FoodX-251

Biasco Anna Marika, Pulerà Francesca e Zarantonello Massimo

Progetto di Visual Information Processing and Management

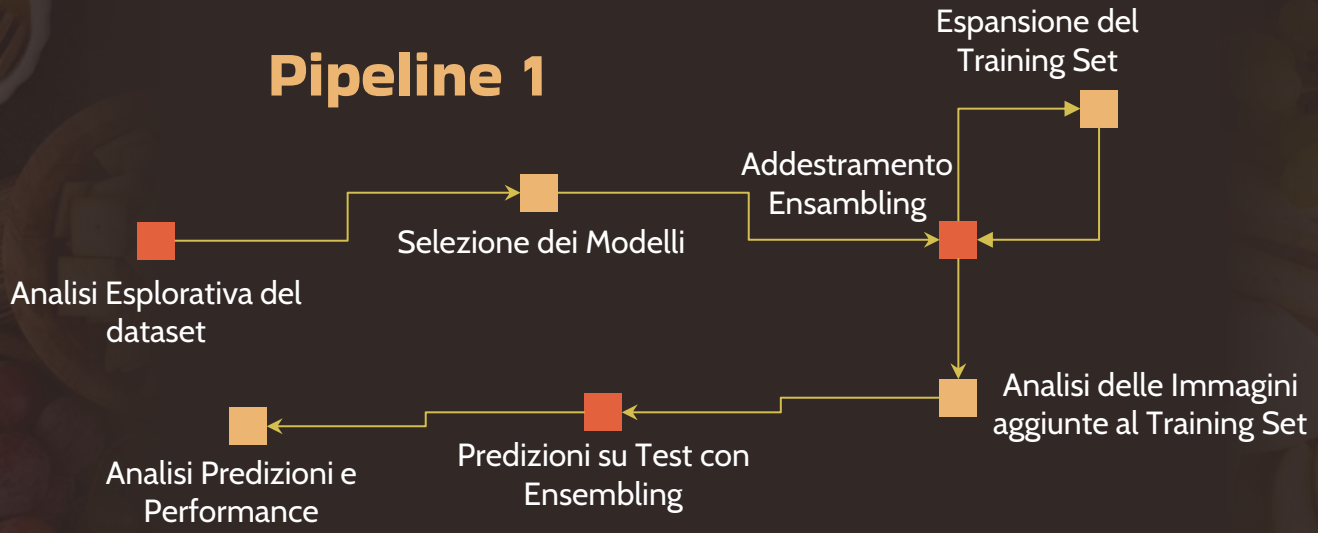
Anno accademico 2024/2025

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA
DISCO DIPARTIMENTO DI INFORMATICA SISTEMISTICA E COMUNICAZIONE

Obiettivo del progetto

Progettare e valutare un sistema di classificazione fine-grained su 251 classi alimentari del dataset FoodX-251.

Pipeline 1



Overview

	immagini
Training set Totale	118.475
Training set Etichettato	5.020
Test set	11.994

Analisi Esplorativa del Dataset

Prima di applicare qualsiasi modello di machine learning, è cruciale comprendere il dataset con cui stiamo lavorando.
Analizzeremo prima il TRAIN set etichettato, e poi il TEST set.

Identificazione dati inconsistenti

come le etichette errate

Esplorazione della qualità delle immagini

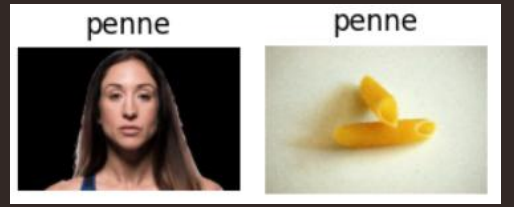
controllando aspetti come dimensioni, scala e formato

Calcolo della varianza del colore delle Immagini

tramite un'analisi della diversità visiva tra le classi per valutare la complessità della classificazione

Identificazione dati inconsistenti – TRAIN

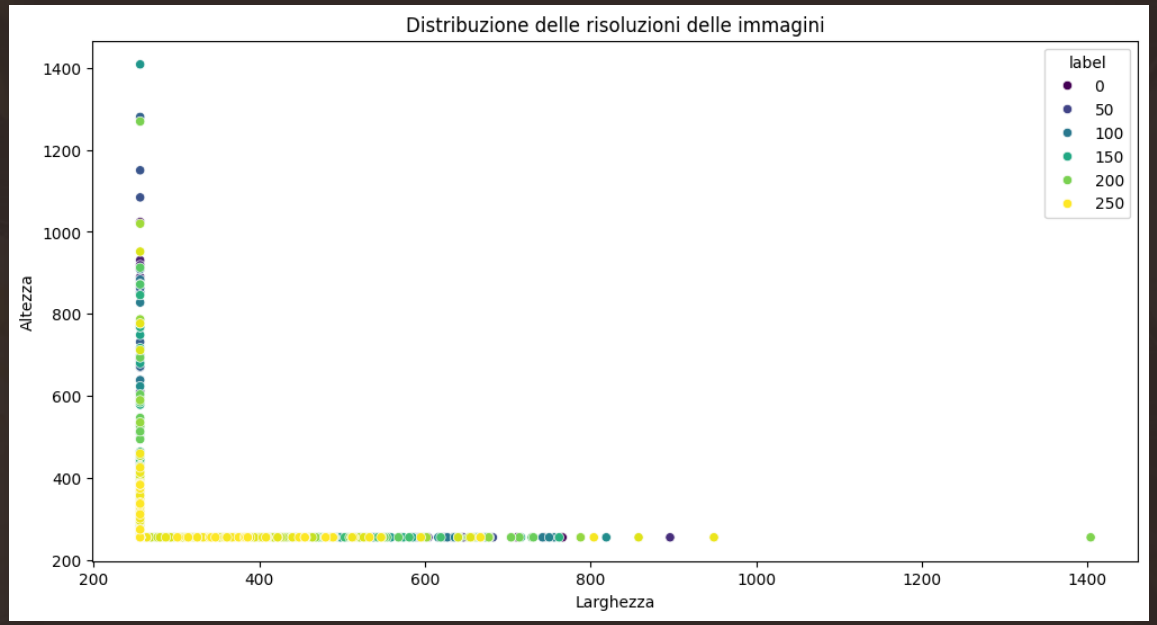
È evidente che il dataset etichettato non è completamente pulito: a volte si possono trovare immagini non correlate al cibo (persone, locali o animali vivi), non rappresentative (ingredienti confezionati o scatole per la preparazione di un piatto), disegni anziché fotografie, utensili e contenitori (come la teiera di un ciambellone) ed etichette errate.



Inoltre, la classe poi (116) non contiene immagini di cibo.

Esplorazione della Qualità delle Immagini - TRAIN

Le immagini variano sia in larghezza che in altezza, e questa variabilità può influire sul processo di preprocessing e sul comportamento del modello.



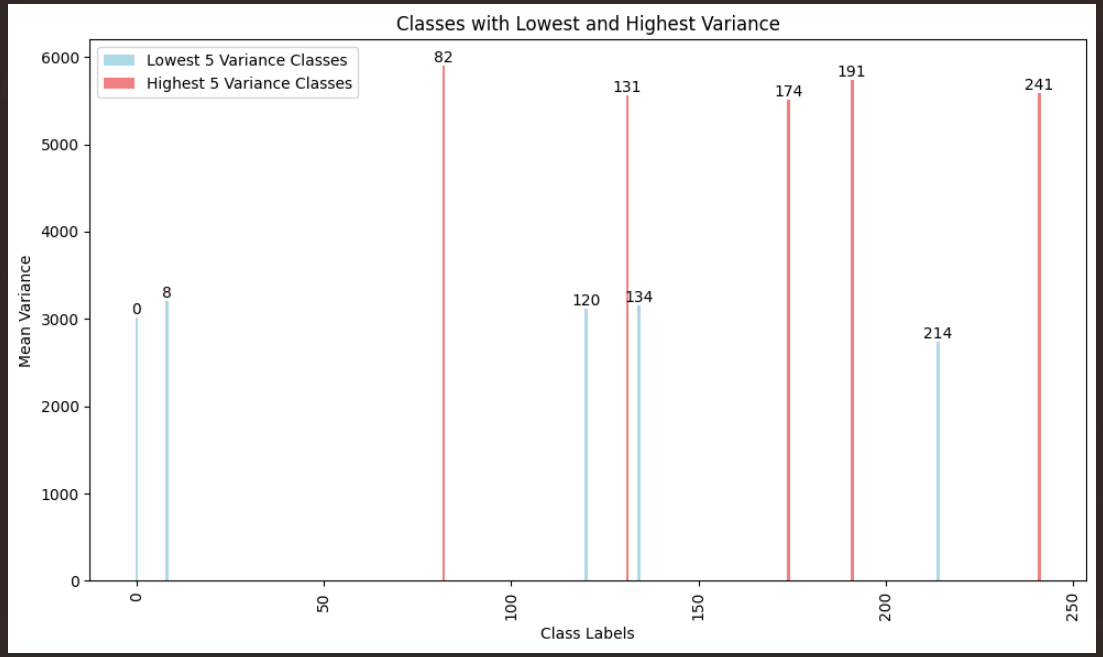
La gestione efficace delle dimensioni delle immagini è cruciale per garantire un addestramento efficace e per evitare che le variazioni nelle dimensioni possano introdurre distorsioni o inefficienze nel modello.

Overview

	immagini	dimensioni medie immagini	range dimensioni
Training set Totale	118.475	341.21x287.14	larghezza (256-2733), altezza (256-2744)
Training set Etichettato	5.020	341.32x286.28	larghezza (256-1404), altezza (256-1408)
Test set	11.994	342.01x287.49	larghezza (256-1035), altezza (256-1274)

- Presenza di outlier con dimensioni molto grandi, che potrebbero richiedere normalizzazione
- Il training set ha un'ampia gamma di dimensioni, mentre il set etichettato e il test set sono più omogenei.
- Le dimensioni medie delle immagini sono simili (~341x287 px), ma ci sono outlier con risoluzioni molto diverse.

Calcolo della Varianza delle Immagini

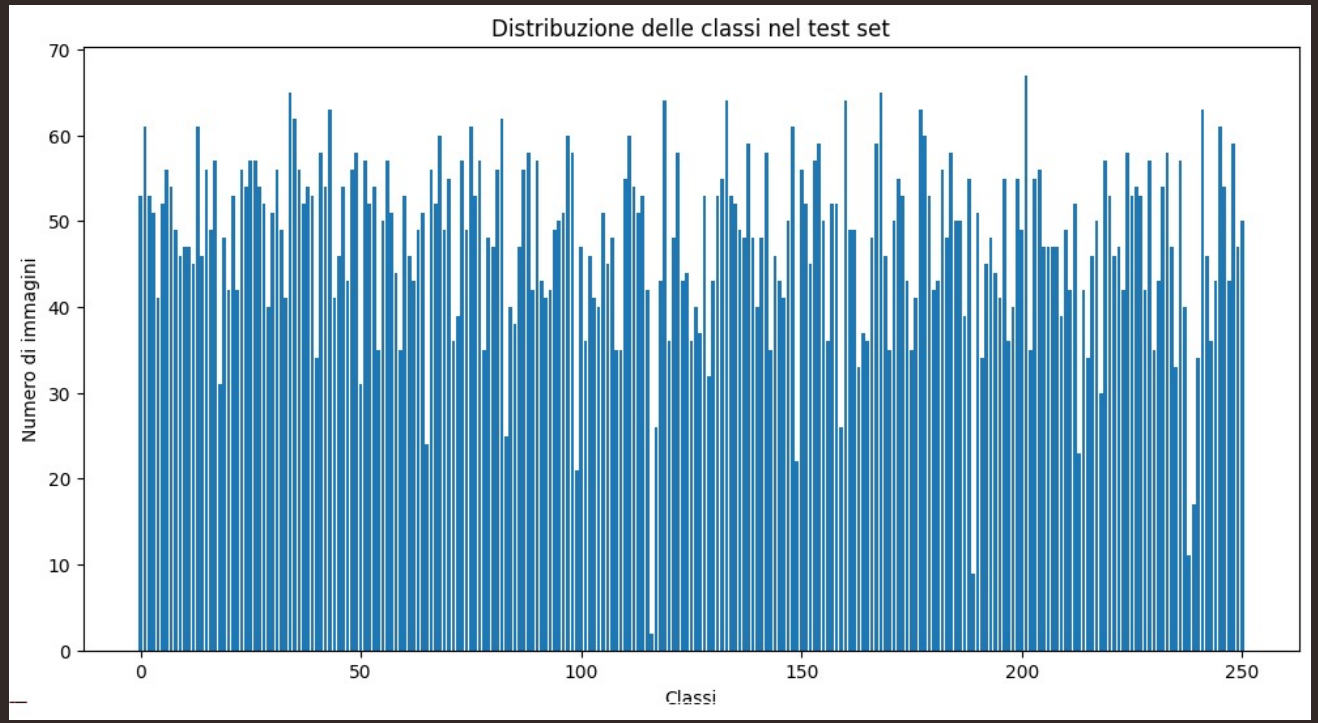


Calcoliamo la complessità visiva utilizzando metodi come la varianza del colore tra le immagini nelle varie classi. Alcune classi potrebbero avere immagini più uniformi (ad esempio piatti con ingredienti simili), mentre altre potrebbero avere una grande variazione visiva. Questo potrebbe aiutare a capire la difficoltà di classificazione per ciascuna classe.

Distribuzione delle Classi – TEST

- Classe con meno immagini: 116 (2 immagini)
- Classe con più immagini: 201 (69 immagini)

C'è uno squilibrio, il quale implica che alcune classi verranno testate su una base dati significativamente più ampia rispetto ad altre.



Identificazione dati inconsistenti – TEST

Per l'insieme di test non sono emerse anomalie. Tutte le immagini risultano coerenti con le etichette assegnate, e a differenza del training set, la classe *poi* contiene immagini di cibo (seppur poche).

Selezione dei Modelli

Siamo partiti valutando le prestazioni su questi tre modelli:

01

SimpleCNN

Massima memoria allocata: 1576.48 MB

Tempo di addestramento: 1876.51 s

Accuratezza del validation: 0.000

Numero di parametri:

02

ResNet18

Massima memoria : 850.67 MB

Tempo di addestramento : 2191.50 s

Accuratezza del validation : 0.15912

Numero di parametri: ≈ 11,315,563

03

ResNet50

Massima memoria : 499.55 MB

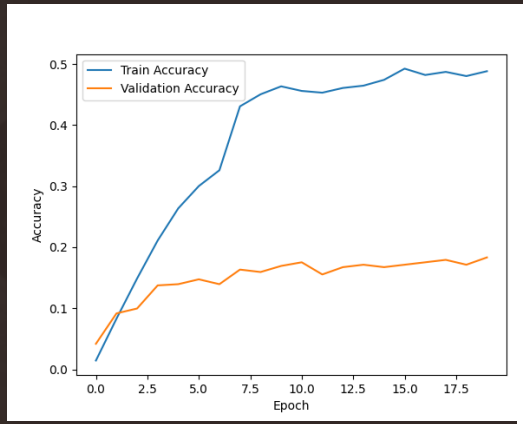
Tempo di addestramento : 2238.57 s

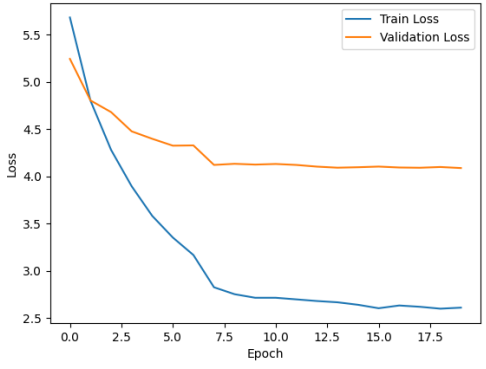
Accuratezza del validation : 0.16967

Numero di parametri ≈ 24,065,099

02

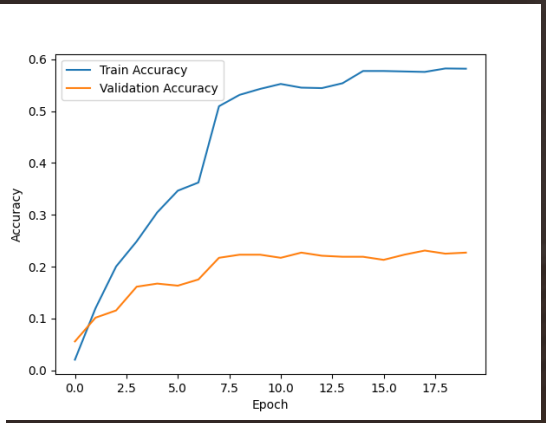
ResNet18

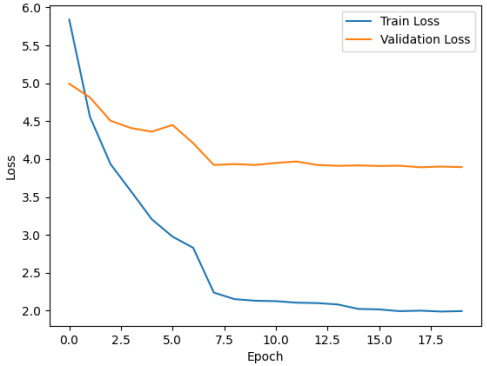

Train Accuracy
Validation Accuracy
Accuracy
Epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Train Loss
Validation Loss
Loss
Epoch
2.5
3.0
3.5
4.0
4.5
5.0
5.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

03

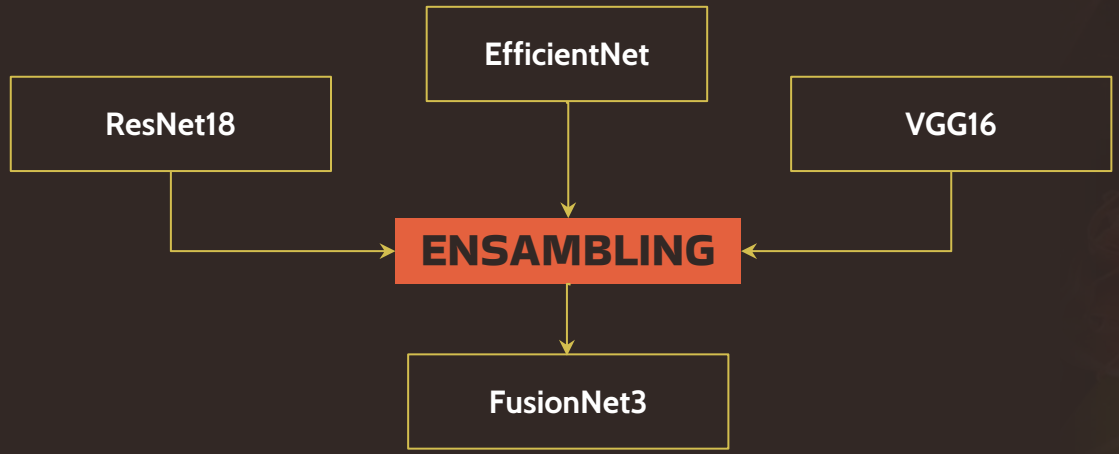
ResNet50


Train Accuracy
Validation Accuracy
Accuracy
Epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Train Loss
Validation Loss
Loss
Epoch
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Selezione dei Modelli

Con tutti e tre i modelli precedenti, l'accuratezza ottenuta è stata inferiore alle aspettative.

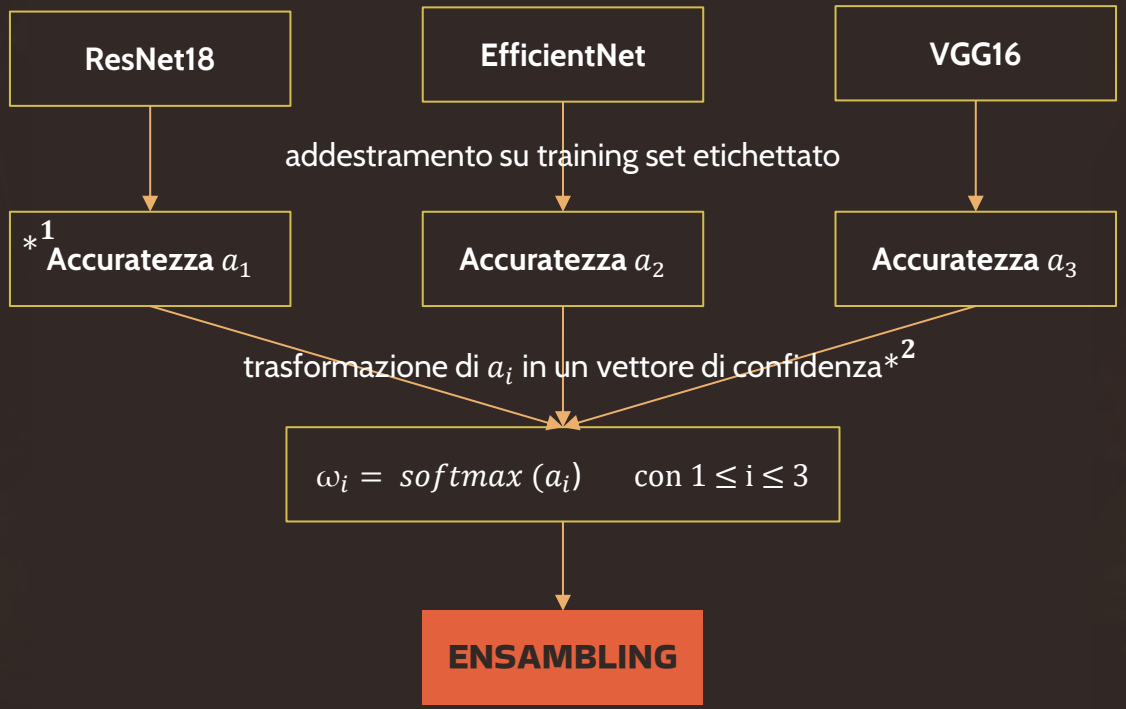


Utilizzando **ResNet18 EfficientNet** e **VGG16**, combinati con il riaddestramento dell'ultimo layer fully connected, siamo riusciti a ottenere le migliori performance in termini di accuratezza e capacità di generalizzazione, segnando un avanzamento significativo nel nostro processo di addestramento.

ENSAMBLING

L'idea è di sfruttare le loro predizioni per **etichettare immagini non annotate**, selezionando solo quelle per cui tutti e tre i modelli **superano una determinata soglia di confidenza** -> ciclo iterativo di **auto-etichettatura e riaddestramento**.

Addestramento Ensambling



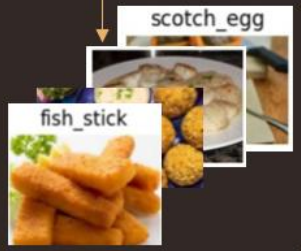
*1 Calcolata su un sottoinsieme di validation

*2 Somma a 1

Espansione del Training Set

batch del training set

ENSAMBLING



batch del training set con le etichette assegnate

immagine unlabel di un batch del training set

ENSAMBLING

SE tutti e tre d'accordo

altrimenti

Calcola confidenza basata sulle prestazioni ottenute fino ad ora

Passa alla prossima immagine

SE media pesata ≥ 80%

etichettatura, rimozione dal training set non etichettato e aggiunta a quello etichettato

Espansione del Training Set

Dataset iniziale: 5020 immagini
Dataset non etichettato: 113,455 immagini
13 cicli

Ciclo 0

Nessuna accuratezza registrata, aggiunte 52 immagini

Cicli 1–3

Prime accuratezze
→ EfficientNet: 21.6%,
→ ResNet18: 18.5%,
→ VGG16: 18.3%

Cicli 4–7

Crescita continua
→ EfficientNet: 24.8%,
→ ResNet18: 23.9%,
→ VGG16: 21.8%

Cicli 8–13

Miglioramenti finali
→ EfficientNet: 28.4%,
→ ResNet18: 26.5%,
→ VGG16: 24.7%

Risultato finale
Dataset di addestramento: 5537 immagini
Dataset non etichettato: 112,935 immagini

Alcuni Esempi

Immagine Etichettata

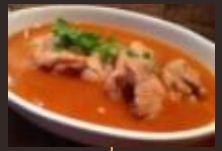
ENSAMBLING

(con ResNet18) (con EfficientNet) (con VGG16)

(0.0002, 0.03, ..., **0.85**, ...) (0.0010, 0.028 ..., **0.80**, ...) (0.008, 0.043 ..., **0.90**, ...)

distribuzioni di probabilità pesate

Immagine NON Etichettata



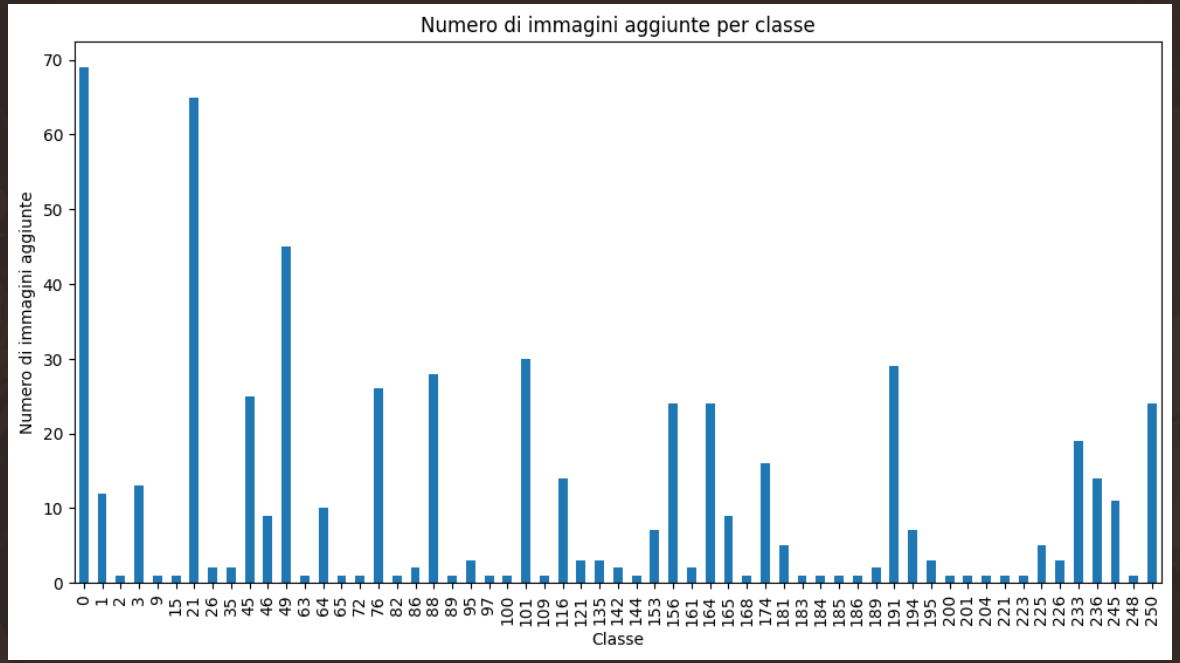
ENSAMBLING

(con ResNet18) (con EfficientNet) (con VGG16)

(0.0002, 0.45, ..., **0.23**, ...) (0.0010, 0.28 ..., **0.5**, ...) (0.006, 0.043 ..., **0.80**, ...)

distribuzioni di probabilità pesate

Analisi delle Immagini Aggiunte al Training Set



Se molte immagini sono state aggiunte alla stessa classe, può essere un segnale che quella classe è più "facile" da classificare con elevata confidenza, probabilmente perché ha caratteristiche visive distintive.

Alcuni Esempi

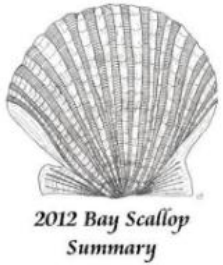
Immagini coerenti con le etichette assegnate dal modello di ensembling

beignet

onion_rings

cupcake

scallop



edamame

mussel

cupcake

poi

poi

edamame

Accuracy che ha l'ensambling quando aggiunge le immagini: **0.928961**

Altri Esempi

Errori rari, simili alle incertezze umane

- **Caso particolare**: nella classe *stuffed_tomato*, alcune immagini mostrano **peperoni ripieni anziché pomodori**

Problemi

✅ **Nuove immagini** aggiunte principalmente da **60 classi specifiche** (es. *macaron, beignet, sashimi, cupcake...*)
⚠️ **Rischio di Overfitting**: squilibrio nella distribuzione → il modello si focalizza troppo su alcune classi
📉 **Conseguenza**: ridotta generalizzazione su classi meno rappresentate

Predizioni su Test con Ensambling

La pipeline ciclica dell'ensambling ha permesso di aggiungere nuove immagini, affinando le predizioni dei modelli e migliorando la generalizzazione.

adesso

Obiettivo: migliorare le predizioni sul **test set** tramite ensambling

Predizione Ponderata con Ensembling

- Le previsioni combinano i risultati di **ResNet18, EfficientNet e VGG16**.
- I pesi di ciascun modello sono basati sulla loro **accuratezza finale**.
- La classe finale è quella con la **probabilità massima** dopo la combinazione ponderata.

Alcuni Esempi

Predizioni Corrette

Confidenza media per le corrette classificazioni: 0.3410472

Predizioni Errate

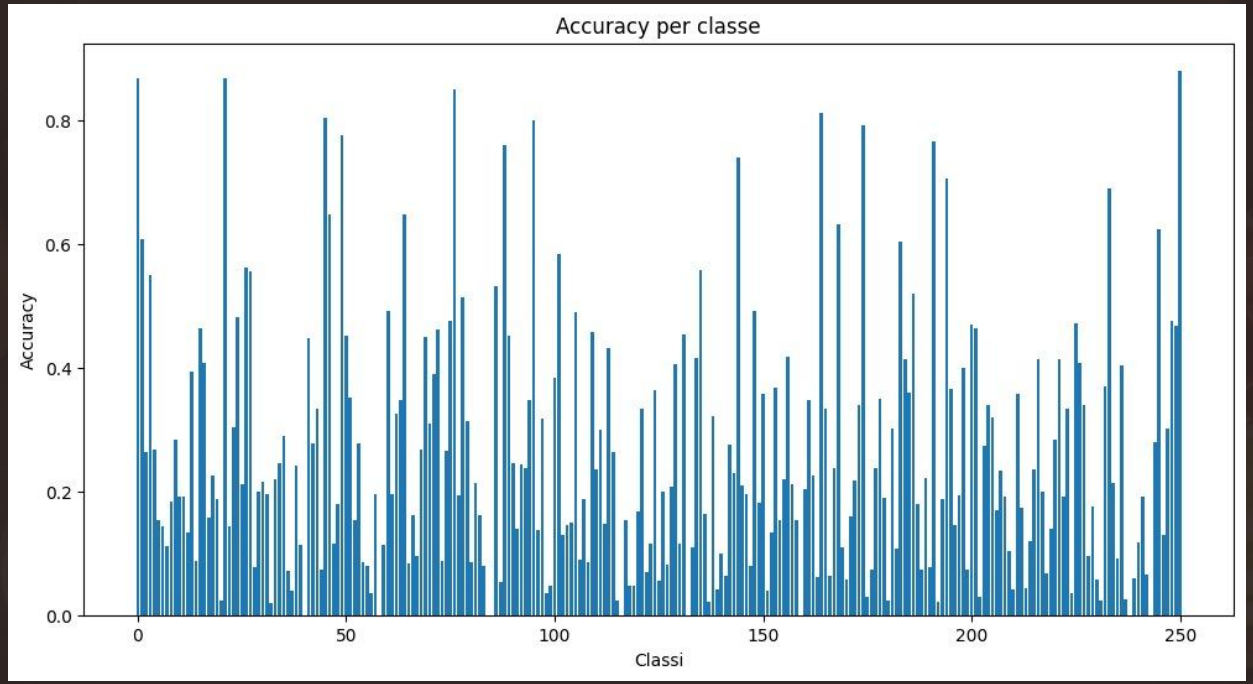
Confidenza media per le classificazioni errate: 0.16867812\

Analisi Predizioni e Performance

- Accuracy iniziale: 27.8% → prestazioni iniziali moderate.
- Accuracy migliorata su un subset di 60 classi: 51% → miglioramento selezionando classi in cui il modello è più sicuro.
- Accuracy migliore raggiunta su un subset di 18 classi: 70.72% → riducendo il numero di classi, la performance aumenta.

Conclusione: la selezione di un sottoinsieme di classi più distinguibili migliora l'accuratezza, riducendo la complessità del problema e ottimizzando l'apprendimento del modello.

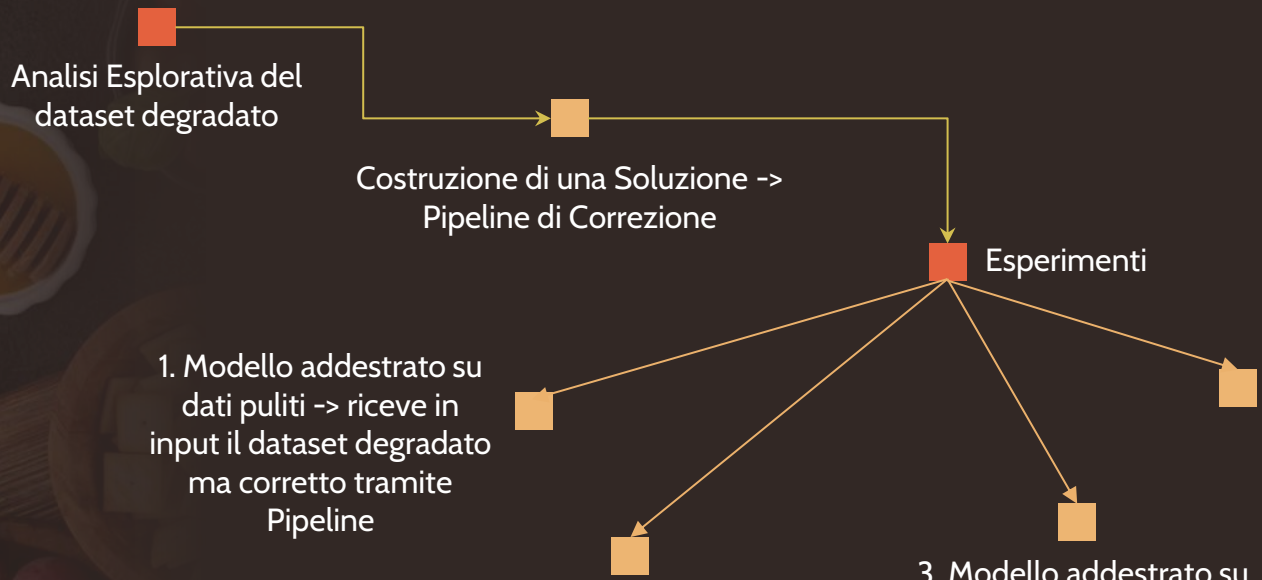
Analisi Predizioni e Performance



Dataset Degradato

Classificazione del validation set degraded

Pipeline 2


Analisi Esplorativa del dataset degradato
Costruzione di una Soluzione -> Pipeline di Correzione
Esperimenti
1. Modello addestrato su dati puliti -> riceve in input il dataset degradato ma corretto tramite Pipeline
2. Applicazione di rumori al train set -> Applicazione pipeline di migliroamento -> Addestramento
3. Modello addestrato su dati puliti -> applicata pipeline di miglioramento per individuare ed escludere immagini degradate
4. K-Nearest Neighbors (KNN) per apprendere funzione di denoising su immagini degradate

Sviluppi Futuri

Metodi di Miglioramento del Rumore

Integrare tecniche avanzate di denoising o identificazione del rumore per selezionare solo immagini di alta qualità nella pipeline di miglioramento, ottimizzando il processo di pseudo-etichettatura.

KNN di Dimensione Ridotta

Esplorare tecniche di ridimensionamento e ottimizzazione del tensore, come l'uso di un autoencoder per comprimere le immagini, o la conversione in scala di grigi per ridurre la complessità computazionale, migliorando così l'efficienza del KNN e facilitando l'elaborazione delle immagini.

Incremento del Ciclo di Aggiunta delle Immagini di Train

Proseguire il ciclo iterativo di pseudo-etichettatura ed espansione del training set, affinando progressivamente il modello. L'obiettivo è migliorare ulteriormente le prestazioni dell'ensemble, incrementando la qualità e la quantità dei dati etichettati con soglie di confidenza sempre più ottimizzate.

Grazie!

Biasco Anna Marika

Pulerà Francesca

Zarantonello Massimo

Bibliografia

ModelliResNet-18 e ResNet-50He, K., Zhang, X., Ren, S., & Sun, J. (2016).
Deep Residual Learning for Image Recognition.
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.
https://doi.org/10.1109/CVPR.2016.90EfficientNetTan, M., & Le, Q. V. (2019).
EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.
Proceedings of the 36th International Conference on Machine Learning (ICML), 6105–6114.
https://arxiv.org/abs/1905.11946VGG16Simonyan, K., & Zisserman, A. (2014).
Very Deep Convolutional Networks for Large-Scale Image Recognition.
Proceedings of the International Conference on Learning Representations (ICLR).
https://arxiv.org/abs/1409.1556DatasetFoodX251Zhang, Y., Sun, Y., Chen, X., & Hu, X. (2020).
FoodX251: A Multi-Label
Dataset for Fine-Grained Food Recognition.
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1306–1315.
https://doi.org/10.1109/CVPR42600.2020.00138