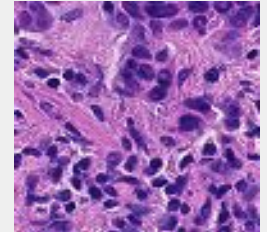
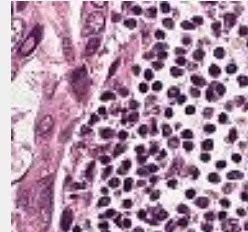
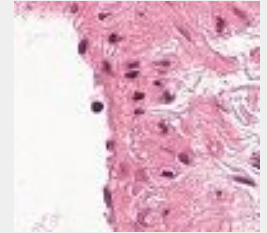
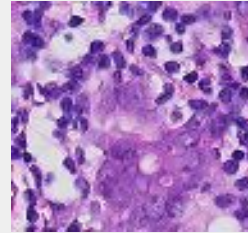


# Histopatologic Cancer Detection

---

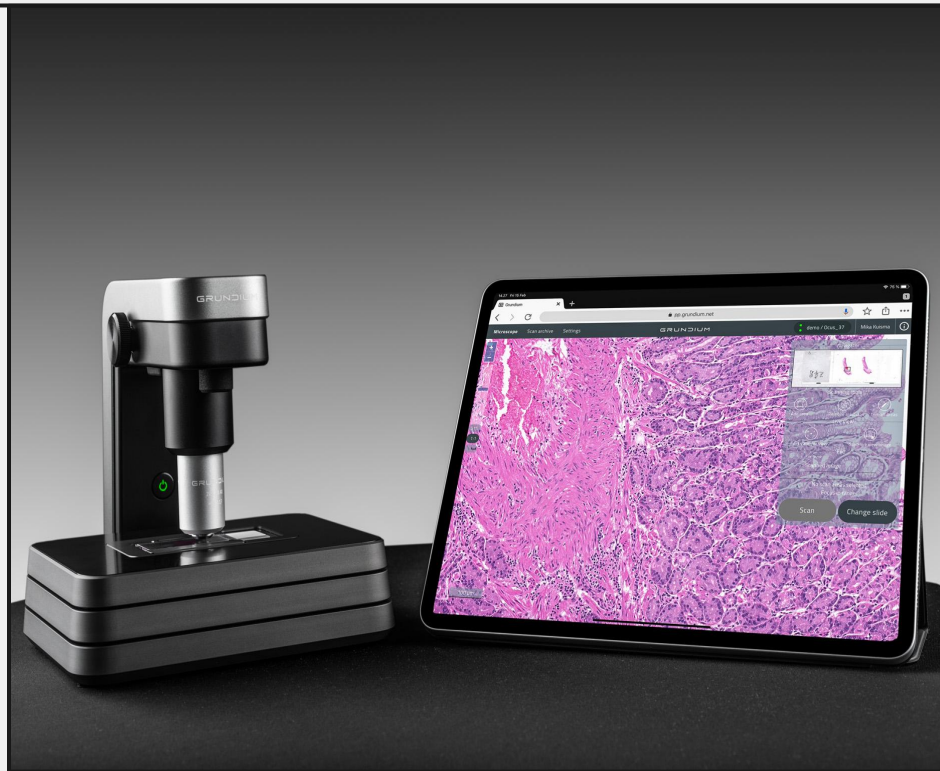
Diagnosi di tumori al sistema linfatico tramite tecniche  
di deep learning applicate alle immagini



# IL CONTESTO

La *patologia digitale* è una branca della patologia che si concentra sulla diagnosi di malattie tramite l'analisi di immagini digitali. Tale pratica grazie all'introduzione del Whole Slide Imaging (**WSI**), sta ricevendo sempre più attenzione nel campo della medicina diagnostica, con l'obiettivo di ottenere analisi efficienti ed economiche.

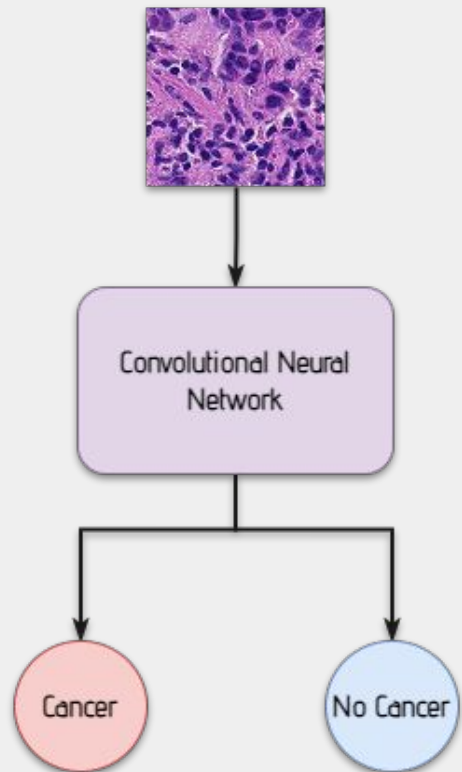
L'utilizzo di modelli neurali in ambito medico è in costante crescita, come dimostra il rapido aumento di strumenti di analisi basati su tecniche di machine learning.



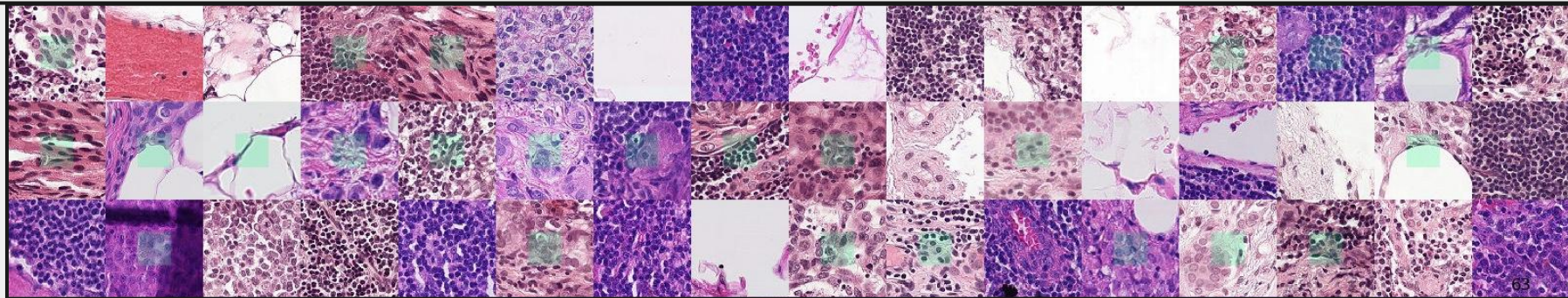
# OBIETTIVO e MOTIVAZIONI

L'obiettivo del progetto è realizzare dei modelli validi nel *riconoscimento dei tumori al sistema linfatico*, partendo da scansioni digitali dei campioni biotici. Si tratta quindi di un task di *classificazione binaria* su immagini, che verrà affrontato addestrando una Convolutional Neural Network, architettura particolarmente adatta all'ambito della Computer Vision.

Questo tipo di applicazioni è molto richiesto in ambito medico in quanto aiuta i patologi nell'effettuare diagnosi sin dai primi stadi della malattia in modo accurato. Ciò permette di prevenire il peggioramento della malattia, quindi è di aiuto anche per gli stessi pazienti.



# IL DATASET



Il dataset utilizzato per lo sviluppo del progetto è reso disponibile dalla piattaforma Kaggle. È un adattamento del dataset originale, pubblicato nel 2018, in quanto non contiene immagini duplicate. Ogni immagine è a colori con una dimensione  $96 \times 96$ px, inoltre la label di riferimento indica la presenza o meno di tessuto metastatico nella *regione centrale* di  $32 \times 32$ px. La regione esterna è fornita per consentire la progettazione di modelli convoluzionali che non utilizzino il padding, garantendo un comportamento coerente quando il modello è applicato a un'immagine WS1 tramite sliding window.

Ai fini del progetto si è deciso di *normalizzare*, per tutte le immagini, il valore dei pixel nel range  $[0,1]$ .

# IL DATASET

---

## Originale



220.025 immagini, *sbilanciate* e senza alcuna divisione in sottocartelle.



## Small

21.326 immagini di training, 3762 di validation e 5000 di test.  
Classi *bilanciate* e divisioni in sottocartelle.

## Medium



100.352 immagini tra training e validation e 18.000 di test.  
Classi *bilanciate* e divisioni in sottocartelle.

# APPROCCIO METODOLOGICO

---

Si è deciso di sperimentare due distinte tipologie di approcci:

- **Fine tuning** di tre reti preaddestrate per il task Imagenet: VGG16, DenseNet-121 e MobileNetV2
- **From Scratch**. Realizzazione di una Convolutional Neural Network ad hoc per il task affrontato

Per l'addestramento si è deciso di fissare un *massimo di 50 epoche*, e di utilizzare la tecnica dell'*EarlyStopping* per fermare l'apprendimento dopo 3 epoche senza miglioramenti nella validation loss. Come loss function la scelta è ricaduta sulla *binary crossentropy* per via della natura binaria del problema.

# DATA AUGMENTATION

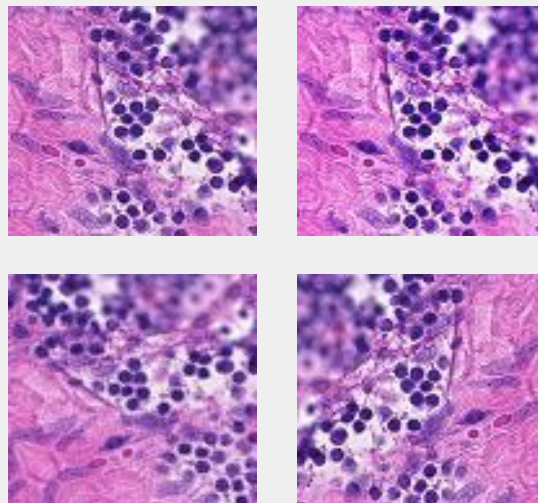
---

Le tecniche di data augmentation permettono di ampliare il dataset a disposizione senza effettivamente raccogliere nuovi elementi. Si è scelto di provare ad introdurre queste tecniche poichè si è immaginato che scanner differenti potessero restituire immagini lievemente diverse.

Nello specifico si è scelto di introdurre le seguenti distorsioni alle immagini:

- ***Random flip***, orizzontale e verticale
- ***Random rotation***
- ***Random contrast***

Queste tecniche sono state applicate soltanto alle immagini utilizzate per il training, e soltanto in una seconda fase, per verificare se potessero introdurre dei miglioramenti significativi.



# OTTIMIZZAZIONE DEGLI IPERPARAMETRI

---

In tutti gli approcci provati, si è scelto di utilizzare tecniche di *hyperparameter optimization* per individuare i parametri ottimali. Questo ha anche consentito di stabilire le migliori scelte architettureali sia nel caso delle reti pre addestrate, per decidere quanti layer fully connected aggiungere e quanti neuroni in ciascuno di essi, sia per la rete from scratch per definirne l'intera struttura.

Per tutti gli approcci si è dunque fatto ricorso all'algoritmo di *ottimizzazione bayesiana*, che nel caso specifico è stata definita per trovare gli iperparametri che minimizzassero la validation loss.

L'algoritmo utilizza ad ogni iterazione un nuovo set di iperparametri, selezionato tenendo conto dei risultati ottenuti agli step precedenti tramite inferenza bayesiana evitando dunque di provare tutte le possibili combinazioni.



# FINE TUNING

---

Per l'approccio di fine tuning si è scelto di utilizzare tre architetture diverse scegliendole in base al numero di parametri, operazioni e accuracy raggiunta nel task originale.

## VGG16

- Sviluppata nel 2014
- Stack convoluzionali 3x3
- +16M parametri
- Accuracy ~0.73

## DENSENET-121

- Sviluppata nel 2018
- DenseBlock
- Collegamenti residuali
- +7M parametri
- Accuracy ~0.75

## MOBILENET-V2

- Sviluppata nel 2017 (Google)
- BottleNeck residual block
- Low resources (per mobile)
- Leggera (14MB)
- +2M parametri
- Accuracy ~0.73

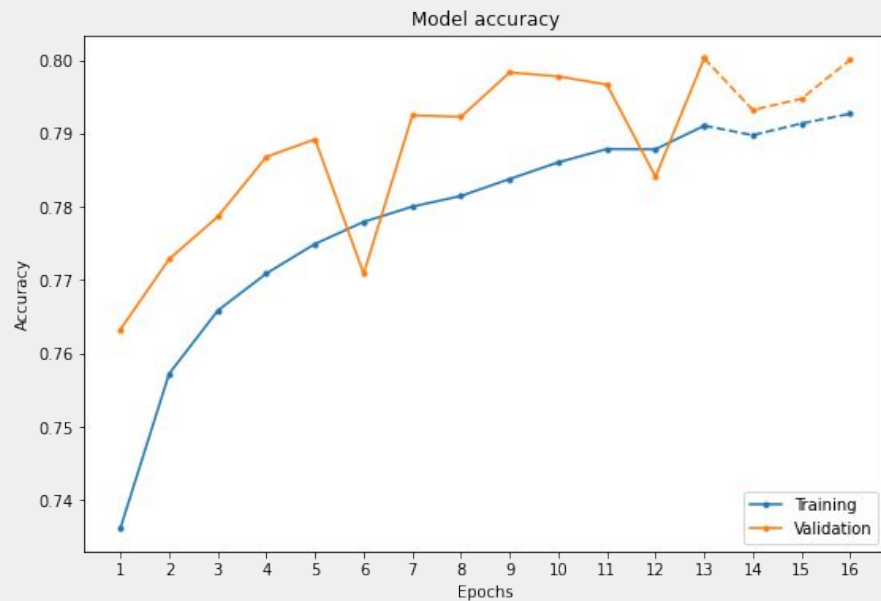
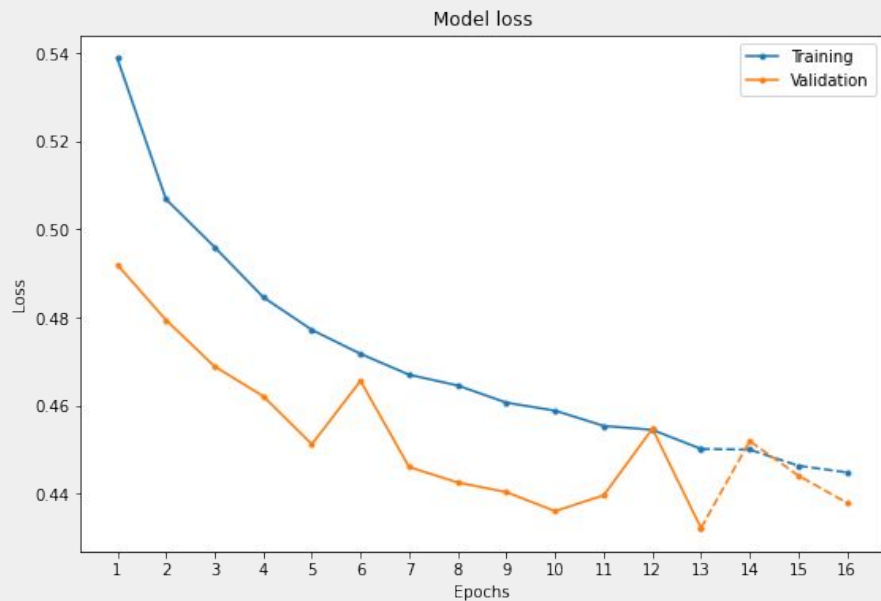
Tutte le architetture sono state tagliate al termine dei blocchi convoluzionali e i pesi frozen. Inoltre sono stati aggiunti nuovi layer FC finali con rispettivamente 64, 32 ed 8 neuroni.

# FINE TUNING: IPERPARAMETRI

Tutte le architetture di fine tuning sono state sottoposte all'ottimizzazione degli iperparametri, valutando anche i possibili layer di data augmentation.

Ottimizzazione degli iperparametri			
	VGG16	DenseNet-121	MobileNet-V2
Configurazione layer FC	[32, 32, 8]	[128, 16, 16, -]	[128, 32, 8]
Probabilità di dropout	[0.0, 0.0, -]	[0.0, 0.0, -, -]	[0.2, 0.5, -]
Learning rate	0.0001	0.01	0.0001
Random flip	False	False	False
Random rotation	[-0.05, 0.05]	0.0	0.0
Random contrast	0.0	0.0	[-0.2, 0.2]
Accuracy	0.75	<u>0.79</u>	0.74

# DENSENET-121: TRAINING



# RETE FROM SCRATCH: ARCHITETTURA

Per la realizzazione della rete from scratch si è deciso di mantenere il formato originale previsto dal dataset, ovvero immagini RGB con dimensione (96,96). Allo scopo di selezionare l'architettura migliore si è definito uno spazio di possibili iperparametri che definissero le caratteristiche dei layer convoluzionali e dei layer fully connected. Di seguito l'elenco dei parametri sottoposti al processo di ottimizzazione e dei rispettivi possibili valori:

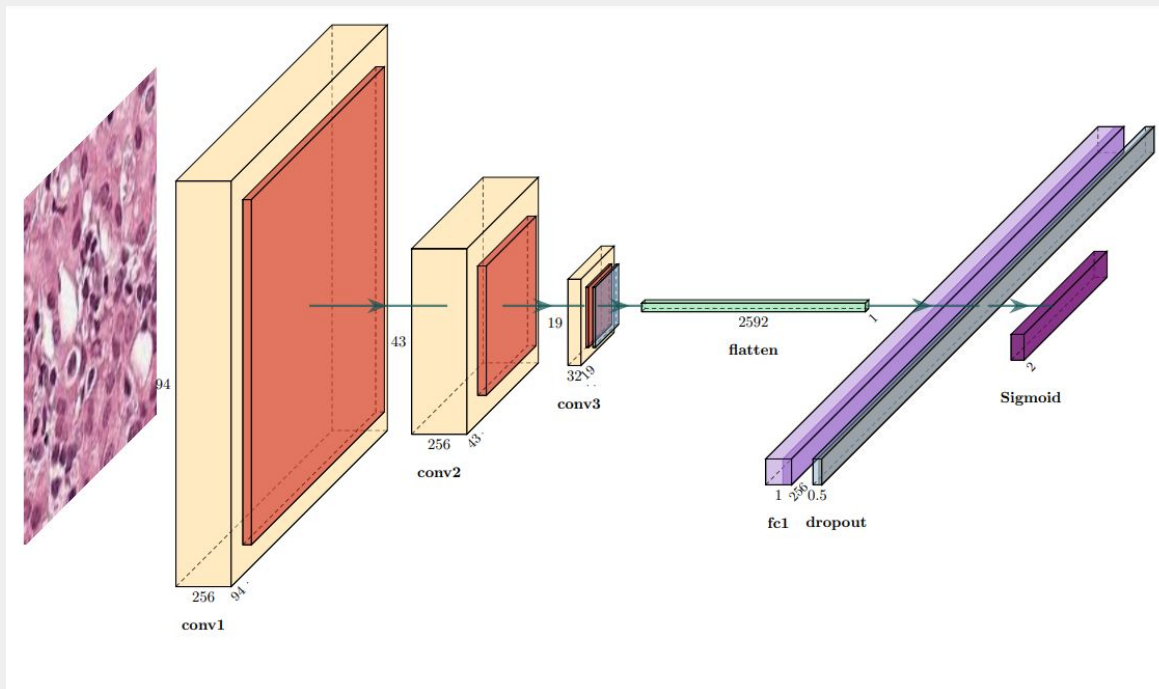
Layer Convoluzionali	
Numero di blocchi convoluzionali (Conv2D + MaxPool2D)	1; 2; 3
Dimensione del kernel (diversa per ciascun layer)	(3,3); (5,5)
Numero di filtri (diverso per ciascun layer)	32; 64; 128; 256
Probabilità di dropout	0; 0.2; 0.5

Layer Fully Connected	
Aggiunta di un layer fully connected prima dell'output layer	True; False
Dimensione del layer	64; 128; 256
Numero di neuroni (diverso per ciascun layer)	32; 64; 128; 256
Probabilità di dropout	0; 0.2; 0.5

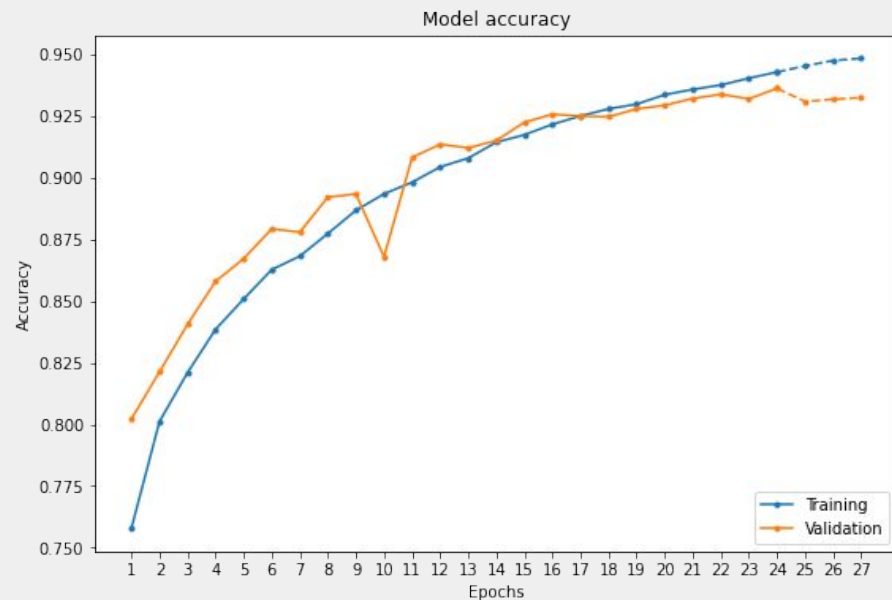
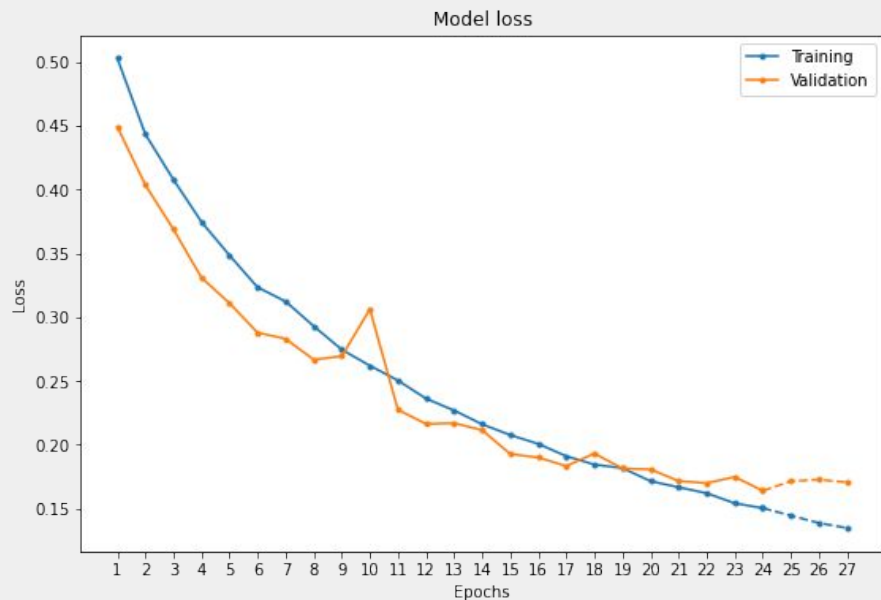
Altro	
Algoritmo di ottimizzazione	SGD; RmsProp; Adam
Learning Rate	1e-2; 1e-3; 1e-4
Funzione di attivazione	ReLu; Leaky-ReLu

# RETE FROM SCRATCH: ARCHITETTURA

A seguito di un processo di *ottimizzazione bayesiana* di 50 iterazioni, è stata individuata la combinazione di iperparametri ottimale

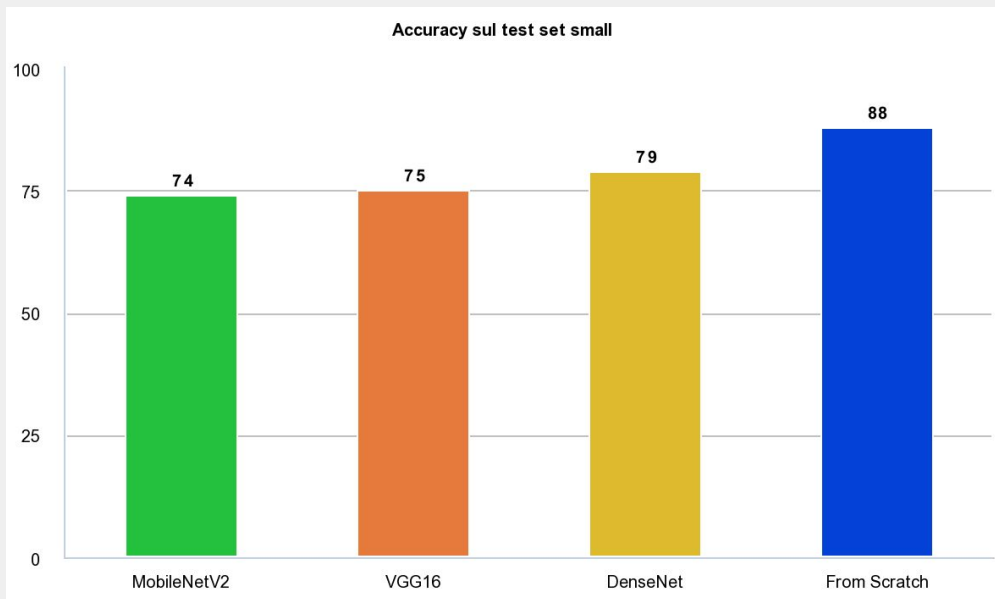


# RETE FROM SCRATCH: TRAINING



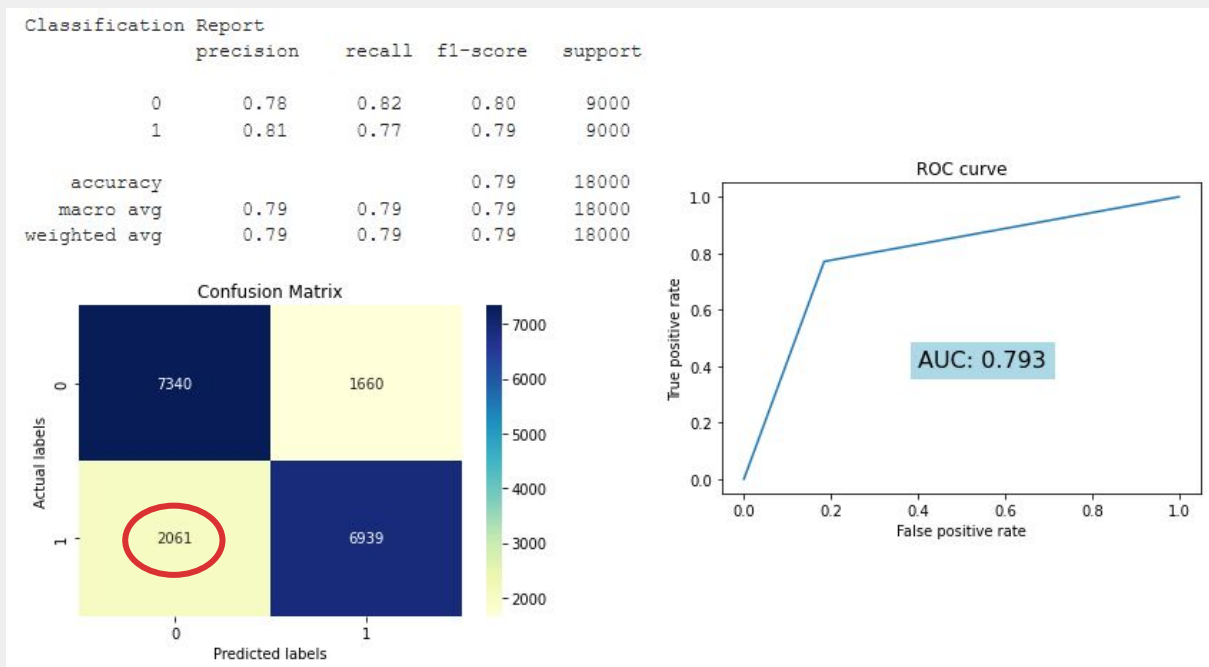
# RISULTATI e VALUTAZIONE

Per valutare le capacità predittive dei diversi modelli, si sono utilizzati gli appositi test set creati in fase di ridimensionamento del dataset. Oltre alle classifiche metriche di *loss* e *accuracy* si è deciso di considerare anche la *recall* della classe positiva, quella delle immagini di tumori.



# RISULTATI e VALUTAZIONE (2)

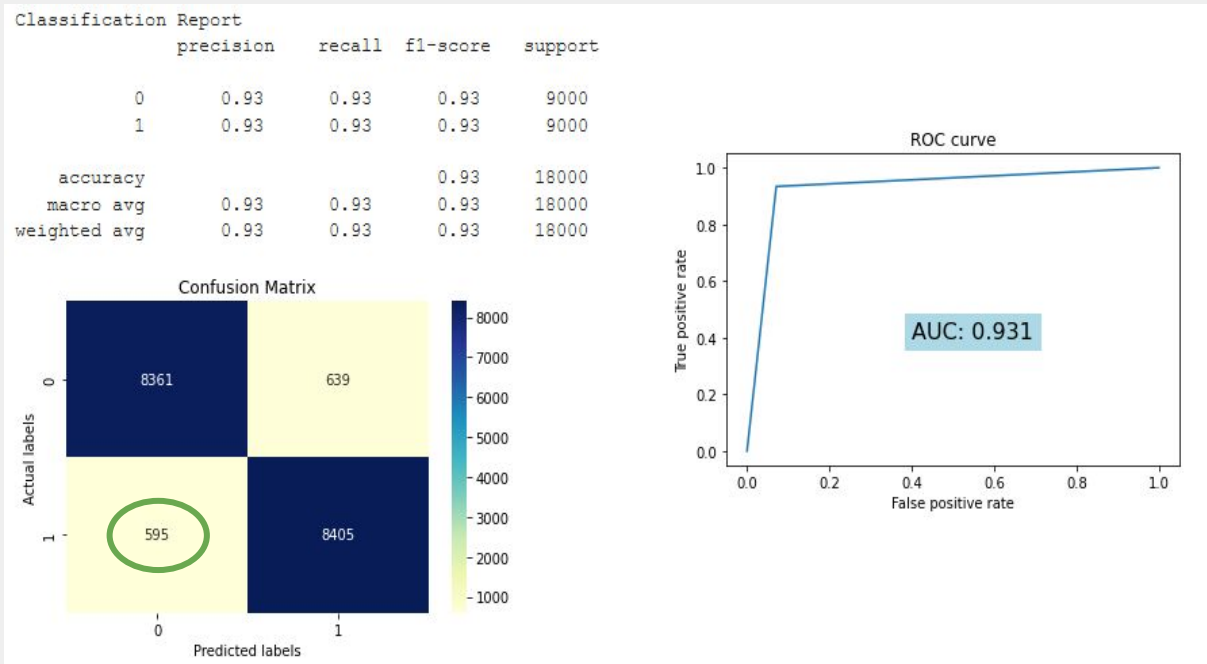
I due modelli che hanno ottenuto le performance migliori sul test set *'small'*, sono poi stati ri-addestrati sul test set *'medium'* e confrontati nuovamente e più in dettaglio. Si evidenziano le performance della rete DenseNet-121:





# RISULTATI e VALUTAZIONE (3)

Il modello from scratch risulta performare notevolmente meglio:



# RISULTATI e VALUTAZIONE (4) – TEMPO

Tutti i modelli creati sono stati valutati anche rispetto al tempo richiesto per effettuare una singola predizione. Questo perchè modelli con predizioni real-time potrebbero essere applicati a macchinari più sofisticati e a scenari più complessi come colonscopie con detection live di eventuali tumori.

Modello	Tempo medio per immagine(*)
From Scratch	0.109s
MobileNetV2	0.157
VGG16	0.193
DenseNet-121	0.218

\* Il tempo di predizione è stato calcolato come media rispetto al tempo necessario per predire un intero batch a parità di potenza computazionale. Inoltre è strettamente dipendente dalle risorse computazionali a disposizione.

# CONCLUSIONI

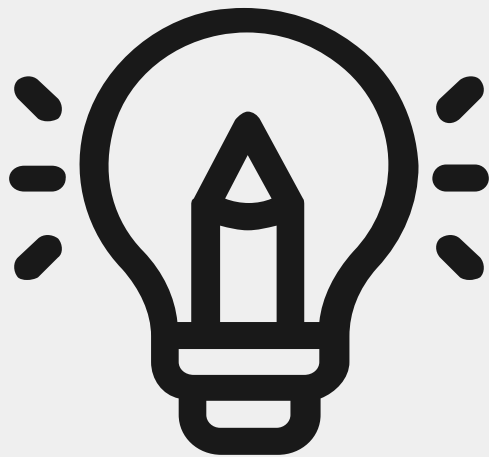
---

La rete from scratch ha ottenuto risultati **soddisfacenti** e incoraggianti.

Troppi falsi negativi, ma si può ancora migliorare:

- Rimozione di **scansioni errate**
- Utilizzo dell'**intero dataset**
- Raccogliere informazioni da un **esperto di dominio**

Un possibile sviluppo futuro è la **compressione** del modello per ottenere predizioni in **real time**.



**DOMANDE?**

**GRAZIE PER L'ATTENZIONE**

# RIFERIMENTI BIBLIOGRAFICI

---

- V. Messina and S. Bianco, “Dispense e slide del corso advanced machine learning,” 2021.
- FDA (US Food and Drug Administration), “FDA allows marketing of first whole slide imaging system for digital pathology,”
- U. Muehlematter, P. Danio, and K. Vokinger, “Approval of artificial intelligence and machine learning-based medical devices in the usa and europe (2015–20): a comparative analysis”
- PCCam, “Histopathologic cancer detection,” 2019.
- K.-S. Lee, S.-H. Son, S.-H. Park, and E. Kim, “Automated detection of colorectal tumors based on artificial intelligence,”
- Magentiq Eye Ltd. (2015) Magentiq eye Ltd.