

# MACHINE LEARNING FOR HUMAN DATA – FINAL EXAMINATION

---

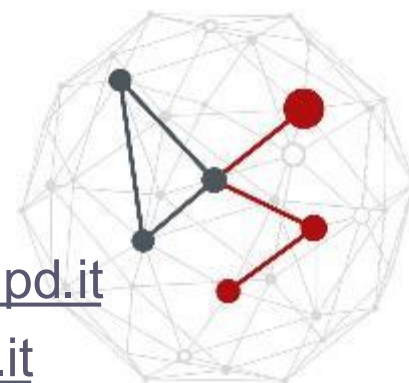
Instructor

**Michele Rossi** - [michele.rossi@unipd.it](mailto:michele.rossi@unipd.it)

Lab. classes

**Francesca Meneghello** - [francesca.meneghello.1@unipd.it](mailto:francesca.meneghello.1@unipd.it)

**Eleonora Cicciarella** - [eleonora.cicciarella@phd.unipd.it](mailto:eleonora.cicciarella@phd.unipd.it)



# 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!**

# Exam dates: important notes

- The exam will be held **in presence**
  - online exams are no longer allowed by UNIPD
- Depending on the number of students we may need to split you into groups that will take the exam **on different days**
  - the exam dates in the previous slide indicate the first days of the session
  - please, be prepared to be scheduled for a different day than the one indicated (try to be available for the day of the exam and the following ones; in case you have unmovable appointments, inform us **as soon as you enroll**)
  - we will send you the schedule some days before the exam when the UNIWEB enrollment will close

# 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<sup>†</sup>, Author two<sup>‡</sup>

# Group self-selection in Moodle

**Deadline:** when you enroll in UNIWEB for the exam  
You can fill it out also before (recommended)

2024-SC2738-003PD-2024-SCQ4106915-N0-SC2738 / Group and project self-selection



GROUP SELF-SELECTION

## Group and project self-selection

Group self-selection

Settings

Groups

More ▼

**Opens:** Wednesday, 30 October 2024, 4:30 PM

Please, **create a new group** by yourself or with one of your colleagues (i.e., max 2 people per group).

As the **group name**, use SurnameA (e.g., Rossi) or SurnameA\_SurnameB (e.g., Meneghello\_Cicciarella) depending on whether you are alone or with a colleague.

As the **group description**, indicate the ID of the project you selected (A1, A2, A3, B1, B2, C1, C2, C3, C4, D1, D2, D3).

Set a **password** for the group and share it with your colleague to enable them to become part of the group (not needed if you are alone).

-----

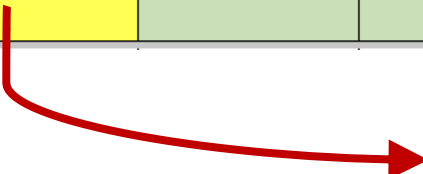
**Available projects**

# 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)

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



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

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

# PART B

# AUDIO SIGNALS

---



# Proposed Projects



## PART A – ON BODY AND ENVIRONMENTAL SENSORS

- 1) A1: Activity recognition with four accelerometers
- 2) A2: Pathological gait recognition
- 3) A3: Motor imagery classification from EEG for brain computer interface

## PART B – AUDIO SIGNALS

- 1) **B1: Speech command recognition (keyword spotting)**
- 2) **B2: Environmental sound classification**

## PART C – IMAGES

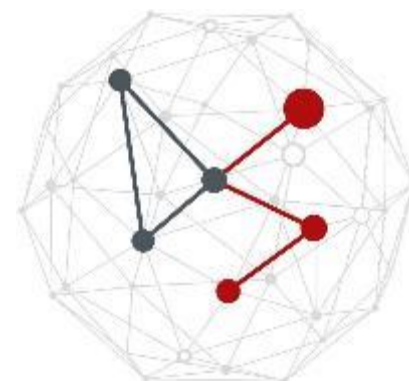
- 1) C1: Sleep posture monitoring
- 2) C2: Bone age prediction from hand radiographs
- 3) C3: Lung disease prediction from X-ray images
- 4) C4: Blood cell type prediction

## PART D – RADIO SIGNALS

- 1) D1: Activity recognition through Wi-Fi devices
- 2) D2: Gesture recognition through radars

# PROJECT B1

---



# Project B1 “Speech recognition”

## Reference papers

[Sainath15] Tara N. Sainath, Carolina Parada, [Convolutional Neural Networks for Small-footprint Keyword Spotting](#), INTERSPEECH, Dresden, Germany, September 2015.

[Warden18] Pete Warden, [Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition](#), arXiv:1804.03209, April 2018.

<https://arxiv.org/abs/1804.03209>

- The authors are from [Google Inc.](#)
- Reference dataset released by Google [Warden18]



# Dataset description

- Reference dataset for small-footprint keyword spotting (KWS)
  - Released in August 2017
  - 65,000 one-second-long utterances of 30 words
  - by thousands of different people
  - released under creative commons 4.0 license
  - collected by AIY (<https://aiyprojects.withgoogle.com/>)

## Google blog

<https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html>

## Speech dataset (2.11 GB uncompressed)

[http://download.tensorflow.org/data/speech\\_commands\\_v0.02.tar.gz](http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz)

# Approaches for implementing a KWS engine

- **LVCSR based KWS** - This approach uses a two-stage process. In the first stage, the transcription of the speech into words is done using a **Large Vocabulary Continuous Speech Recognition (LVCSR)** engine, outputting formatted text. In the second stage, a textual search for the key-words within the text is performed. Using this approach, results from LVCSR and the text search are combined to spot the key-words
- **Phoneme Recognition based KWS** - This approach also uses a two-stage process. In the first stage, the speech is transformed to a sequence of phonemes. In the second stage, the application searches for phonetically transcribed key-words in the phoneme sequence obtained from the first stage
- **Word Recognition based KWS [Sainath15]** - This approach searches for the key-words in a **one stage operation**. The recognition is phoneme-based and the KWS engine looks for the keyword in the speech stream based on a target sequence of phonemes representing the key-word

# CNN model from [Sainath15]

- Features are obtained from raw audio data
- **40-dimensional log Mel filterbanks coefficients**
  - audio frame length 25 ms
  - with a 10 ms time shift
- **At every new audio frame**
  - **Feature vector** is obtained
  - And stacked with 23 frames to the left and 8 to the right (32 frames total)
  - This returns 32 frames at a time, **spanning over  $31 \times 10 \text{ ms} + 25 \text{ ms} = 0.335 \text{ s}$**
- **A Convolutional Neural Network (CNN) is used to detect words**
- **Input to the CNN is a matrix of size  $t \times n = 32 \times 40 = 1,280$  elements**
  - $t$  represents the number of elements in time (number of audio frames)
  - $n$  represents the number of elements in the frequency domain (Mel features)

# CNN model from [Sainath15]

- 27-44% improvement for KWS with respect to traditional neural networks
- The paper focus is on
  - Devising CNN architectures with small memory footprint
  - Playing with CNN parameters (number of kernels, strides, pooling, etc.)

# Possible project developments

- **Experiment with different audio features**
  - Type of coefficients (e.g., discrete Wavelet transform)
  - Design of Mel filterbanks
- **Play with a standard/deep CNN using**
  - dropout, regularization
- **Investigate recent/new ANN architectures**
  - Autoencoder-based (CNN/RNN autoencoder + following SVM)
  - Attention mechanism and/or inception-based CNN networks
  - Comparison of different architectures: memory vs accuracy

# Useful resources

## Recent developments

[[Chorowski15](#)] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, [Attention-Based Models for Speech Recognition](#), Conference on Neural Information and Processing Systems (NIPS), Montréal, Canada, 2015.

[[Tang18](#)] R. Tang and J. Lin, [Deep residual learning for small-footprint keyword spotting](#), in IEEE ICASSP, Calgary, Alberta, Canada, 2018.

[[Andrade18](#)] D. C. de Andrade, S. Leo, M. L. D. S. Viana, and C. Bernkopf, [A neural attention model for speech command recognition](#), arXiv:1808.08929, 2018. <https://arxiv.org/pdf/1808.08929.pdf>

**White Paper:** “Key-Word Spotting - The Base Technology for Speech Analytics”  
<https://pdfs.semanticscholar.org/e736/bc0a0cf1f2d867283343faf63211aef8a10c.pdf>

**Example code:**

[https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/speech\\_commands/](https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/speech_commands/)

# MACHINE LEARNING FOR HUMAN DATA – FINAL EXAMINATION

---

Instructor

**Michele Rossi** - [michele.rossi@unipd.it](mailto:michele.rossi@unipd.it)

Lab. classes

**Francesca Meneghello** - [francesca.meneghello.1@unipd.it](mailto:francesca.meneghello.1@unipd.it)

**Eleonora Cicciarella** - [eleonora.cicciarella@phd.unipd.it](mailto:eleonora.cicciarella@phd.unipd.it)

