

# FOOD RECOGNITION

*Laurea magistrale in Ingegneria e scienze informatiche*  
Corso di Teorie e tecniche del riconoscimento  
A.A. 2019/20

Alessia Bodini – VR451051

# DESCRIZIONE del PROGETTO

- ❑ **SCOPO:** identificazione di piatti/cibi da ogni parte del mondo a partire da una sola foto
- ❑ **MOTIVAZIONI:**
  - Indicazione degli ingredienti che compongono cibi non conosciuti
  - Descrizione dei valori nutrizionali
  - Elencazione dei possibili allergeni
- ❑ **OBBIETTIVO FINALE:** analizzare e decretare il metodo di classificazione più adatto



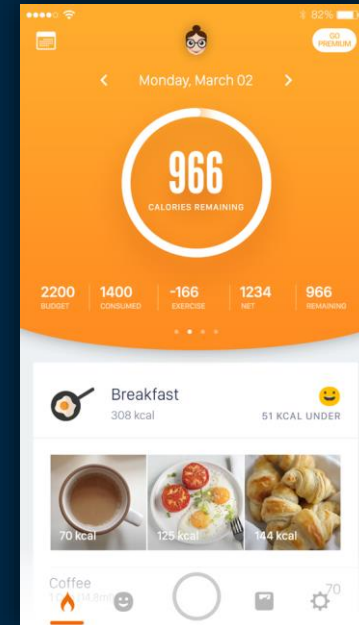
# STATO dell'ARTE

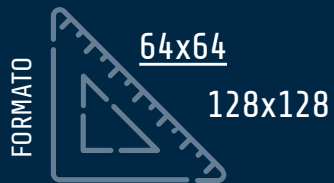
L'applicazione più conosciuta in questo campo è **Calorie Mama**.

- Fa uso di Food AI API, basata sul deep learning
- Si occupa di *instant food recognition* ma non solo
- App disponibile per Apple e Android



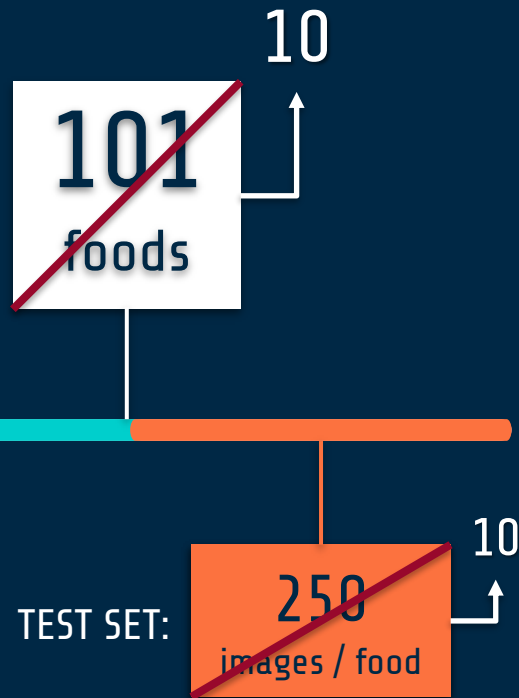
CALORIE  
MAMA





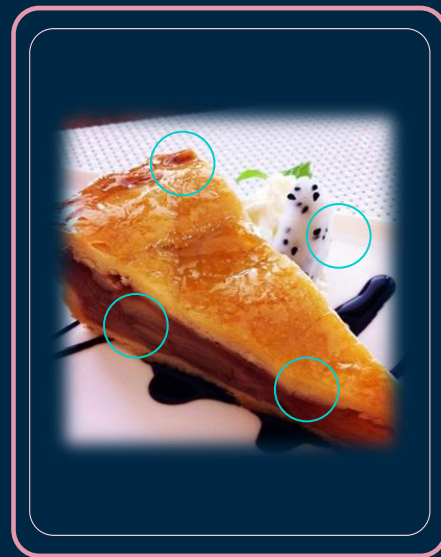
# DATASET

Food-101 by K. Scott Mader  
(<https://www.kaggle.com/kmader/food41>)

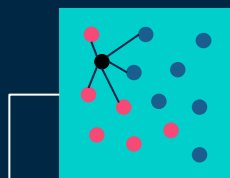


# ESTRAZIONE delle FEATURES

- Tramite la rete neurale ResNet-50  
(dal file "resnet.py")
- 2048 features per ogni immagine

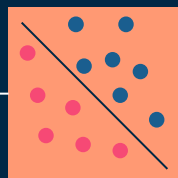


# METODI di RICONOSCIMENTO



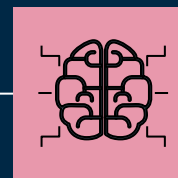
01

KNN



02

SVM



03

Deep NN

# KNN

(K-Nearest Neighbors)


## Metriche:

- Distanza euclidea
- Distanza di Minkowski
- Distanza del coseno
- Correlazione

**K:** 1, 3, 7

Miglior valore di accuratezza ottenuto rispetto a tutti i modelli!

(Precision = 0.33, Recall = 0.34)



Metric \ K	1	3	7
Euclidean	18%	17%	9%
Minkowski	18%	17%	9%
Cosine	<u>20%</u>	19%	12%
Correlation	19%	16%	14%

# SVM

## (Support Vector Machine)

### Kernel:

- Lineare
- Polinomiale
- RBF (con diversi valori per  $\gamma$  e  $C$ )

**N° di iterazioni:** 10, 100, 1000

I valori di accuratezza non sono stabili:  
rieseguendo i test diverse volte ci possono essere  
oscillazioni anche del 10%!

Kernel \ Iter.	10	100	1000
Linear	10%	14%	15%
Polynomial	10%	7%	5%

Kernel \ Iter.	10	100	1000
$\gamma = \text{scale}, C = 1$	15%	<u>18%</u>	16%
$\gamma = \text{scale}, C = 0.1$	8%	17%	16%
$\gamma = \text{auto}, C = 1$	8%	16%	14%
$\gamma = \text{auto}, C = 0.1$	9%	13%	16%

Miglior risultato: 18%  
(Precision = 0.33, Recall = 0.60)



# DNN

## (Deep Neural Network)

### Topologia:

- Conv2d: 3  $\rightarrow$  6 (3x3)
- MaxPool2d (2x2)
- Conv2d: 6  $\rightarrow$  16 (3x3)
- Linear: 16 x 14 (30) x 14 (30)  $\rightarrow$  2048
- Linear: 2048  $\rightarrow$  1024
- Linear: 1024  $\rightarrow$  10

**Funzione di perdita:** *CrossEntropyLoss*

**Ottimizzatore:** *SGD*

**N° di epoche:** 2, 5, 10

I valori di accuratezza non sono stabili:  
anche in questo caso possono oscillare  
fino al 5% ogni volta!

Size \ Epochs	2	5	10
64x64	11%	11%	10%
128x128	7%	<u>16%</u>	16%

TRAIN SET: 100 vs TEST SET: 10

Size \ Epochs	2	5	10
64x64	12,60%	12,20%	10,60%
128x128	12,16%	11,16%	10,64%

TRAIN SET: 750 vs TEST SET: 250

# Conclusioni

- I risultati raggiunti sono piuttosto scarni
- KNN si è dimostrato, nonostante questo, un semplice nonché miglior metodo di classificazione



THE END