelle



Senza e Con erase_colors

Imprecisioni ed Errori



VDM



SFA



Pelle non rimossa e Colore rimosso



Capelli rimossi

Contributi

- ▶ Eleonora
 - ▶ Funzioni di preprocessing
 - ▶ Funzioni di postprocessing
 - ▶ Acquisizione feature LBP
- ▶ Michele
 - ▶ Acquisizione video: cam, singlethread e multithread + Script di setup
 - ▶ Funzioni di pre e post processing
 - ▶ Diff con background e processing del frame
- ▶ Mattia
 - ▶ Classificatore
 - ▶ Acquisizione e manipolazione dataset
 - ▶ Tester