Sistema di riconoscimento piante

Progetto di visione Artificiale

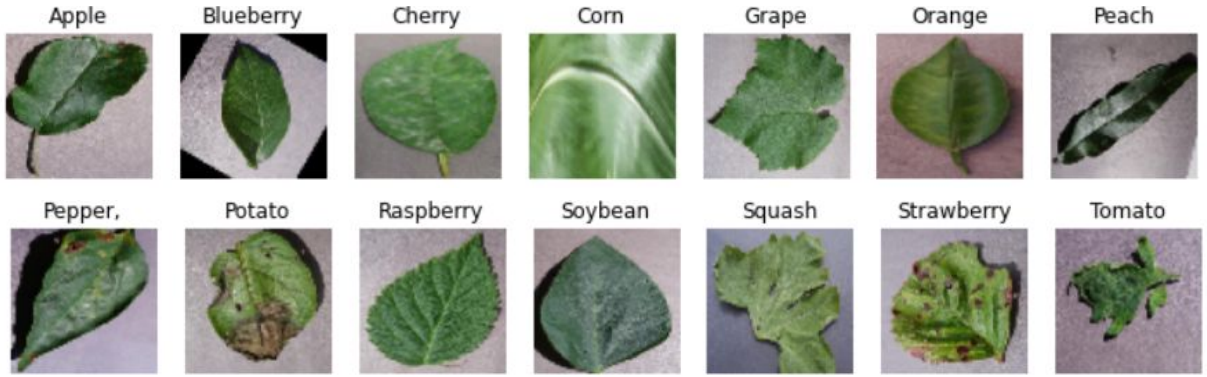
Vitali Anna e Yan Elena

Introduzione

- per il progetto di visione artificiale si è pensato di realizzare un sistema in grado di **riconoscere** diverse **tipologie di piante** a partire da una foto della loro foglia
- sono state realizzate tre diverse soluzioni
  - riconoscimento delle 14 diverse piante tramite **feature *Hand Crafted***
  - riconoscimento delle 14 diverse piante tramite una **rete neurale**
  - riconoscimento delle diverse **piante e malattie** che le possono affliggere (38 classi) tramite una **rete neurale**

Dataset utilizzato

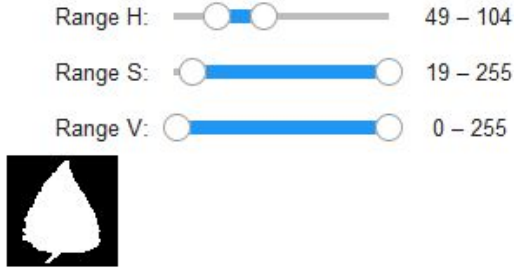
- per risolvere il problema è stato utilizzato il **dataset**: Plant Diseases disponibile sulla piattaforma **Kaggle**
- il dataset contiene in tutto **87K immagini RGB** di foglie con malattie e sane riferite a **14** diverse **tipologie di piante**
- presenta un totale di **38 differenti classi**



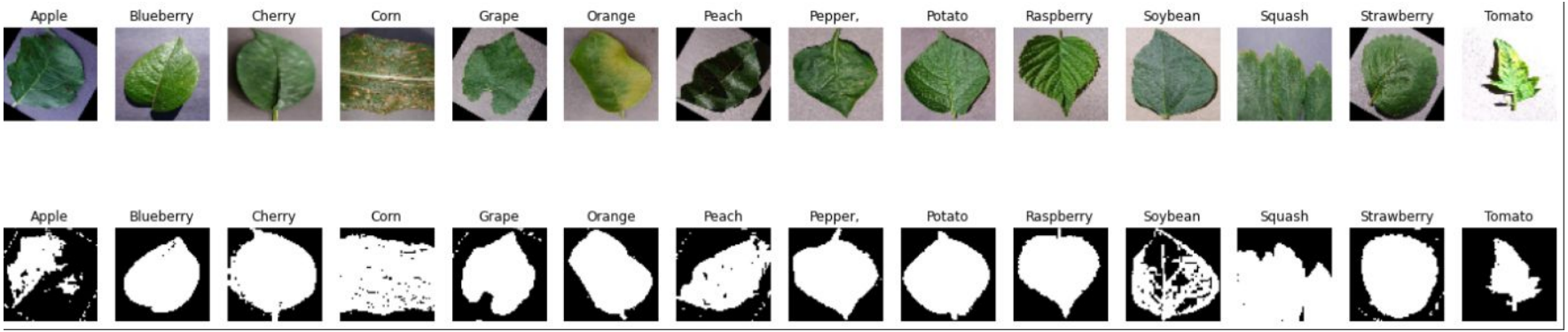
Riconoscimento piante - feature Hand Crafted

- per **riconoscere** le diverse **tipologie di piante** si è deciso di **estrarre** dalle immagini le **feature**
  - di **forma**
  - **colore**
  - di **tessitura**
- per **ogni feature** viene prodotta una **classifica** delle **immagini più simili** a quella da classificare

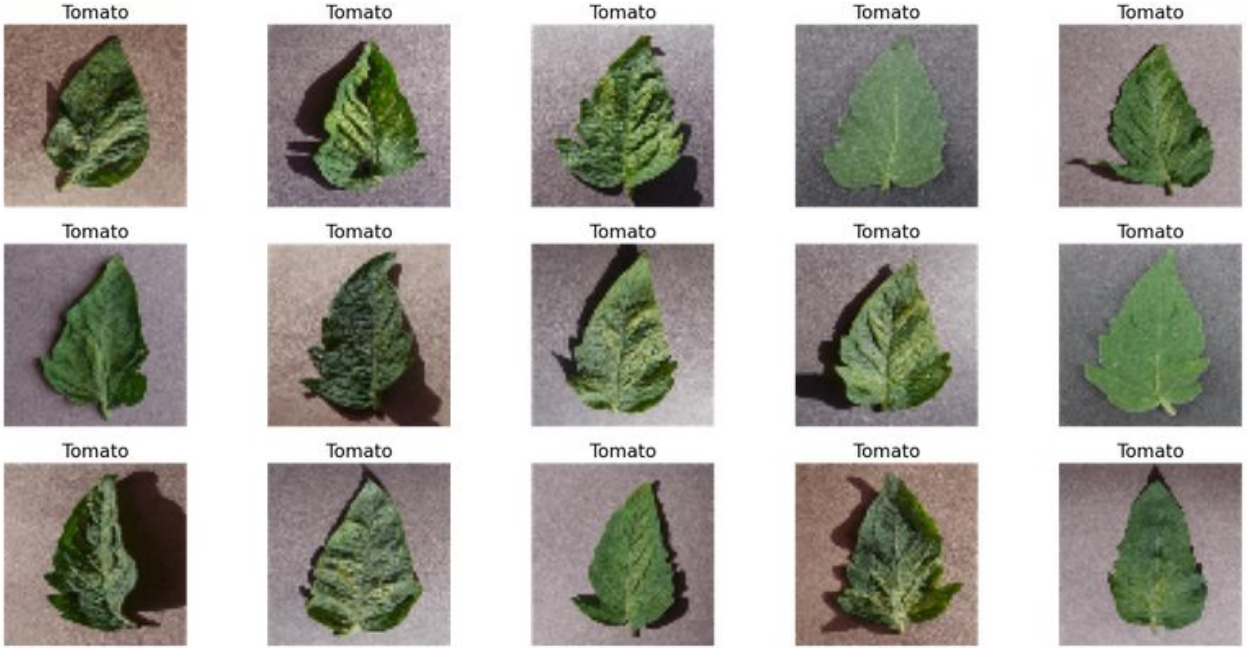
Feature di forma



- l'estrazione delle feature di forma è stata effettuata:
  1. calcolando la **rappresentazione** dell'immagine nello **spazio HSV**
  2. **tarando** i valori di questo spazio ottenendo una **maschera binaria** che individuasse la **foglia**
  3. **applicando** questa **maschera binaria** all'immagine, per estrarre la forma della foglia,
  4. applicando un'operazione di **chiusura morfologica** per riempire eventuali buchi



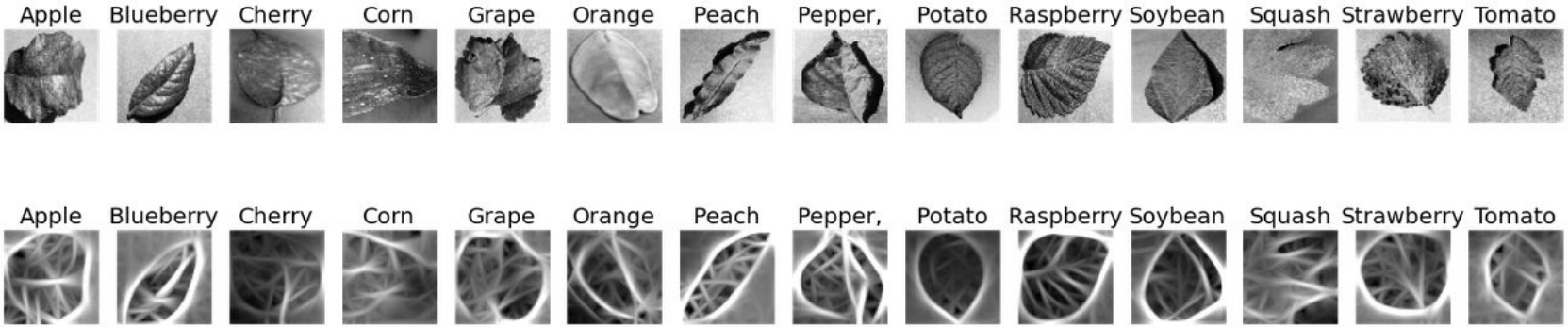
Risultati - feature di forma



Feature di texture

Per poter estrarre le feature relative alla texture si è deciso di utilizzare i filtri di **Gabor**

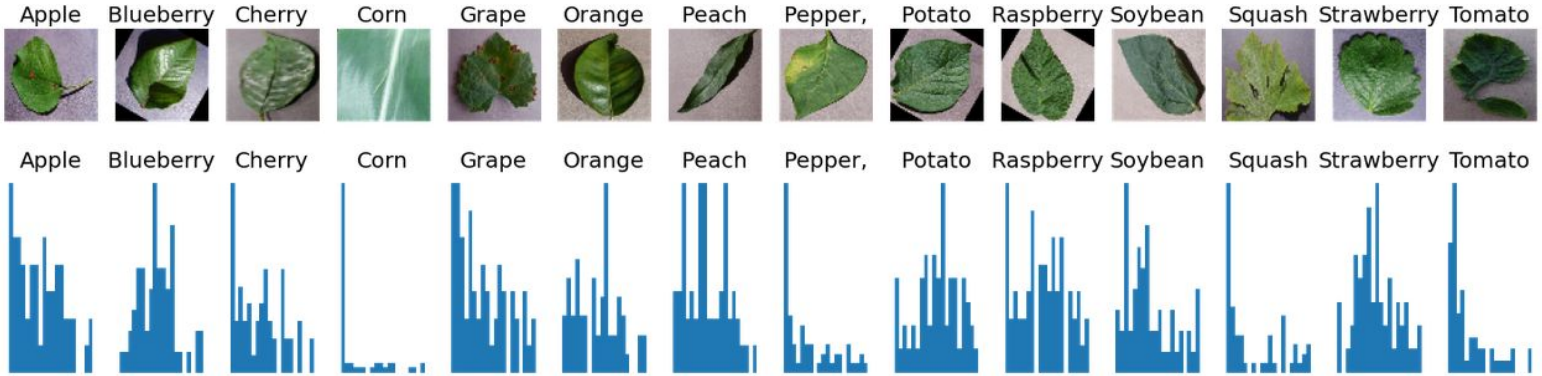
- un filtro di Gabor è una **sinusoide** ottenuta progressivamente a partire da una **gaussiana** e regolata da tre parametri principali
  - **ampiezza** della gaussiana;
  - **frequenza**;
  - **orientazione** rispetto allo spazio x,y.



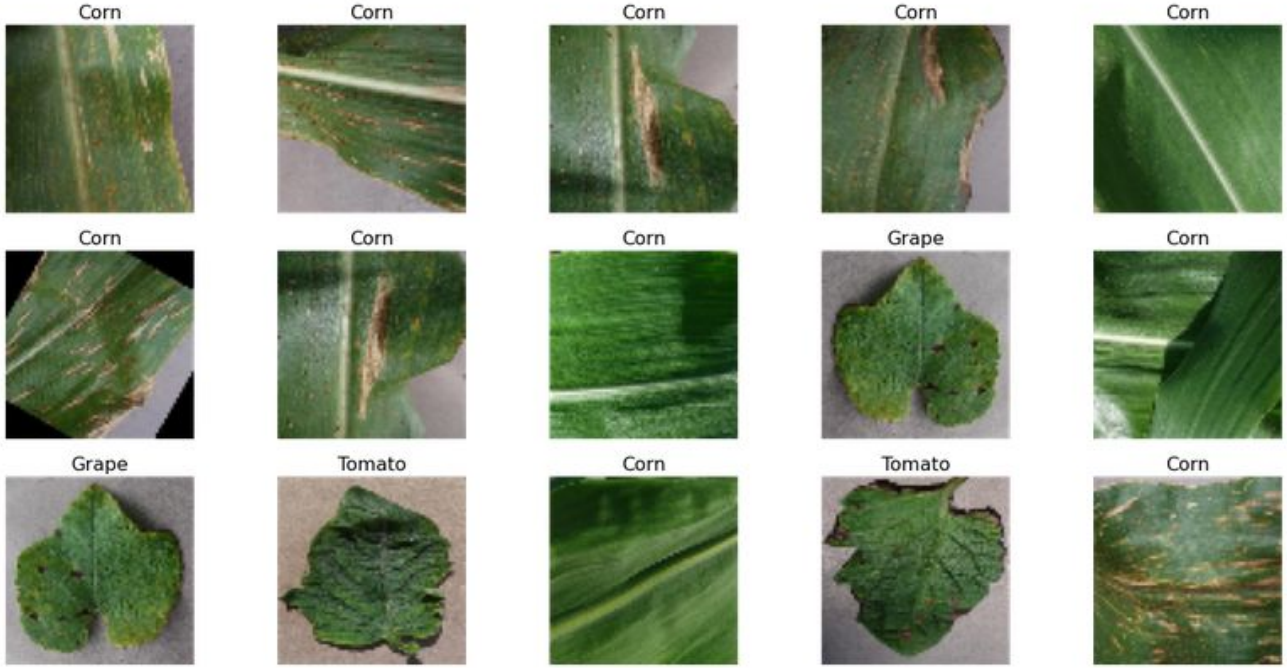
Risultati - feature di texture

Feature colore

- È vero che le foglie non differiscono molto fra loro per il **colore**, ma queste feature possono comunque **contribuire** nella **distinzione** delle diverse classi;
- Per il calcolo dell'istogramma colore è stato utilizzato lo spazio colore **HSV** e un numero di **bin** pari a **20**
    - i **range**, dei diversi parametri dello spazio, sono gli stessi che sono stati utilizzati per individuare la forma della foglia.



Risultati - feature di colore

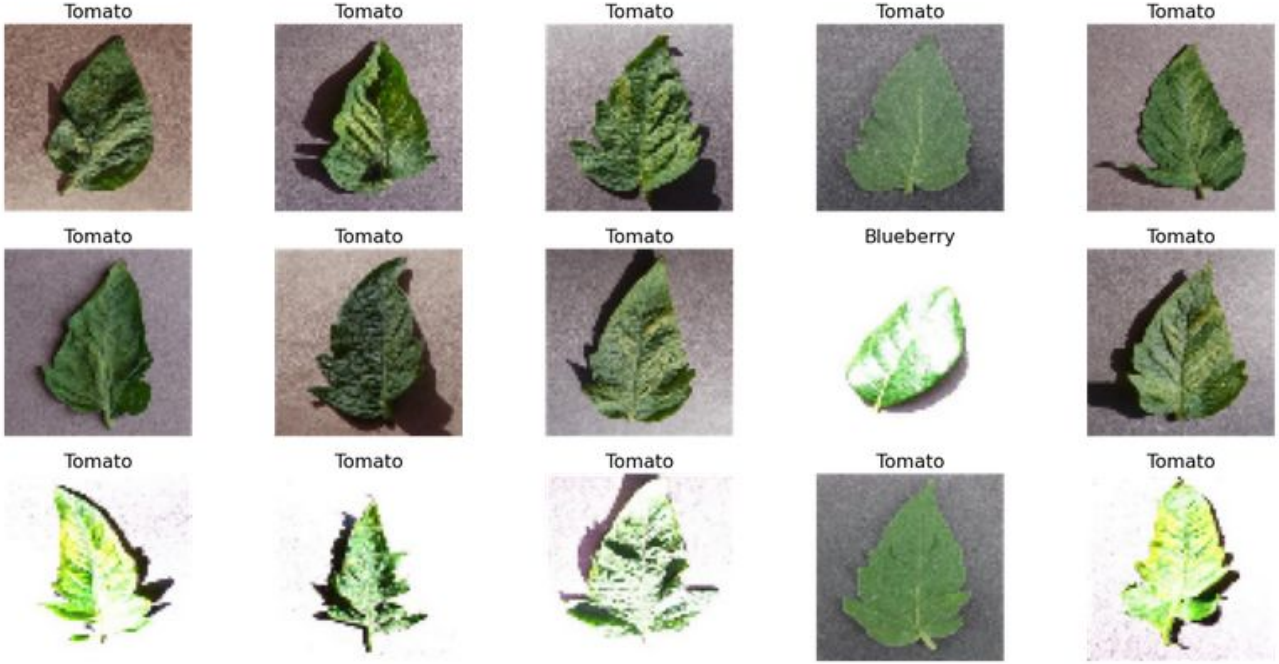


Fusione degli score

- le diverse **classifiche** vengono convertite in **punteggi,** attribuendo un **peso** diverso alle differenti feature
- punteggi vengono **sommati** e associati alle diverse classi del problema
- all'immagine viene **associata** la **classe** con il punteggio più **elevato**

```
tot_scores = borda_count([color_ranking, shape_ranking, texture_ranking], [  0.1, 0.6, 0.3])
tot_ranking = np.array(tot_scores).argsort()[:: -1]
```

Classifica finale



Valutazione delle prestazioni- Validation e Test

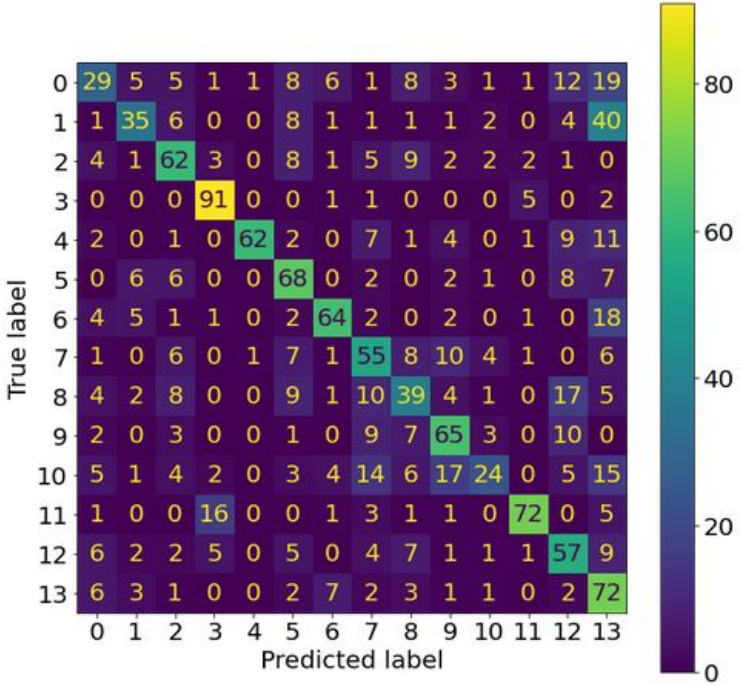
Una volta classificate le diverse immagini si è effettuato il calcolo:

- dell'**accuratezza** → circa **60%** su Validation set e **63**% sul Test set
- della **matrice di confusione**, per valutare l'**entità** degli **errori** commessi.

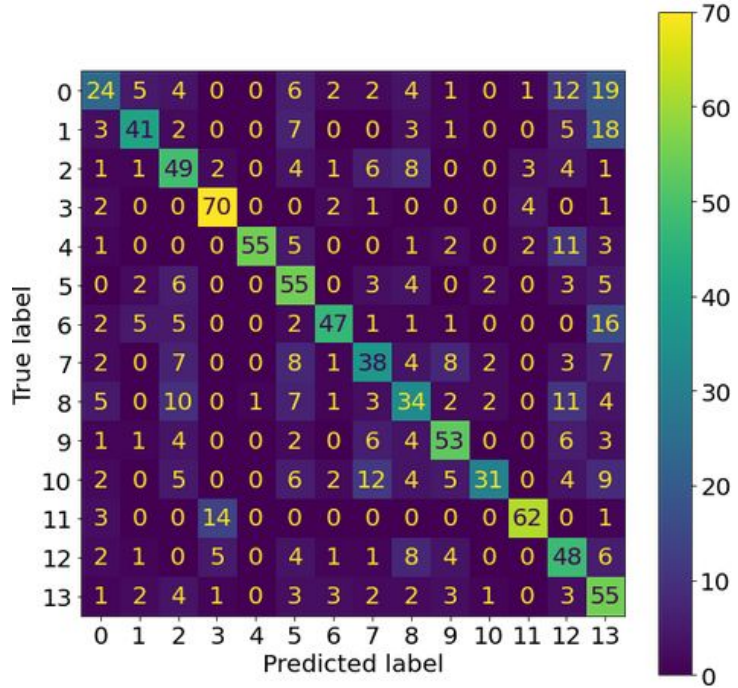
Principali cause di errore:

- l'immagine è di **piccole** dimensioni
- foglie con **forme simili** fra loro

Matrice di confusione



Validation Set



Test Set

Classification Report

```
accuratezza per classe: [0.29 0.35 0.62 0.91 0.62 0.68 0.64 0.55 0.39 0.65 0.24 0.72 0.57 0.72]
accuratezza media: 0.5678571428571428
              precision    recall  f1-score   support

       Apple       0.45      0.29      0.35       100
   Blueberry       0.58      0.35      0.44       100
      Cherry       0.59      0.62      0.60       100
        Corn       0.76      0.91      0.83       100
       Grape       0.97      0.62      0.76       100
      Orange       0.55      0.68      0.61       100
       Peach       0.74      0.64      0.68       100
      Pepper,      0.47      0.55      0.51       100
      Potato       0.43      0.39      0.41       100
   Raspberry       0.58      0.65      0.61       100
     Soybean       0.60      0.24      0.34       100
      Squash       0.86      0.72      0.78       100
  Strawberry       0.46      0.57      0.51       100
      Tomato       0.34      0.72      0.47       100

    accuracy                           0.57      1400
   macro avg       0.60      0.57      0.56      1400
weighted avg       0.60      0.57      0.56      1400
```

Validation Set

```
accuratezza per classe: [0.3    0.5125 0.6125 0.875  0.6875 0.6875 0.5875 0.475  0.425  0.6625
 0.3875 0.775  0.6    0.6875]
accuratezza media: 0.5910714285714285
              precision    recall  f1-score   support

       Apple       0.49      0.30      0.37        80
   Blueberry       0.71      0.51      0.59        80
      Cherry       0.51      0.61      0.56        80
        Corn       0.76      0.88      0.81        80
       Grape       0.98      0.69      0.81        80
      Orange       0.50      0.69      0.58        80
       Peach       0.78      0.59      0.67        80
      Pepper,      0.51      0.47      0.49        80
      Potato       0.44      0.42      0.43        80
   Raspberry       0.66      0.66      0.66        80
     Soybean       0.82      0.39      0.53        80
      Squash       0.86      0.78      0.82        80
  Strawberry       0.44      0.60      0.51        80
      Tomato       0.37      0.69      0.48        80

    accuracy                           0.59      1120
   macro avg       0.63      0.59      0.59      1120
weighted avg       0.63      0.59      0.59      1120
```

Test Set

Problematiche e migliorie

Problematiche della soluzione adottata:

- la classificazione **computazionalmente costosa** e richiede **diverso tempo**
  - non si è utilizzato un **classificatore**, a causa dei **limiti** di **memoria** imposti dal framework CoLab

Possibili **migliorie**:

- provare **altre tecniche** per l'estrazione delle feature
- provare ad **aumentare** la **dimensione** delle immagini;
- **migliorare** la tecnica di acquisizione della **forma**;
- sfruttare per la computazione della classe la **GPU**;

Riconoscimento piante - Deep Learning

Come modello di partenza si è deciso di utilizzare **VGG16** addestrata con i pesi di **imagenet**:

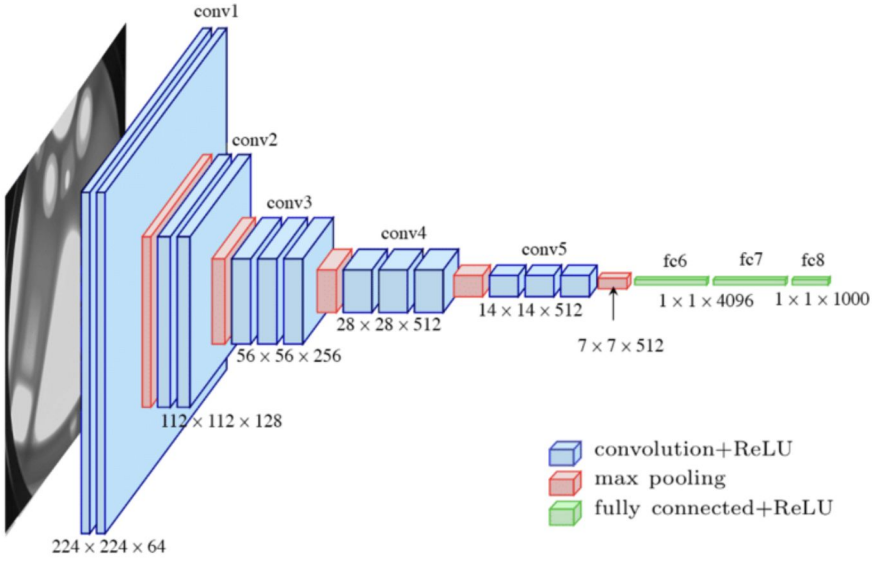
- la dimensione delle immagini di input è 224 x 244 x 3

È stato definito un nuovo modello in questo modo:

- aggiungendo un nuovo livello **Fully Connected**, adattando il numero di classi (14 piante);
    - funzione di attivazione → *soft_max*.

Per l'addestramento del modello è stato utilizzato

- **Adam** -> come **ottimizzatore**
- **sparse_categorical_crossentropy** -> come funzione di **loss**.



Modello rete VGG16

Valutazione delle prestazioni - Training e Validation set

Per l'addestramento del modello sono state utilizzate **3 epoche**

- già con questo numero si ottengono un buon risultato per la **loss** e per l'**accuratezza**.

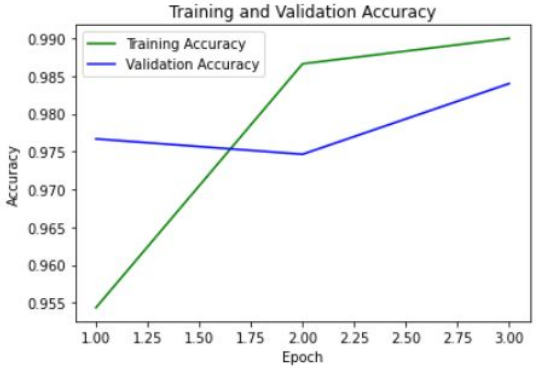
L'**accuratezza**:

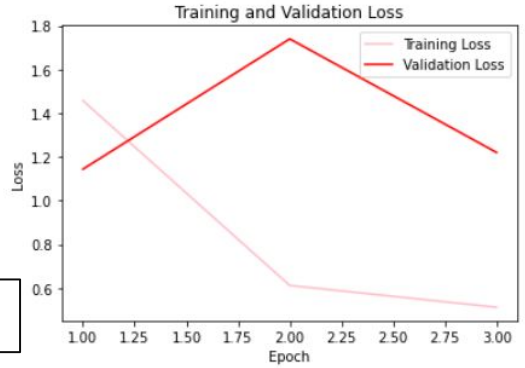
- sul *training set* → **aumenta** notevolmente fin dalla prima epoca, per poi **rallentare**
- sul *validation set* → si **abbassa** leggermente fino alla seconda epoca per poi **aumentare**

La **loss**:

- per il *training set* → tende a **diminuire** man mano che le **epoche aumentano**
- per il *validation set* → **aumenta** durante la prima epoca per poi **diminuire** nella seconda

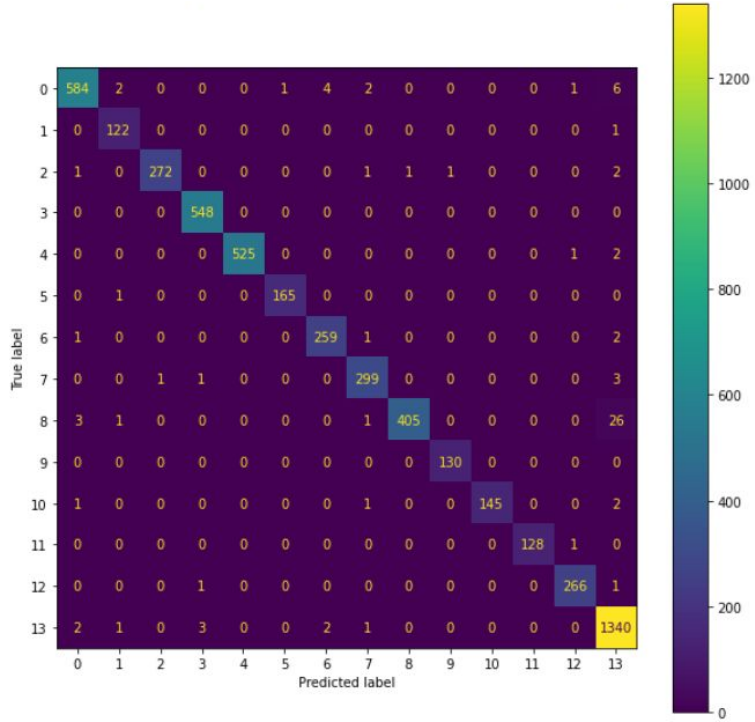
```
loss: 0.5117 - accuracy: 0.9900 - val_loss: 1.2203 - val_accuracy: 0.9840
```





Valutazione delle prestazioni - Test set

	precision	recall	f1-score	support
Apple	0.99	0.97	0.98	600
Blueberry	0.96	0.99	0.98	123
Cherry	1.00	0.98	0.99	278
Corn	0.99	1.00	1.00	548
Grape	1.00	0.99	1.00	528
Orange	0.99	0.99	0.99	166
Peach	0.98	0.98	0.98	263
Pepper,	0.98	0.98	0.98	304
Potato	1.00	0.93	0.96	436
Raspberry	0.99	1.00	1.00	130
Soybean	1.00	0.97	0.99	149
Squash	1.00	0.99	1.00	129
Strawberry	0.99	0.99	0.99	268
Tomato	0.97	0.99	0.98	1349
accuracy			0.98	5271
macro avg	0.99	0.98	0.99	5271
weighted avg	0.98	0.98	0.98	5271



Problematiche e possibili modifiche

Problematiche della soluzione adottata:

- la *loss* sul validation set **non** ha un **andamento stabile**;
- se si **diminuiscono** le **dimensioni** delle immagini, le **prestazioni calano**, sebbene non di molto.

Possibili **modifiche**:

- adottare un'altra funzione per misurare la ***loss***;
- modificare l'ottimizzatore;
- aumentare le epoche.

Riconoscimento delle malattie - Deep Learning

In questo caso sono state considerate **tutte** e **38** le **classi** del problema

Come modello di partenza si è deciso di utilizzare ***MobileNet*** addestrata con i pesi di **imagenet**:

- la dimensione delle immagini di input è di 224 x 224 x 3

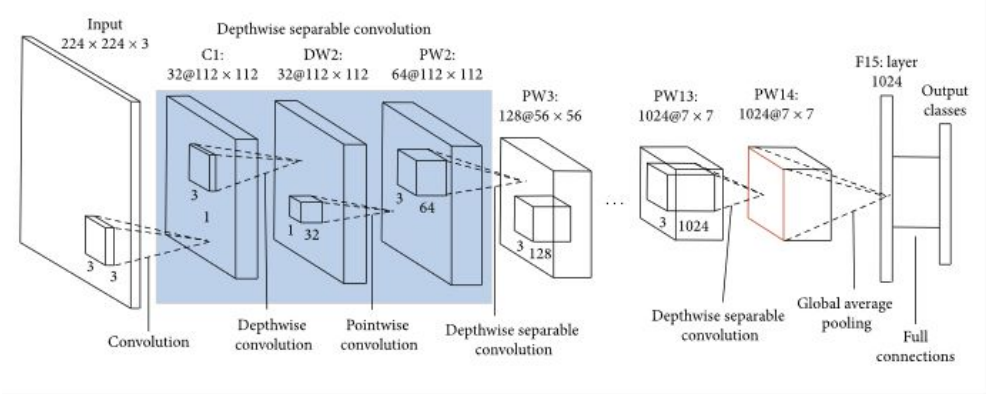
È stato definito un nuovo modello in questo modo:

- **aggiungendo** un nuovo livello ***Fully Connected*** adattando il numero di classi
  - funzione di attivazione → *soft_max*.

Per l'addestramento del modello è stato utilizzato:

- **Adam** → come ottimizzatore
- **sparse_categorical_crossentropy**

  → come funzione di loss



Valutazione delle prestazioni - Training e Validation set

Per l'addestramento del modello sono state utilizzate **3 epoche**

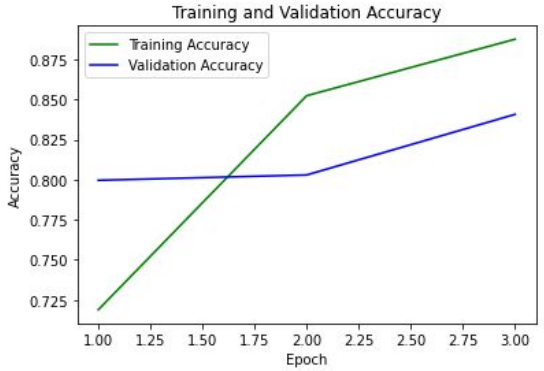
- si è visto che aumentando questo numero, anche se l'accuratezza migliora, la **loss** invece peggiora.

L'**accuratezza**:

- sul *training set* → **aumenta** notevolmente fin dalla prima epoca, per poi **rallentare**
- sul *validation set* → fino alla seconda epoca rimane stabile e poi aumenta

La **loss**:

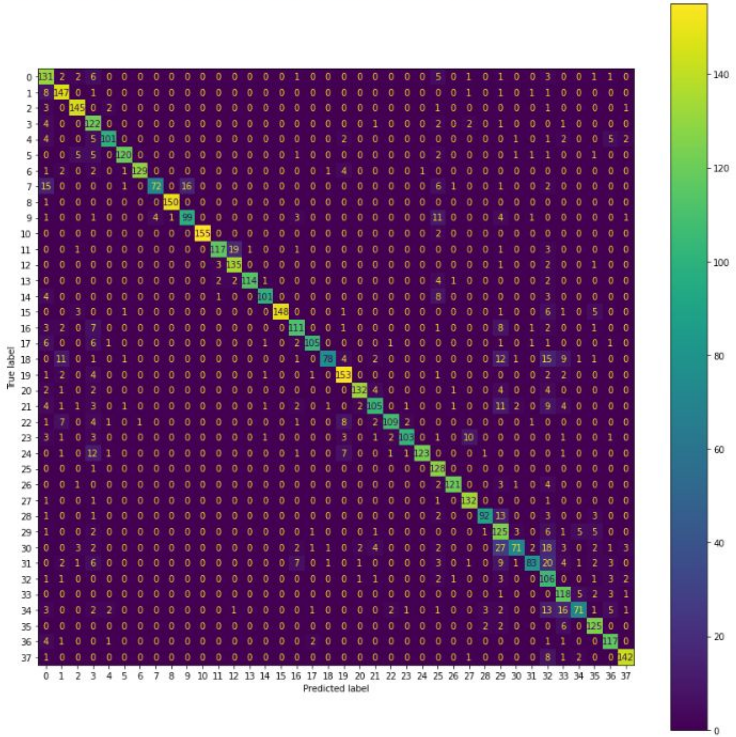
- per il *training set* → tende a **diminuire** man mano che le **epoche aumentano**
- per il *validation set* → **aumenta** durante la prima epoca per poi **diminuire** nella seconda




```
loss: 2.2044 - accuracy: 0.8876 - val_loss: 4.1122 - val_accuracy: 0.8407
```

Valutazione delle prestazioni - Test set

	precision	recall	f1-score	support
Apple___Apple_scab	0.64	0.85	0.73	154
Apple___Black_rot	0.81	0.92	0.86	160
Apple___Cedar_apple_rust	0.90	0.95	0.92	153
Apple___healthy	0.61	0.92	0.73	133
Blueberry___healthy	0.93	0.82	0.87	123
Cherry_(including_sour)___Powdery_mildew	0.96	0.88	0.92	137
Cherry_(including_sour)___healthy	0.99	0.91	0.95	141
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0.95	0.63	0.76	114
Corn_(maize)___Common_rust_	0.99	0.99	0.99	151
Corn_(maize)___Northern_Leaf_Blight	0.86	0.79	0.82	126
Corn_(maize)___healthy	1.00	0.99	0.99	157
Grape___Black_rot	0.95	0.82	0.88	143
Grape___Esca_(Black_Measles)	0.86	0.95	0.90	142
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	0.99	0.90	0.95	126
Grape___healthy	0.95	0.86	0.91	117
Orange___Haunglongbing_(Citrus_greening)	1.00	0.89	0.94	166
Peach___Bacterial_spot	0.83	0.81	0.82	137
Peach___healthy	0.95	0.83	0.89	126
Pepper,_bell___Bacterial_spot	0.95	0.57	0.71	137
Pepper,_bell___healthy	0.84	0.92	0.87	167
Potato___Early_blight	0.96	0.88	0.92	150
Potato___Late_blight	0.87	0.70	0.78	149
Potato___healthy	0.95	0.80	0.87	137
Raspberry___healthy	0.95	0.79	0.87	130
Soybean___healthy	0.99	0.83	0.90	149
Squash___Powdery_mildew	0.69	0.99	0.82	129
Strawberry___Leaf_scorch	0.96	0.92	0.94	132
Strawberry___healthy	0.89	0.97	0.93	136
Tomato___Bacterial_spot	0.93	0.80	0.86	115
Tomato___Early_blight	0.54	0.83	0.65	151
Tomato___Late_blight	0.88	0.49	0.63	144
Tomato___Leaf_Mold	0.91	0.57	0.70	145
Tomato___Septoria_leaf_spot	0.45	0.87	0.59	122
Tomato___Spider_mites Two-spotted_spider_mite	0.69	0.91	0.78	130
Tomato___Target_Spot	0.84	0.57	0.68	125
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.83	0.93	0.87	135
Tomato___Tomato_mosaic_virus	0.84	0.92	0.88	127
Tomato___healthy	0.93	0.92	0.92	155
accuracy			0.84	5271
macro avg	0.87	0.84	0.84	5271
weighted avg	0.87	0.84	0.84	5271



Problematiche e possibili modifiche

Problematiche della soluzione adottata:

- la *loss* sul validation set **non** ha un **andamento stabile**;
- necessario **diverso tempo** per effettuare le differenti epoche;
- se **diminuiscono** le dimensioni delle immagini, le **prestazioni calano** di molto.

Possibili **modifiche**:

- aumentare la **dimensione** dell'immagine;
- **aggiungere** ulteriori livelli al modello, cercando di ridurre l'*overfitting*, ad esempio adottando il **Dropout** o ulteriori livelli di **Pooling**;
- adottare un'altra funzione per misurare la ***loss***;
- modificare l'ottimizzatore;
- aumentare le epoche.

The End