Visual Processing and Information Management

Gabriele Fumagalli (845108)
Veronica Morelli (839257)
Davide Valoti (846737)

FoodX-251:

A Dataset for Fine-grained Food Classification

- Kaur, P., et al. "Foodx-251: A dataset for fine-grained food classification. arXiv 2019." arXiv preprint arXiv:1907.06167.
- Immagini di cibo appartenenti a 251 classi
- Il dataset è suddiviso in:
  - training set: 118475 immagini
  - validation set: 11994 immagini
  - validation set degradato: 11994 immagini

Campionamento

- Abbiamo creato un account *colab pro* per avere una GPU più veloce e più spazio in memoria RAM.
- Abbiamo creato tre account su Azure Machine Learning studio per poter avere una GPU più veloce.

Nonostante ciò è stato necessario effettuare dei campionamenti.

	Immagini Train per classe	Immagini Validation per classe	Immagini totali Train	Immagini totali Validation
Campionamento 1	100	20	25034	4979
Campionamento 2	30	9	7530	2252
Campionamento 3	20	5	5020	1252

NB: Il numero di immagini per classe non è equidistribuito.

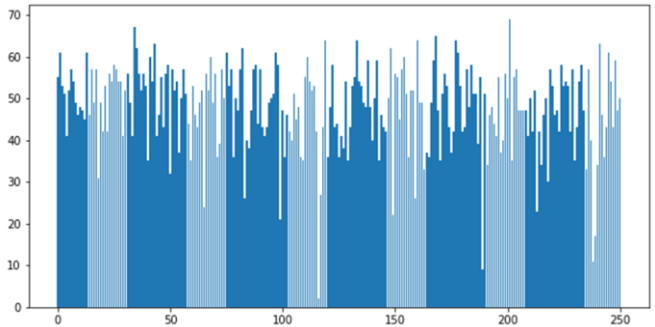
Approccio utilizzato

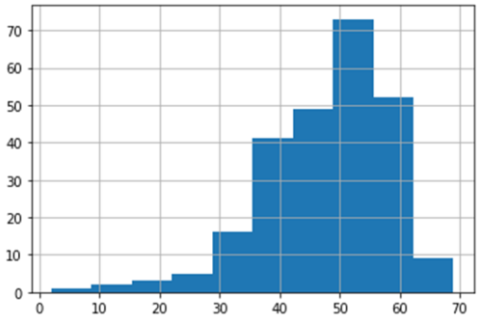
- Pre-processing
  - Pipeline 1
  - Pipeline 2
- Classificazione
  - Classificazione from scratch
  - Transfer learning fine-tuning
  - Transfer learning feature extraction
- Content Based Image Retrieval
  - CNN-based features – MobileNetV2
  - Estrazione keypoints tramite AKAZE
  - Siamese network

Analisi del dataset: validation set degradato

Il numero di immagini per ogni classe varia da:

- un minimo di 2 immagini (classe 250)
- un massimo di 69 (classe 201).





Immagini che mostrano la distribuzione delle immagini nelle varie classi

Pre-processing: Obiettivo

Dopo un'analisi dell'intero dataset è stato riscontrato che, all'interno del validation set degradato, le immagini presentano alterazioni che riducono notevolmente la capacità di classificazione delle stesse.

Obiettivo → Applicazione di tecniche di elaborazione delle immagini al fine di ottenere una migliore classificazione

Criticità →
- Immagini sfocate
- Immagini rumorose
- Alterazioni cromatiche

Pre-processing: Pipeline previste

Pipeline 1:

- Unblur dell'immagine con ESRGAN2
- White balancing
- Median filter
- Bilateral filter
- Gamma correction
- Unblur dell'immagine con NAFNet
- Sharpen filter
- Metriche di qualità

Pipeline 2:

- Unblur dell'immagine con ESRGAN2
- White balancing
- Bilateral filter
- Gamma correction
- Sharpen filter
- Metriche di qualità

Unblur dell'immagine

Le immagini degradate, in entrambe le pipeline, come primo step riducono l'effetto blur mediante l'uso della rete ESRGAN2.

Immagine 'Original'

Immagine 'Degraded'

Immagine 'Unblurred'

Kupyn, Martyniuk, Wu, Wang, et al. "DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better." arXiv preprint arXiv:1908.03826.

White balancing

Sia per la ‘Pipeline 1’ che per ‘Pipeline 2’ si procede con il white balancing in modo che i colori che percepiremmo come bianchi vengano resi come bianchi nell'immagine finale.

Immagine ‘Original’

Immagine ‘Unblurred’

Immagine ‘White balancing’

Barata, Celebi, "Improving Dermoscopy Image Classification Using Color Constancy." ieeexplore.ieee.org/abstract/document/6866131

Median filter

Nella Pipeline 1 viene applicato un filtro mediano 3x3 per rimuovere il rumore di tipo ‘salt and pepper’, che potrebbe generare artefatti con l’utilizzo della rete NAFNet.

Immagine ‘Original’

Immagine ‘White balancing’

Immagine ‘Median’

Bilateral filter

In entrambe le pipeline, viene applicato un filtro bilaterale per rimuovere il rumore che potrebbe eventualmente essere ancora presente. Nel caso in cui invece, l'immagine risultasse peggiore (secondo la metrica scelta, SSIM o BRISQUE) tale filtro non verrebbe applicato.

Immagine 'Original'

Immagine 'Median'

Immagine 'Bilateral'

Gamma correction

Entrambe le pipeline prevedono anche l'applicazione di una gamma correction adattiva per la regolazione della luminanza e quindi ridurre il contrasto complessivo.

Immagine 'Original'

Immagine 'Median'

Immagine 'Gamma'

AGCWD: S. Huang, F. Cheng and Y. Chiu, „Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution," in IEEE Transactions on Image Processing, vol. 22, no. 3, pp. 1032-1041, March 2013

Unblur con NAFNet

La prima pipeline prevede una seconda fase di unblur dell'immagine, questa volta utilizzando la rete NAFNet

Immagine 'Original'

Immagine 'Gamma'

Immagine 'NAFNet'

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun, et al. "Simple Baselines for Image Restoration." arXiv preprint arXiv:2204.04676.

Sharpen filter

Al termine delle due pipeline viene applicato uno sharpen filter con lo scopo di accentuare gli edges e quindi contrastare la perdita dei dettagli provocata dal blur.

Immagine 'Original'

Immagine 'NAFNet'

Immagine 'Sharpen'

Pipeline1


Immagine ‘Original’


Immagine ‘Degraded’


Immagine ‘Unblurred’


Immagine ‘White balancing’


Immagine ‘Median’


Immagine ‘Bilateral’


Immagine ‘Gamma’


Immagine ‘NAFNet’


Immagine ‘Sharpened’

Pipeline 2


Immagine ‘Original’


Immagine ‘Degraded’


Immagine ‘Unblurred’


Immagine ‘White balancing’


Immagine ‘Bilateral’


Immagine ‘Gamma’


Immagine ‘Sharpened’

Metriche di qualità

Metriche utilizzate per la valutazione della qualità:

- SSIM (metrica Full-reference)
- BRISQUE (metrica No-reference)

Pipeline 1

Step	SSIM
Degraded	0.517
Deblurred	0.520
White balancing	0.529
Median filter	0.521
Gamma correction	0.576
NAFNet	0.571
Sharpen	0.485

Step	BRISQUE
Original	17.315
Degraded	65.542
Deblurred	35.784
White balancing	35.406
Median filter	41.448
Gamma correction	33.739
NAFNet	42.205
Sharpen	25.079

Pipeline 2

Step	SSIM
Degraded	0.517
Deblurred	0.520
White balancing	0.529
Gamma correction	0.576
Sharpen	0.463

Step	BRISQUE
Original	17.315
Degraded	65.542
Deblurred	35.784
White balancing	35.406
Gamma correction	33.739
Sharpen	25.385

Classificazione step

- Comprensione del dataset
- Impostazione dell'ambiente con importazione librerie utili
- Caricamento dati + Campionamento
- Analisi Esplorativa
- Preprocessing (resize + rescaling)
- Classificazione: Reti Neurali from scratch e Transfer Learning
- Selezione e confronto tra i modelli migliori
- Predizioni sul test set degratato migliorato dopo il pre-processing

Reti Neurali from scratch

Nome del modello Nome della rete	Architettura + Proprietà	Categorical Accuracy	Note
Model v1 Net1	2 CNN (32, 64)	Cat_Acc: 0.14 Val_Cat_Acc: 0.03	
Model v2 Net2	2 CNN (32, 64) + batch normalization	Cat_Acc: 0.15 Val_Cat_Acc: 0.04	
Model v3 Net3	3 CNN(32, 64, 128) + batch normalization	Cat_Acc: 0.70 Val_Cat_Acc: 0.04	Aumento Categorical Accuracy sul dataset di training
Model v4 l1 Net4l1	3 CNN (32, 64, 128) + batch normalization + l1 (regolarizzazione lasso)	Cat_Acc: 0.50 Val_Cat_Acc: 0.04	
Model v4 l2 Net4l2	3 CNN (32, 64, 128) + batch normalization + l2 (regolarizzazione ridge)	Cat_Acc: 0.71 Val_Cat_Acc: 0.04	
Model v4_5 Net4_5	3 CNN (32, 64, 128) + batch normalization + l2 (regolarizzazione lasso) + 1 Dense (64)	Cat_Acc: 0.71 Val_Cat_Acc: 0.03	
Model v5 Net5	3 CNN (32, 64, 128) + batch normalization + l2 (regolarizzazione lasso) + 2 Dense (128, 64)	Cat_Acc: 0.38 Val_Cat_Acc: 0.03	Diminuzione Categorical Accuracy sul dataset di training, rete troppo complessa
Model v6 Net6	3 CNN (32, 64, 128) + batch normalization + l2 (regolarizzazione lasso) + Inizializzazione pesi He	Cat_Acc: 0.73 Val_Cat_Acc: 0.03	
Model v7 Net7	3 CNN (32, 64, 128) + batch normalization + l2 (regolarizzazione lasso) + Reduce learning Rate	Cat_Acc: 0.49 Val_Cat_Acc: 0.08	Diminuzione Categorical Accuracy sul dataset di training ma leggero aumento sul validation
Model v8 Net8	3 CNN (32, 64, 128) + batch normalization + l2 (regolarizzazione lasso) + Dropout	Cat_Acc: 0.59 Val_Cat_Acc: 0.03	

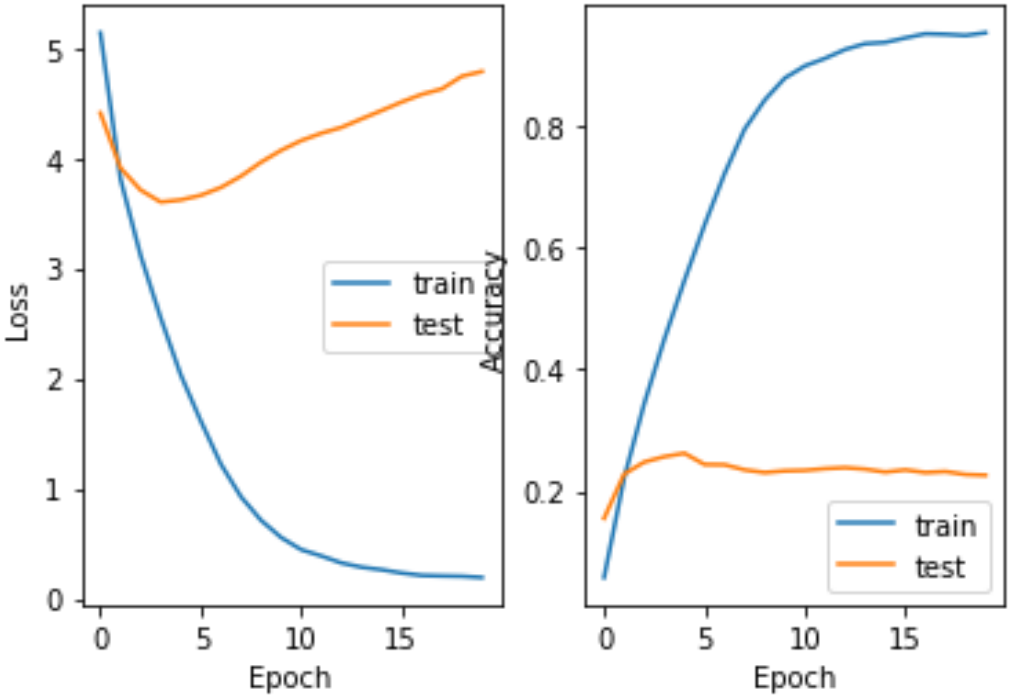
Transfer Learning

Nome del modello Nome della rete	Architettura + Proprietà	Categorical Accuracy	Note
ResNet50 v1 Model1	ResNet50 tutti layer congelati + 2 Dense (128, 64) + Dropout	Cat_Acc: 0.57 Val_Cat_Acc: 0.23	Categorical Accuracy sul validation cresciuta di molto rispetto alle reti neurali from scratch
ResNet50 v2 Model 2	ResNet50 tutti layer congelati + 7 Dense (256, 128, 64, 32, 16) + Dropout	Cat_Acc: 0.45 Val_Cat_Acc: 0.13	Categorical Accuracy diminuita sia sul training che sul validation. Blocco fully connected molto complesso.
ResNet50 v3 Model 3	ResNet50 layer congelati fino al 134 + Addestramento pesi per restanti layer + Dropout	Cat_Acc: 0.95 Val_Cat_Acc: 0.23	Categorical Accuracy sul dataset di training molto alta. Modello migliore
MobileNetV2 v1 Model1	MobileNetV2 tutti layer congelati + 2 Dense (128, 64) + Dropout	Cat_Acc: 0.70 Val_Cat_Acc: 0.22	

ResNet50 V3

Cat_Acc: 0.95

Val_Cat_Acc: 0.23



Feature Extraction

Features	Dimensione	Architettura	Categorical Accuracy	Note
Features estratte dall'ultimo dense layer della MobileNetV2 Net1_fe	(1,1280)	2 Dense layer (32, 16)	Cat_Acc: 0.11 Val_Cat_Acc: 0.06	Performance molto basse
Features estratte dall'ultimo dense layer della MobileNetV2 – V1 Net1_feim	(7, 7, 1280)	3 CNN (32, 64, 128)	Cat_Acc: 0.82 Val_Cat_Acc: 0.17	Categorical Accuracy aumentata sia sul training che sul validation. In linea con i risultati ottenuti dai modelli precedenti
Features estratte dall'ultimo dense layer della MobileNetV2 – V2 Net2_feim	(7, 7, 1280)	3 CNN (32, 64, 128) + Dropout	Cat_Acc: 0.66 Val_Cat_Acc: 0.19	

Prediction

Risultati della prediction del modello ResNet50v3 calcolate sul validation set restored:

<table>
<tr><th>Validation set originale</th><th>Validation set degradato</th><th colspan="2">Validation set restored</th></tr>
<tr><td rowspan="4">Validation categorical accuracy: 0.230</td><td rowspan="4">Validation categorical accuracy: 0.065</td><td rowspan="2">Pipeline 1:</td><td>SSIM: 0.123</td></tr>
<tr><td>BRISQUE: 0.088</td></tr>
<tr><td rowspan="2">Pipeline 2:</td><td>SSIM: 0.112</td></tr>
<tr><td>BRISQUE: 0.096</td></tr>
</table>

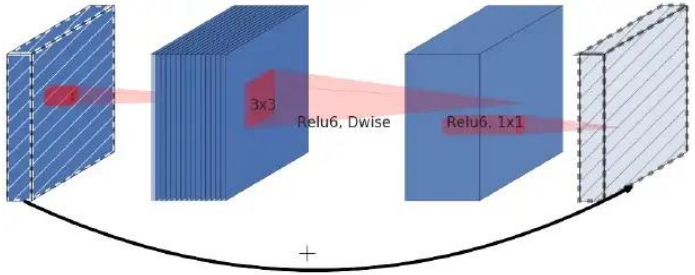
CBIR: Approcci

- 1st: CNN-based features
- 2nd: Estrazione keypoints tramite AKAZE
- 3rd: Siamese network

CNN-based features
MobileNetV2

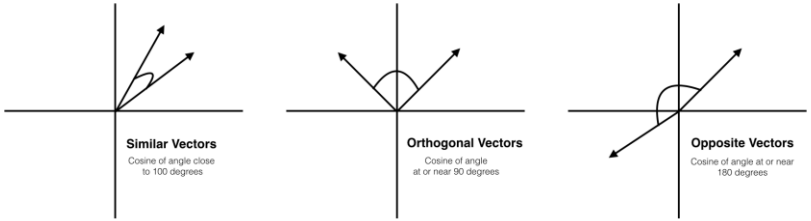
- 25K immagini di training
- 100 immagini per ogni classe di cibo
- Estrazione delle features dalla rete preaddestrata Mobilnet_v2
- Selezione del primo layer Dense come feature extractor (1280 nodi)

Input	Operator	t	c
$224^2 \times 3$	conv2d	-	32
$112^2 \times 32$	bottleneck	1	16
$112^2 \times 16$	bottleneck	6	24
$56^2 \times 24$	bottleneck	6	32
$28^2 \times 32$	bottleneck	6	64
$14^2 \times 64$	bottleneck	6	96
$14^2 \times 96$	bottleneck	6	160
$7^2 \times 160$	bottleneck	6	320
$7^2 \times 320$	conv2d 1x1	-	1280
$7^2 \times 1280$	avgpool 7x7	-	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k

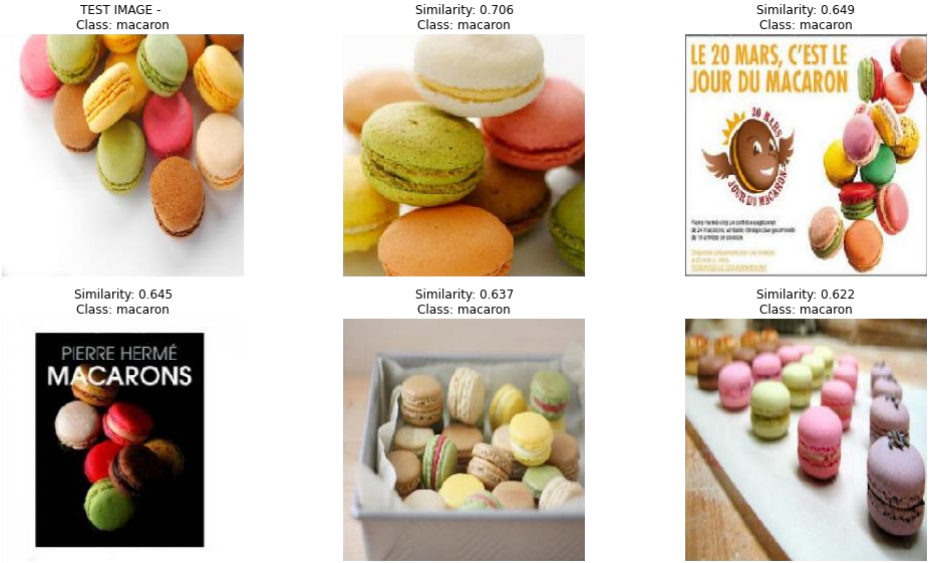


Misura di similarità

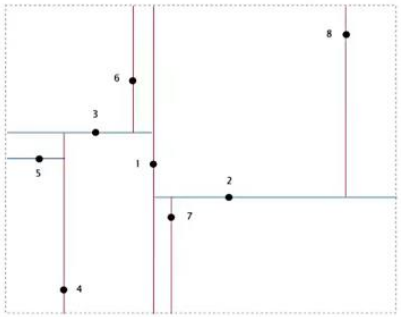
- Cosine similarity come misura di similarità

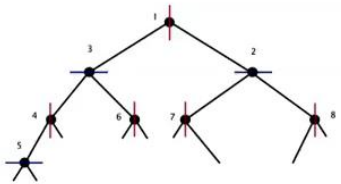


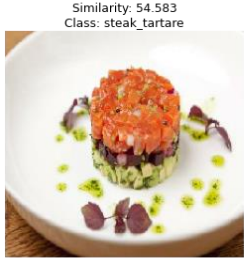
- Struttura for-loop per controllare ogni possibile combinazione
- Top-5 accuracy: 31%



- Seconda versione: aggiunta della ricerca tramite KDTree
- Search engine ottimizzato che permette di evitare for-looping (sliding midpoint rule)
- Migliori search performance: O(logn) invece di O(n)
- Diminuzione 50% tempo query
- Top-5 accuracy: 24%







Estrazione keypoints tramite AKAZE

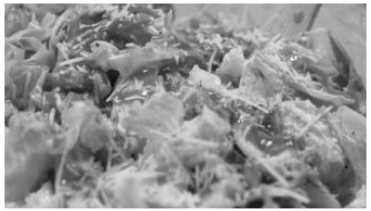
- 25K immagini di training
- 100 immagini per ogni classe di cibo
- Estrazione dei descrittori AKAZE di ciascuna immagine

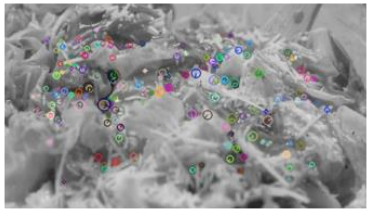
A sinistra: blurring di tipo Gaussiano per itentificazione e descrizione delle features 2D, usato ad esempio per SIFT.

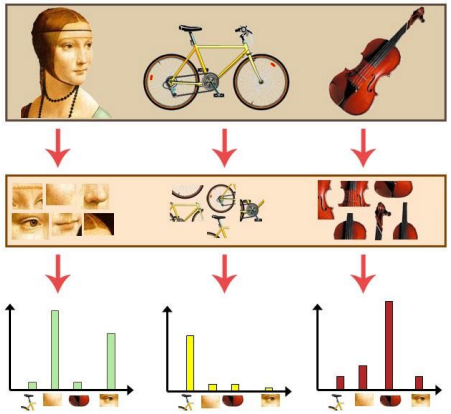
A destra: applicazione spazio non lineare, in grado di mantenere bordi naturali, usato ad esempio per AKAZE.

STEPS

- Estrazione dei descrittori utilizzando una soglia fissata a 0.01
- Clustering dei 500k descrittori ottenuti (k-means, con k=251)
- Definizione della BoWA (Bag of Words AKAZE descriptors)
- Top-5 accuracy: 7%



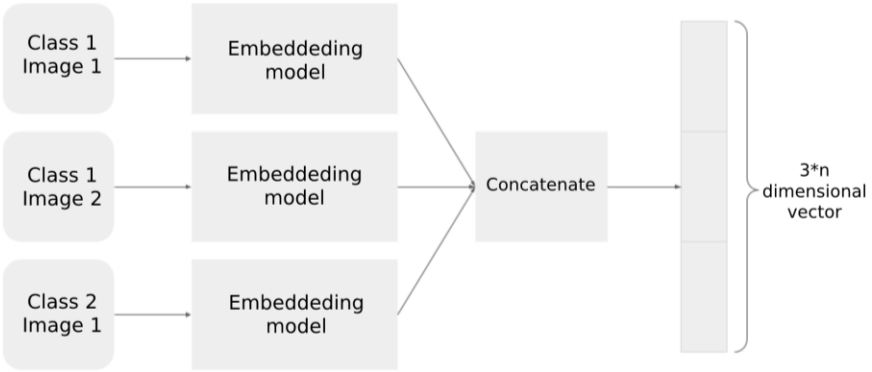




Usman M., Weiqiang W., Abdenour H., Shahbaz P., "Bag of words KAZE (BoWK) with two-step classification." doi: 2018.5069

Siamese network

- 25K immagini di training
- 100 immagini per ogni classe di cibo
- Definizione di una rete embedded per la classificazione
- Miglioramento delle performance tramite triplet loss



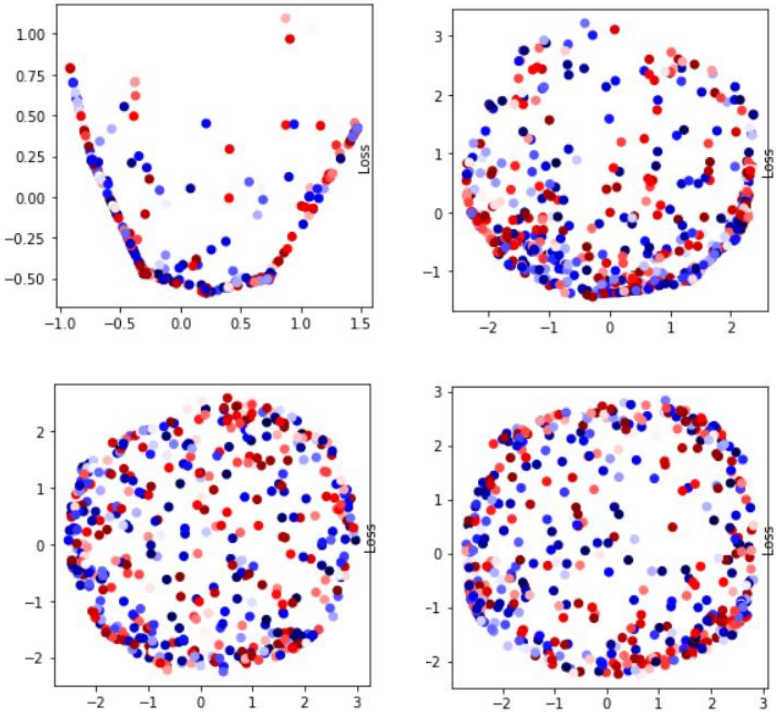
STEPS:

- Lettura immagine di dimensione 100x100x3
- Identificazione esempio positivo e negativo per triplet loss (alpha=0.2)

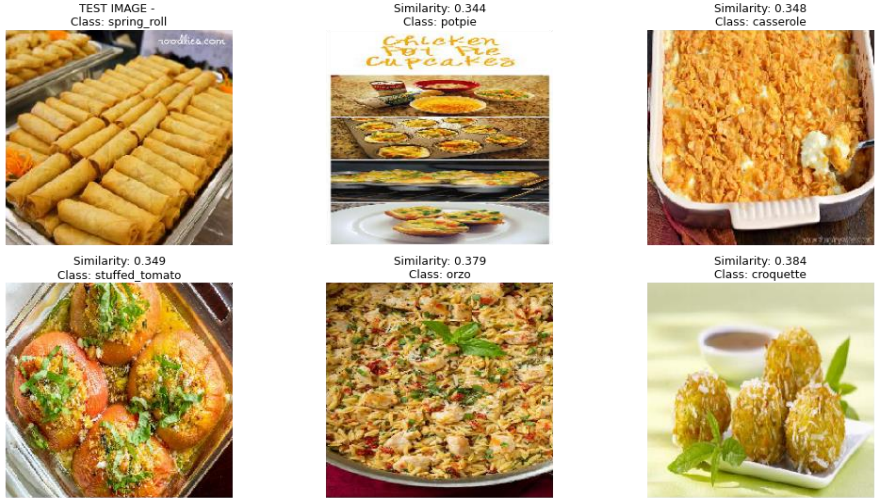
$$Loss = \sum_{i=1}^{N} \left[\| f_i^a - f_i^p \|_2^2 - \| f_i^a - f_i^n \|_2^2 + \alpha \right]_+$$

- Definizione rete con input di dimensione 1x30000 per ciascuna delle tre immagini, seguita da un layer Dense con 128 neuroni e attivazione RELU e un layer Dense con 251 neuroni con attivazione Sigmoid
- Applicazione KNN per identificare immagini più simili

- Evoluzione del training tramite PCA



- Top-5 accuracy: 15%



Problemi riscontrati

- Limitate risorse computazionali (GPU e RAM)
- Presenza di immagini non inerenti al cibo (es. occhio, scatole, negozi)

Class ‘Gingerbread’

Class ‘Gyro’

Conclusioni e sviluppi futuri

Conclusioni:

- Migliori performance ottenute sul modello ResNet50_v3 (validation categorical accuracy = 0.230)
- Ricostruzione tramite metrica full reference (SSIM, VCA=0.123) del dataset degradato porta a performance migliori rispetto alla metrica no reference (BRISQUE, VCA=0.096)
- Entrambi i metodi migliorano l'accuracy ottenuta sul dataset degradato (0.065)

Sviluppi futuri:

- Aggiunta di un relevance feedback
- Prevedere una classe di rigetto in cui catalogare tutte le immagini non relative alle 251 classi da classificare

Grazie per la vostra attenzione

Gabriele Fumagalli – Veronica Morelli – Davide Valoti