

# Robust Food Classification Project

Visual Information Processing and Management  
Università degli Studi di Milano-Bicocca

Matteo Breganni 869549  
Lorenzo Monti 869960



# Esplorazione del Dataset

- 251 Categorie
- 20 Immagini per classe
- Immagini di **dimensioni** diverse
- Immagini **errate**
- Immagini non esattamente appartenenti alla categoria

Image 2 - Size: 256x768

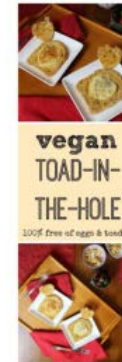


Image 2 - Size: 224x224



Image 1 - Size: 256x309



Image 2 - Size: 256x331



Image 3 - Size: 384x256



Image 4 - Size: 384x256



Image 5 - Size: 340x256



Image 6 - Size: 387x256



Image 7 - Size: 256x384



Image 8 - Size: 256x384



# Esplorazione del Dataset

- Categorie a volte molto **ampie**

Dal validation set:

Image 1 - Size: 466x256



Image 2 - Size: 256x341



Image 3 - Size: 256x384



Image 4 - Size: 341x256



Image 5 - Size: 409x256



Image 6 - Size: 488x256



Image 7 - Size: 256x271



Image 8 - Size: 454x256

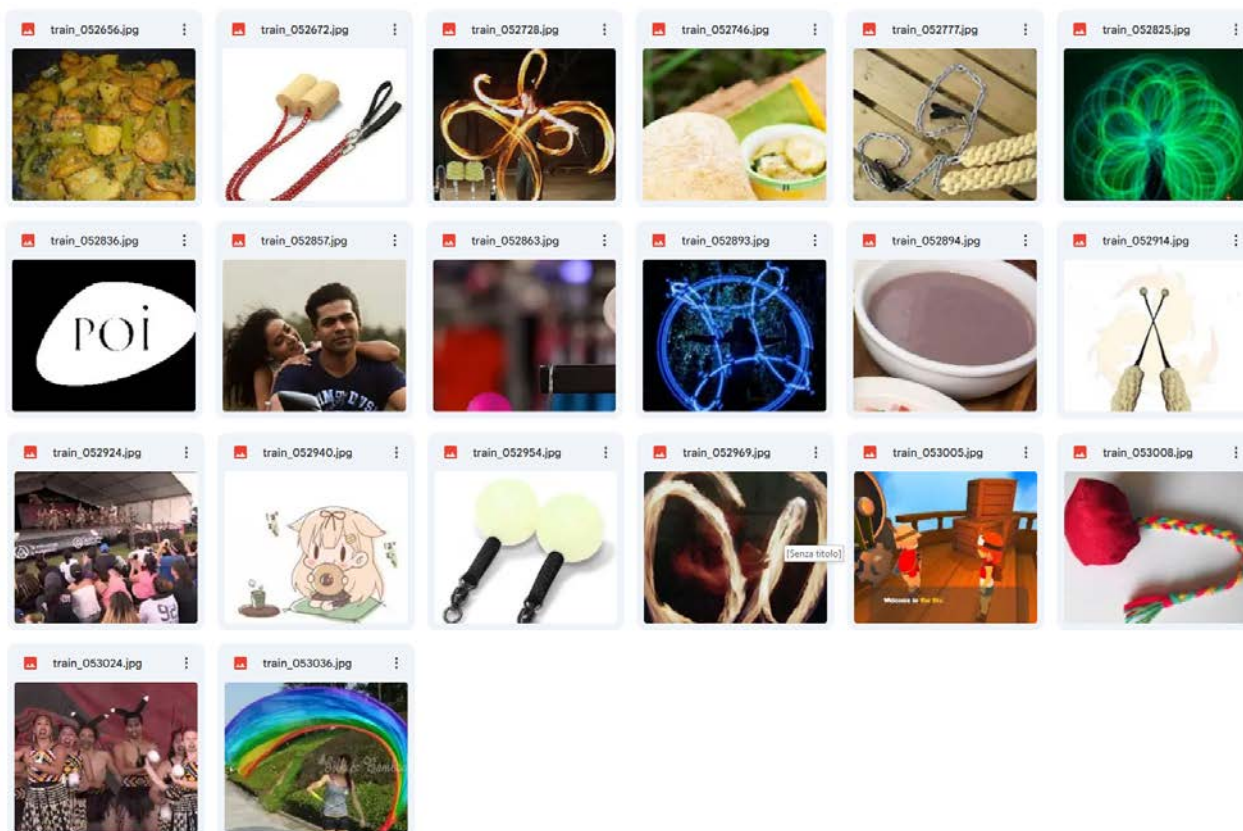




# Esplorazione del Dataset

- Categorie con principalmente **immagini sporche**
- Relativo validation set **piccolo**

Training set:



Validation set:

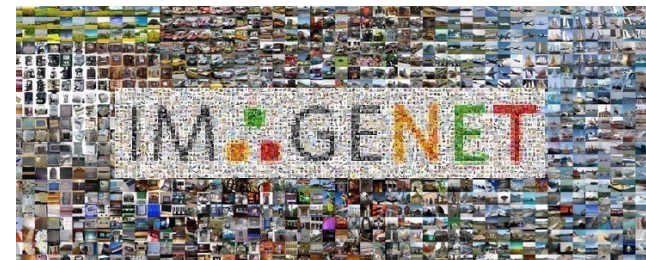


# Approcci Iniziali: Feature Extraction

- **Features extraction** con il modello pre-trainato di MobileNetV2 sul dataset Imagenet

## Metodi:

- **KNN**
  - 15% di Accuracy
- **SVM**
  - 27% di Accuracy
  - Con ottimizzazione parametri: 28% di Accuracy
- Tentativo di utilizzare il penultimo layer al posti dell'ultimo per feature extraction:
  - ~10GB di array di features vs ~80MB



# Approcci Iniziali: CNN Fatta a Mano

## Approccio generale:

- Data loaders per ridurre la RAM utilizzata
- Validation set ridotto durante il training per ridurre le computazioni (50 batches)
- Immagini pre-processate e labels con one-hot encoding, softmax output

## CNN creata e allenata da zero:

- Normalizzazione tra 0 e 1
- Performance molto basse causate dalla bassa grandezza del dataset
  - Test Loss: ~9.70
  - Test Accuracy (Top-1): ~1.60%
  - Top-5 Accuracy: ~5.70%

Layer	Output shape	Param #
Input	(224, 224, 3)	/
Conv, 16, (3, 3), stride 2	(111, 111, 16)	448
Batch normalization	(111, 111, 16)	64
Conv, 32, (3, 3), stride 2	(55, 55, 32)	4'640
Batch normalization	(55, 55, 32)	128
Conv, 64, (3, 3), stride 2	(27, 27, 64)	18'496
Batch normalization	(27, 27, 64)	256
Conv, 128, (3, 3), stride 2	(13, 13, 128)	73'856
Conv, 128, (3, 3), stride 2	(6, 6, 128)	147'584
Flatten	(4608)	0
Dense, 256, 50% dropout	(256)	1'179'904
Output (softmax)	(251)	64'507

# Approcci Iniziali: CNN Fine-Tunata

## CNN Fine tunata MobileNetV2:

- Pesi congelati sulla prima parte
- Layer fully connected con dropout aggiunto
- Buona performance
  - Test Loss: ~3.49
  - Test Accuracy (Top-1): ~25%
  - Top-5 Accuracy: ~50%
- Totale pesi: 3'042'619
- Totale pesi allenabili: 784'635

Layer	Output shape	Param #
Input	(224, 224, 3)	/
MobileNetV2	(7, 7, 1280)	2'257'984
GlobalAveragePooling	(1280)	0
Dense, 512, 65% dropout	(512)	655'872
Output	(251)	128'763

## CNN Fine tunata DenseNet121:

- Stesso funzionamento
- Rete più grande
  - Totale pesi: 7'691'067
  - Totale pesi allenabili: 653'563
- Performance superiori
  - Test Loss: ~3.28
  - Test Accuracy: ~26%
  - Top-5 Accuracy: ~53%
- Tempi simili per il training ma più alti per l'inferenza

Layer	Output shape	Param #
Input	(224, 224, 3)	/
DenseNet121	(7, 7, 1024)	7'037'504
GlobalAveragePooling	(1024)	0
Dense, 512, 65% dropout	(512)	524'800
Output	(251)	128'763

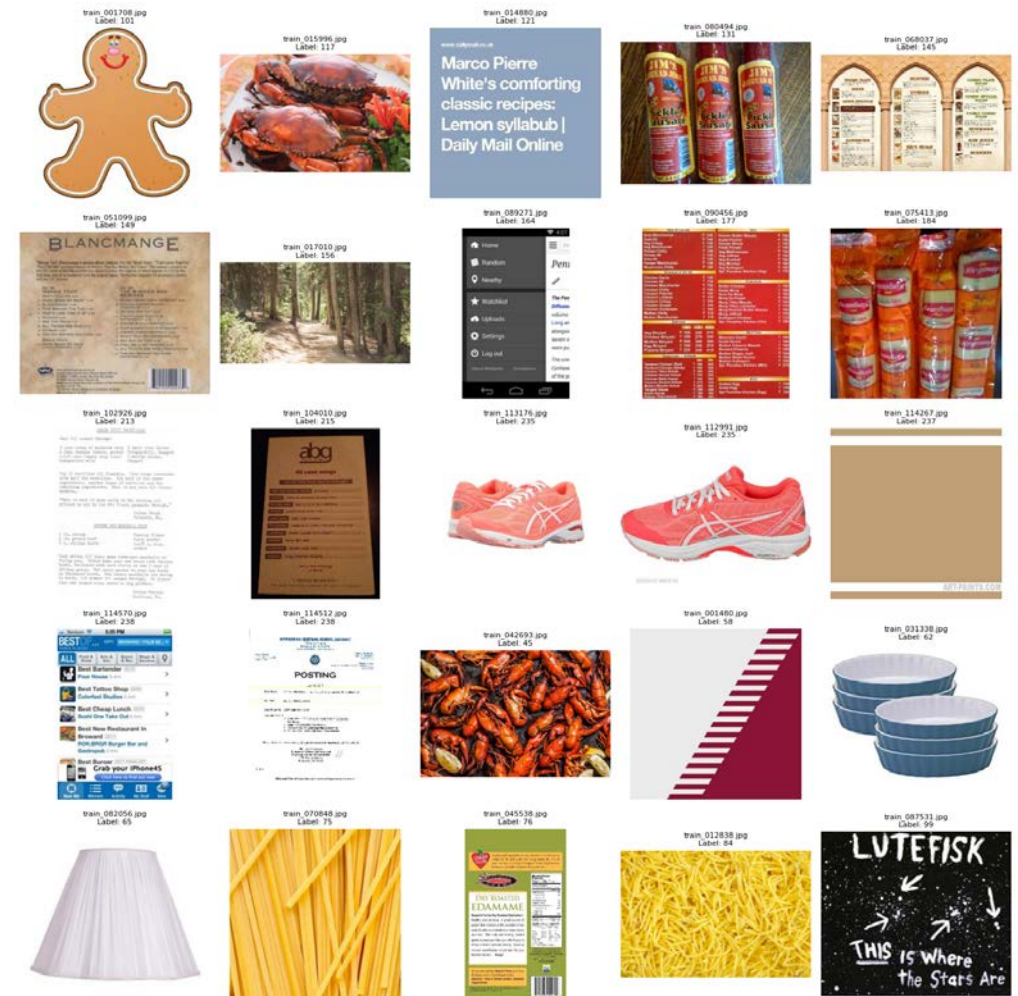


# Pulizia del Training Set: Metodi

- Metodi per il riconoscimento degli outliers nel train set
  - Obiettivo: rimuovere le immagini sporche per migliorare la performance
- Test per valutare l'efficacia di ciascun metodo:
  - Tuning della sensibilità di ciascun metodo per ritornare 25 outliers
  - Calcolo manuale di 2 score verificando sul val set:
    - Numero di immagini correttamente flaggate
    - Stesso score, sommato agli edge cases

Metodi:		score1	score2
• K-Means	->	7/25	9/25
• DBSCAN	->	15/25	17/25
• Isolation Forest	->	15/25	17/25
• NearestNeighbors	->	20/25	22/25
• SVM	->	6/25	7/25
• KD-Tree	->	8/25	10/25

- Performance diverse e non ottime
- La sensibilità di ciascun metodo può essere adattata per ritornare un diverso numero di immagini.



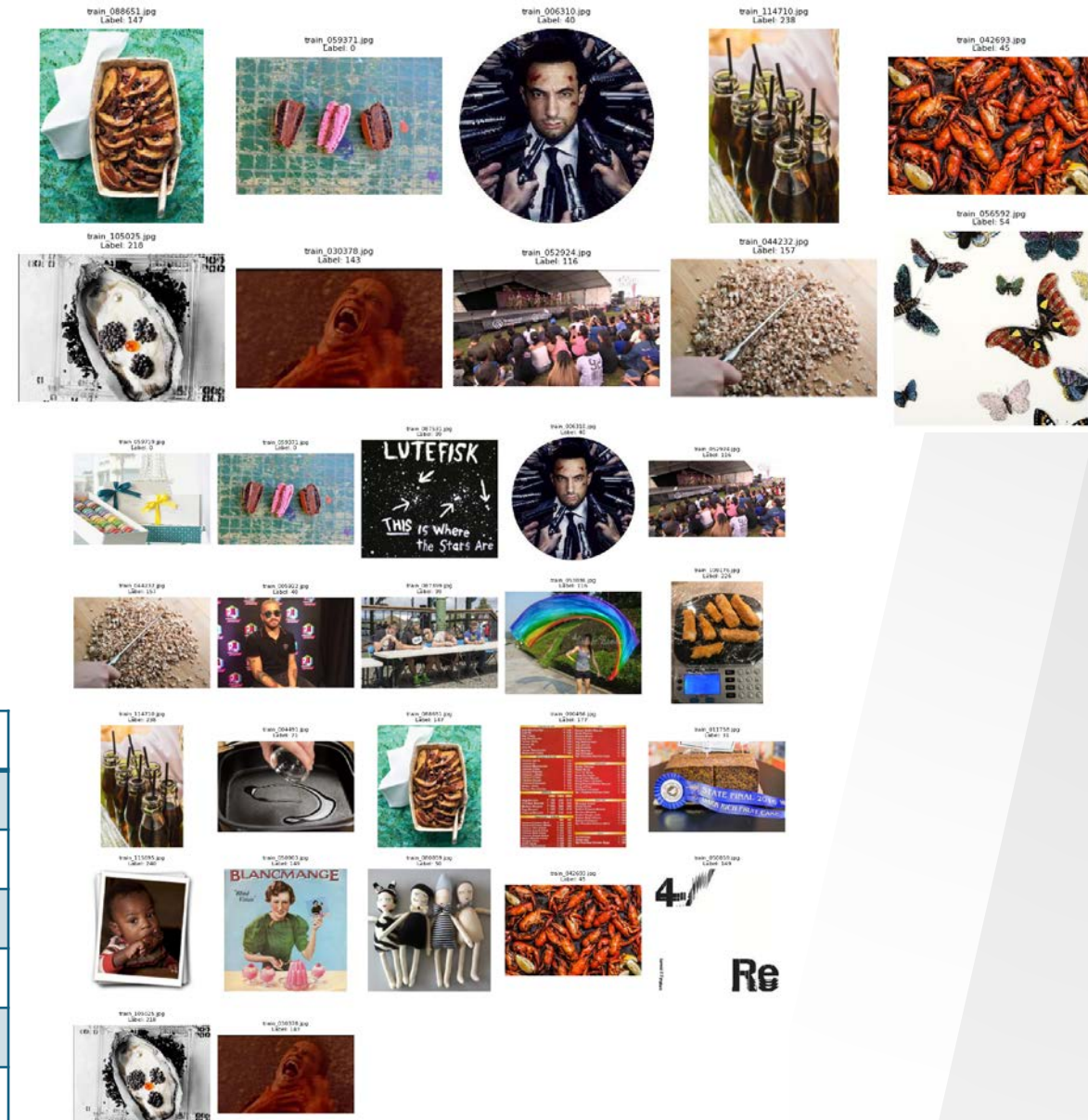
25 outliers ritornati dal metodo NearestNeighbors



# Pulizia del Training Set: Consenso

- Performance dei metodi troppo basse per essere usati
- Rischio di rimuovere troppe immagini buone supera i benefici di togliere quelle sporche
- Metodi di consenso:
  - Metodi diversi possono commettere errori diversi
  - Outliers trovati da almeno n metodi
    - Risultato con n=3 sui test da 25 immagini --->
    - Metodo non ottimo perché ogni metodo ha la stessa capacità di voto
  - Consenso pesato maggiore di n
    - Utilizzando gli score calcolati come stima di efficacia
    - Calcolo pesi (somma a 1)
      - Risultato test con soglia 0.3 ----->

Metodo	Score1	Score2	Peso1	Peso2
K-Means	7/25	9/25	<b>0.1</b>	0.11
DBSCAN	15/25	17/25	<b>0.21</b>	0.21
Isolation Forest	15/25	17/25	<b>0.21</b>	0.21
NearestNeighbors	20/25	22/25	<b>0.28</b>	0.27
SVM	6/25	7/25	<b>0.08</b>	0.09
KD-Tree	8/25	10/25	<b>0.11</b>	0.12



# Pulizia del Training Set: Risultati

- 500 immagini per metodo
  - Cambiando la sensibilità di ciascuno
- Applicando la funzione di consenso pesata
- Threshold a 0.5, rimuovendo 422 immagini:
  - Test Loss: ~3.40
  - Test Accuracy: ~25%
  - Top-5 Accuracy: ~51%
    - Performance peggiorata, i benefici di rimuovere le immagini sporche non superano i problemi del togliere immagini buone
- Threshold a 0.8, rimuovendo 91 immagini:
  - Test Loss: ~3.32
  - Test Accuracy: ~ 26%
  - Top-5 Accuracy: ~ 53%
    - Performance simili a prima
- Come migliorare?
  - Provare un numero diverso di outliers da ritornare per metodo (richiede tempo per fare tuning manual dei parametri)
  - Provare threshold diversi per la funzione di consenso
  - Sfruttare più dati per avere outliers migliori

# Pulizia del Training Set: Unlabelled

- Idea: sfruttare i dati dell'unlabelled training set per migliorare gli outliers trovati da ciascun metodo
  - Più dati aiutano a definire i clusters meglio
  - Il dataset unlabelled sembra avere molte meno immagini sporche
  - 30% del dataset unlabelled usato, per motivi di computazione (34'036 immagini)
- Ricalcolo dei 500 outliers per metodo
  - Funzione di consenso pesata, con gli stessi pesi precedenti
    - Supponendo che l'efficienza relativa di ogni metodo rimanga invariata
- Threshold 0.90 -> 95 immagini ritornate
  - Il threshold necessario per avere un numero simile di immagini è aumentato, indice che la «confidenza» del metodo è migliorata
- Performance raggiunte:
  - Test Loss: ~3.22 (-3%~\*)
  - Test Accuracy: ~28% (+7%~\*)
  - Top-5 Accuracy: ~55% (+4%~\*)
- Possibili miglioramenti:
  - Tuning numero outliers per metodo, e threshold della funzione di consenso
  - Ricalcolo dei pesi di efficacia relativa

# Data Augmentation

- Può essere utile dato il piccolo numero di immagini nel train set
- Augmentation applicata al miglior dataset pulito

Trasformazioni provate singolarmente:

- **Traslazioni**
  - Translation 0.1 -> Test Loss: 3.4143; Test Accuracy: 0.2496; Top-5 Accuracy: 0.5113
  - **Translation 0.2** -> Test Loss: 3.4257; Test Accuracy: 0.2505; Top-5 Accuracy: 0.5135
  - Translation 0.23 -> Test Loss: 3.5944; Test Accuracy: 0.2491; Top-5 Accuracy: 0.5079
- **Luminosità**
  - RandomBrightness 0.1 -> Test Loss: 3.2584; Test Accuracy: 0.2629; Top-5 Accuracy: 0.5381
  - **RandomBrightness 0.12** -> Test Loss: 3.4441; Test Accuracy: 0.2684; Top-5 Accuracy: 0.5307
  - RandomBrightness 0.2 -> Test Loss: 3.4787; Test Accuracy: 0.2480; Top-5 Accuracy: 0.5011
- **Rotazione**
  - RandomRotation 0.15 -> Test Loss: 3.6059; Test Accuracy: 0.2473; Top-5 Accuracy: 0.5088
  - **RandomRotation 0.2** -> Test Loss: 3.4162; Test Accuracy: 0.2540; Top-5 Accuracy: 0.5055
  - RandomRotation 0.6 -> Test Loss: 3.5778; Test Accuracy: 0.2214; Top-5 Accuracy: 0.4676
- **Contrasto**
  - **RandomContrast 0.1** -> Test Loss: 3.4983; Test Accuracy: 0.2405; Top-5 Accuracy: 0.4946
  - RandomContrast 0.15 -> Test Loss: 3.5525; Test Accuracy: 0.2250; Top-5 Accuracy: 0.4748
  - RandomContrast 0.2 -> Test Loss: 3.5525; Test Accuracy: 0.2250; Top-5 Accuracy: 0.4748
- Combinazioni di trasformazioni diverse



# Modello Unfrozen e Modello Pesato

## Modello iniziale migliorato:

- Dataset migliore dal data cleaning
- Modello ottimizzato ulteriormente

Test Loss: ~3.08 (-4%~\*)

Test Accuracy: ~30% (+7%~\*)

Top-5 Accuracy: ~57% (+4%~\*)

## Modello Unfrozen:

- Fine tuning ulteriore
  - Modello base unfrozen (DenseNet)
  - Learning rate più basso
- Tempi di training molto lunghi

Test Loss: ~2.96 (-4%~\*\*)

Test Accuracy: ~32% (+7%~\*\*)

Top-5 Accuracy: ~60% (+5%~\*\*)

## Modello pesato:

- Assegnazione di pesi alle classi, relativamente alla distribuzione
  - Procedimento di cleaning ha sbilanciato il dataset
  - Verrà sbilanciato ulteriormente nella sezione successive
- Performance che dovrebbero migliorare ma classi problematiche che hanno perso esempi (come la 116) hanno ancora immagini sporche
- Questo modello diventa la baseline e viene utilizzato nelle successive sezioni

Test Loss: ~3.07 (0%~\*\*)

Test Accuracy: ~30% (0%~\*\*)

Top-5 Accuracy: ~58% (+2%~\*\*)

\*Aumento percentuale rispetto al modello migliore del data cleaning con dataset unlabelled

\*\*Aumento percentuale rispetto al modello iniziale migliorato

# Train Set Unlabelled

- Aumento dimensione train set aggiungendo dal dataset senza label
- Metodi per selezionare dati da aggiungere, usando il modello pesato per trovare le labels:
  - Confidenza della top class  $>$  soglia
  - Confidenza della top class con margine  $>$  soglia rispetto alla confidenza della seconda classe
- Approccio iterativo:
  - Set della confidenza per ritornare immagini classificate bene (iniziando con confidenza alta, abbassata dopo ogni iterazione)
  - Filtrare le immagini ritornate dal metodo, togliendo quelle già aggiunte
  - Limitare al massimo 3 immagini aggiunte per classe ogni iterazione per ridurre lo sbilanciamento
  - Re-training del modello con il train set nuovo e ri-calcolando i pesi delle classi
- Dopo aver esaurito le immagini a confidenza molto alta, usare il secondo metodo
- Solo il 10% del dataset senza labels è stato utilizzato
  - Sempre lo stesso 10%, non random ad ogni iterazione, per esaurire effettivamente le immagini ad alta confidenza

# Train Set Unlabelled

Test iniziale del metodo, usando 1% del dataset senza label, con soglia di confidenza al 90%:

- 19 immagini ritornate
  - Prime 12:

21



49



101



76



21



189



229



51



71



15



193



68



- Controllate manualmente rispetto al validation set, tutte corrette tranne la quinta

# Train Set Unlabelled

Iterazioni utilizzando sempre lo stesso 10% del dataset senza labels (performance scalerebbero di conseguenza usando l'intero set di dati:

Metodo	Soglia	Immagini trovate	Meno imgs già usate	Massimo 3 per classe	Dimensione dataset	Loss	Top-1	Top-5
/	/	/	/	/	4'925	~3.07	~30%	~58%
Threshold	0.9	219	219	127	5'052	~3.05	~31%	~58%
Threshold	0.8	497	370	205	5'257	~3.04	~31%	~58%
Threshold	0.75	768	438	228	5'485	~3.03	~31%	~59%
Threshold	0.70	982	427	233	5'718	~3.04	~31%	~58%
Threshold	0.65	1500	711	354	6'072	~3.01	~32%	~59%
Margin	0.60	526	517	391	6'463	~3.01	~32%	~59%
Margin	0.60	526	126	126	6'589	<b>~3.00</b>	<b>~33%</b>	<b>~60%</b>

**Modello finale:** ultimo modello, togliendo i freeze, fine-tunato ulteriormente:

Test Loss: ~2.93 (-1%~\*)

Test Accuracy: ~**34%** (+6%~\*)

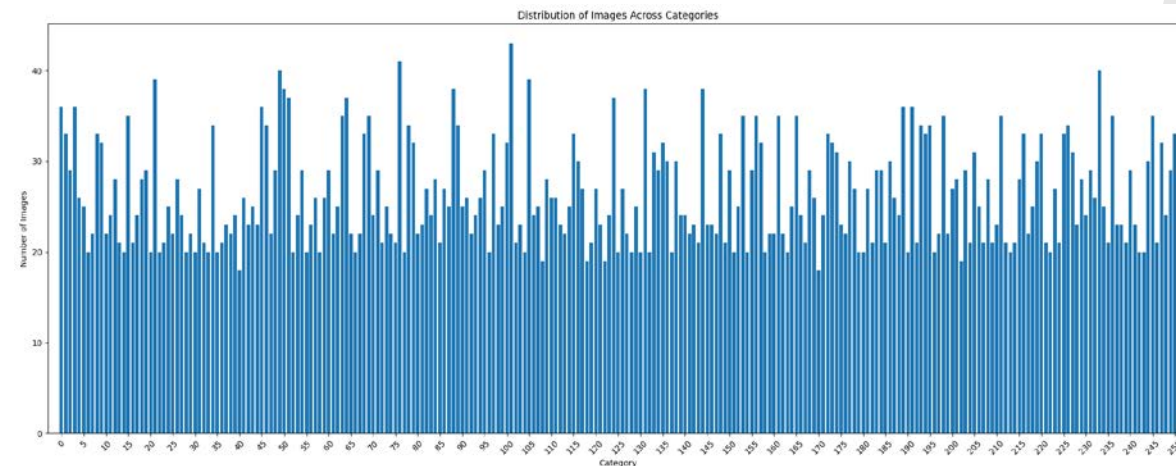
Top-5 Accuracy: ~**61%** (+2%~\*)

\*Aumento percentuale rispetto al modello unfrozen precedente, ovvero quello dopo la pulizia del training set



# Valutazione Generale

- Valutazioni sul dataset in generale
- Distribuzione delle classi nel dataset finale ----->



- Confusion matrix e classification report illeggibili
- Dataframe navigabili
- Ordinato per top-1 accuracy:

Class	Top-1 Accuracy	Top-k Accuracy
88	0.775862	0.862069
164	0.756757	0.972973
76	0.754717	0.905660
168	0.723077	0.876923
4	0.707317	0.878049
124	0.704545	0.886364
250	0.700000	0.940000
144	0.695652	0.826087
45	0.695652	0.826087
63	0.693878	0.918367

Ordinato per top-5 accuracy:

Class	Top-1 Accuracy	Top-k Accuracy
164	0.756757	0.972973
60	0.566038	0.943396
250	0.700000	0.940000
63	0.693878	0.918367
134	0.666667	0.907407
76	0.754717	0.905660
204	0.561404	0.894737
124	0.704545	0.886364
174	0.534884	0.883721
191	0.676471	0.882353

- Nessuna class ha una top-1 accuracy sopra all'80%
- 6 classi con top-5 accuracy sopra al 90%

# Valutazione Generale

- Ordinamento per top-5 accuracy, invertito
- 6 classi con top-1 accuracy 0
- 2 classi con top-5 accuracy 0
- Questi dataframe ci danno indicazioni molto utili su quali classi andare ad analizzare

Class	Top-1 Accuracy	Top-k Accuracy
213	0.000000	0.000000
239	0.000000	0.000000
85	0.000000	0.078947
238	0.000000	0.090909
99	0.047619	0.095238
84	0.050000	0.100000
65	0.041667	0.125000
40	0.000000	0.142857
132	0.000000	0.181818
58	0.022727	0.181818

# Valutazione Classi Migliori

- Migliore classe per top-1: classe 88
- Train set:

train\_006245.jpg



train\_007324.jpg



train\_007435.jpg



train\_007513.jpg



train\_007437.jpg



train\_072618.jpg



train\_007625.jpg



train\_007315.jpg



train\_007383.jpg



train\_007231.jpg



train\_007212.jpg



train\_007300.jpg



- Val set:

val\_003888.jpg



val\_001333.jpg



val\_003385.jpg



val\_001683.jpg



val\_009512.jpg



val\_007114.jpg



val\_007297.jpg



val\_005160.jpg



val\_000811.jpg



val\_007449.jpg



val\_008394.jpg



val\_005415.jpg



# Valutazione Classi Migliori

- **Migliore classe per top-1: classe 88**
- Classi predette per le immagini di test:

Classe	n
88	30
2	8
142	2
161	2
170	1
194	1
...	...

- Classe confusa più spesso nel top-1:
  - Classe 2

train\_062381.jpg



train\_062538.jpg



train\_062644.jpg



train\_062692.jpg



train\_062412.jpg



train\_016651.jpg



train\_062491.jpg



train\_062756.jpg



train\_062726.jpg



train\_062605.jpg



train\_062665.jpg



train\_062761.jpg





# Valutazione Classi Migliori

- **Migliore classe per top-1: classe 88**
- Quali categorie appaiono più spesso nella top-5 delle immagini di test di questa categoria:

Classe	n
<b>88</b>	<b>50</b>
133	21
226	16
155	12
2	8
153	7
...	...

- Classe con più similarità:
  - Classe 133
  - Sembra identica

- Man mano diventano meno simili

val\_003792.jpg



val\_003681.jpg



val\_003131.jpg



val\_004525.jpg



val\_004554.jpg



val\_003081.jpg



val\_005572.jpg



val\_002356.jpg



val\_003648.jpg



val\_009936.jpg



train\_108807.jpg



train\_010193.jpg



train\_109120.jpg



Classe 226:

train\_054068.jpg



train\_054263.jpg



train\_054336.jpg



Classe 155:

# Valutazione Classi Migliori

- Seconda migliore classe per top-1: classe 164
- Train set:



- Quali categorie appaiono più spesso nella top-5 delle immagini di test di questa categoria:

Classe	n
164	36
26	31
100	25
232	11
106	11
201	5
...	...



# Valutazione Classi Migliori

- Tutte classi di panini, che man mano diventano meno simili

- Classe 26:

train\_033629.jpg



train\_033826.jpg



train\_033739.jpg



train\_033884.jpg



train\_033689.jpg



train\_004110.jpg



- Classe 100:

train\_004084.jpg



train\_004234.jpg



train\_033738.jpg



train\_004426.jpg



train\_004388.jpg



train\_004183.jpg



- Classe 232:

train\_111988.jpg



train\_058623.jpg



train\_112118.jpg



train\_112072.jpg



train\_111750.jpg



train\_111726.jpg



- Classe106:

train\_066785.jpg



train\_066536.jpg



train\_066867.jpg



train\_066665.jpg



train\_066631.jpg



train\_066757.jpg



# Valutazione Classi Peggiori

- **Peggior classe per top-5: classe 213**
- Mai presente nella top-5 delle immagini di test della sua classe
- Train set: (il train set prima dell'incremento è molto simile)

train\_102950.jpg



train\_102846.jpg



train\_102848.jpg



train\_102933.jpg



train\_102878.jpg



train\_102908.jpg



train\_102850.jpg



train\_103091.jpg



train\_102895.jpg



train\_103008.jpg



train\_102926.jpg



train\_102817.jpg



- Val set: immagini molto diverse tra di loro

val\_012114.jpg



val\_012123.jpg



val\_011420.jpg



val\_011687.jpg



val\_011351.jpg



val\_010379.jpg



val\_011174.jpg



val\_012014.jpg



val\_011011.jpg



val\_010692.jpg



val\_012046.jpg



val\_010992.jpg





# Valutazione Classi Peggiori

- **Peggior classe per top-5: classe 213**
- Classi più predette sulle immagini di test della classe 213 molto sparse ----->

Classe	n
128	15
76	14
28	13
242	10
53	8
97	7
...	...

- Classi più presenti nella top-5:

171: 2  
130: 1  
196: 1  
242: 1  
128: 1  
15: 1  
234: 1  
190: 1  
108: 1  
139: 1  
160: 1  
28: 1  
42: 1  
14: 1  
120: 1  
231: 1  
175: 1  
49: 1  
237: 1  
158: 1  
185: 1  
125: 1

# Valutazione Classi Peggiori

- **Peggior classe per top-5: classe 213**
- Classi più presenti nella top-5:
  - Contengono elementi sparsi presenti nel training set della classe 213
  - Classe 128:

train\_077952.jpg



train\_078064.jpg



train\_078021.jpg



train\_078006.jpg



train\_076159.jpg



train\_078039.jpg



- **Classe 76:**

train\_004011.jpg



train\_045425.jpg



train\_045145.jpg



train\_045328.jpg



train\_045461.jpg



train\_045253.jpg



- **Classe 28:**

train\_056072.jpg



train\_056265.jpg



train\_056253.jpg



train\_056162.jpg



train\_055797.jpg



train\_056264.jpg



# Valutazione Classi Peggiori

- Seconda peggiore classe per top-5: classe 239
  - Immagini di categorie diverse nel training set (anche prima dell'aumento)
  - 5 immagini di biscotti e 9 di torte
  - Train set originale:



- Validation set:





# Valutazione Classi Peggiori

- Seconda peggiore classe per top-5: classe 239
  - Classi più presenti nella top-5:
  - Mix di biscotti e torte

Classe	n
101	8
68	8
31	8
115	5
...	...

Classe 101:



Classe 68:



Classe 31:





# Valutazione Classi Peggiori

- 2 classi peggiori successive nel top-1, con 0 top-1 accuracy
  - Classe 85: mix di immagini diverse (già dal train set originale)

train\_058561.jpg



train\_058728.jpg



train\_058666.jpg



train\_058627.jpg



train\_058383.jpg



train\_058703.jpg



train\_058562.jpg



train\_058427.jpg



train\_058461.jpg



train\_058559.jpg



train\_058492.jpg



train\_058512.jpg



- Classe 238: mix di immagini diverse (già dal train set originale)

train\_114867.jpg



train\_114736.jpg



train\_114512.jpg



train\_114792.jpg



train\_114654.jpg



train\_114786.jpg



train\_114661.jpg



train\_114545.jpg



train\_114553.jpg



train\_114641.jpg



train\_114590.jpg



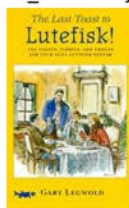
train\_114621.jpg



# Valutazione Classi Peggiori

- **Classe 99:** maggioranza di immagini sporche nel dataset iniziale
  - L'aumento del dataset di train aggiunge ulteriori immagini sporche, dato che questa è la classe di maggioranza
  - Train set originale:

train\_087382.jpg



train\_087532.jpg



train\_087523.jpg



train\_087510.jpg



train\_087214.jpg



train\_087531.jpg



train\_087399.jpg



train\_087232.jpg



train\_087237.jpg



train\_087363.jpg



train\_087317.jpg



train\_087347.jpg



train\_087292.jpg



train\_087535.jpg



train\_087336.jpg



train\_087511.jpg



train\_087362.jpg



train\_087219.jpg



train\_087431.jpg



train\_087403.jpg





# Valutazione su Test Set Reale

- Immagine 1: pizza fatta in casa

Original image: dataset/test\_set\_real/1.jpg



Top-5 predizioni:

Classe	Confidenza
236	52,08%
107	4,74%
183	4,48%
57	3,66%
222	3,47%

Classe 236:

val\_010618.jpg



val\_011688.jpg



val\_011347.jpg



val\_011541.jpg



Classe 107:

val\_000620.jpg



val\_007405.jpg



val\_007615.jpg



val\_000615.jpg



Classe 183:

val\_000316.jpg



val\_007573.jpg



val\_007930.jpg



val\_000126.jpg



# Valutazione su Test Set Reale

- Immagine 2: basilico per pesto

Original image: dataset/test\_set\_real/2.jpg



Top-5 predizioni:

Classe	Confidenza
15	69,82%
76	8,13%
204	6,70%
114	3,28%
94	2,17%

Classe 15:

val\_001405.jpg



val\_007498.jpg



val\_003642.jpg



val\_005145.jpg



Classe 76:

val\_006688.jpg



val\_003523.jpg



val\_008748.jpg



val\_008968.jpg



Classe 204:

val\_002542.jpg



val\_008018.jpg



val\_005702.jpg



val\_006400.jpg





# Valutazione su Test Set Reale

- Immagine 3: strozzapreti fatti in casa

Original image: dataset/test\_set\_real/3.jpg



Top-5 predizioni:

Classe	Confidenza
88	20,85%
243	7,85%
92	5,37%
117	5,19%
153	4,86%

Classe 88:

val\_002372.jpg



val\_006909.jpg



val\_006424.jpg



val\_007114.jpg



Classe 243:

val\_010329.jpg



val\_011239.jpg



val\_010994.jpg



val\_011414.jpg



Classe 92:

val\_000779.jpg



val\_007726.jpg



val\_003844.jpg



val\_005979.jpg



# Valutazione su Test Set Reale

- Immagine 4: lasagne fatte in casa

Original image: dataset/test\_set\_real/4.jpg



Top-5 predizioni:

Classe	Confidenza
157	19,41%
31	11,47%
240	7,28%
119	6,56%
206	6,27%

Classe 157:



Classe 31:



Classe 240:





# Valutazione su Test Set Reale

- Immagine 5: tagliata di manzo

Original image: dataset/test\_set\_real/5.jpg



Classe 217:

val\_011883.jpg



val\_010452.jpg



val\_011283.jpg



val\_010820.jpg



Classe 74:

val\_008642.jpg



val\_000111.jpg



val\_006363.jpg



val\_003734.jpg



Top-5 predizioni:

Classe	Confidenza
217	53,73%
74	8,03%
231	7,24%
158	6,93%
122	2,62%

Classe 231:

val\_012099.jpg



val\_010790.jpg



val\_012101.jpg



val\_010333.jpg

