Progetto DSIM

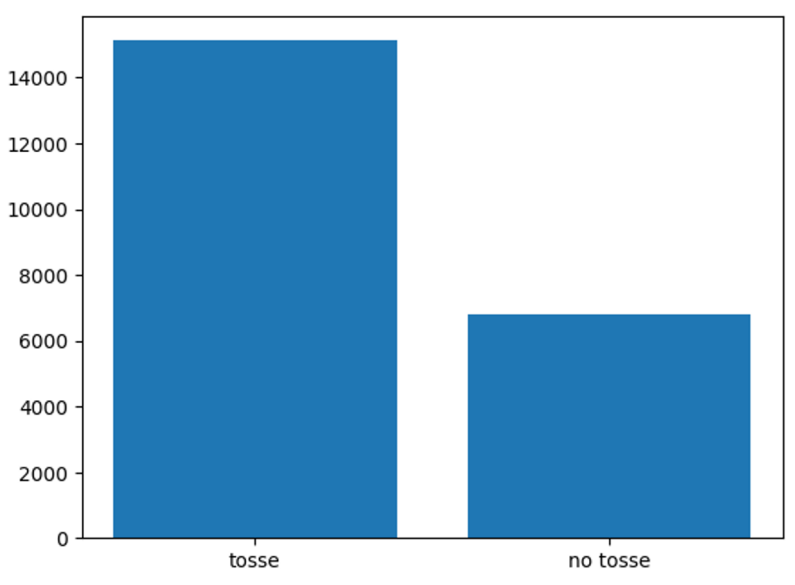
Premate Andrea 829777

Task 1: Audio classification

Dataset

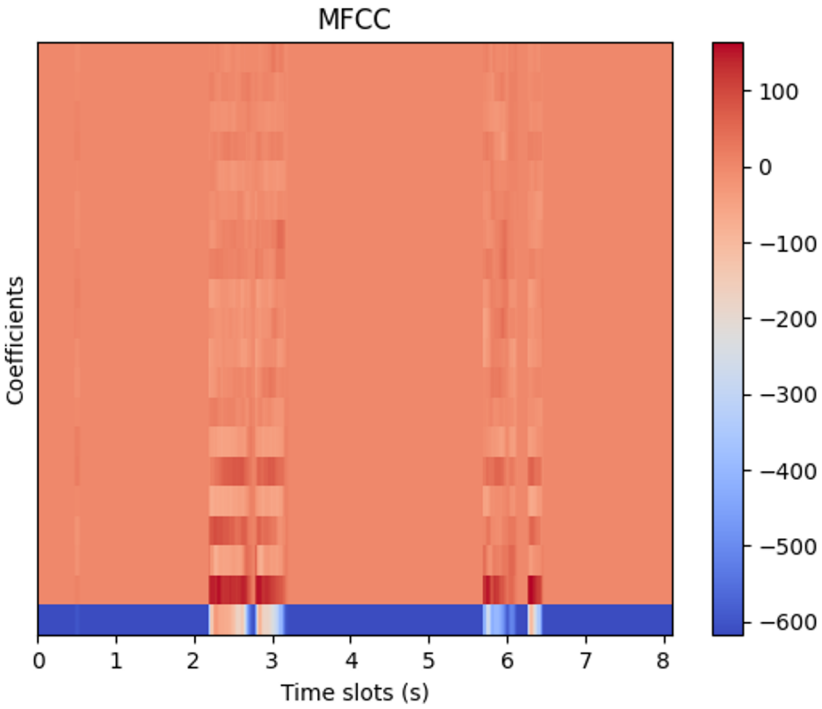
(https://www.kaggle.com/datasets/andrewmvd/covid19-cough-audio-classification)

- Il dataset contenente 27 550 file audio di tosse e diversi attributi come:
  - probabilità tosse
  - età
  - provenienza geografica
  - malattia
  - …
- Classificazione: detector tosse
- Classe no tosse: < 20% prob
  Classe tosse: > 80% prob
- Bilanciamento del dataset

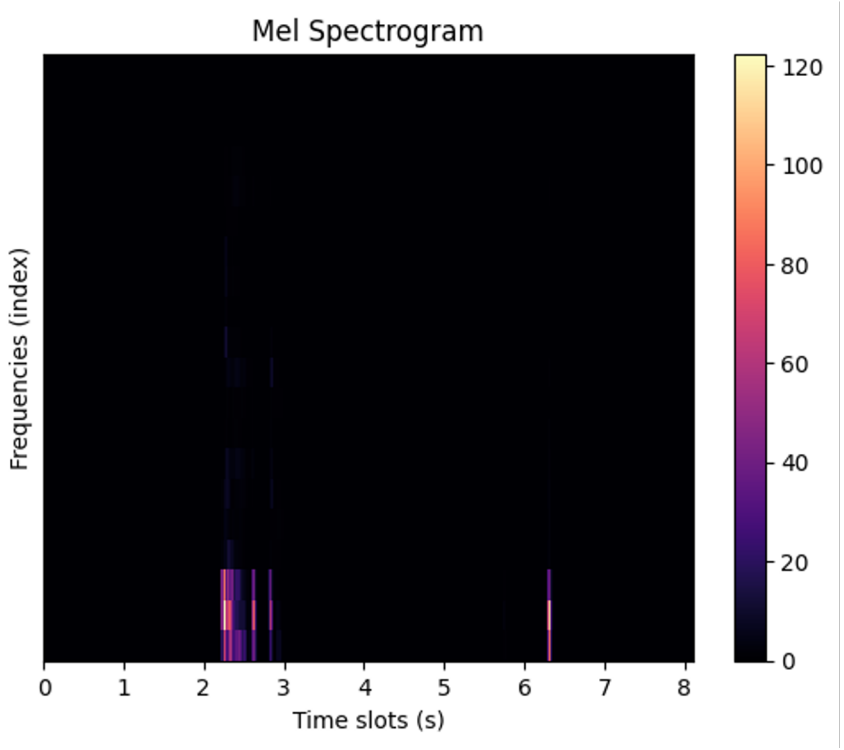


Estrazione features (tosse)

Mel-Frequency Cepstral Coefficients

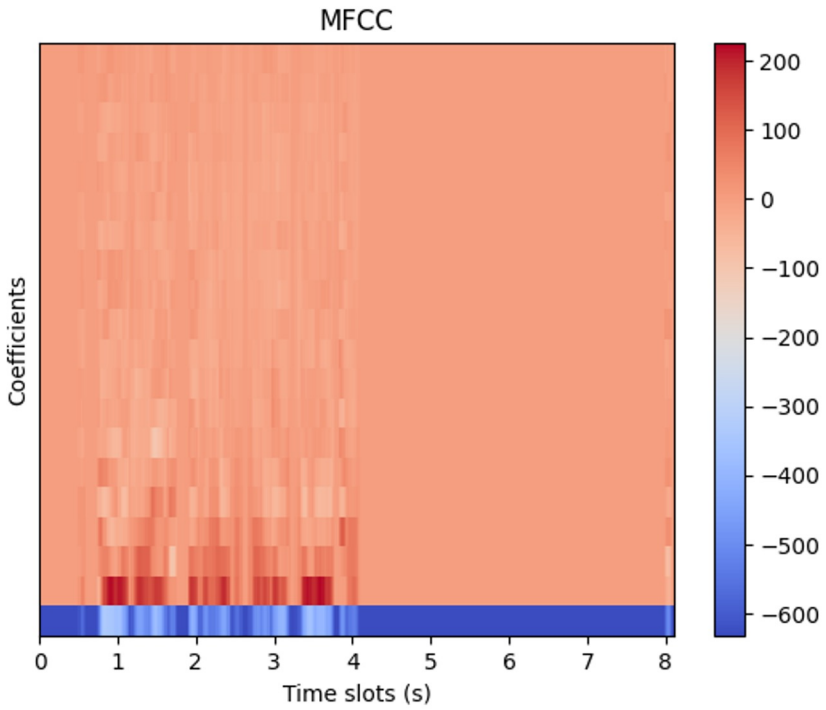


Mel Spectrogram

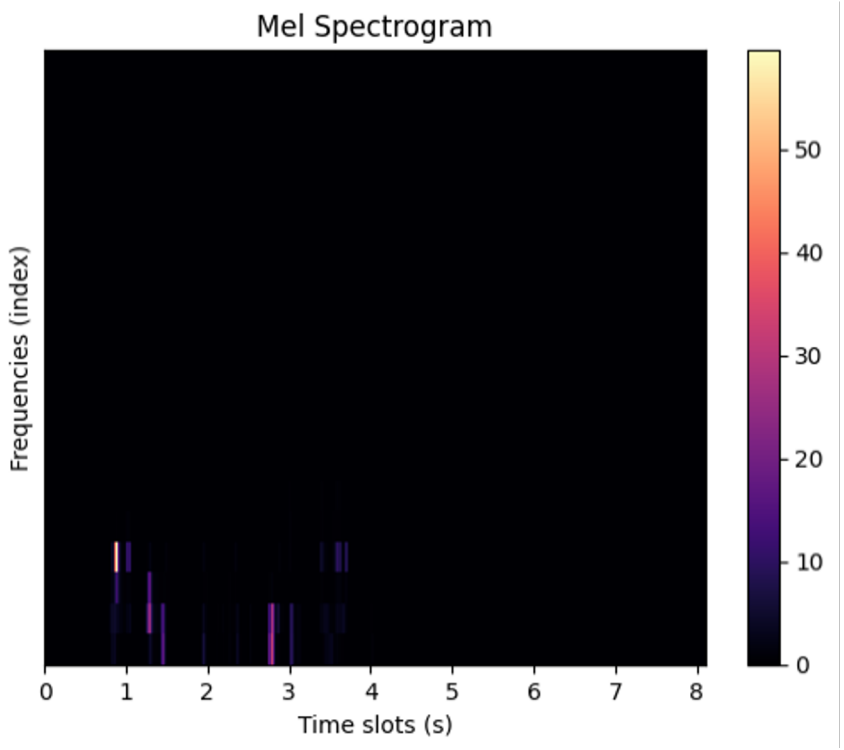


Estrazione features (no tosse)

Mel-Frequency Cepstral Coefficients

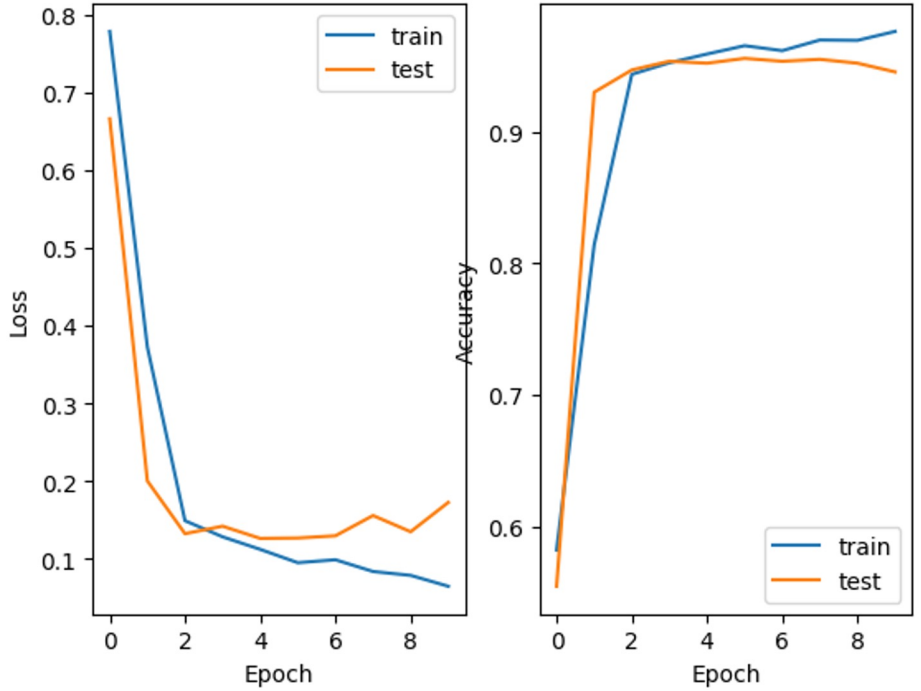


Mel Spectrogram



Modello 1: CNN su MFCC

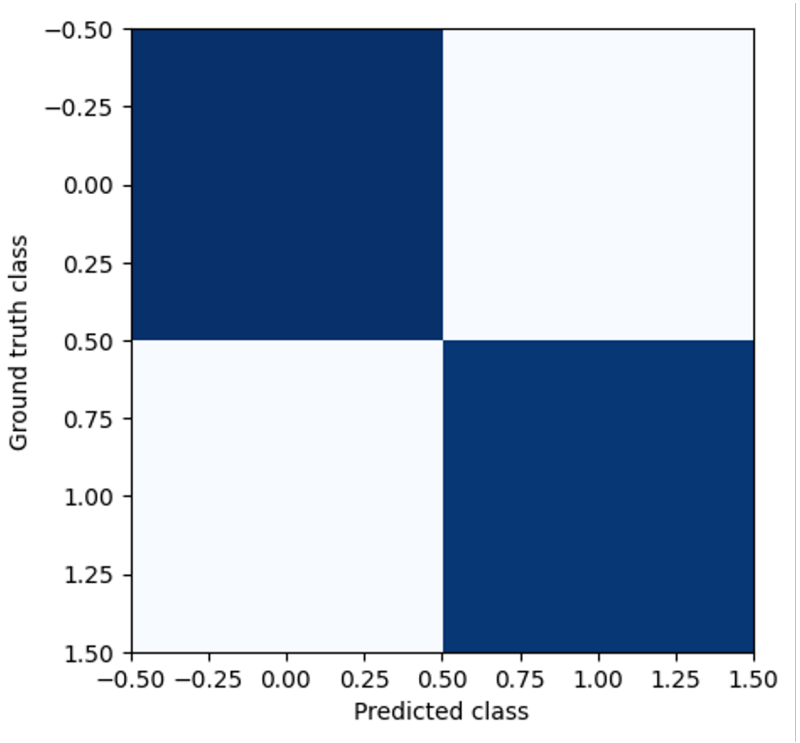
```
x = inputs
x = keras.layers.Conv2D(32, 5, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.MaxPooling2D(5, strides=5, padding='same')(x)

x = keras.layers.Conv2D(64, 2, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.MaxPooling2D(2, strides=2, padding='same')(x)

x = keras.layers.Conv2D(128, 2, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.MaxPooling2D(2, strides=2, padding='same')(x)

x = keras.layers.Flatten()(x)
x = keras.layers.Dense(units = 16, activation = 'relu')(x)
outputs = keras.layers.Dense(1, activation='sigmoid')(x)
```



Modello 1: CNN su MFCC (test)

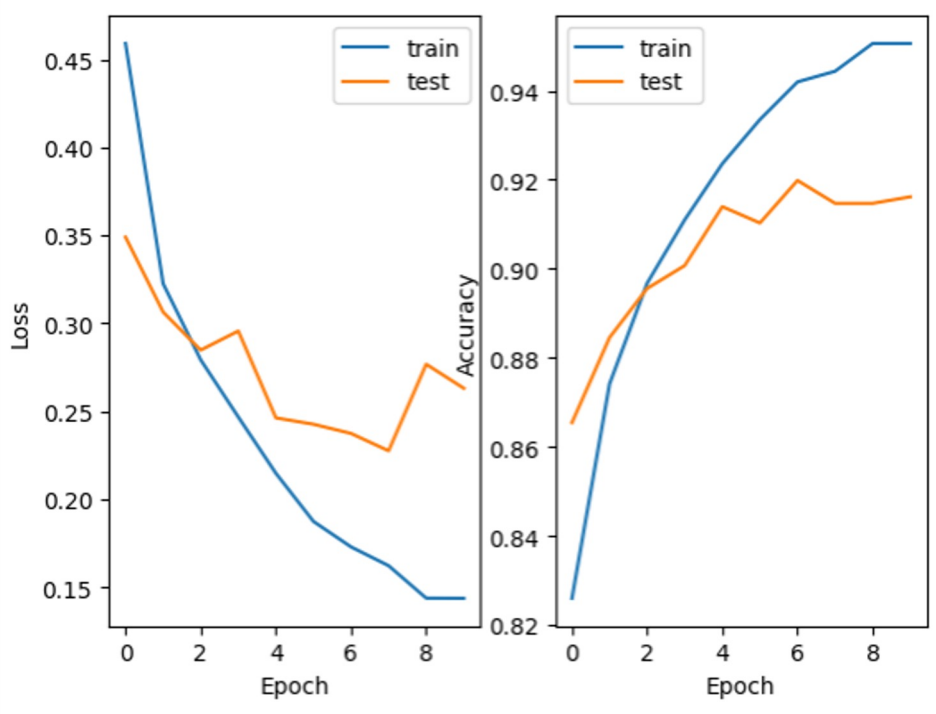
	precision	recall	f1-score	support
0	0.96	0.96	0.96	689
1	0.96	0.96	0.96	671
accuracy			0.96	1360
macro avg	0.96	0.96	0.96	1360
weighted avg	0.96	0.96	0.96	1360

```
[[663  26]
 [ 27 644]]
```



Modello 2: CNN su Mel Spectrogram

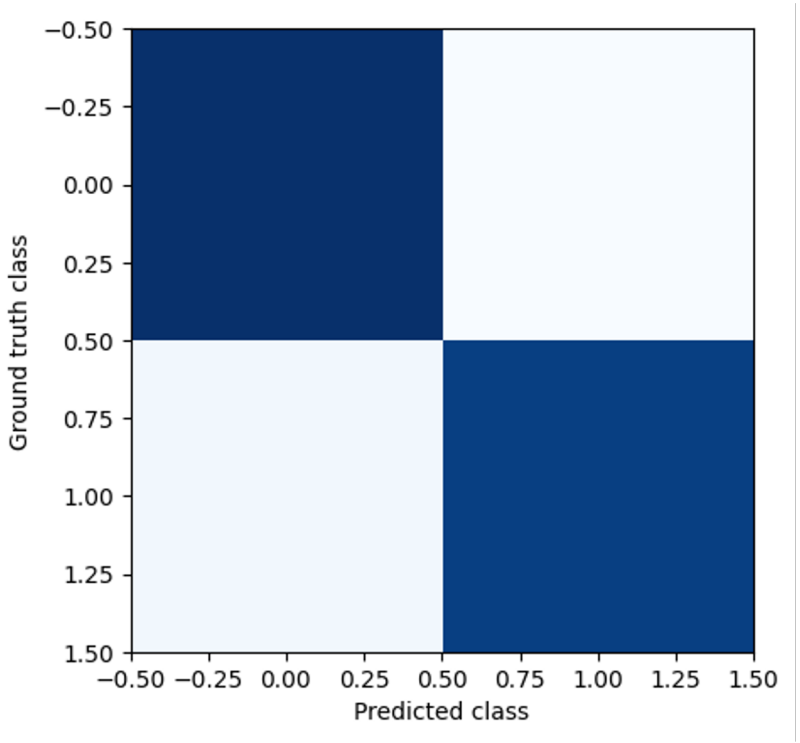
- Rete uguale al punto precedente



Modello 2: CNN su Mel Spectrogram (test)

	precision	recall	f1-score	support
0	0.93	0.95	0.94	689
1	0.95	0.92	0.94	671
accuracy			0.94	1360
macro avg	0.94	0.94	0.94	1360
weighted avg	0.94	0.94	0.94	1360

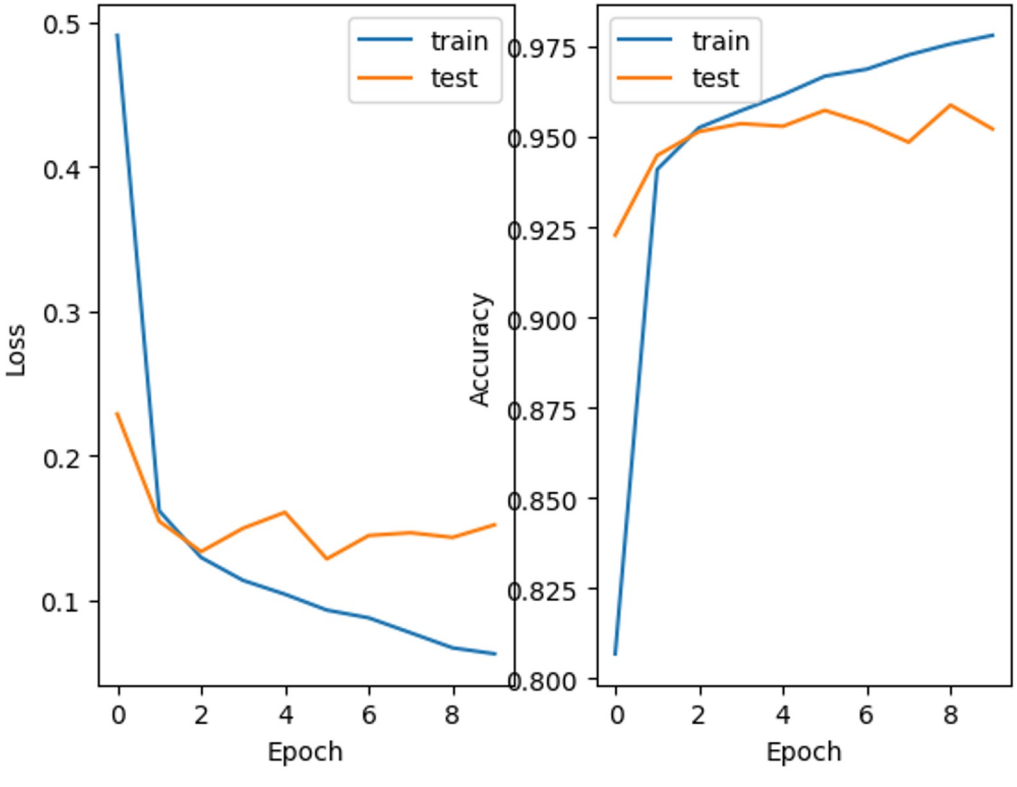
```
[[656  33]
 [ 52 619]]
```



Modello 3: CNN su MFCC + Mel Spectrogram

Rete uguale al punto precedente ma:

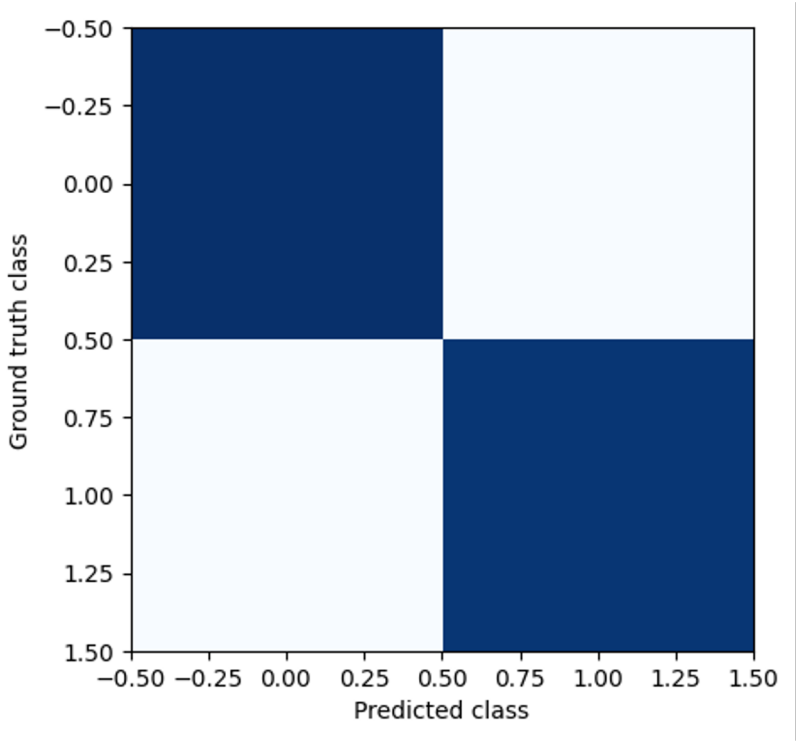
- input con due canali
- fc 32 anziché 16 nel penultimo layer



Modello 3: CNN su MFCC + Mel Spectrogram (test)

	precision	recall	f1-score	support
0	0.96	0.96	0.96	689
1	0.96	0.96	0.96	671
accuracy			0.96	1360
macro avg	0.96	0.96	0.96	1360
weighted avg	0.96	0.96	0.96	1360

```
[[660  29]
 [ 27 644]]
```



Task 2: Image classification

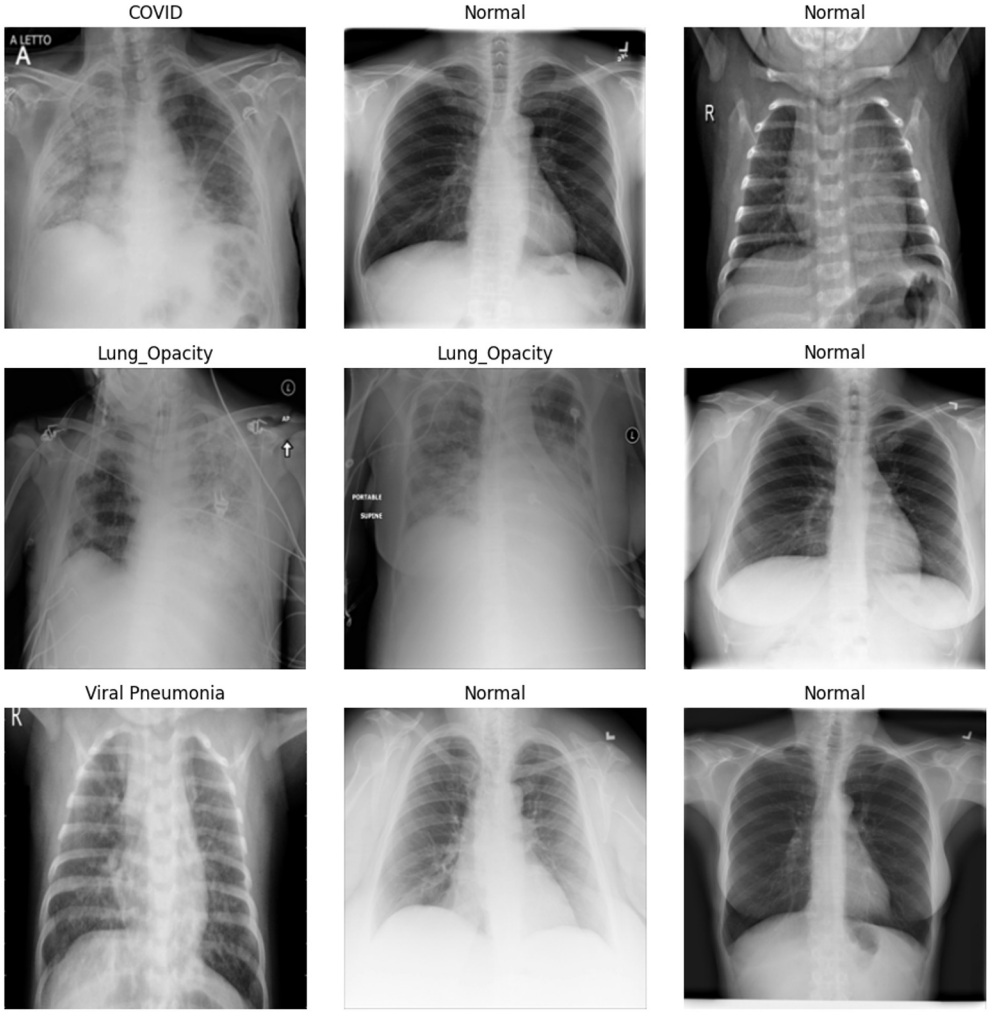
Dataset

(https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database)

- Il dataset contiene radiografie polmonari di:
  - 3616 COVID-19 positive
  - 10192 Normal
  - 6012 Lung Opacity (Non-COVID lung infection)
  - 1345 Viral Pneumonia
- Relative mask di segmentazione dei polmoni
- Classificazione: ipotesi diagnostica

Dataset

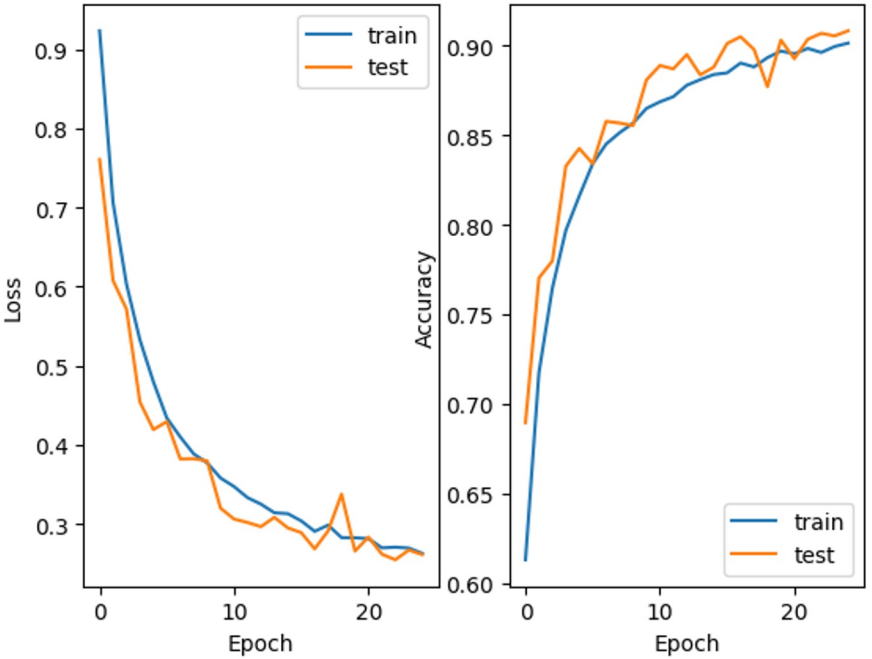
- Immagini a scala di grigi
- Le radiografie provengono da differenti dataset



Modello 1: CNN built from scratch

```
x = inputs
x = keras.layers.Conv2D(32, 5, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.MaxPooling2D(5, strides=5, padding='same')(x)
x = keras.layers.Dropout(0.1)(x)

x = keras.layers.Conv2D(64, 5, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.MaxPooling2D(5, strides=5, padding='same')(x)
x = keras.layers.Dropout(0.1)(x)

x = keras.layers.Conv2D(128, 3, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.MaxPooling2D(3, strides=3, padding='same')(x)
x = keras.layers.Dropout(0.1)(x)

x = keras.layers.Conv2D(256, 2, padding='same')(x)
x = keras.layers.Activation('relu')(x)
x = keras.layers.Flatten()(x)
x = keras.layers.Dropout(0.1)(x)
x = keras.layers.Dense(units = 16, activation = 'relu')(x)

outputs = keras.layers.Dense(4, activation='softmax')(x)
```

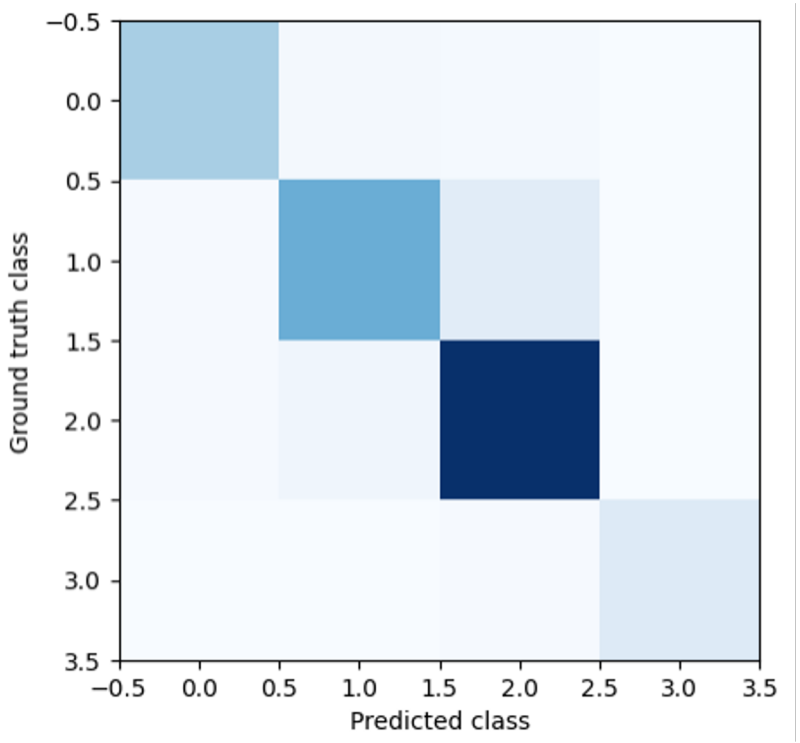
- Data augmentation: flip orizzontale, rotazione, zoom, luminosità



Modello 1: CNN built from scratch (test)

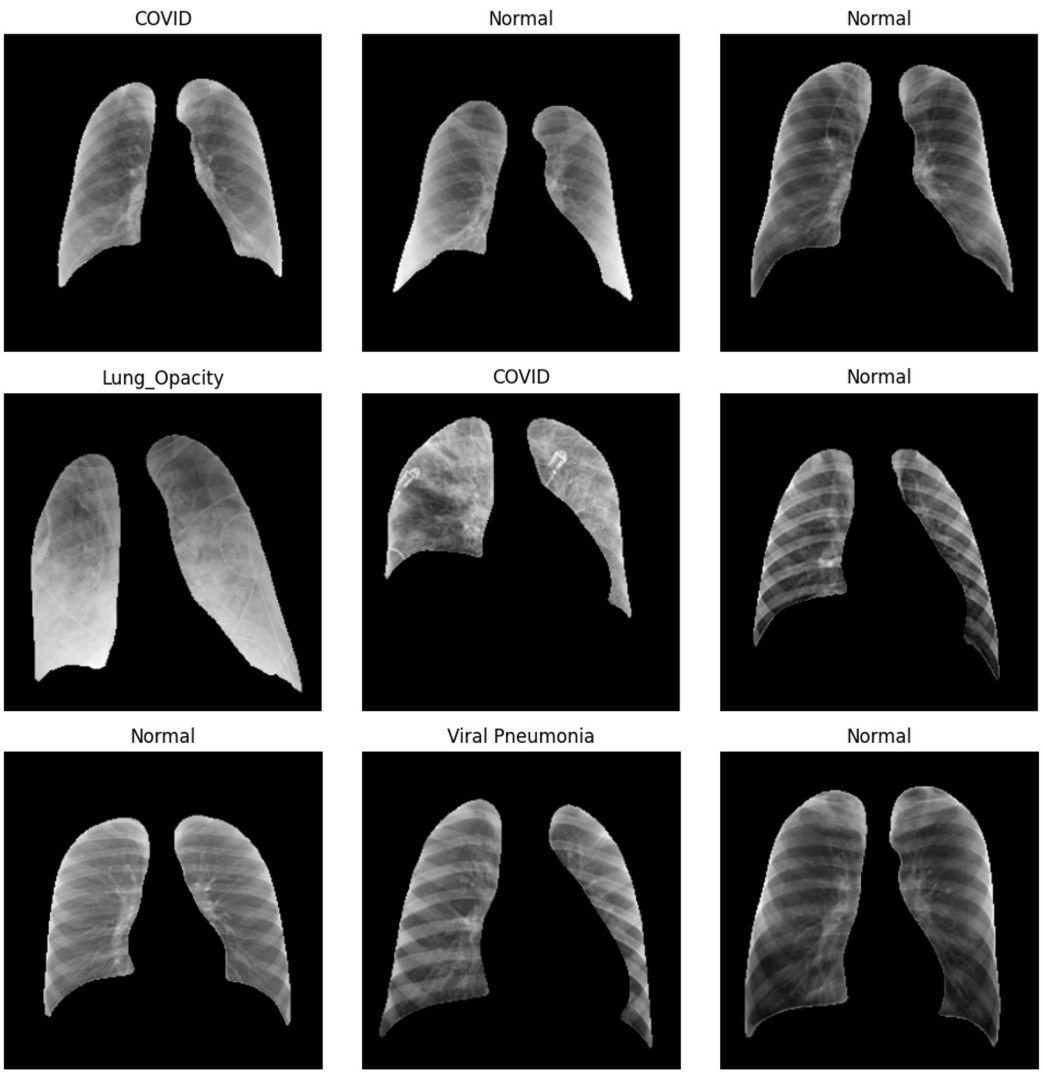
	precision	recall	f1-score	support
COVID	0.94	0.91	0.93	362
Lung_Opacity	0.89	0.81	0.85	602
Normal	0.88	0.95	0.91	1019
Viral Pneumonia	0.96	0.93	0.95	134
accuracy			0.90	2117
macro avg	0.92	0.90	0.91	2117
weighted avg	0.90	0.90	0.90	2117

```
[[330  17  13   2]
 [ 10 485 106   1]
 [ 10  41 966   2]
 [  0   0   9 125]]
```



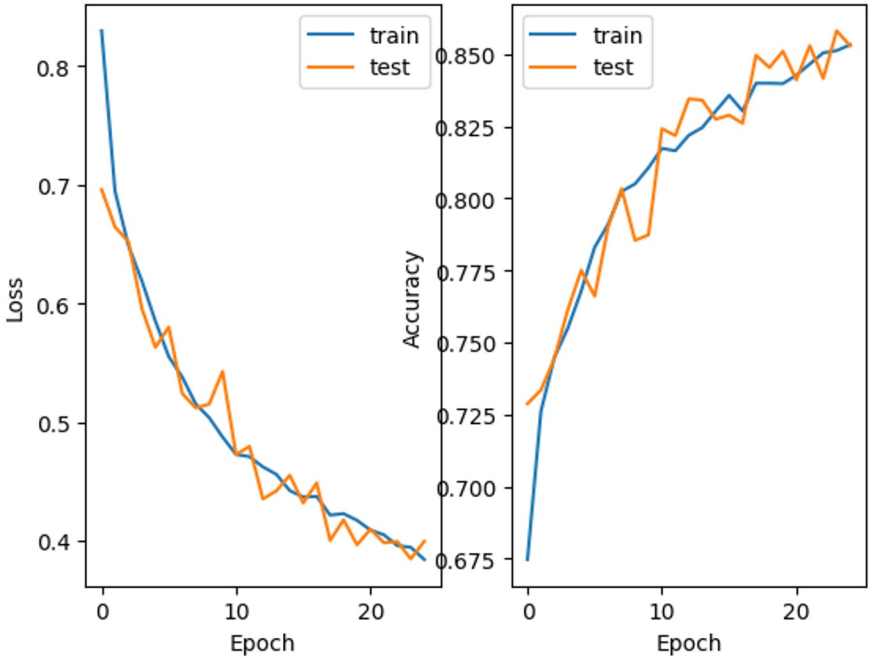
Utilizzo delle mask

- Utilizzo della stessa rete ma solo su parti di immagini segmentate.
- Aspettative: training facilitato almeno in fase iniziale.



Modello 2: CNN built from scratch + mask

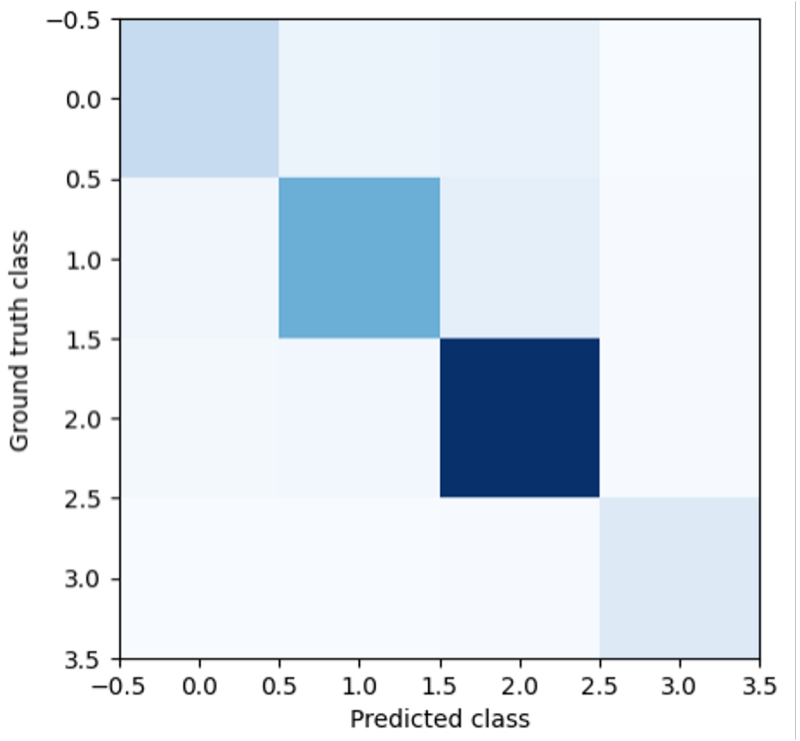
- Stessa rete utilizzata al punto precedente e stessa augmentation



Modello 2: CNN built from scratch + mask (test)

	precision	recall	f1-score	support
COVID	0.82	0.66	0.73	362
Lung_Opacity	0.85	0.80	0.82	602
Normal	0.86	0.95	0.90	1019
Viral Pneumonia	0.93	0.93	0.93	134
accuracy			0.86	2117
macro avg	0.86	0.83	0.85	2117
weighted avg	0.85	0.86	0.85	2117

```
[[238  59  65   0]
 [ 31 481  85   5]
 [ 19  28 967   5]
 [  1   1   7 125]]
```



No mask vs Mask

Performance migliori senza mask, cause?

- Overfitting su diversi formati di dataset di origine (notazioni sulla rx) e relativa distribuzione sulle label.
- Overfitting su altri elementi del busto statisticamente più probabili per ipotesi diagnostica. Esempio ipotetico:

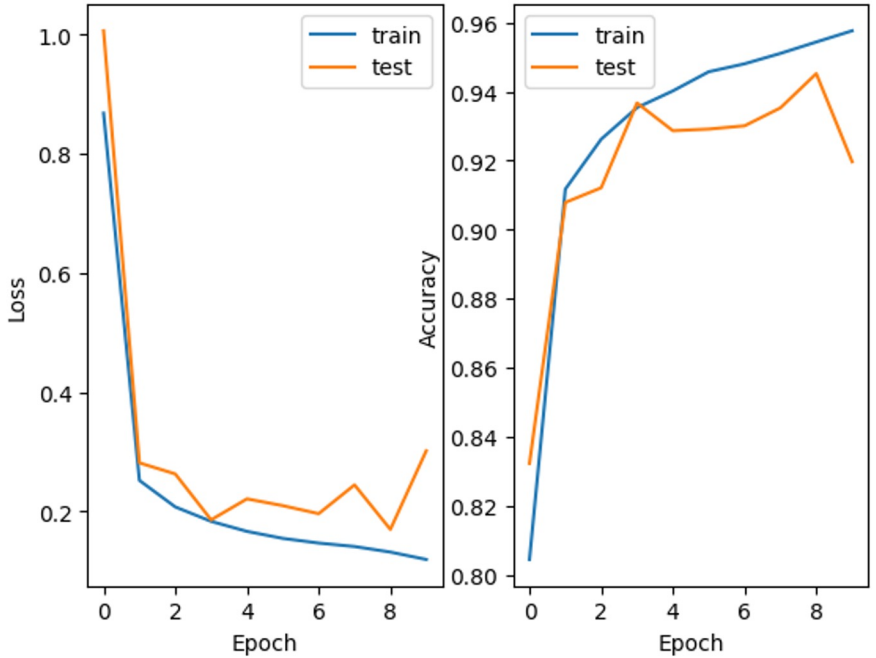
UNWANTED FEATURES

donne hanno distribuzione diversa sulle label → riconoscimento del seno dalla rx e learning conseguente in base a tale feature.

- Segmentazione approssimativa

Modello 3: ResNet50 + fine tuning

- ResNet50 + fc1028 + fc515 + fc32 + fc4
- Freeze fino a circa metà rete (conv4_block4_out)



Modello 3: ResNet50 + fine tuning (test)

	precision	recall	f1-score	support
COVID	0.98	0.95	0.97	362
Lung_Opacity	0.95	0.88	0.91	602
Normal	0.92	0.98	0.95	1019
Viral Pneumonia	0.99	0.93	0.96	134
accuracy			0.94	2117
macro avg	0.96	0.94	0.95	2117
weighted avg	0.94	0.94	0.94	2117

```
[[345  10   7   0]
 [  5 529  68   0]
 [  3  18 997   1]
 [  0   0   9 125]]
```