

Content Based Image Retrieval su abiti

Progetto VIPM - anno accademico 20/21

- Christian Bernasconi - 816423
- Gabriele Ferrario - 817518
- Riccardo Pozzi - 807857
- Marco Ripamonti - 806785

Appello 18-01-2021

Dataset



140K



30K

Dataset

- giubbetti
- giacche eleganti
- pantaloni lunghi
- pantaloni corti
- scarpe
- scarpe col tacco
- felpe
- t-shirts
- borse



30K

Obiettivi - Classificazione



Obiettivi - CBIR



Here some similar clothes

11:25

Obiettivi - Applicazione stile e CBIR



=

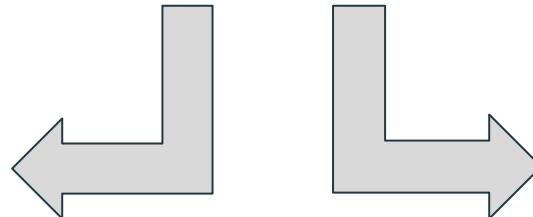


Here some similar clothes to the one generated!

11:45

Obiettivi - Applicazione effetti

Disegno

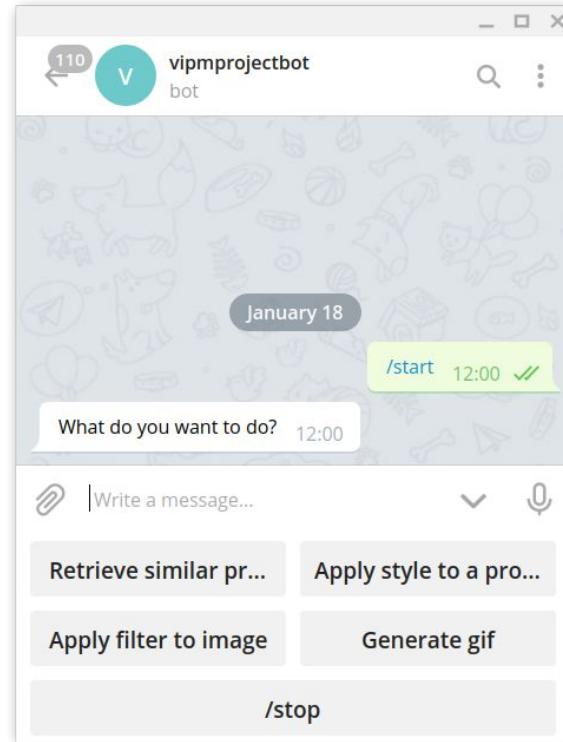


Mosaico



Telegram BOT

- implementazione **python** [1]
- interattivo
- database **chiave valore** [2] per gestire gli stati dei diversi utenti

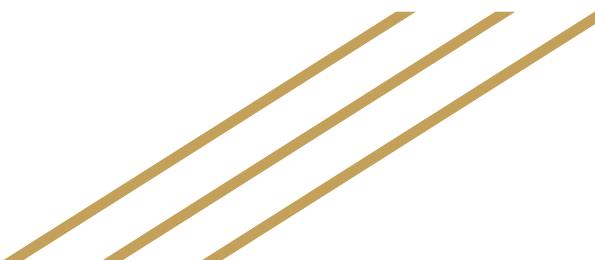


[1] <https://github.com/dros1986/python-bot>

[2] <https://pythonhosted.org/pickleDB/index.html>



Controllo input



Controllo input - Immagini sfocate

Primo approccio:

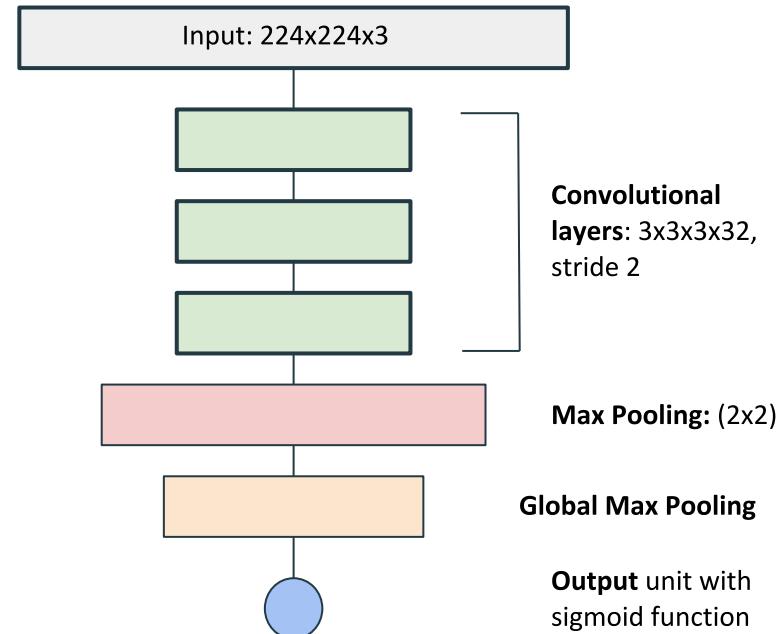
- filtro laplaciano e controllo su varianza output
- non sempre efficace

Secondo approccio:

- rete convoluzionale
- classificazione foto sfocate e nitide

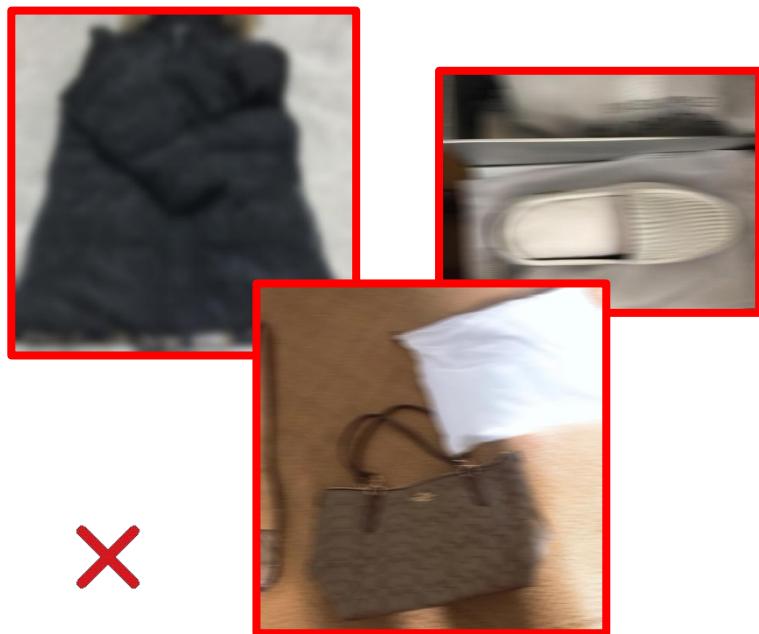
Controllo input - Immagini sfocate

- dataset con immagini reali sfocate/mosse e nitide [3].
- **augmentation** con filtro gaussiano e filtro per immagini mosse.
- **accuracy** su test set (20% dei dati): 0.89



[3] <https://www.kaggle.com/kwentar/blur-dataset>

Controllo input - Immagini sfocate

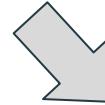


Controllo input - Immagini scure

Immagine input



YCbCr



Controllo skewness

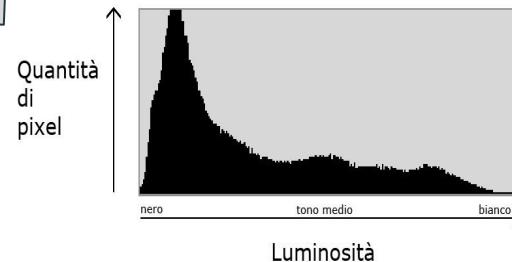


Immagine output



Adaptive Gamma
Correction



Controllo input - Immagini scure

$$Output = 255 * \left(\frac{Input}{255} \right)^{2 \left(\frac{128 - Mask}{128} \right)}$$

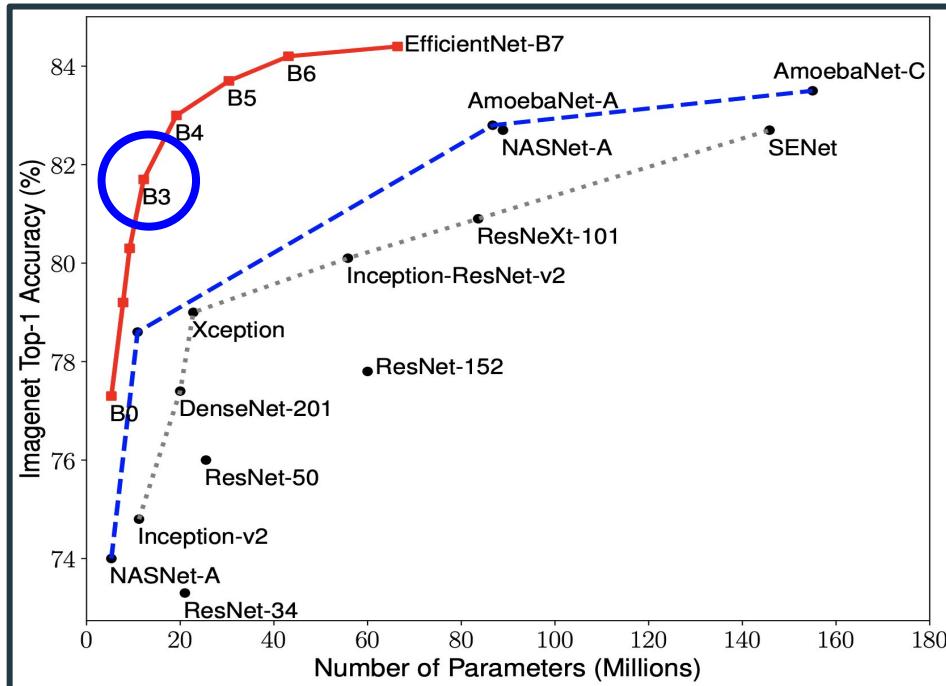
[4]

- **Maschera:** input grayscale invertito + filtro bilaterale
- **Base** dell'esponente variabile in base alla media dell'intensità (**$2 - 0.8 * (media/threshold)$**)

[4] Moroney, Nathan. "Local color correction using non-linear masking." Color and Imaging Conference. Vol. 2000. No. 1. Society for Imaging Science and Technology, 2000.

Controllo input - Classificazione

- CNN: *EfficientNet-B3* (pretrained su *ImageNet*)



Controllo input - Classificazione

- **CNN:** *EfficientNet-B3* (pretrained su *ImageNet*)
- allenamento **ultimo dense layer** congelando il resto della rete data la presenza di vestiti nel train originale
- **classi considerate:** 9 categorie precedenti + 1 unknown



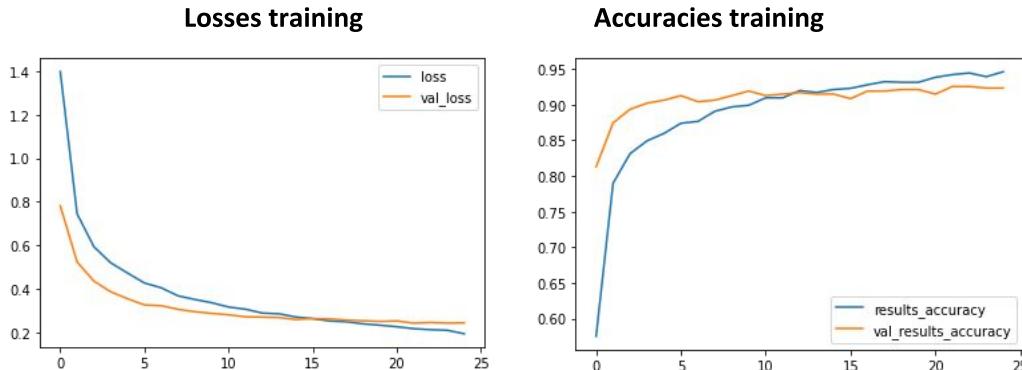
Controllo input - Classificazione

Training:

- sottoinsieme di circa 150-300 immagini per ogni classe
- **data augmentation**
 - rotazione random (fino a 360°)
 - capovolgimento orizzontale
 - capovolgimento verticale
 - rumore gaussiano
 - blur tramite filtro gaussiano

Controllo input - Classificazione

- training, validation(16%) e test set (20%)

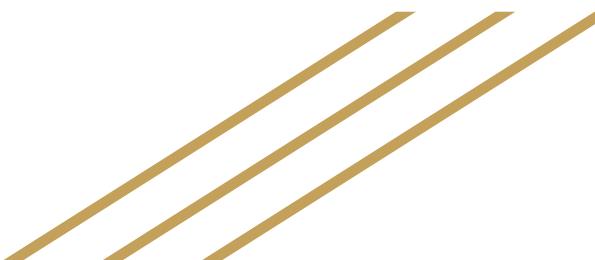


Evaluation on test:

- **Accuracy:** 92.5 %
- **F1-Score:** 91.6 %



Ricerca similarità

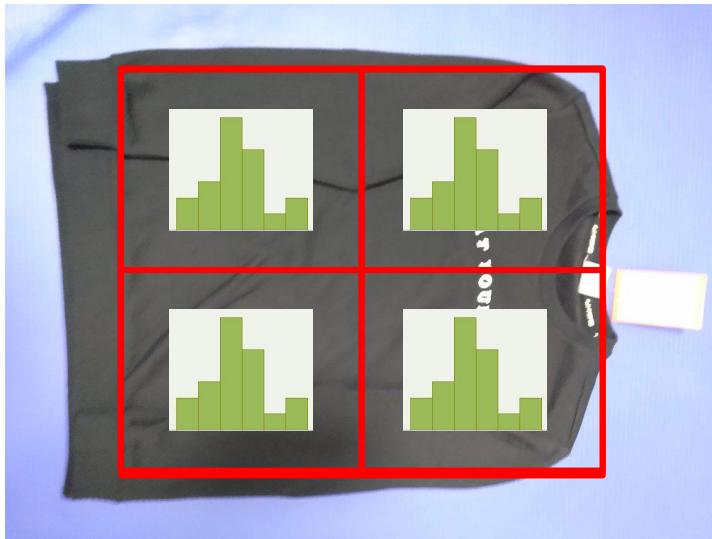


Ricerca similarità - Feature extraction

Estrazione feature secondo **diversi obiettivi** di ricerca di similarità:

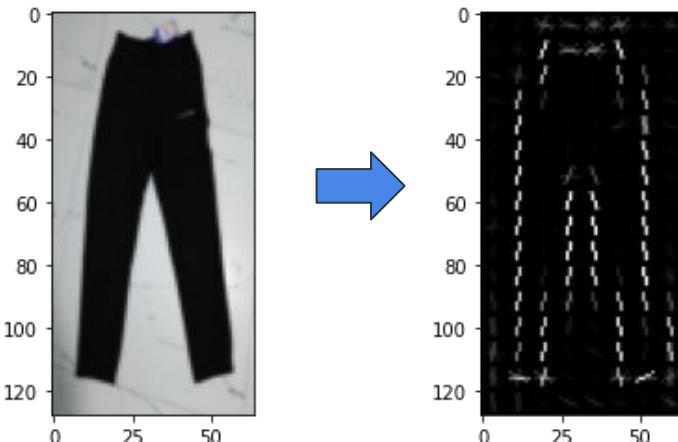
- feature **colore**: Istogramma colore
- feature **shape**: Histogram of oriented Gradients (HoG)
- feature **neurali**: [EfficientNet](#), ResNet50

Ricerca similarità - Colore



- **HSV** per poter privilegiare colore rispetto a luminosità
- divisione immagine in 4 sottoregioni centrali
- bins per canale:
 - 16 Hue
 - 18 Saturation
 - 2 per Brightness.
- $|features| = 2304$

Ricerca similarità - Shape



Histogram of Oriented Gradients (HOG)

- ridimensionamento immagine (64x128)
- istogrammi gradienti
- $|features| = 3780$

[5] DALAL, Navneet; TRIGGS, Bill. Histograms of oriented gradients for human detection. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. IEEE, 2005. p. 886-893.

Ricerca similarità - Feature neurali

CNN pretrained per ricerca semantica:

- **EfficientNet-B3**
- **ResNet50**

Feature estratte dell'ultimo global max pooling layer:

- $|\text{EfficientNet features}| = 1536$
- $|\text{ResNet features}| = 2048$

Ricerca similarità - Ulteriori features

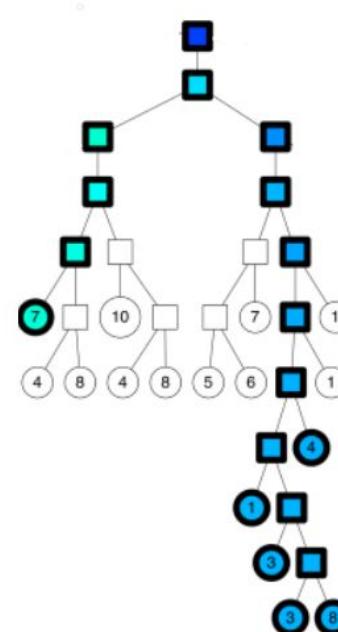
- **Bag of Visual Words:**
 - costruzione vocabolario con **SIFT** e k-means
- **Local binary patterns** (uniform): applicata alla parte centrale dell'immagine

Hanno portato **scarsi risultati**; non sono stati inclusi nell'applicazione.

Ricerca similarità - Indicizzazione e Retrieval

Indicizzazione immagini tramite **Annoy**[6]
(Approximate Nearest Neighbors Oh Yeah):

- **ricerca approssimata** basata su foresta di alberi binari
- ottime performance con features < 100
- buone performance anche con features dell'ordine delle migliaia



[6] <https://github.com/spotify/annoy>

Ricerca similarità - Indicizzazione e Retrieval

Creazione indici features immagini:

- **PCA** per ridurre dimensionalità delle features di HOG e reti neurali
- **cosine distance** per features neurali
- **euclidean distance** per features su istogrammi

Ricerca similarità - Valutazione CBIR

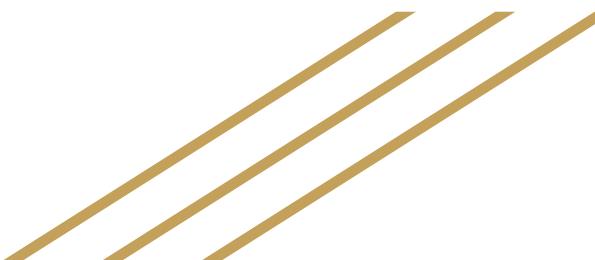
- conoscenza **incompleta** della rilevanza
- metriche che approssimano average precision: **bpref**, **bpref-10**, **indAP** [7]
- 20 immagini query

	m_bpref	m_bpref10	m_indAP
RESNET	0.6580	0.6815	0.6387
EFFICIENTNET	0.8215	0.8358	0.8073
COLOR	0.4745	0.4948	0.4476
SHAPE	0.4710	0.4880	0.4456

[7] Yilmaz, Emine & Aslam, Javed. (2006). Estimating average precision with incomplete and imperfect judgments. International Conference on Information and Knowledge Management, Proceedings. 102-111. 10.1145/1183614.1183633.



Applicazione stile



Applicazione stile - Controllo sfondo



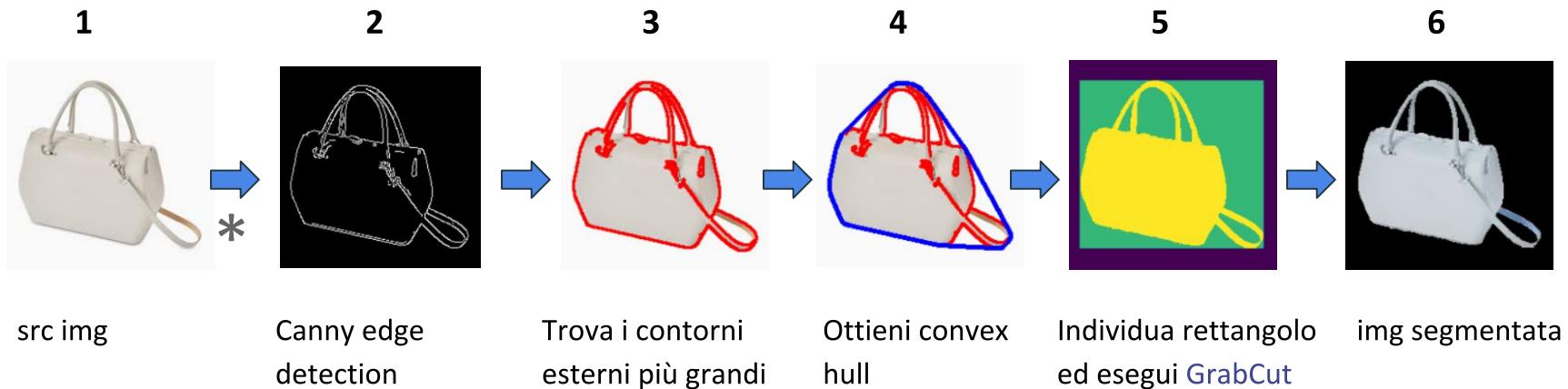
✗



✓

- applicazione **filtro bilaterale**
- estrazione **cornice** immagine
- edge detection sulla cornice tramite algoritmo **Canny**

Applicazione stile - Segmentazione automatica



* se l'immagine fornita in input non presenta sfondo uniforme (immagine da catalogo), viene effettuato uno step aggiuntivo di filtro bilaterale per rendere più efficace la segmentazione

Applicazione stile - Creazione nuovo capo

src img 1



estrai dettagli



crea maschera
colorata su parti più
chiare ($x > 235$)



effettua somma pesata su
maschera colorata (0.4) e
stile (0.6)



src img 2



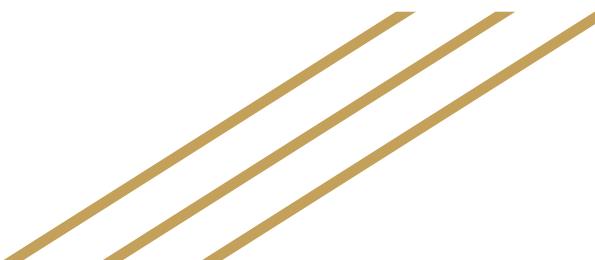
Applicazione stile - Retrieval

Alcuni esempi di retrieval per features neurali sui nuovi capi:



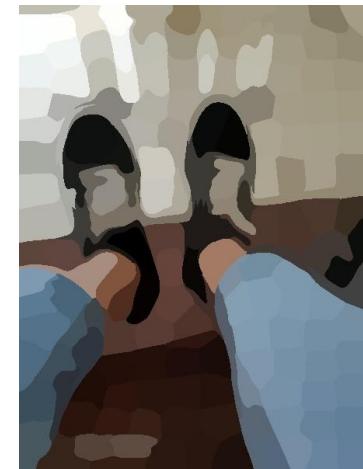


Applicazione effetti



Applicazione effetto - Effetto mosaico

Segmentazione in **superpixel** con Simple Linear Iterative Clustering (**SLIC**).



Applicazione effetto - Effetto disegno



Gray +
Gauss



Gray +
Gauss +
Invert



Edges

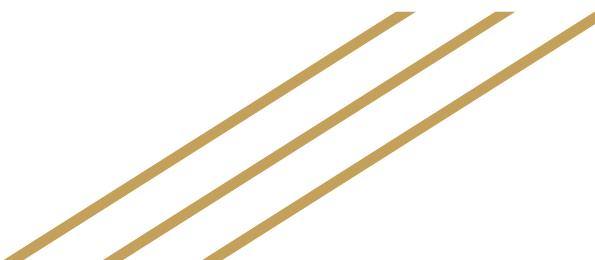


Combine



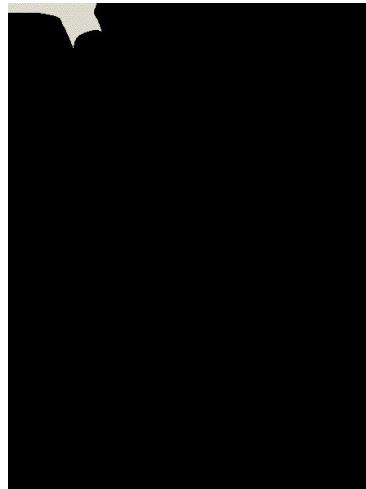


Generazione GIF



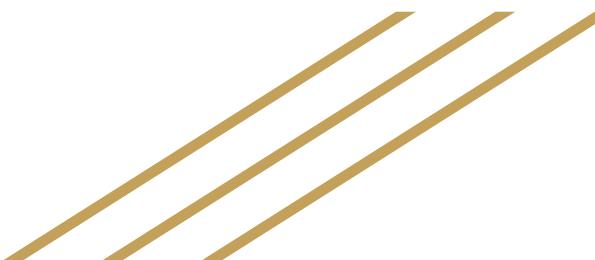
Applicazione effetto - Effetto mosaico

Creazione sequenza di frames andando a riempire i superpixels di un'immagine.





Sviluppi futuri



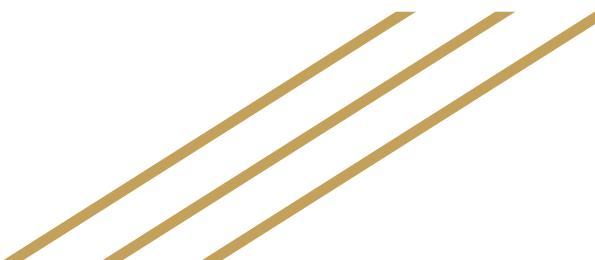
Sviluppi futuri

- miglioramento **blur detection**
- **combinazione features** per rendere più robuste singole feature
- valutazione del **neural style transfer** in sostituzione all'attuale applicazione di stile
- **object identification** dei singoli vestiti indossati da una persona per restituire outfit simili



Grazie per l'attenzione





Slide extra

Filtro Gaussiano - Filtro Bilaterale

Filtro Gaussiano

- Filtro passa basso
- I coefficienti del kernel campionati dal prodotto di due **distribuzioni Gaussiane**
 - diminuiscono all'aumentare della distanza dal centro del kernel
- Deviazione standard e grandezza del filtro influiscono sull'effetto di blur
- Blur globale, dettagli non preservati

Filtro Bilaterale

- Rimpiazza l'intensità di ogni pixel con la media pesata del loro vicinato
- I pesi possono essere campionati con una distribuzione Gaussiana
- I pesi dipendono, oltre che dalla distanza tra pixel, anche dalle **differenze tra intensità colore o profondità**
- Permette di preservare edge

Filtro Laplaciano

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Il filtro laplaciano è un **edge detector**.

Permette di computare le **derivate seconde** di un'immagine andando a misurare la velocità con cui le derivate cambiano.

Questo filtro permette di evidenziare quelle zone a rapido cambiamento di intensità. Se la **varianza** dell'output di una convoluzione è elevata significa che sono presenti sia zone che identificano edge che non edge.

Skewness

La skewness permette di misurare l'asimmetria di una distribuzione di probabilità. La skewness può essere calcolata in diversi modi. La misura adottata in questo progetto è la **Skewness di moda di Pearson**:

$$\frac{\text{mean} - \text{mode}}{\text{standard deviation}}$$

Applicando questo concetto alla **distribuzione della luminosità** di un'immagine possiamo capire che:

- se **skewness > 0**: distribuzione right skewed → immagine **Low Key**
- se **skewness < 0**: distribuzione left skewed → immagine **High Key**

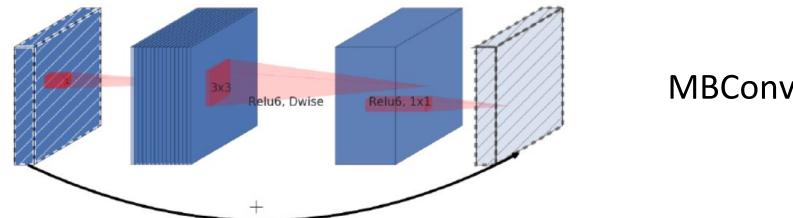
ResNet

Reti con un elevato numero di layer forniscono elevate performance, ma la profondità rende il task di training arduo a causa di problemi di Vanishing Gradient. I layer dei livelli più bassi rischiano di non essere aggiornati correttamente durante l'algoritmo di Back Propagation.

Le **ResNet** sono state introdotte nel 2015 e cercano di ovviare a questo problema. ResNet introduce il residual mapping. ResNet utilizza i layer per il training di una funzione residua.

EfficientNet

- Nuova famiglia di reti
- Pubblicata nel 2019 da **Google**
- Questa famiglia di reti permette di incrementare le prestazioni in termini di accuracy e allo stesso tempo cerca di **ridurre** i parametri e le Floating Point Operation Per Seconds (**FLOPS**)
- L'obiettivo è offrire una rete con elevate prestazioni utilizzabile anche su dispositivi limitati in termini di risorse



Back

Scale Invariant Features Transform (SIFT)

- identifica **keypoints** e ne descrive feature locali
- invariante a **rotazione, traslazione, scala, illuminazione e punto di vista**
- **feature locali**: resistenti a occlusioni e rumore
- computazionalmente **efficiente**
- **discriminante**: oggetti diversi rappresentati in modo molto diverso

Bag of Visual Words

Bag of Words

- Prende spunto dal concetto di **BOW** nel contesto del NLP.
- Rappresentare un documento in base alla **frequenza di parole** che compaiono in un **vocabolario**

Bag of Visual Words

- Costruire un **vocabolario** utilizzando un gruppo di immagini significative. (**SIFT + clustering**)
- Rappresentare un'immagine in base alla frequenza con cui una parola (centroide di un cluster) compare nelle vicinanze dei keypoints estratti dall'immagine

Back

Sobel

- Operatore spaziale 2D che permette di misurare i gradienti sulla luminosità di un'immagine andando ad enfatizzare regioni ad elevata frequenza spaziale, gli edge.

-1	0	+1
-2	0	+2
-1	0	+1

Edge orizzontali

+1	+2	+1
0	0	0
-1	-2	-1

Edge verticali

Principal Component Analysis (PCA)

Tecnica di **riduzione della dimensionalità** dello spazio delle features.

- Mantenere le variabili più importanti che descrivono la maggior parte dei dati
- Assicurare che le variabili siano tra di loro indipendenti
- Perdita di alcune delle informazioni originali
- **Curse of Dimensionality!**
 - Modelli in overfitting
 - Clustering più difficile
 - Misure di similarità meno precise

Simple Linear Iterative Clustering (SLIC)

Algoritmo che permette di generare superpixel andando a clusterizzare i pixel in base alla loro similarità colore e prossimità.

- Clustering nello spazio dimensionale **labxy**
 - lab → spazio colore
 - xy → posizione pixel
- 1. Inizializza K cluster campionando pixel su una griglia regolare e muovendoli verso le zone con gradiente più basso in un vicinato 3x3.
- 2. Ogni pixel è associato ad uno di questi cluster.
- 3. Ricalcolo centroidi in base al vettore labxy medio dei pixel in ogni cluster
- 4. Ripetere fino a che la distanza tra vecchi e nuovi centri è relativamente piccola

Annoy - funzionamento

Costruzione:

- viene costruito un insieme di **alberi binari**
- ogni albero è costruito facendo uno **split** con iperpiano equidistante da due punti **random** nello spazio delle features
- **ricorsivamente** si ripete su ogni sottoregione ottenuta

Query:

- **ricerca** dei punti candidati in tutti gli alberi binari della foresta
- al raggiungimento del numero prefissato di candidati **rimuove duplicati**
- calcola **distanze** dei candidati e **riordina**
- restituisce i più vicini alla query

Canny

Algoritmo di edge detection che si compone delle seguenti fasi:

1. applicazione **filtro gaussiano** per **riduzione rumore**
2. ricerca **gradiente luminosità** per individuare contorni orizzontali/verticali/diagonali utilizzando 4 filtri
3. **rimozione** dei punti **non massimi** locali
4. i punti restanti vengono confrontati con **due soglie**:
 - valori inferiori alla soglia bassa vengono scartati
 - valori superiori alla soglia alta vengono mantenuti
 - valori intermedi vengono mantenuti solo se contigui a punti già accettati

Segmentazione GrabCut

Dopo aver individuato i contorni tramite ricerca sull'immagine binaria degli edges [10] ed individuato il rettangolo contenente il convex hull, viene eseguito l'algoritmo **GrabCut**[11] per segmentare l'immagine:

1. parte da **bounding box** individuata dal rettangolo: ciò che sta fuori è considerato come background certo
2. stima la distribuzione del **colore** di **BG/FG** tramite **Gaussian Mixture Model**
3. costruisce **Markov Random Fields** sulle labels BG/FG dei pixels: vengono favorite regioni connesse aventi stessa label
4. applica ottimizzazione **graph cut** per raggiungere segmentazione ottimale

[10] Suzuki, Satoshi. "Topological structural analysis of digitized binary images by border following." *Computer vision, graphics, and image processing* 30.1 (1985): 32-46.

[11] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "" GrabCut" interactive foreground extraction using iterated graph cuts." *ACM transactions on graphics (TOG)* 23.3 (2004): 309-314.

Histogram of Oriented Gradients (HoG)

HOG è un descrittore di caratteristiche usato in computer vision ed in elaborazione delle immagini per il riconoscimento di oggetti.

Algoritmo:

- l'immagine viene divisa in celle
- in ogni cella per ogni pixel dell'immagine viene calcolato il gradiente nella direzione x e nella direzione y
- per ogni cella viene calcolato un istogramma dei gradienti orientati
- gli istogrammi vengono normalizzati

Vantaggi: invariante alle trasformazioni geometriche ma non all'orientamento degli oggetti

Local Binary Patterns (LBP)

LBP è un descrittore di caratteristiche usato in computer vision ed in elaborazione delle immagini per la classificazione delle texture.

Algoritmo:

- per ogni pixel viene selezionato un vicinato e viene calcolato un array binario
- la codifica del vettore ottenuto corrisponde al nuovo valore assegnato al pixel centrale nella matrice LBP
- partendo dalla matrice LBP viene calcolato un istogramma rappresentante la codifica dell'immagine

Nota: l'uniformità fornisce un ulteriore livello di rotazione e invarianza della scala di grigi

Metriche CBIR

Considerando una query con R risultati si definiscono:

$$\text{bpref} = \frac{1}{R} \sum_r \left(1 - \frac{\text{number of } n \text{ above } r}{R} \right)$$

$$\text{bpref-10} = \frac{1}{R} \sum_r \left(1 - \frac{\text{number of } n \text{ above } r}{10 + R} \right)$$

$$\text{indAP} = \frac{1}{R} \sum_r \left(1 - \frac{\text{number of } n \text{ above } r}{\text{rank}(r)} \right)$$

dove n indica un risultato non rilevante, r un risultato rilevante e rank(r) la posizione di r.