Master's degree ICT Internet Multimedia Engineering

MACHINE LEARNING FOR HUMAN DATA – FINAL EXAMINATION

Instructor

Michele Rossi - michele.rossi@unipd.it

Lab. classes

Francesca Meneghello - francesca.meneghello.1@unipd.it

Eleonora Cicciarella - eleonora.cicciarella@phd.unipd.it

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

1222·2022 800 ANNI

UNIVERSITÀ DEGLI STUDI DI PADOVA

General guidelines

The final exam is project-based

- This does not mean that you can avoid understanding the theory…see the next slides

1. Pick a project among the 11 options that we provide to you: each consists of a challenge and an associated dataset
2. Design one/more original solution(s) to the problem based on neural networks, implement it/them in TensorFlow and evaluate and compare the performance
3. Prepare a report and a presentation describing your work

- You can work in a group with another student
  - max 2 people per group
  - you are free to arrange the groups
  - both members have to contribute to the work

Exam dates and submission deadlines

- Exam: January 28-29, 2025
  - report+code submission deadline: Jan. 25, 2025
- Exam: February 18-19, 2025
  - report+code submission deadline: Feb. 15, 2025
- Exam: June 18-19, 2025
  - report+code submission deadline: June 15, 2025
- Exam: July 2-3, 2025
  - report+code submission deadline: June 29, 2025
- Exam: September 18-19, 2025
  - report+code submission deadline: Setp. 15, 2025

IMPORTANT NOTES on next page!

For the final examination you must

GROUP SELF-SELECTION
Group and project self-selection

1. Fill out the group selection form in Moodle indicating the students (1 or 2) in your group (we will send the instructions through the Moodle's news channel) → remember to do that!

ASSIGNMENT
Project report and code upload - January 28-29, 2025

2. Upload in Moodle (following the instructions about naming etc.)

FILE
Project report - Latex template

   A. a report (use the LaTex template available on Moodle)
   B. the code of the implementation in **TensorFlow**

3. Prepare a presentation through slides (20 minutes strict, possibly including a demo) for the day of the exam

The report

1. Should be done in LaTex following the template available on Moodle
2. Should be written in a clear and organized manner
3. Should include graphical presentations of your approaches
4. Should clearly show and discuss the results

FILE

Project report - Latex template

"We Rock the Hizzle and Stuff"
hints on how to write a nice research essay

Michele Rossi[†], Author two[‡]

Evaluation

- The evaluation will consider different aspects
  - about the report, the presentation and the project itself

project					written report			oral	
originality (10)	preprocessing (10)	learning architecture (15)	comparison / performance analysis (10)	live demo (10)	clarity of exposition (10)	completeness of results (accuracy, complexity, time) (10)	technical soundness (15)	duration (10)	clarity (10)

Show how the approach works on some examples (using pre-trained networks) or a walkthrough

- Your grade will be computed by:
  1. Summig the points (max 110)
  2. Multiplying the sum by 0.424242
  3. Subtracting 11.69
  4. Limiting the score in [0, 32]

see the details in the LaTex template for the final report (on Moodle)

Guidelines

- Prepare the project and the report considering the grid we use for the evaluation (see previous slide)
  - pay attention to the pre-processing phase (normalize the data)
  - create original neural network architectures
  - compare the performance of different approaches (use the correct metrics...check about data balancing)
  - evaluate the performance of the algorithms in terms of running time and complexity (memory occupation)

Guidelines

- Be creative!
  - We provided you with some ideas for possible project developments, but original works are always welcome!
  - You can use the neural network architectures seen during the labs and/or experiment with new approaches!
  - We provide you with some references but try to explore a bit other contributions in the literature that may be helpful (search for them in https://scholar.google.it/)
  - Pre-processing techniques may be useful
  - Implement your own neural network architecture…DO NOT use pre-trained models from Keras: the objective of the project is that you put into practice the things you learned during the theoretical lessons, not to improve your skills about reusing networks/code developed by others :)

Guidelines

- The use of TensorFlow is mandatory
  - Pytorch is only allowed for spiking neural networks through snntorch
- The use of pretrained networks is not allowed
  - You can use them for comparison but cannot be the main architectures
- During the exam we will ask you the reasoning behind using the specific architectures (e.g., CNN/RNN/attention…)
  - **Do not use the NN functions as black boxes**: you need to understand why you are using the specific architectures
  - **REMEMBER**: Python is not intelligent, it takes something as input and provides an output, it only checks the shape of the data → pay attention and use your theoretical knowledge

Common mistakes to avoid

- Data not correctly normalized
  - This is an important step for ML algorithms to not have biases in the algorithm
- Preprocessing not considered
  - In addition to ML you may need to apply some signal processing algorithms to clean the data before NN
- Train/validation/test sets not correctly split
  - The three sets do not have to overlap: no data from training should be used during validation or test
- Validation performed on a small number of samples that is not statistically significant
  - e.g., evaluation performed on 1 or 2 samples…
- Complexity of the algorithms in terms of time and memory not analyzed
- Use wrong input data
  - e.g., for IMU datasets, obtain the activity prediction by using single IMU measurements and not a sequence of measurements

MIME
Master's degree ICT Internet Multimedia Engineering

PROJECT C3

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

800 ANNI 1222·2022

UNIVERSITÀ DEGLI STUDI DI PADOVA

Project C3 “Lung disease prediction from X-ray images”

Reference papers

[Jiancheng2023] Yang, Jiancheng, et al. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification, Scientific Data 10.1 (2023): 41.

[Xiaosong2017] Wang, Xiaosong, et al. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Dataset (2 GB uncompressed)

https://medmnist.com/ dataset+code

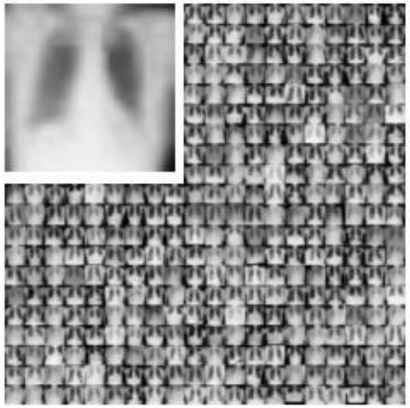
https://zenodo.org/records/10519652 download the chestMNIST dataset

Three versions: image sizes 1x64x64, 1x128x128, and 1x224x224

Dataset description

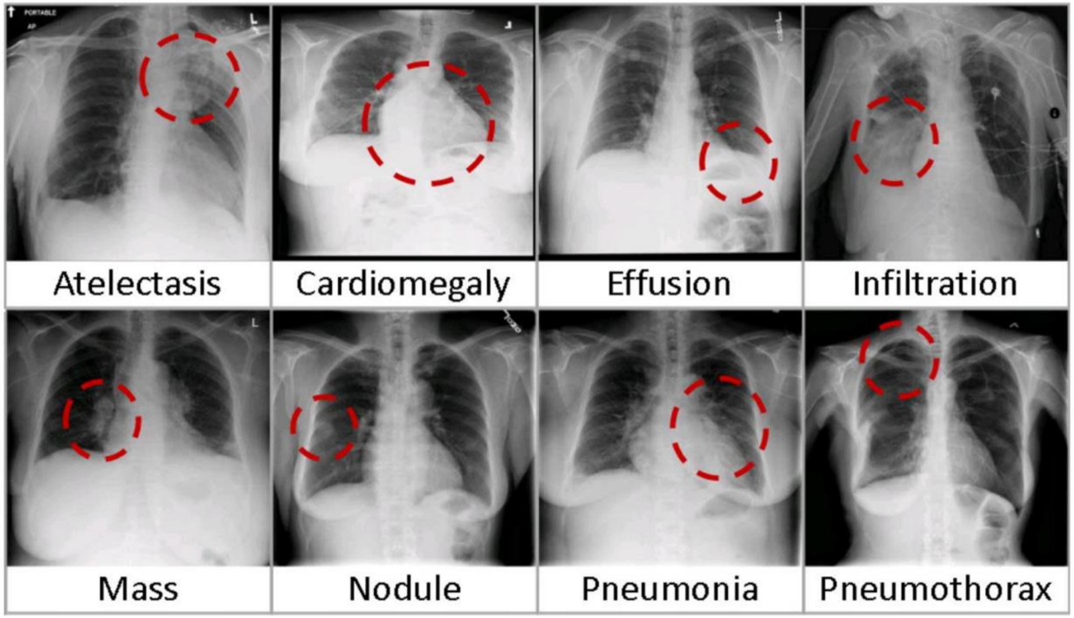
- 112,120 frontal X-ray images
- 30,805 unique patients
- 14 different classes of diseases

ChestMNIST

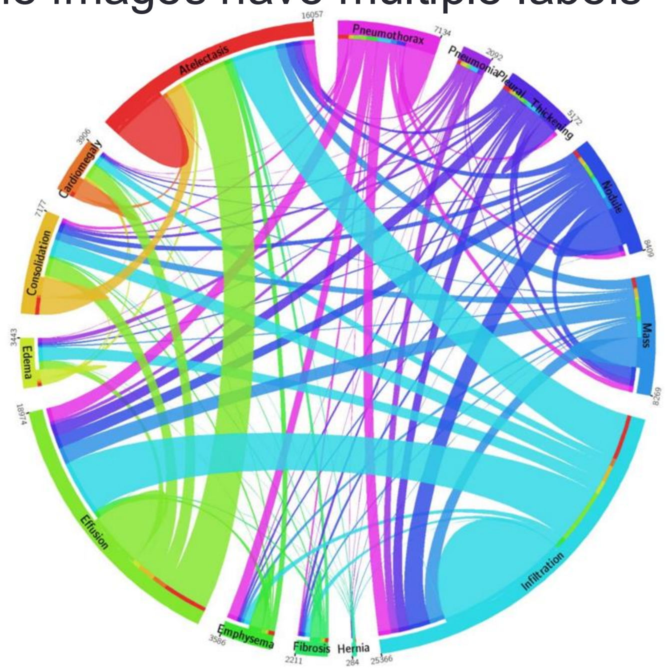


MedMNIST2D	Data Modality	Tasks (# Classes/Labels)	# Samples	# Training / Validation / Test
ChestMNIST	Chest X-Ray	Multi-Label (14) Binary-Class (2)	112,120	78,468 / 11,219 / 22,433

B. Eight visual examples of common thorax diseases



some images have multiple labels



Approach in [Xiaosong2017]

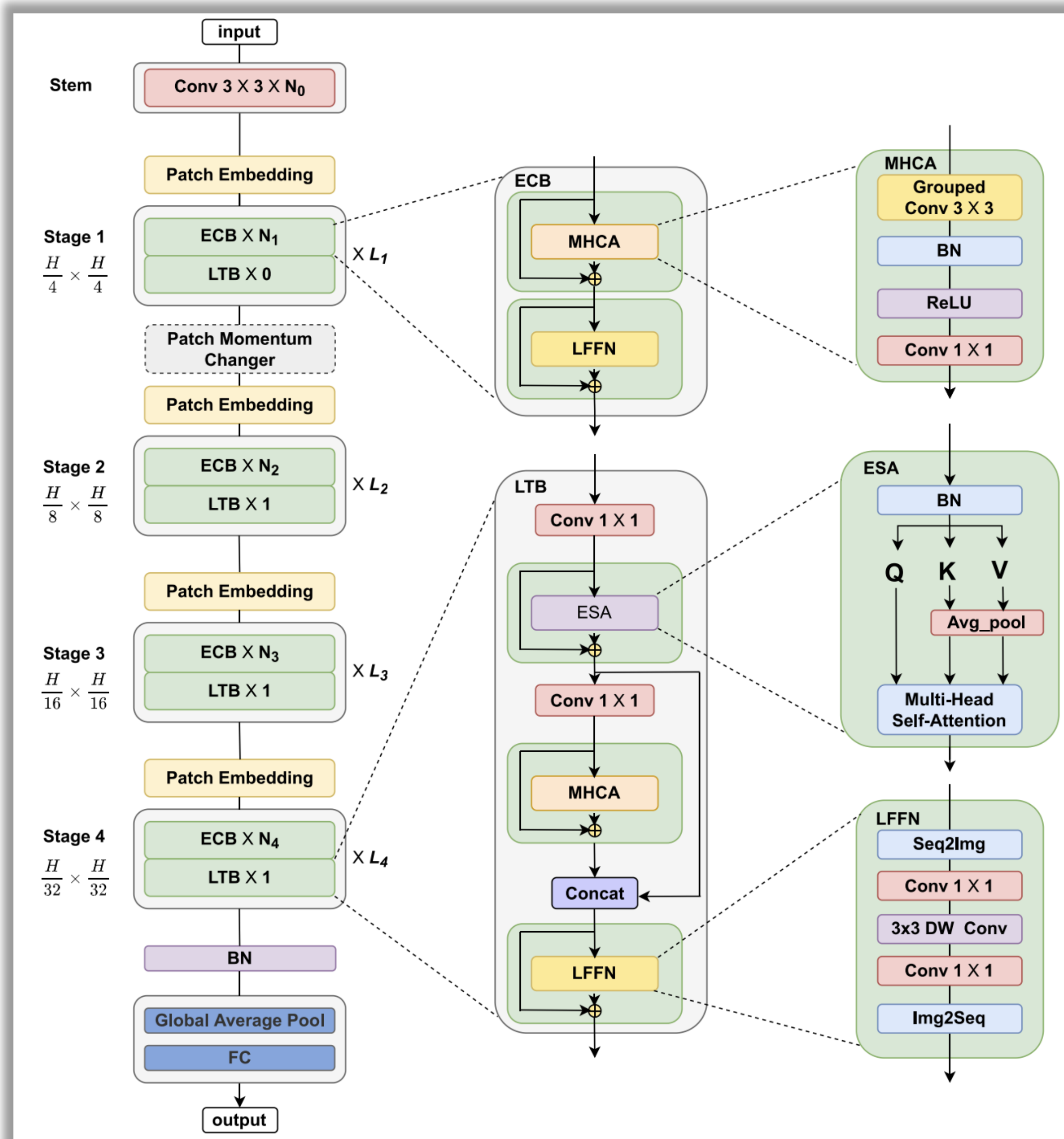
- Classification considering 8 classes
- 4 different pre-trained models: AlexNet, GoogLeNet, VGG and ResNet-50

Setting	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pneumothorax
Initialization with different pre-trained models								
AlexNet	0.6458	0.6925	0.6642	0.6041	**0.5644**	0.6487	0.5493	0.7425
GoogLeNet	0.6307	0.7056	0.6876	0.6088	0.5363	0.5579	0.5990	0.7824
VGGNet-16	0.6281	0.7084	0.6502	0.5896	0.5103	0.6556	0.5100	0.7516
ResNet-50	**0.7069**	**0.8141**	**0.7362**	**0.6128**	0.5609	**0.7164**	**0.6333**	**0.7891**
Different multi-label loss functions								
CEL	0.7064	0.7262	0.7351	0.6084	0.5530	0.6545	0.5164	0.7665
W-CEL	0.7069	0.8141	0.7362	0.6128	0.5609	0.7164	0.6333	0.7891

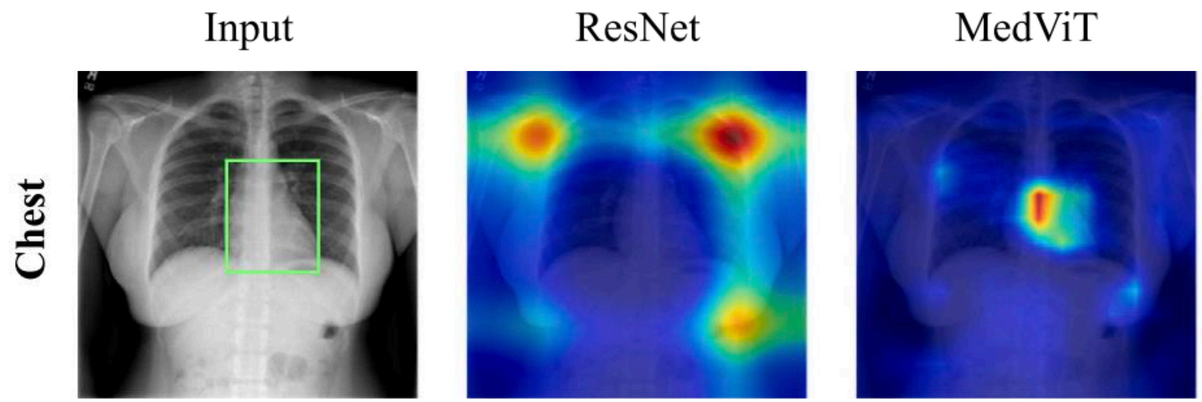
Table 3. *AUCs of ROC curves for multi-label classification in different DCNN model setting.*

Approach in [Jiancheng2023]

- MedViT: composed of a patch embedding layer, transformer blocks and a series of stacked convolution in each stage



Approach in [Jiancheng2023]



Methods	ChestMNIST	
	AUC	ACC
ResNet-18 (28) [20]	0.768	0.947
ResNet-18 (224) [20]	0.773	0.947
ResNet-50 (28) [20]	0.769	0.947
ResNet-50 (224) [20]	0.773	0.948
auto-sklearn [75]	0.649	0.779
AutoKeras [76]	0.742	0.937
Google AutoML [77]	0.778	0.948
MedVIT-T (224)	0.786	0.956
MedVIT-S (224)	0.791	0.954
MedVIT-L (224)	0.805	0.959

Possible project developments

- Classification task
  - use row images or extract features
  - use the 8 classes considered in [Xiaosong2017] or 14 classes as done in [Jiancheng2023]
  - try with different image sizes (64x64, 128x128, 224x224)
  - possible approaches: classify the entire image or use subpatches and then apply a decision fusion mechanism
  - use attention mechanisms
  - use spiking neural networks

- Architectures
  - CNN, RNN, attention, …

MiME.
Master's degree ICT Internet Multimedia Engineering

MACHINE LEARNING FOR HUMAN DATA – FINAL EXAMINATION

Instructor

Michele Rossi - michele.rossi@unipd.it

Lab. classes

Francesca Meneghello - francesca.meneghello.1@unipd.it

Eleonora Cicciarella - eleonora.cicciarella@phd.unipd.it

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

800 ANNI 1222·2022

UNIVERSITÀ DEGLI STUDI DI PADOVA