VIPM Project

FoodX-251 Classification and Other Tasks

Vittorio Haardt (853268)
Luca Porcelli (853189)

Universita degli studi di Milano-Bicocca

9 Febbraio 2024

Outline

1. Analisi Dataset
2. Pulizia Train
3. Classificazione
4. Classificazione Immagini PDF

Outline: Analisi Dataset

1. Analisi Dataset
   1. Distribuzione Data Set
   2. Problemi Training Set
   3. Test Set Degradato
2. Pulizia Train
3. Classificazione
4. Classificazione Immagini PDF

1. Distribuzione Data Set

Dataset FoodX-251

251 classi di immagini di cibo

Training set

~118,000 immagini

Test Set

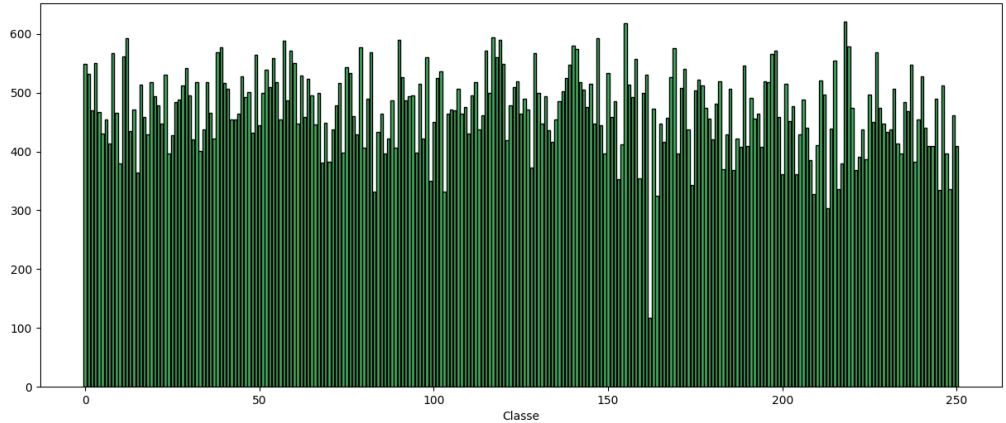
~11,000 immagini

Test Set Degradato

~11,000 immagini

1.1 Distribuzione Data Set

Classi sbilanciate nel training, andrà gestito per la classificazione



1.2 Problemi Training Set

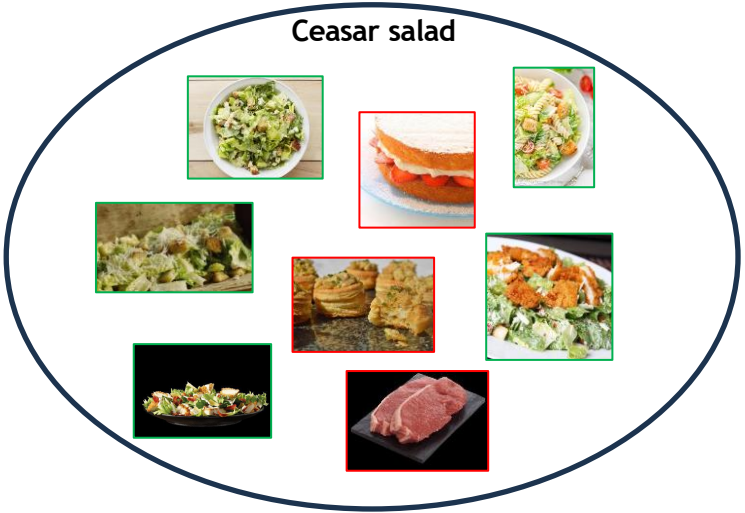
Il training set presenta diversi problemi che vanno gestiti in modo da poter addestrare adeguatamente i classificatori.

- ~20% di osservazioni assegnate ad una label sbagliata
- Immagini non appartenenti a nessuna classe
- Immagini degradate
- Similarita tra classi
- Dissimilarita e rumore intra classi

1.2 Problemi Training Set: label sbagliate

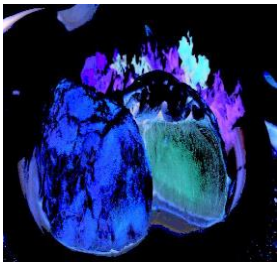
Il problema principale riguarda gli elementi classificati male nel training set. Porta ad inaffidabilita della ground truth. Ingestibile manualmente per la mole di osservazioni.

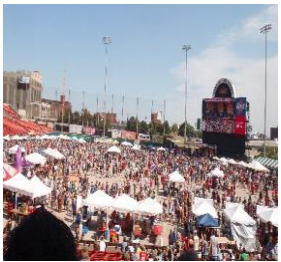
Necessità di un approcio automatizzato



1.2 Problemi Training Set: immagini fuori tema o degradate

Le immagini non riguardanti cibo portano a rumore, cosi come quelle degradate, è necessario per questo motivo rimuoverle.





1.2 Problemi Training Set: similarità tra classi

Le classi simili tra di loro portano una challenge ulteriore nella classificazione, questo problema tuttavia non è risolvibile, dato che non si vuole rimuovere o aggregare classi.

Stuffed Tomatoes

Stuffed Peppers

1.2 Problemi Training Set: dissimilarità intra classi

La stessa classe di cibo può contenere immagini estremamente differenti, comunque cosiderate corrette.

1.3 Test Set Degradato

Il Test Set degradato contine le stesse osservazioni del Test Set base con delle degradazioni.

L'obbiettivo principle è avere le pefomance più alte possibili sul Test degradato. Per questo motivo è bene osservare le possibili degradazioni presenti in modo da sviluppare la strategia più adeguata.

- Gaussian Filter
- Blurring
- Jpeg compression

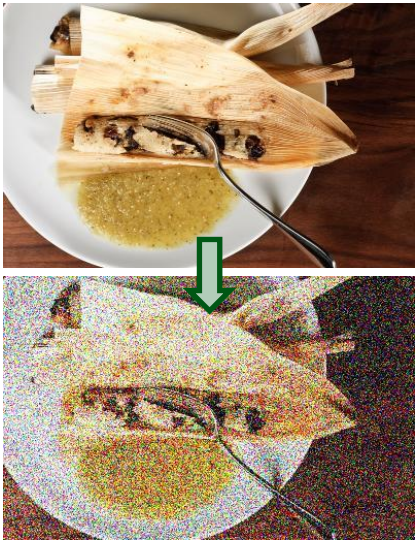
1.3 Test Set Degradato: gaussian noise

Il Gaussian noise segue una distribuzione di probabilità normale, con valori di intensità dei pixel concentrati attorno ad una media e diffusi in modo simmetrico.

Comporta piccole variazioni casuali nell'intensità dei pixel, portando l'immagine ad essere meno fedele all'originale.

Possibile risoluzione con:

- Filtraggio
- Denoising



1.3 Test Set Degradato: blurring

Il blurring, o sfocatura, è un effetto che riduce la nitidezza e la chiarezza di un'immagine, facendola sembrare più morbida o meno definita.

L'effetto di sfocatura comporta una perdita di dettagli e definizione nell'immagine, facendo sembrare che i bordi siano meno definiti e meno nitidi.

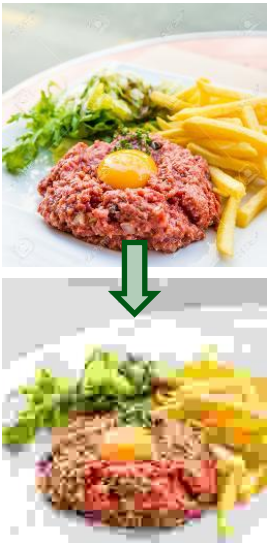
1.3 Test Set Degradato: jpeg compression

La compressione JPEG è una tecnica ampiamente utilizzata per ridurre la dimensione dei file delle immagini digitali, ma può introdurre artefatti e rumore nell'immagine.

Comporta perdita di dettagli e alla comparsa di artefatti visivi.

Tipici segni di rumore:

- Blocchi (quadrati o blocchi visibili nelle zone di transizione di colore)
- Ringing (linee di contorno intorno ai bordi ad alto contrasto)



Outline: Pulizia Train

1. Analisi Dataset
2. **Pulizia Train**
   1. **Approccio Iniziale**
   2. **Pulizia Effettiva**
   3. **Valutazione Empirica**
3. Classificazione
4. Classificazione Immagini PDF

2 Pulizia Train

Per garantire delle perfomance di classificazione adeguate è necessaria una pulizia del Training Set che risolva i problemi visti prima.

In particolare è cruciale avere una ground truth quanto più possibile affidabile.

Assunzioni fatte:

- Impossbilità di valutazione oggettiva risultato pulizia
- Maggior parte degli per classe corretti
- Immagini degradate considerate scorrette

2.1 Approccio Iniziale

L'idea iniziale era di applicare un algoritmo di **Semantic Image Clustering** [1] sul Training Set, per poi ispezionare i cluster e osservarne la composizione.

Idea: ogni cluster composto dai veri elementi di ogni classe.

Problemi

- Silmilarità tra classi
- Elementi non appartenenti a nessuna classe
- Numero di cluster difficile da definire
- Ispezione dei cluster manuale (time consuming)

[1] Khalid Salama, Semantic Image Clustering: Semantic Clustering by Adopting Nearest neighbors (SCAN) algorithm.

2.2 Pulizia Effettiva: idea

Assumiamo che classi reali nel train siano dei cluster, derivanti da una clusterizzazione su feature di alto livello (colore, texture). Stimando i centroidi di ogni cluster possiamo applicare una soglia e rimuovere gli elementi più distanti, assunti come non appartenenti alla classe.

Lavoriamo su una classe per volta.

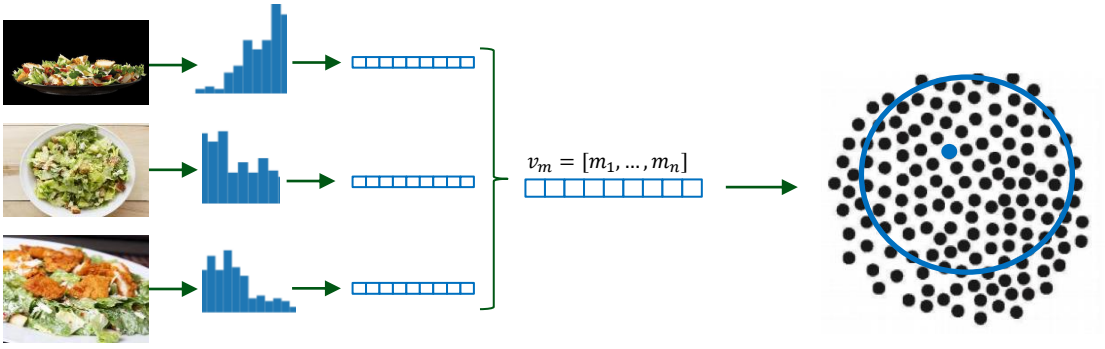
Componenti

- Feature: Color Difference (CDH L*a*b*) e Texture (LBP) [2][3]
- Immagini come vettori di feature
- Centroide di classe vettore mediano
- Distanza: Camberra Distance (enfasi sulle fearute più dissimili)

[2] AdityaShaha, Image Retrieval using Color Difference Histogram (https://github.com/AdityaShaha/CBIR-Using-CDH).
[3] Fauzan Firdaus, Texture Feature Extraction Using LBP (https://github.com/faoezanf/Texture-Shape-And-Color-Extraction).

2.2 Pulizia Effettiva: schema

Il centroide è il vettore mediano, per garantire più stabilita.
Il quartile degli elementi più distanti (25%) viene considerato con label sbagliata quindi rimosso.



2.2 Pulizia Effettiva: step pulizia

Step:

Per ogni classe (label = k)

Ridimensionamento immagini 64x64

CHD:
1. Calcolo istogramma CHD per ogni osservazione in k
2. Calcolo centroide come vettore mediano
3. Calcolo distanza vettori osservazioni e centroide
4. Separo il 25% delle ossecazioni più lontante

LBP:
1. Calcolo istogramma LBP per ogni osservazione in k
2. Calcolo centroide come vettore mediano
3. Calcolo distanza vettori osservazioni e centroide
4. Separo il 25% delle ossecazioni più lontante

Rimozione osservazioni separate alemeno una volta [possibili alternative]

2.3 Valutazione Empirica

Impossibile attuare una valutazione oggettiva (no ground truth).
Valutazione “manuale” sulla classe 94 (Caesar salad).

Pre pulizia

~16% osservazioni sbagliate:

- Classe sbagliata
- Degradate
- Non di cibo

Post pulizia

~5% osservazioni sbagliate

Drawback: rimozione di molte osservazioni corrette

Outline: Classificazione

1. Analisi Dataset
2. Pulizia Train
3. Classificazione
    1. Pulizia Test Set
    2. Data Augmentation
    3. Modelli
    4. Evaluation
4. Classificazione Immagini PDF

3 Classificazione

Obbiettivo: performance più alte possibili sul test set degradato.

Sono state testati modelli diversi con architetture diverse.
Nella fase di addestramento si è usato il 20% del training set come validation per l'ottimizzazione dei parametri.

Strategie:

- Bilanciamento classi (normalized class weight: $w_{norm} = \frac{\sum w_{class}}{w_{class} \times N}$)
- Miglioramento test set degradato
- Data agumentation

3.1 Pulizia Test Set

Il test degradato presenta diversi problemi, si è sperimentato un approcio di pulizia automatico.

Idea: soglie su diversi paramenti, in modo da riconoscere il problema ed attuare una modifica specifica per il problema in questione.

Soglie

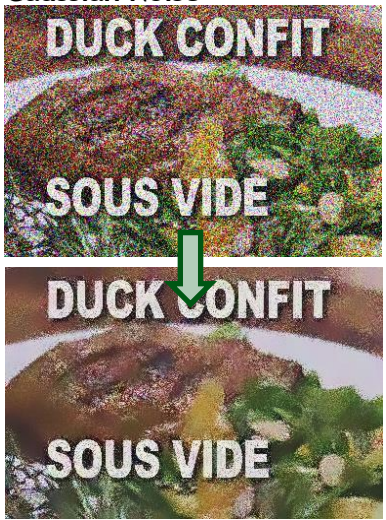
- Blur → Laplacian var < 20
- Gaussian noise → Estremo destro istogramma saturazione > 2000
- Compressione jpeg → BRISQUE > 90

Soglia ottimizzata su random sample di 100, classificato manualmente, e poi applicato all'intero test set.

Acc	π normali	π noise	π blur	π jpeg comp
0.87	0.95	0.67	1.00	0.46

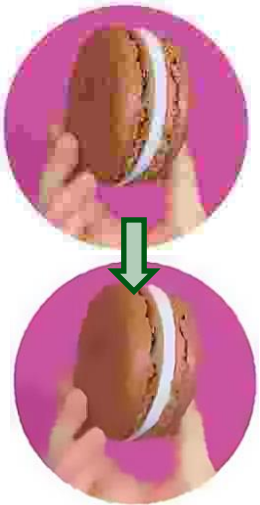
3.1 Pulizia Test Set: risultati pulizia

Gaussian Noise



Blur

Compressione jpeg

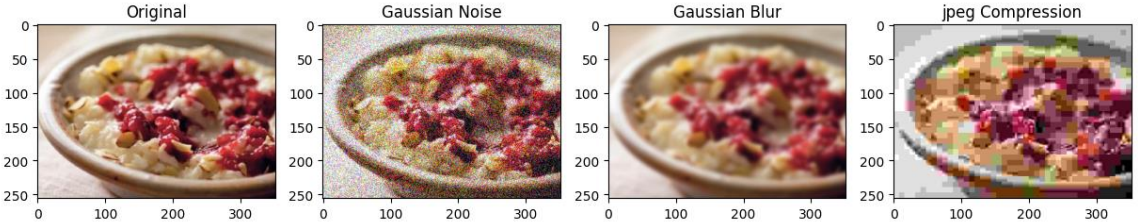


3.2 Data Augmentation

Data agumentation per addestrare i modelli su immagini aventi gli stessi problemi del test set degradato.

Vantaggi

- Aumento generizabilità
- Aumento osservazioni
- Riduzione overfitting



3.3 Modelli

Data la mole di dati, la difficoltà del task e i limiti computazionali, si è optato per un addestramento dei modell in **fine-tuning**. [4]

I modelli sono stati addestrati con e senza data agumentation.
Le perfromance sono state valutate sui tre test set (set base, set degradato, set ripulito).
Lo obbiettivo è quello di ottenere le perfromance più alte possibili sul set degradato (o ripulito).

Architetture

- ResNet50 (baseline)
- ResNet101
- EfficentNetB0

[4] Kaur P.et al., FoodX-251: A Dataset for Fine-grained Food Classification

3.3 Modelli: fine tuning

Modelli addestrati in fine tuning (pesi da ImageNet [5]).
Freeze dei pesi fino all'ultimo blocco convluzionale dei modelli pre addestrati.
Valutazione finale attuata con **Accuracy Top1** e **Accuracy Top 5**.

Modifiche Architettura

- \- Input layer
- \- Output layer
- \+ Input layer (224, 224, 3)
- \+ Global average pooling 2D
- \+ Dropout 0.3 (per ridurre l'overfitting)
- \+ Fully connected layer, con softamax (251)

Parametri

- Epoche = 10
- Batch size = 64
- Learning rate = 1e-4
- Optimizer = Adam
- Decay = 0.1 (ogni 5 epoche)
- Regularizer = 1E-2 (ultimo layer)
- Loss function = Categorical cross entropy

[5] Russakovsky, Olga, et al. Imagenet large scale visual recognition challenge

3.4 Evaluation

	Top 1 Acc Test set	Top 5 Acc Test set	Top 1 Acc Test set degradato	Top 5 Acc Test set degradato	Top 1 Acc Test set ripulito	Top 5 Acc Test set ripulito
ResNet50	0.3972	0.6737	0.2825	0.5165	0.2721	0.4986
ResNet50 d.a.	0.3318	0.6124	0.2659	0.5275	0.2480	0.4924
ResNet101	0.4982	0.7605	0.3314	0.5414	0.3319	0.5428
ResNet101 d.a.	0.5193	0.7826	**0.4745**	**0.7412**	0.4273	0.6940
EfficentNetB0	0.5548	0.8173	0.3740	0.6005	0.3652	0.5880
EfficentNetB0 d.a.	0.4928	0.7719	0.4467	0.7231	0.4025	0.6675

3.4 Evaluation: conclusione

Sebbene il modello con le performance migliori sul test set risulti la EfficentNetB0, l'obbiettivo è quello di ottenere le perfomance più alte possibili sul test set degradato (o ripulito).

Il modello migliore risulta quindi la **ResNet101** con data agumentation (acc = 47%).

Il migliormento del del test set degradato è risultato fallimentare.

Motivazione Performance

Sin dall'inizio erano attese delle perfromance modeste, questo è dovuto alla difficolta del task

- Numero di classi
- Limitatezza computazionale
- Risoluzione parziale problemi training set

Outline: Classificazione Immagini PDF

1. Analisi Dataset
2. Pulizia Train
3. Classificazione
4. Classificazione Immagini PDF
    1. Estrazione immagini
    2. Classificazione

4.1 Estrazione Immagini: immagini pagine

Per l'estrazione delle immagini dal pdf si è optato per un approcio che non richedesse l'utilizzo di modelli per il riconoscimento di immagini.
Piuttosto si è costruito "da zero" un algortimo, che è risultato perfomante (nel caso studio)

Step:

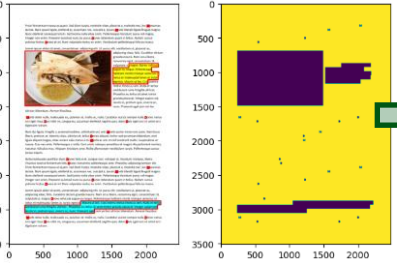
Ogni pagna del pdf in un immagne
Per ogni pagina

1. Maschera basata sulla luminosità
2. Separazione maschere con griglia verticale e orizzontale (+ rimozione impurità)
3. Estrazione singole immagini (+ quadratizzazione maschera)
4. Rimozione immagini con pochi colori (di testo)

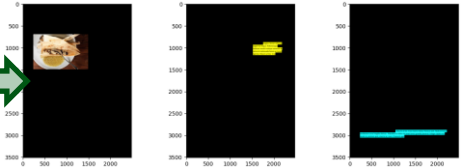
Salvataggio immagini

4.1 Estrazione Immagini: schema

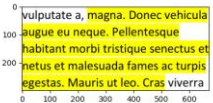
1.



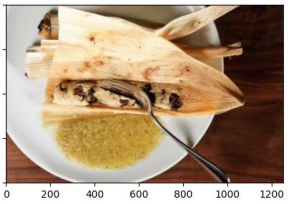
2.



3.



4.

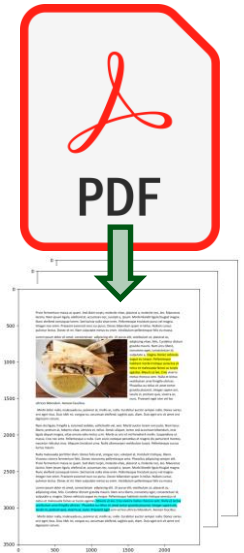


4.1 Estrazione Immagini: pagine in immagini

Ogni pagina viene convertita in un Pixmap.

La Pixmap contiene informazioni pixel per pixel sull'immagine (larghezza e l'altezza dell'immagine, i dati dei pixel, e il formato di colore utilizzato).

L'oggetto Pixmap viene convertito in un array NumPy. Questo array rappresenta l'immagine nella pagina.

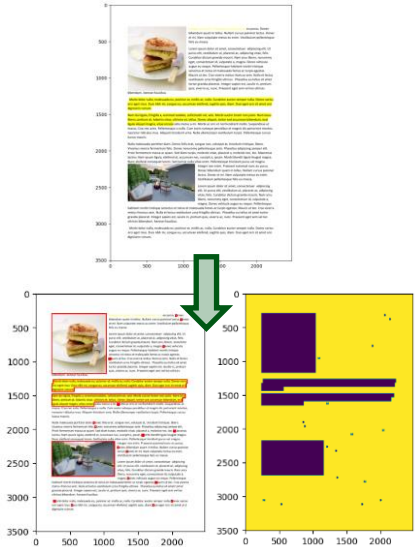


4.1 Estrazione Immagini: maschera luminosità

L'immagine viene convertita in scala di grigi.
Si crea una maschera binaria sui valori di luminosità dell'immagine in scala di grigi (>0.99).

Apertura morfologica sulla maschera binaria. L'apertura è una combinazione di erosione seguita da dilatazione, e utilizzata per rimuovere piccoli dettagli da un'immagine binaria.

- Vengono selezionate anche sezioni di testo evidenziate
- Rimangono piccole imperfezioni nella maschera

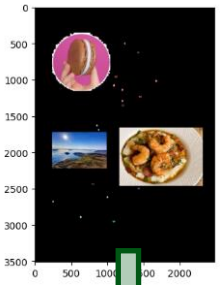


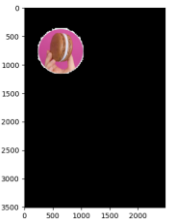
4.1 Estrazione Immagini: separazione maschere

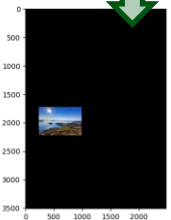
Applicazione di due filtri per la separazione delle maschere, che attuano un controllo separando i gruppi di 0 nella maschera binaria.

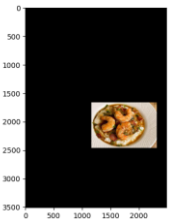
Prima in orizzontale e poi in verticale. In modo da avere una maschera per immagine.

Con pulizia dalle maschere più piccole di 1/100 Dell'intera pagina









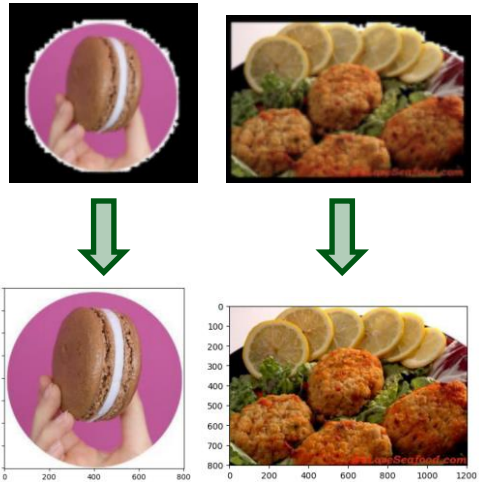
4.1 Estrazione Immagini: estrazione e quadratizzazione

La funzione ConvexHull trova l’involucro convesso dei vertici del contorno.

Vengono estratti i vertici dell’involucro.

Viene calcolato il rettangolo delimitatore dell'immagine basato sui vertici del contorno.

Viene estratta un'immagine ritagliata dall'immagine di della pagina del PDF utilizzando il rettangolo delimitatore calcolato.

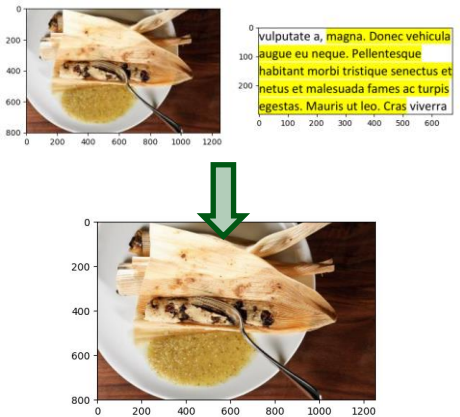


4.1 Estrazione Immagini: rimozione testi

Vengono identificati i colori unici presenti nell'immagine.

I colori unici sono pixel con almeno almeno un canale differente.

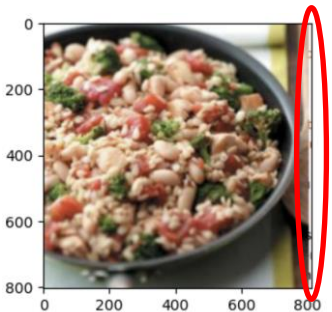
Vengono eliminate le immagini con un numero di colori unici <10000 (considerate testo).



4.1 Estrazione Immagini: conclusione

Vengono correttamente estratte tutte le immagini dal PDF e correttamente nessun testo viene considerato immagine.

Vengono riscontrati due problemi:



4.2 Classificazione

Obbiettivo: classificazione delle immagini estratte dal PDF in food/non-food.

Si è scelto di usare l'architettura più performante sul test set (EfficentNetB0).

Il modello è stato addestrato sul dataset **Food-5K image dataset.** [6]

Data Augmentation

- Rotazioni (40)
- Zoom (0.2)
- Flip orizzontale
- Flip verticale
- Range luminosità (0.5, 1.5)

Parametri

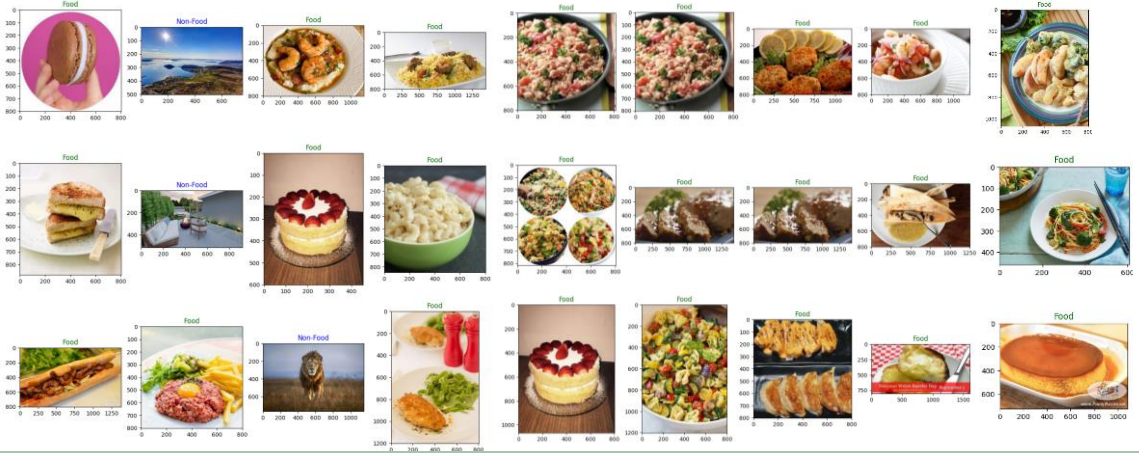
- Epoche = 50 (early stop a 34)
- Batch size = 64
- Learning rate = 1e-4
- Optimizer = Adam
- Decay = 0.8 (ogni 5 epoche)
- Regularizer = 1E-2 (ultimo layer)
- Loss function = Categorical cross entropy

	Top 1 Acc
EfficentNetB0	0.9890

[6] Alekstander Antonov, Food-5k image dataset (https://www.kaggle.com/datasets/trolukovich/food5k-image-dataset)

4.2 Classificazione: applicazione sul PDF

Il modello classifica correttamente tutte le immagini prsenti nel PDF caso studio.



Grazie per l’attenzione!
Domande?