Vision Transformer for femur fracture classification

Leonardo Tanzi[1], Andrea Audisio[2], Giansalvo Cirrincione[3], Alessandro Aprato[2], Enrico Vezzetti[2]

[1]Polytechnic University of Turin
[2]University of Turin
[3]University of Picardie Jules Verne

Published Paper on Injury

Article on Medium

Table of Contents

1. Introduction

1.1 Introduction

Musculoskeletal diseases represent the most common cause of long-term disability worldwide

The correct evaluation and classification of fractures by specialists strongly affect future patients' treatment

In particular, in 2010 the estimated incidence of hip fractures was 2.7 million patients per year globally

Common issues

- Superimposition of soft tissues in obese patients
- Complex patients' positioning
- Stressful working environment of Emergency Departments
- Second opinion not always available
- Intrinsic complexity of the classification

Idea

Implementing a CAD (Computer Assisted Diagnosis) system in doctors' workflow might directly impact patients' outcomes

1.2 Overview

Initial approaches for fracture classification focused on conventional machine learning techniques

After the introduction of AlexNet in **2012**, the research community switched to CNN

Most of the work focuses on binary classification

Low impact on specialist diagnosis

Just 4 previous works tackled multi-class classification

Results are still non-optimal

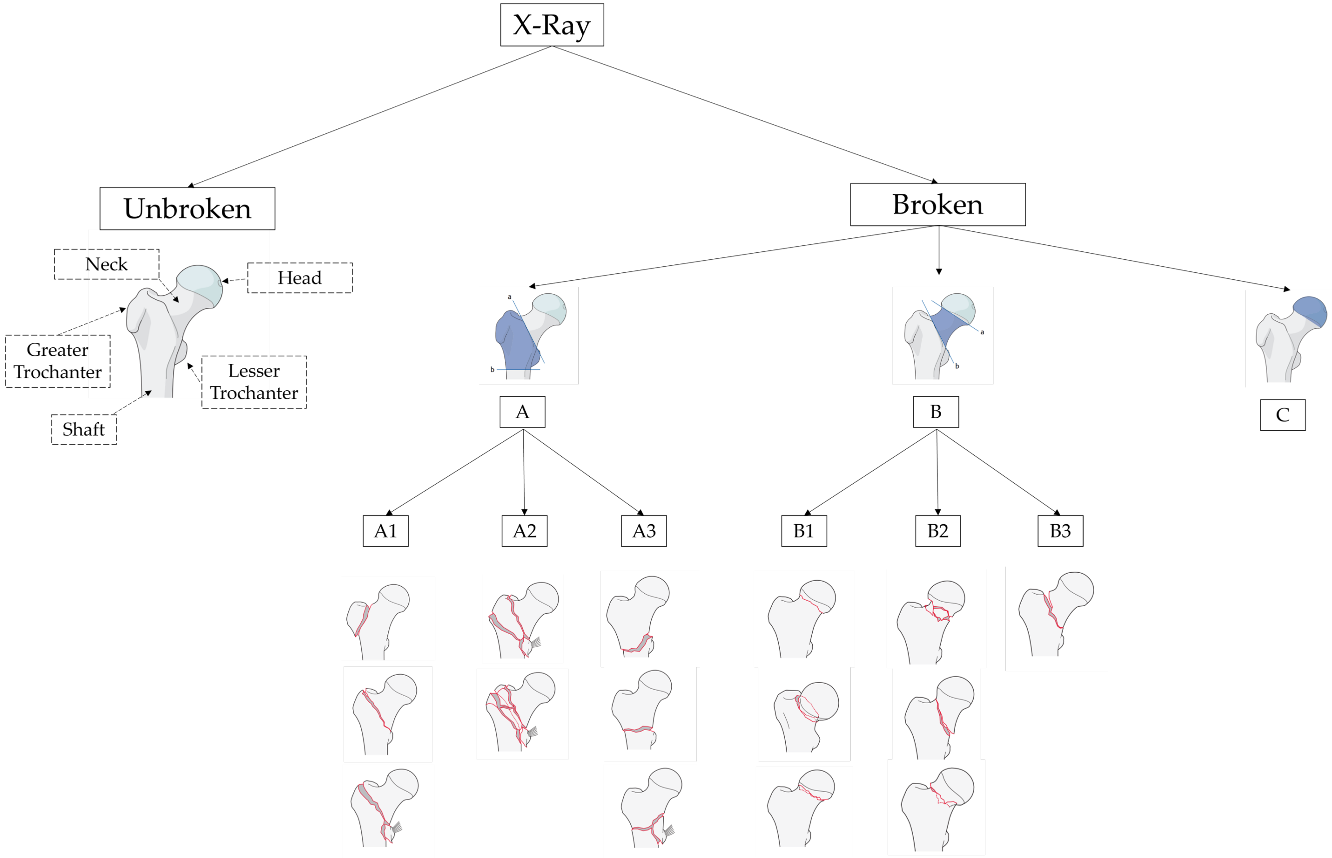
A new paradigm called Transformer in **2017**, based on self-attention, is introduced in NLP

In **2020**, the Vision Transformer was proposed, the first vision architecture based on **self-attention** which reaches the SOTA in many tasks

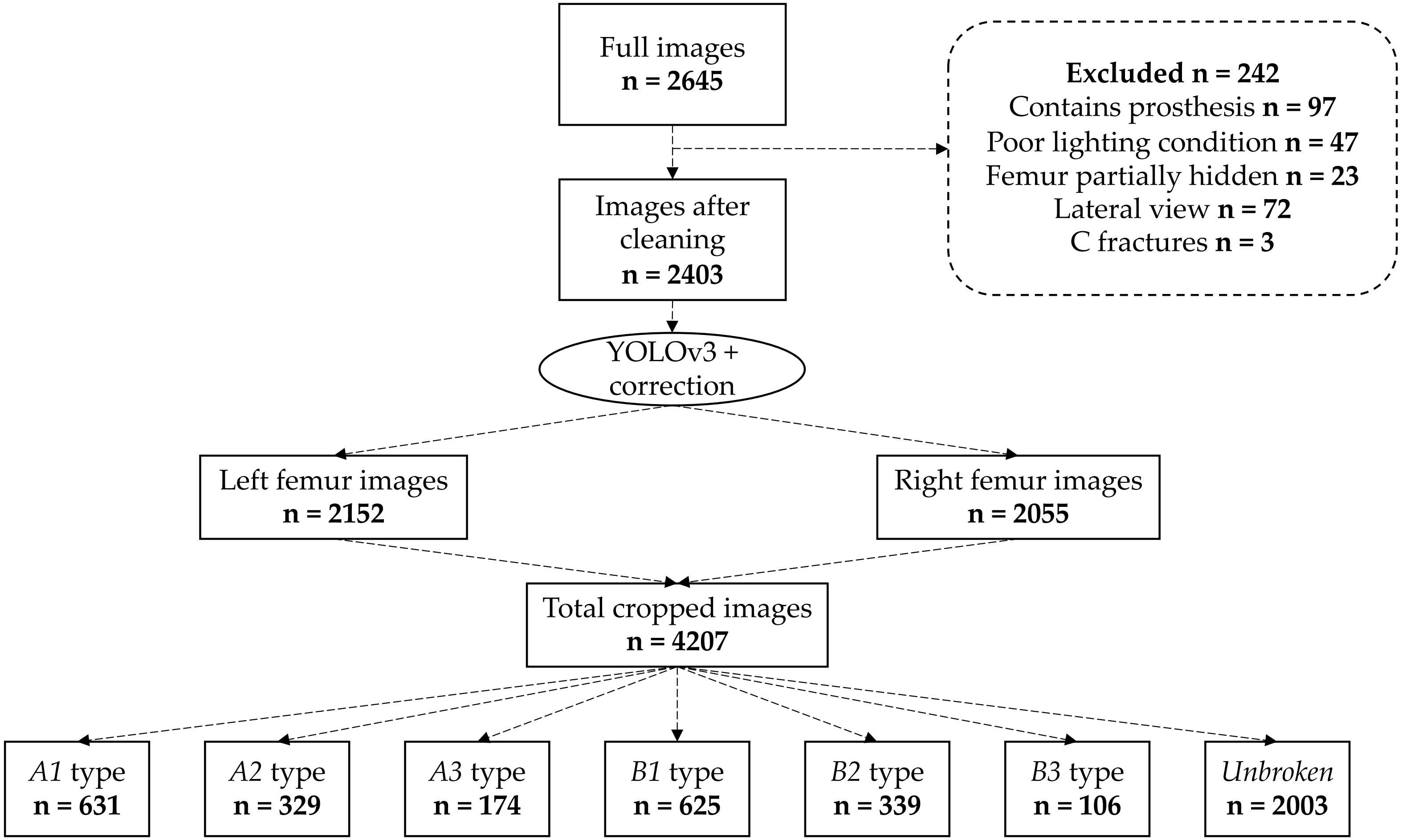
Can we apply self-attention to femur fracture classification?

1.3 AO Classification

The AO classification is hierarchical and provides a well-defined methodology for assessing fractures correctly



1.4 Dataset Creation



1.5 Dataset Samples

A1

A2

A3

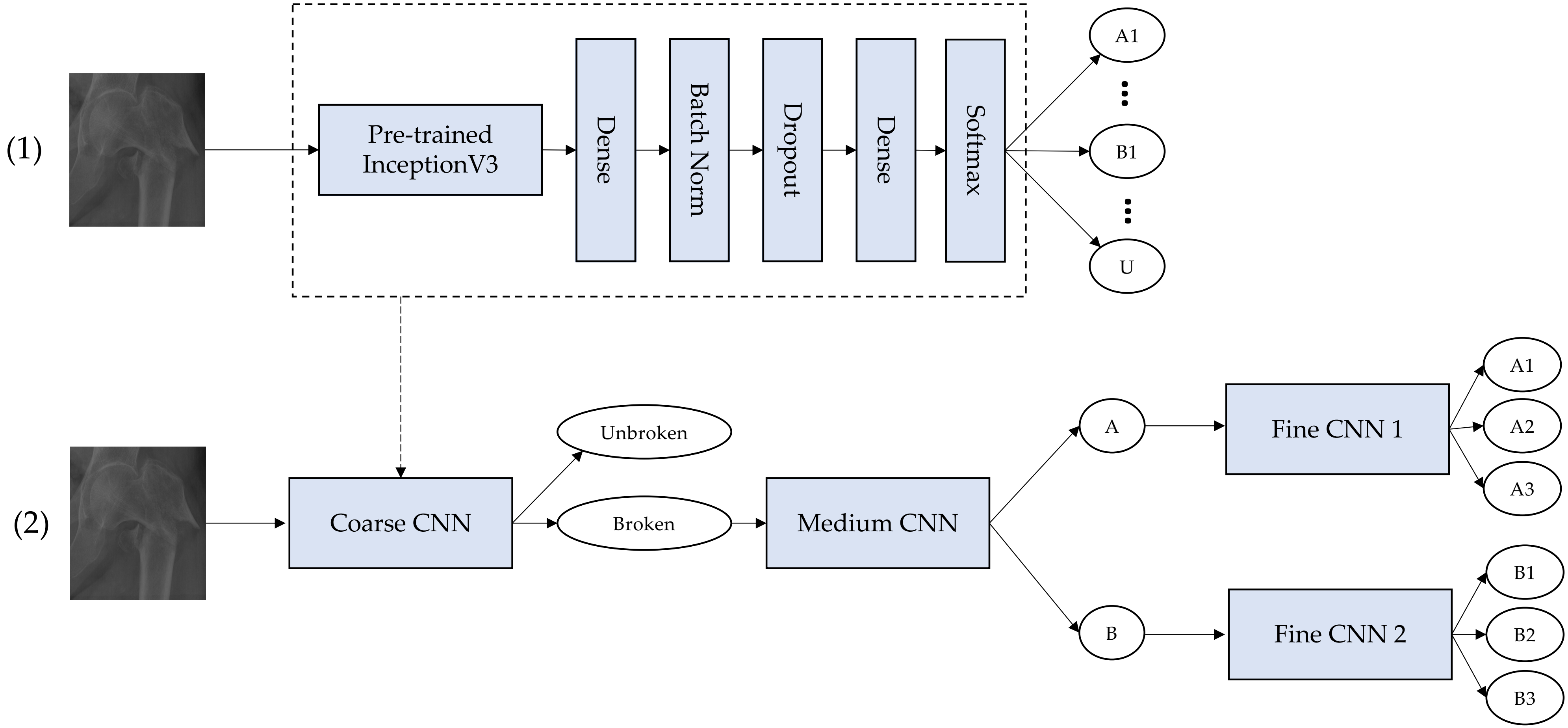
B1

B2

B3

Unbroken

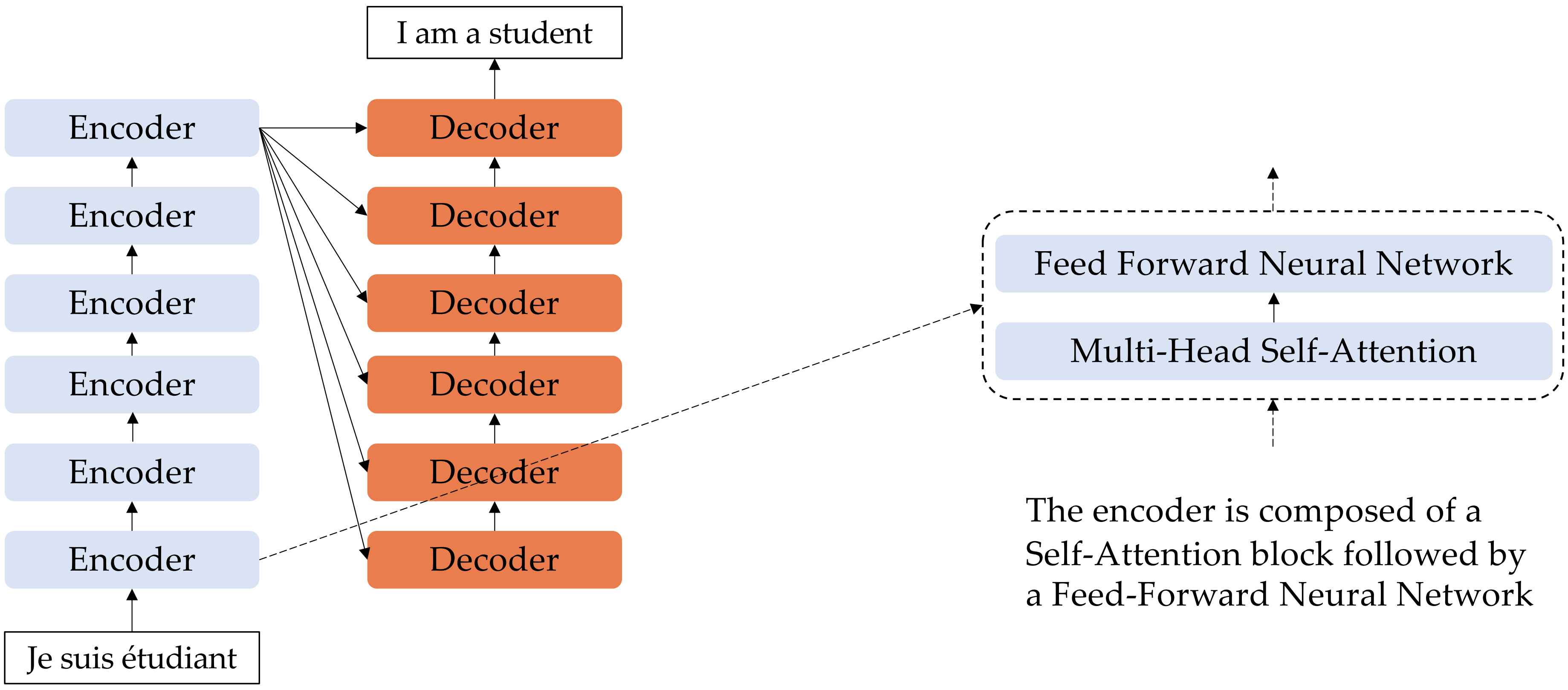
1.6 Baselines



2. Methods

2.1 Transformer Intuition

The transformer's original architecture was composed of a series of encoders followed by a series of decoders

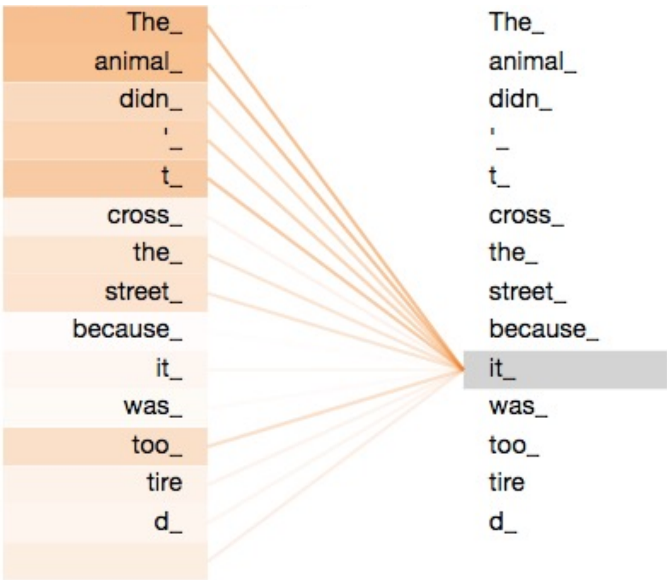


The encoder is composed of a Self-Attention block followed by a Feed-Forward Neural Network

2.2 Self-Attention Intuition

Self-Attention is a method to understand the relevant words in a sentence in relation to the one you're currently processing.

*"The animal didn't cross the street because **it** was too tired"* → What does *"it"* in this sentence refer to?



When the model is processing the word *"it"*, self-attention allows to associate *"it"* with *"animal"* (and other relevant words)

Self-Attention relies on three matrixes: **Query (Q)**, **Key (K),** and **Value (V)**

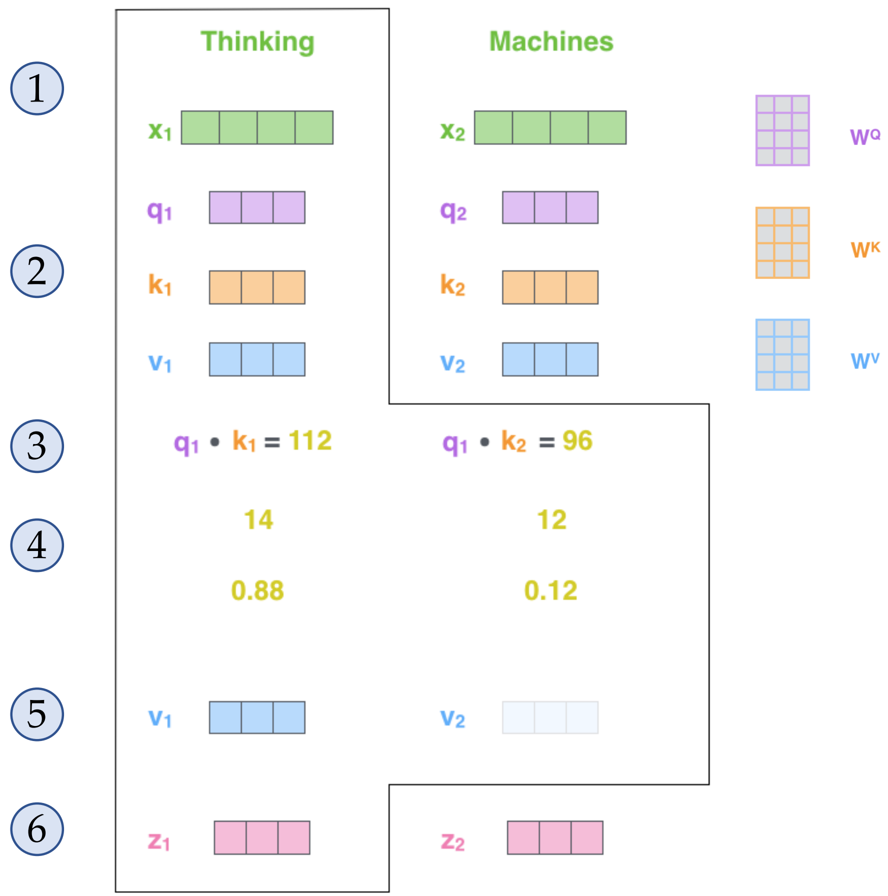
"A crude analogy is to think of it as searching through a filing cabinet. The Query is the note with a tag of the topic you're researching. The Keys are the labels of the folders inside the cabinet. When you match the tag with a note, we take out the content of that folder, this content is the Values vector"

2.3 Self-Attention in Details

We want to apply self-attention to the sentence *"Thinking machines"*

1. Embed words to tokens
2. For each token, compute Query, Key and Value by multiplying each token for three learnable and shared matrixes W_q, W_k and W_v
3. To compute self-attention for the first token, multiple its Q by all the other K to obtain the **Score (S)**
4. Divide by $\sqrt{d_k}$ and pass through a Softmax
5. Multiply the results for the V vector
6. Sum all the V to obtain the final Z vector

$$SA = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

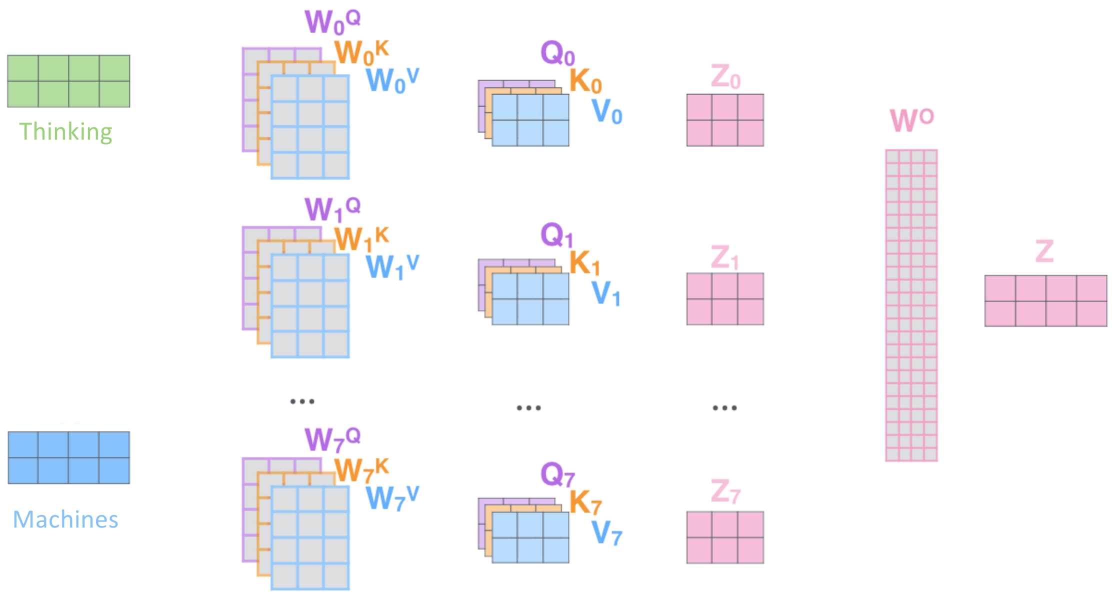


2.4 Multi-Head Self-Attention

Considerations on **Q**, **K**, **V**, and **S**:

- Q is a representation of the current token used to score against all the other token
- K can be seen as a set of labels that we match against Q in our search for relevant words
- V contains actual word representation
- S determines the amount of focus to put on each token

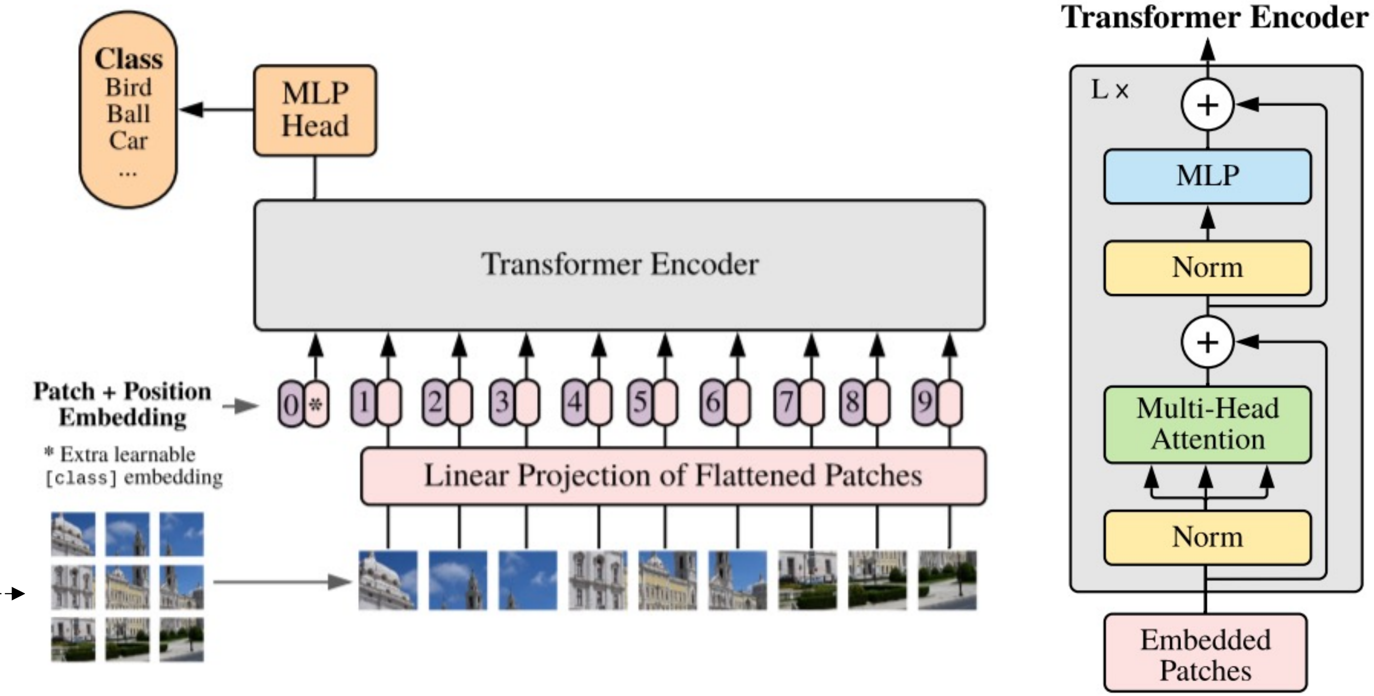
In practice, more than one head is used (**multi-head self-attention**), in order to have multiple representation. The multiple output are multiplied by a learnable matrix W_0 used to keep the dimension fixed



2.5 Vision Transformer

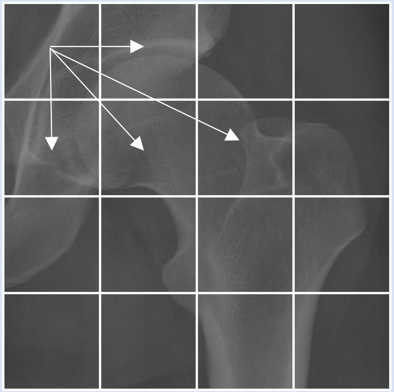
ViT was applied in this paper, one of the first vision solutions leveraging self-attention

The particularity of this architecture is that, to handle image data, it divides the images into grids and focuses on small patches.



2.6 ViT for femur fractures

The main idea behind the use of the Transformer in this work is its global attention



Comparing each token with each other token is an approach very similar to the one used by specialists

2.7 Model Selection

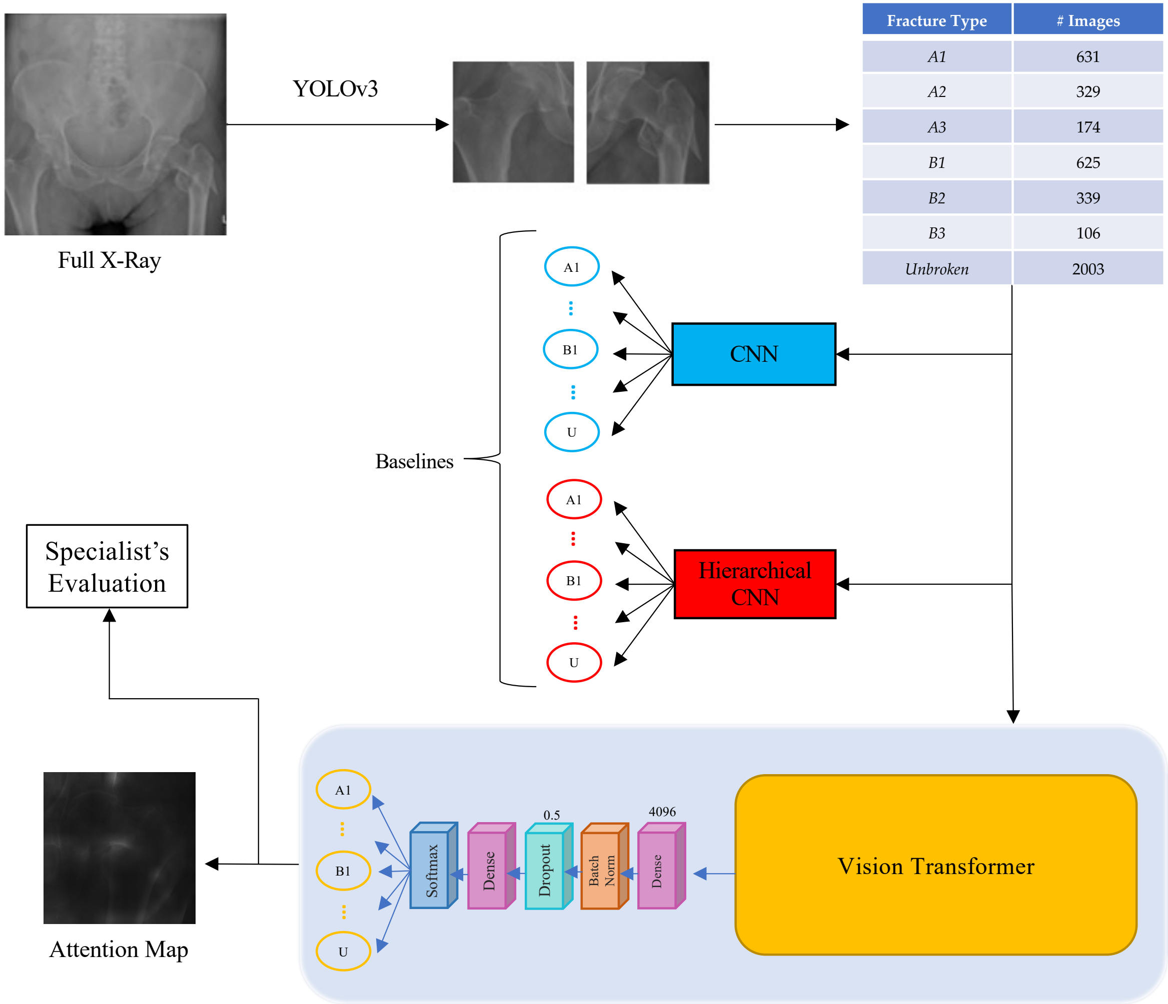
ViT has been shown not to work well with small datasets

Alternatives

- Use pre-trained networks
- Add a convolution step before self-attention

Architecture	Precision	Recall	F1-score
B16	0.77 (CI 0.67-0.88)	0.74 (CI 0.59-0.88)	0.75 (CI 0.63-0.87)
B32	0.67 (CI 0.51-0.83)	0.65 (CI 0.48-0.83)	0.65 (CI 0.49-0.81)
L16	0.77 (CI 0.64-0.90)	0.76 (CI 0.62-0.91)	0.77 (CI 0.64–0.89)
L32	0.71 (CI 0.59-0.83)	0.65 (CI 0.48-0.82)	0.66 (CI 0.53-0.80)
CCT	0.39 (CI 0.18-0.59)	0.38 (CI 0.12-0.65)	0.38 (CI 0.15-0.60)

2.8 Full Pipeline



Fracture Type	# Images
A1	631
A2	329
A3	174
B1	625
B2	339
B3	106
Unbroken	2003

3. Results

3.1 Comparison with Baselines

CNN

CNN	Precision	Recall	F1-score	# Images
A1	0.42	0.53	0.47	91
A2	0.65	0.43	0.51	94
A3	0.65	0.60	0.63	25
B1	0.59	0.61	0.60	90
B2	0.43	0.45	0.44	49
B3	0.43	0.19	0.26	16
Unbroken	0.85	0.89	0.87	282
Macro AVG	0.57 (CI 0.42-0.72)	0.53 (CI 0.33-0.72)	0.54 (CI 0.36-0.71)	

Hierarchical CNN

Hierarchical CNN	Precision	Recall	F1-score	# Images
A1	0.36	0.51	0.42	91
A2	0.50	0.21	0.30	94
A3	0.20	0.32	0.24	25
B1	0.51	0.70	0.59	90
B2	0.59	0.20	0.30	49
B3	0.11	0.06	0.08	16
Unbroken	0.87	0.88	0.87	282
Macro AVG	0.44 (CI 0.21-0.68)	0.41 (CI 0.14-0.69)	0.40 (CI 0.15–0.64)	

ViT-L16

ViT	Precision	Recall	F1-score	# Images
A1	0.66 (↑24%)	0.66 (↑23%)	0.66 (↑19%)	91
A2	0.77(↑12%)	0.66 (↑23%)	0.71 (↑20%)	94
A3	0.92 (↑30%)	0.92 (↑32%)	0.92 (↑29%)	25
B1	0.74 (↑15%)	0.93 (↑23%)	0.82 (↑22%)	90
B2	0.79 (↑ 20%)	0.69 (↑24%)	0.74 (↑30%)	49
B3	0.56 (↑13%)	0.56 (↑37%)	0.56 (↑30%)	16
Unbroken	0.95 (↑8%)	0.94 (↑5%)	0.95 (↑8%)	282
Macro AVG	0.77 **(↑20%)** (CI 0.64-0.90)	0.76 **(↑23%)** (CI 0.62-0.91)	0.77 **(↑23%)** (CI 0.64–0.89)	

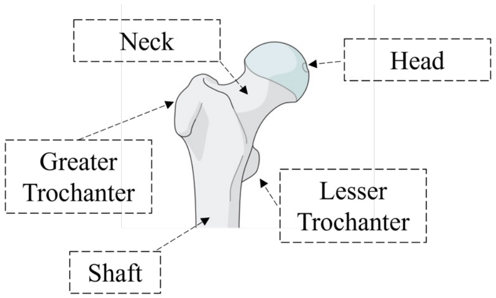
3.2 Specialists Evaluation

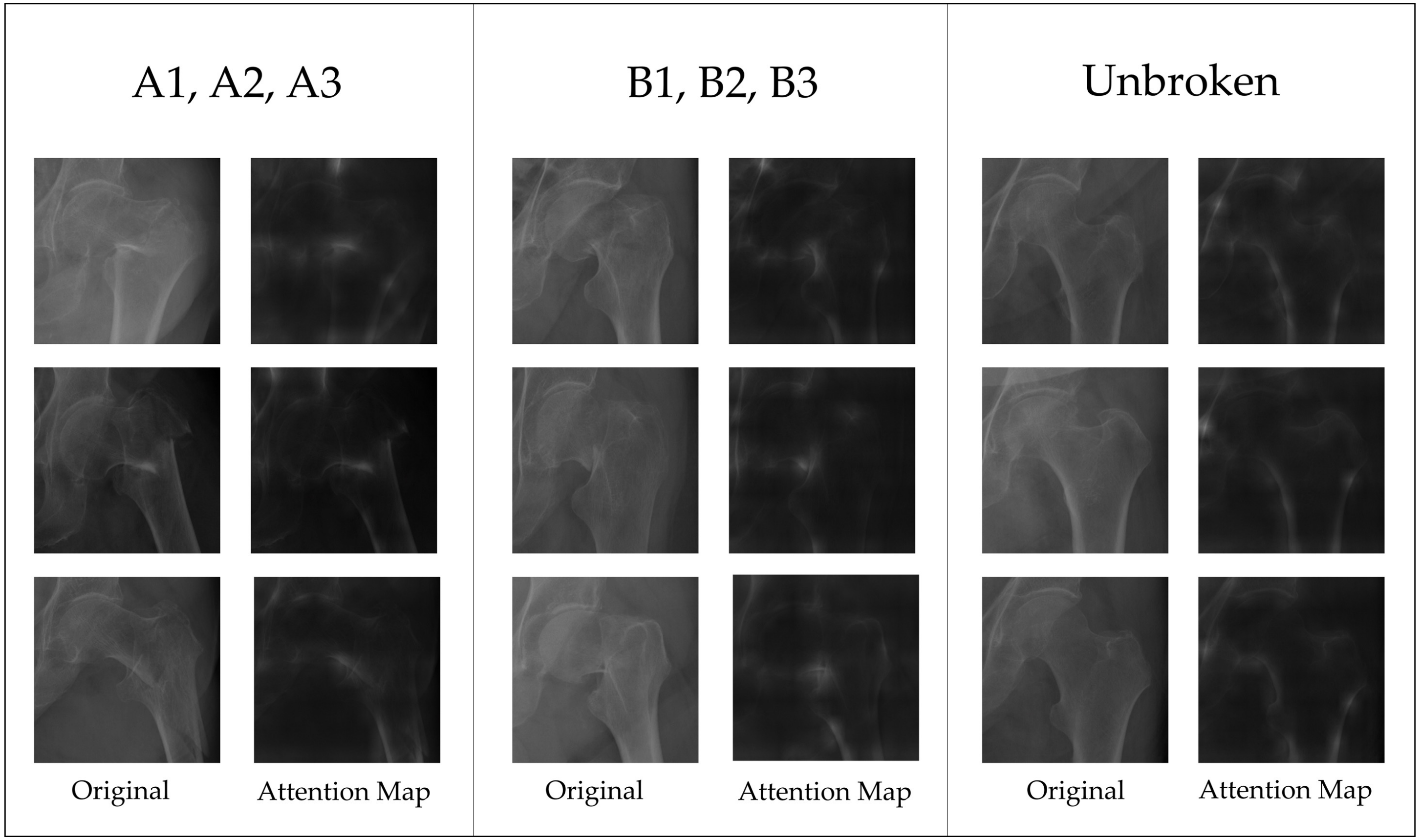
Specialist	Years of experience	Accuracy without CAD	Accuracy with CAD	Accuracy Improvement
Resident #1	2	0.55	0.90	0.35
Resident #2	1	0.55	0.89	0.34
Resident #3	2	0.53	0.98	0.45
Resident #4	4	0.69	1.00	0.31
Resident #5	3	0.63	0.98	0.35
Resident #6	2	0.53	0.96	0.43
Resident #7	3	0.64	0.98	0.34
Radiologist #1	10	0.81	1.00	0.19
Radiologist #2	15	0.90	1.00	0.10
Radiologist #3	7	0.80	1.00	0.20
Radiologist #4	13	0.87	1.00	0.13
Residents' Average		0.58 (CI 0.53 – 0.65)	0.96 (CI 0.92 – 0.99)	0.37 (CI 0.32 – 0.42)
Radiologists' Average		0.84 (CI 0.77 – 0.92)	1.00	0.15 (CI 0.08 – 0.23)
Total Average		**0.68 (CI 0.59 – 0.78)**	**0.97 (CI 0.94 – 1.00)**	**0.29 (CI 0.12 – 0.37)**

4. Discussion

4.1 Attention Maps

The attention maps, after being analyzed by a team of clinicians, showed how the network correctly focused on the trochanteric area for the *A* class, the neck and the greater trochanter for the *B* class, and around the whole cortex for the *Unbroken* class.





4.2 Comparation with SOTA

Paper	Method	Dataset	F1-score	Additional Notes
Our	ViT	2043 samples	0.77	
Lee et al. [1]	InceptionV3 followed by FCN and LSTM	786 samples, with 1, 6, and 8 samples, respectively, used to validate classes *B3*, *B1*, and *A3*	0.50	It also leverage text annotations, which are usually very hard to collect
Kazi et al. [2]	Attention module to locate the femur area followed by an InceptionV3 network	1173 samples, with 15 samples for A3 fractures	0.68	The class unbroken was not considered for multi-class classification

[1] Lee C, Jang J, Lee S, Kim YS, Jo HJ, Kim Y. Classification of femur fracture in pelvic X-ray images using meta-learned deep neural network. Scientific Reports, 2020

[2] Kazi A, Albarqouni S, Sanchez AJ, Kirchhoff S, Biberthaler P, Navab N, et al. Automatic classification of proximal femur fractures based on attention models. Machine learning in medical imaging, 2017

4.3 Limitations and future works

1) The evaluation was done through a web interface

Limitation	Possible Solution
• Specialists were not in a situation of stress • The short two weeks period between the two evaluations may have created a bias	• Further clinical studies in everyday routine • Wait more time before the second evaluation or utilize different sets of images with the same level of difficulty

4.3 Limitations and future works

2) Dataset imbalance and under-represented classes

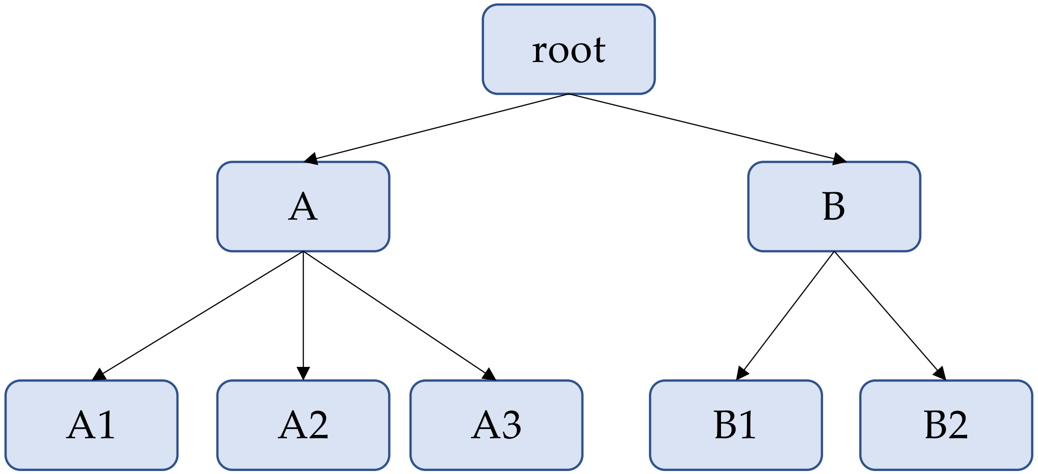
Limitation	Possible Solution
• Not enough samples • Data augmentation create fake fractures	Generative Adversarial Networks. How to use them to augment data in a reliable way?

4.3 Limitations and future works

3) Not leveraging the hierarchical structure

Limitation	Possible Solution
We are ignoring some information that could be very useful, especially for under-represented leaf nodes	• Hierarchical loss • Stop before prediction • New metric

4.3 Limitations and future works



Different level have different set of labels, in this case we have:

$$s_1 = \{A, B\}$$
$$s_2 = \{A1, A2, A3, B1, B2\}$$

We can define a loss as the sum of two weighted cross-entropy losses:

$$L = \alpha L_{fine} + L_{coarse}$$

4.3 Limitations and future works

Stop prediction at a certain level

Use a confidence score at each level to understand if it makes sense to continue in the classification

Adapt metric

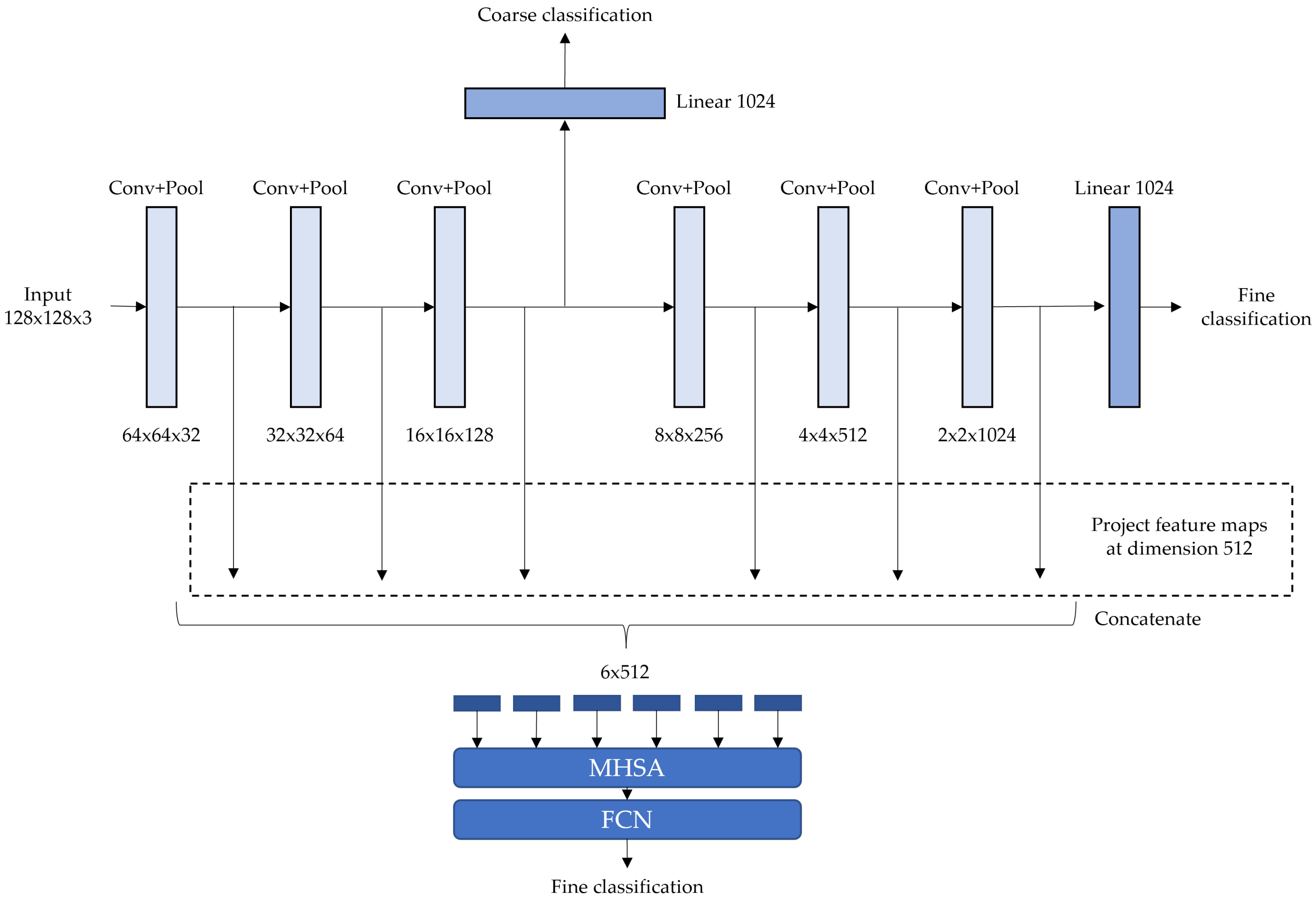
Metric which tells how far you are (in terms of hierarchy) from the actual leaf

Accuracy at leaf node it is not enough

$$h\_acc(n1, n2) = \frac{d_{CA(n_1,n_2)}}{h}$$

A metric that can help to understand the accuracy at different levels is the ratio between the depth of the common ancestor and the height of the tree

4.3 Limitations and future works



Extract intermediate representations and exploit the parallelism between subsequent layers in CNN and hierarchy by extracting tokens and use them to train a self-attention module

This idea was inspired from: Koo, J., Klabjan, D., and Utke, J. (2018). [Combined convolutional and recurrent neural networks for hierarchical classification of images.](#)

Thank you for your attention

Contacts
leonardo.tanzi@polito.it
Website
Medium