

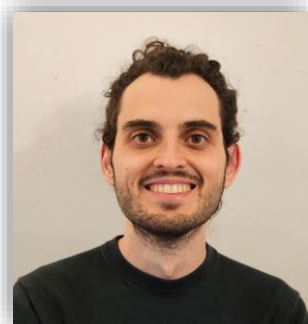


Deep learning for medical imaging school

Hands-on session

Foundation Models

Julio
Silva-Rodríguez

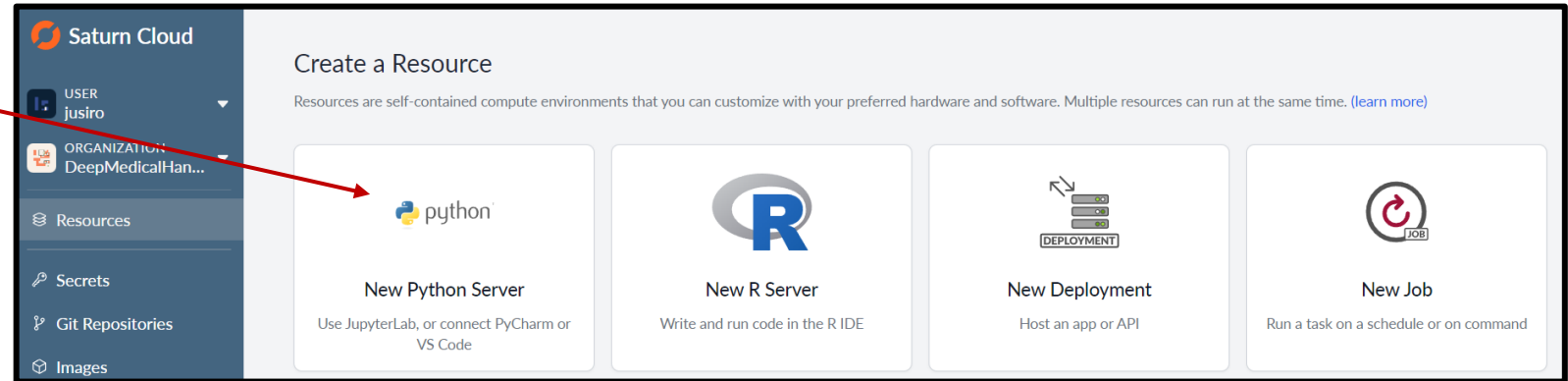


Jose Dolz

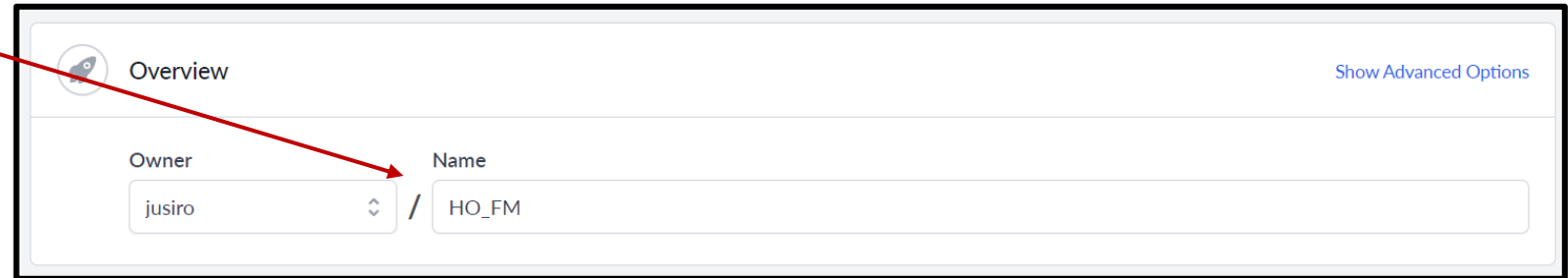


Setup your
SaturnCloud
enviroment

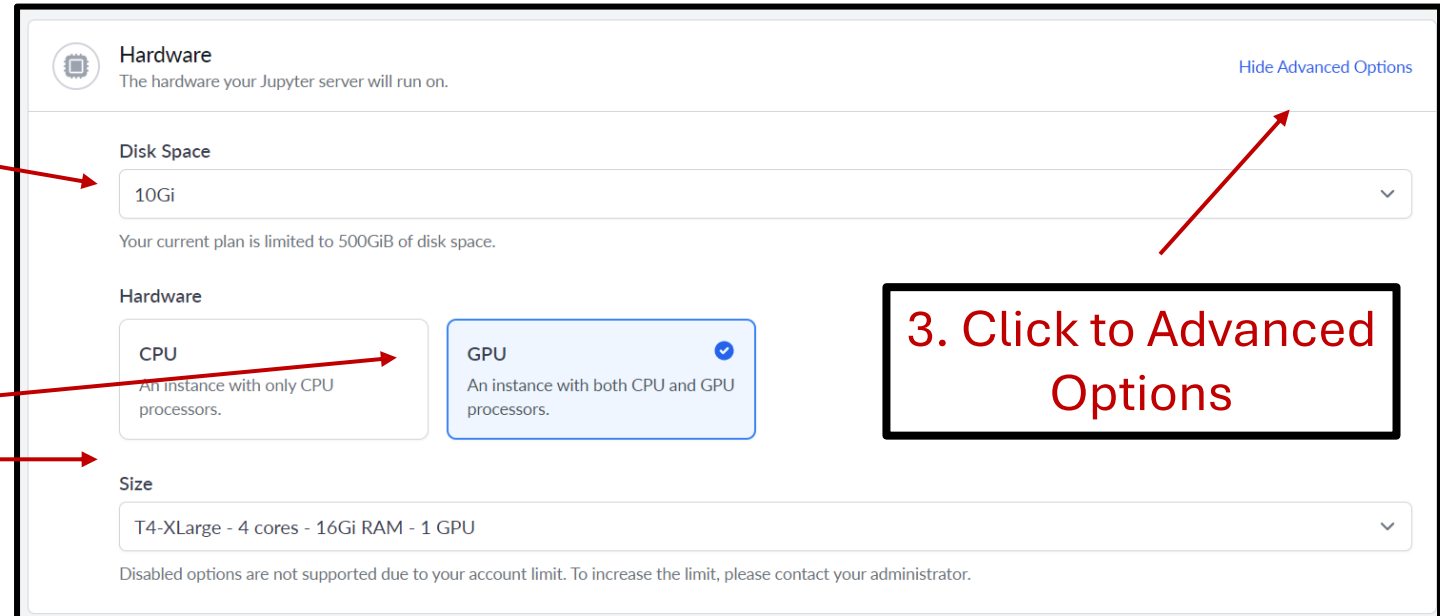
1. Click to “New Python Server”



2. Add Name



4. 10Gb of Disk Space



5. Select GPU Hardware

3. Click to Advanced Options

6. Add libraries
using pip

Environment
The software your Jupyter server will use. This includes libraries, packages, environment variables, and other attributes. [Show Advanced Options](#)

Image saturncloud/saturn-python-pytorch **Version** 2023.09.01

Extra Packages
Extra packages are installed every time the resource starts up - right before the start script. Use spaces to separate packages. If you find yourself adding the same packages to lots of resources, you may want to permanently add packages to a custom image instead. (?)

☐ Conda ☒ Pip ☐ Apt

transformers
torchvision
openpyxl

☐ This is a requirements.txt
If enabled, this should be valid requirements file describing all packages to be installed by `pip`, such as the content produced by `pip freeze`. See the [pip documentation for details](#).

The packages together will run the following script:
`pip install transformers torchvision openpyxl`

7. Select duration
of project

Additional features
Optional settings for your Jupyter server. [Show Advanced Options](#)

☐ Allow SSH Connections
Use SSH to directly connect to the server, including through VSCode, PyCharm, and other tools (?)

Shutoff After
3 days

Create **Cancel**

8. Create

Jupyter Server

stopped

Start



T4-XLarge - 4 cores - 16Gi RAM - 1 GPU - 10Gi Disk

Auto Shutoff: 3 days

Spot Instance: No

URLs: (server not running)

SSH URL: (not enabled) (?)

9. Start project

Jupyter Lab



Jupyter Server

running

Stop



T4-XLarge - 4 cores - 16Gi RAM - 1 GPU - 10Gi Disk

Auto Shutoff: 3 days

Spot Instance: No

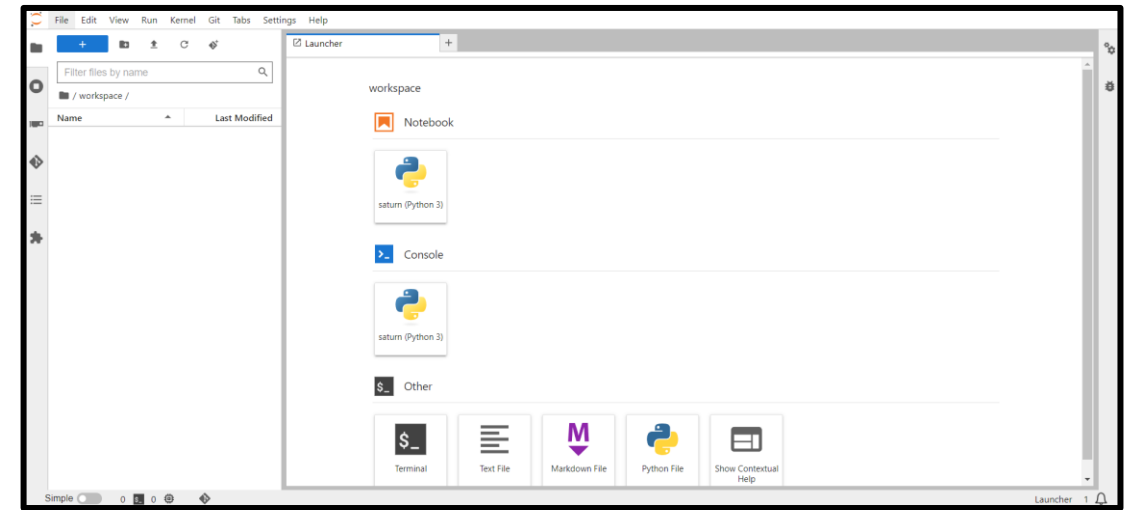
URLs:

<https://w-jusir-hofm-5edff54e9b8049f3a4c3d6d922d2d37d.community.saturnenterprise.io>

SSH URL: (not enabled) (?)

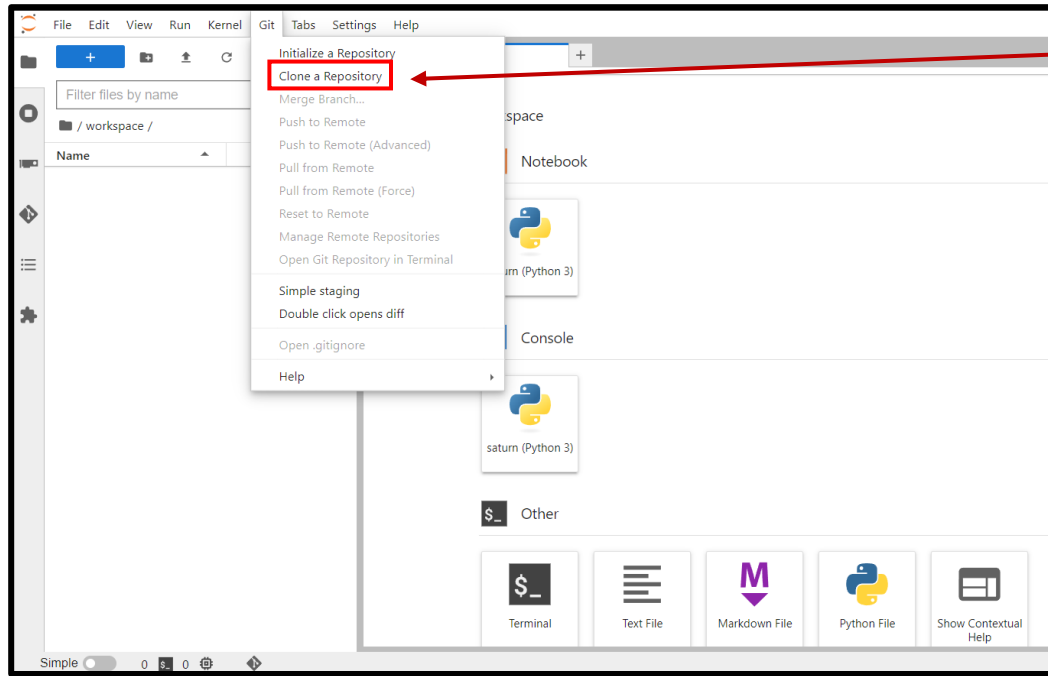
10. Go to Lab

Jupyter Lab

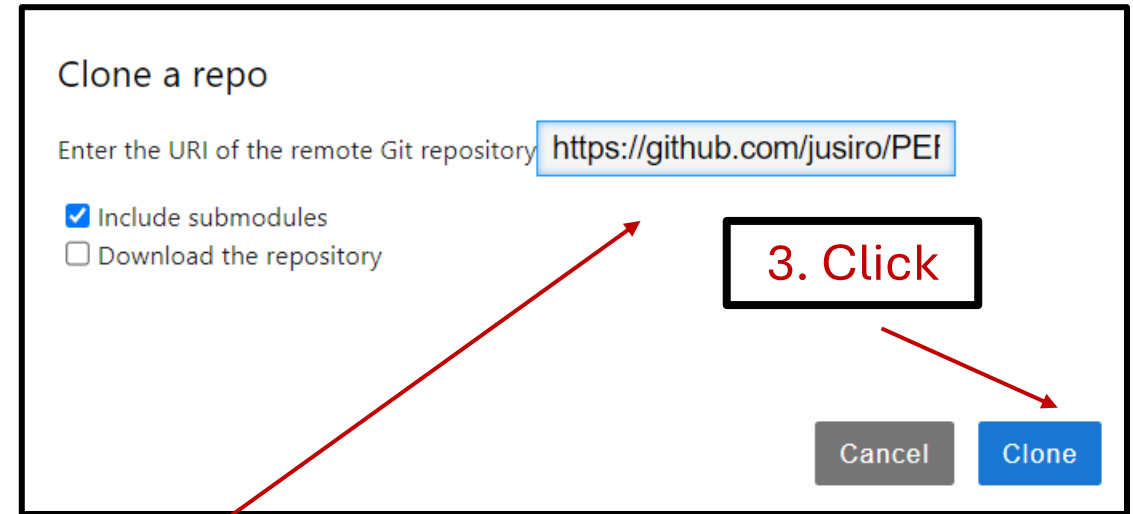


Ready to start!

**Code
Download**

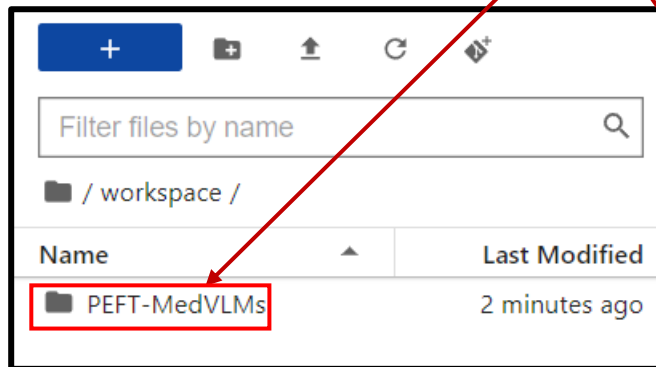


1. Git - Clone



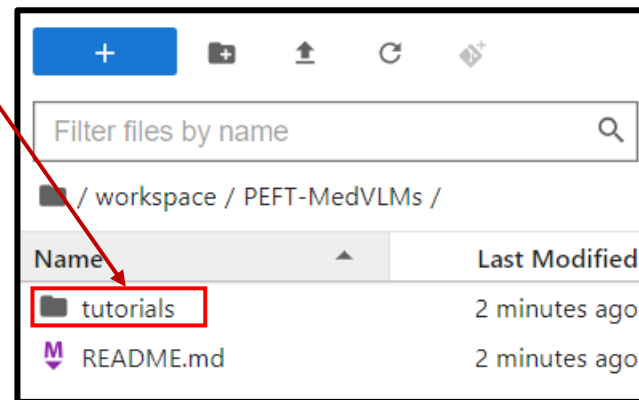
3. Click

4. Click

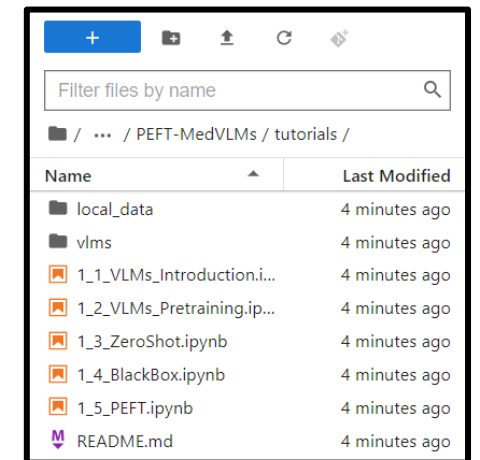


2. Add URL

<https://github.com/jusiro/PEFT-MedVLMs>



Ready to start!



Datasets

1. Go to the link

⌚ 5 minutes

<https://data.mendeley.com/datasets/9xxm58dvs3/2>

Mendeley Data

SICAPv2 - Prostate Whole Slide Images with Gleason Grades Annotations

Published: 22 October 2020 | Version 2 | DOI: 10.17632/9xxm58dvs3.2
Contributor: Julio Silva-Rodríguez

Description

A database containing prostate histology whole slide images with both annotations of global Gleason scores and path-level Gleason grades.

Data associated with the paper:

Silva-Rodríguez, J., Colomer, A., Sales, M. A., Molina, R., & Naranjo, V. (2020). Going deeper through the Gleason scoring scale : An automatic end-to-end system for histology prostate grading and cribriform pattern detection. Computer Methods and Programs in Biomedicine, 195. <https://doi.org/10.1016/j.cmpb.2020.105637>

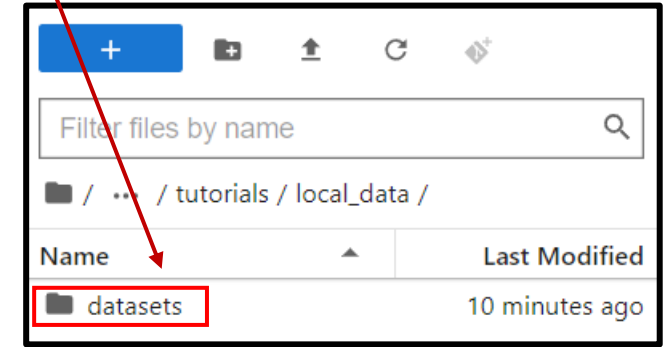
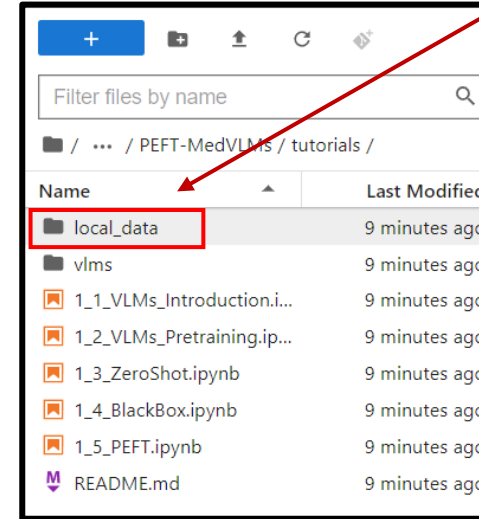
[Download All 2.01 GB](#) ⓘ

Files

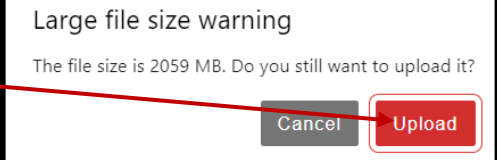
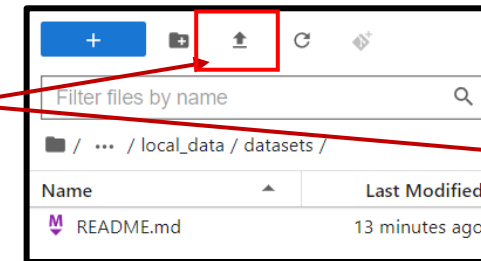
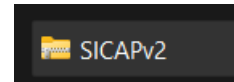
SICAPv2.zip 2.01 GB

2. Download

3. Go to dataset folder



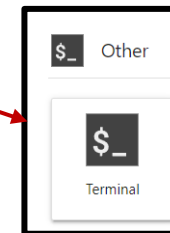
5. Upload



Uploading...

⌚ 5 minutes

6. Unzip



```
~/workspace/PEFT-MedVLMs/tutorials/local_data/datasets$ unzip SICAPv2.zip
```

HANDS-ON

3. Auxiliary functions

Right click in the
README + “Show...”

```
# DLMI24_HO_FM

### Foundation models for medical imaging: Hands on!

This project includes required codebase for an introductory hands-on session on vision-language foundation models for medical image analysis. Also, we include the particular details for Datasets preparation in this file (see below), and instructions for setting the Saturn Cloud resources (see below).

Also, an introductory video is available.

## Overview

Foundation models are large-scale n heterogeneous sources, which can be ef of downstream tasks.

In particular, this tutorial is focus medical foundation models. This is, trained networks. For example, we will work over models pre-trained uniquely on histology data. Such specialized vision models provide specially efficient transferability to new tasks/datasets. For example, they can be adapted using only few labeled examples (so-called shots), and requiring minimal parameter tuning.

In this tutorial, due to the large resources required for pre-trained, we will focus on the adaptation stage. Nevertheless, we will introduce toy examples to introduce the student to typical losses and pipelines employed. Regarding the adaptation, you will learn how to achieve state-of-the-art performance on your image classification dataset employing minimum data and computing resources.
```

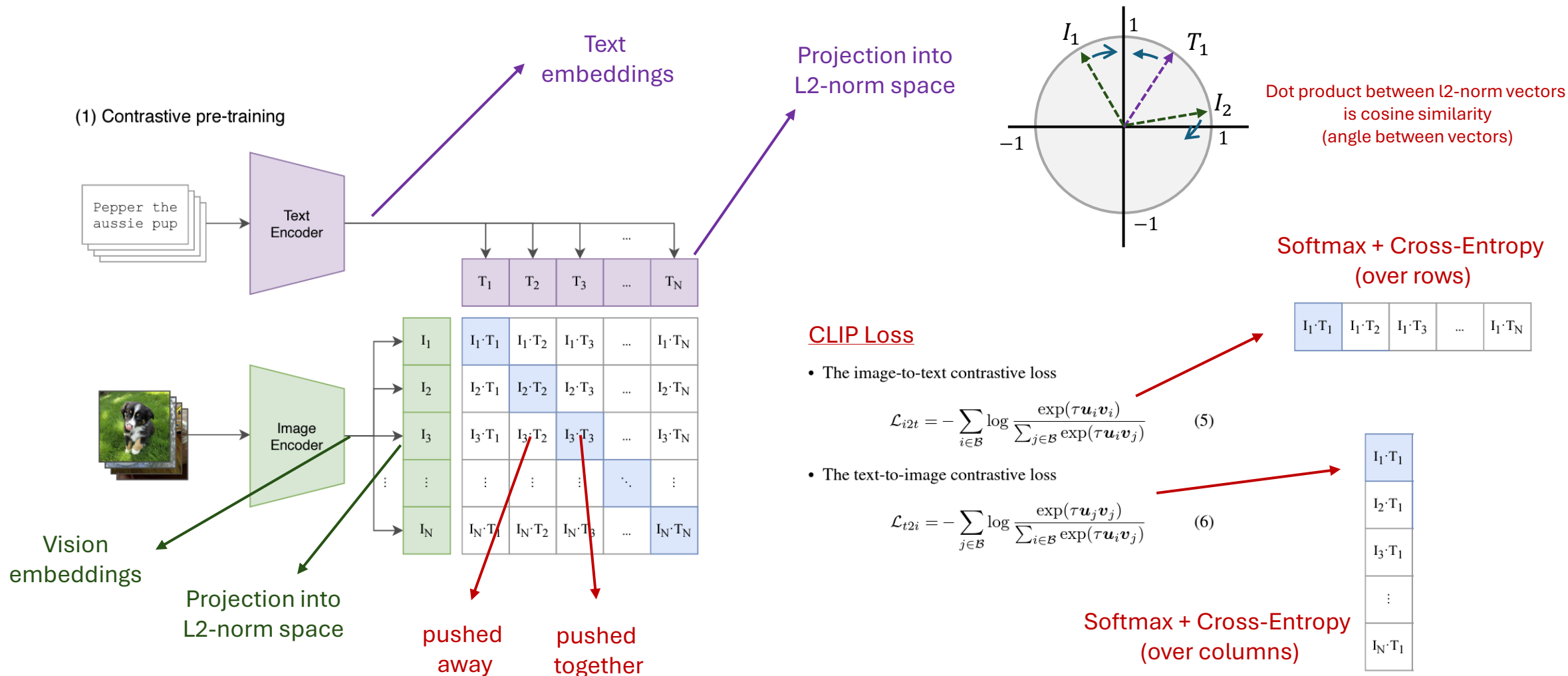
1. README

Filter files by name	
/ ... / PEFT-MedVLMs / tutorials /	
Name	Last Modified
local_data	24 minutes ago
vlms	24 minutes ago
1_1_VLMs_Introduction.i...	24 minutes ago
1_2_VLMs_Pretraining.ip...	24 minutes ago
1_3_ZeroShot.ipynb	24 minutes ago
1_4_BlackBox.ipynb	24 minutes ago
1_5_PEFT.ipynb	24 minutes ago
README.md	24 minutes ago

2. Tutorials

Introduction to Vision-Language Models and PEFT

CONTRASTIVE VISION-LANGUAE PRE-TRAINING (CLIP)



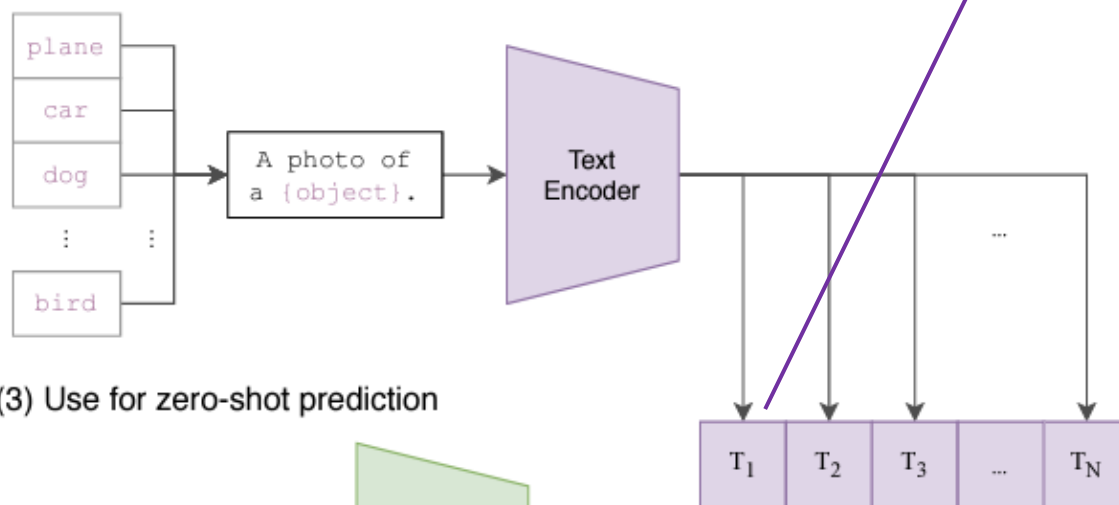
ZERO-SHOT PREDICTIONS

Note: Text embeddings for target categories are also called **class prototypes**, or **zero-shot prototypes**.

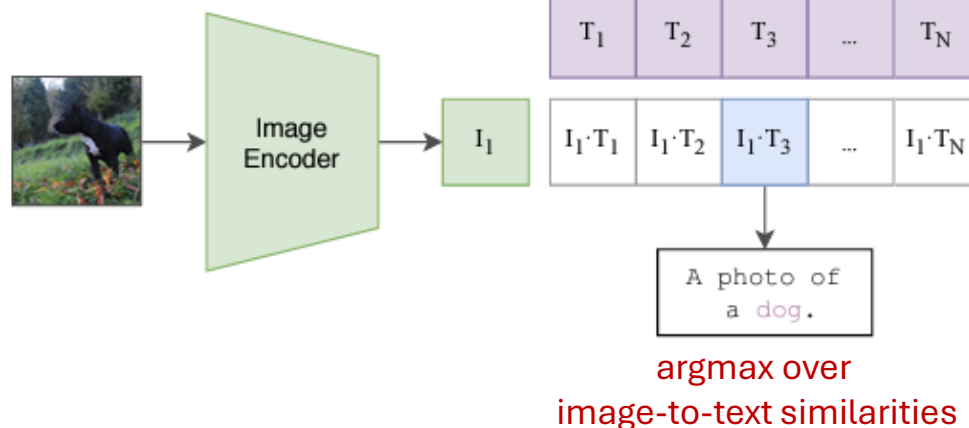
They do not require image samples to compute this reference embedding, but only text, and that is why they are called “zero-shot”. Images with similar representations will be more likely to belong to this category.

Note that they are equivalent to a Linear output layer!
 W (classes, features).

(2) Create dataset classifier from label text

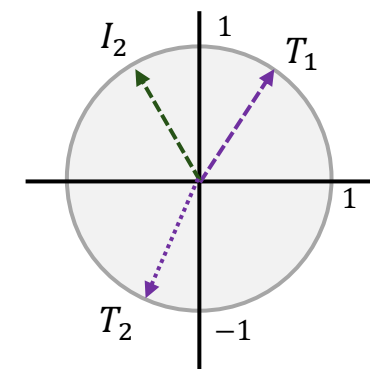


(3) Use for zero-shot prediction



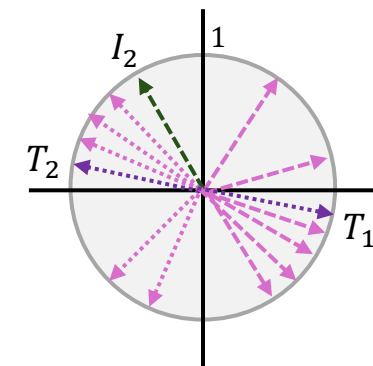
A. Single Prompt

“a photo of [CLS]”



B. Prompt Ensemble

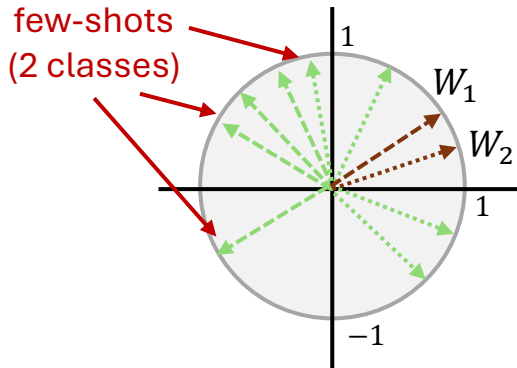
“a photo of [CLS]”
 “a sketch of [CLS]”
 “small animal with black hair”
 “has four legs”
 “two animals playing”
 ...



BLACK-BOX ADAPTERS

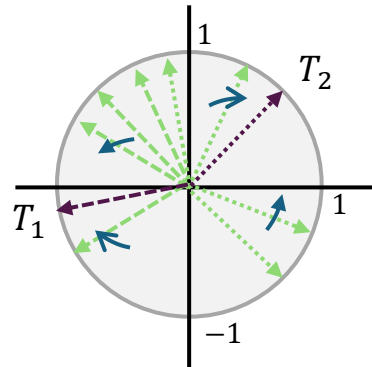
- Work over pre-computed vision features - They are backbone-agnostic.
- May profit zero-shot prototypes for the target tasks.
- They are backbone-agnostic.
- Very efficient, do not even require GPU.
- Potentially, they do not require access to pre-trained weights (similar to ChatGPT).

A. Linear Probing



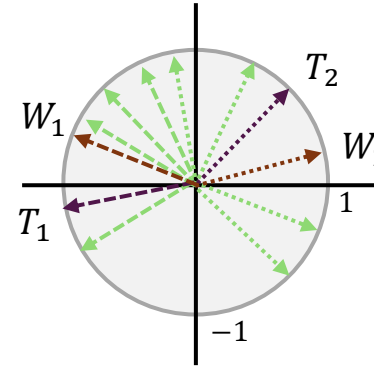
Train W_1, W_2 to
minimize cross-entropy
We train class prototypes

B. CLIP-Adapter



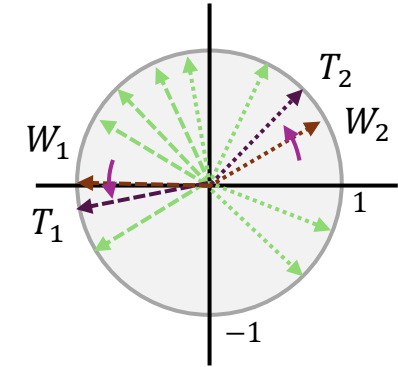
Train an mlp to modify vision features
keep class prototypes as zero-shot

C. ZS-Linear Probe



Train W_1, W_2 to
minimize cross-entropy
Initialize then to zero-shot

D. Class-Adaptive
Linear Probe (CLAP)



Train W_1, W_2 to
minimize cross-entropy
Constraint them to remain close to zero-shot

PARAMETER-EFFICIENT FINE-TUNING

- Train a subset of parameters to modify deep features.
- Two types: selective, and additive.
- More efficient than full-finetuning, and more flexible than black-box Adapters.
- If carefully designed, they can avoid catastrophic forgetting.

A. Affine-Layer Norm

$$y = \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}} * \gamma + \beta$$

We only tune these
from the whole encoder

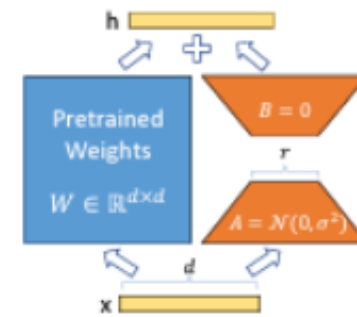
B. Bias Tuning

$$y = xA^T + b$$

$$y = \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}} * \gamma + \beta$$

C. Low-Rank Adapters

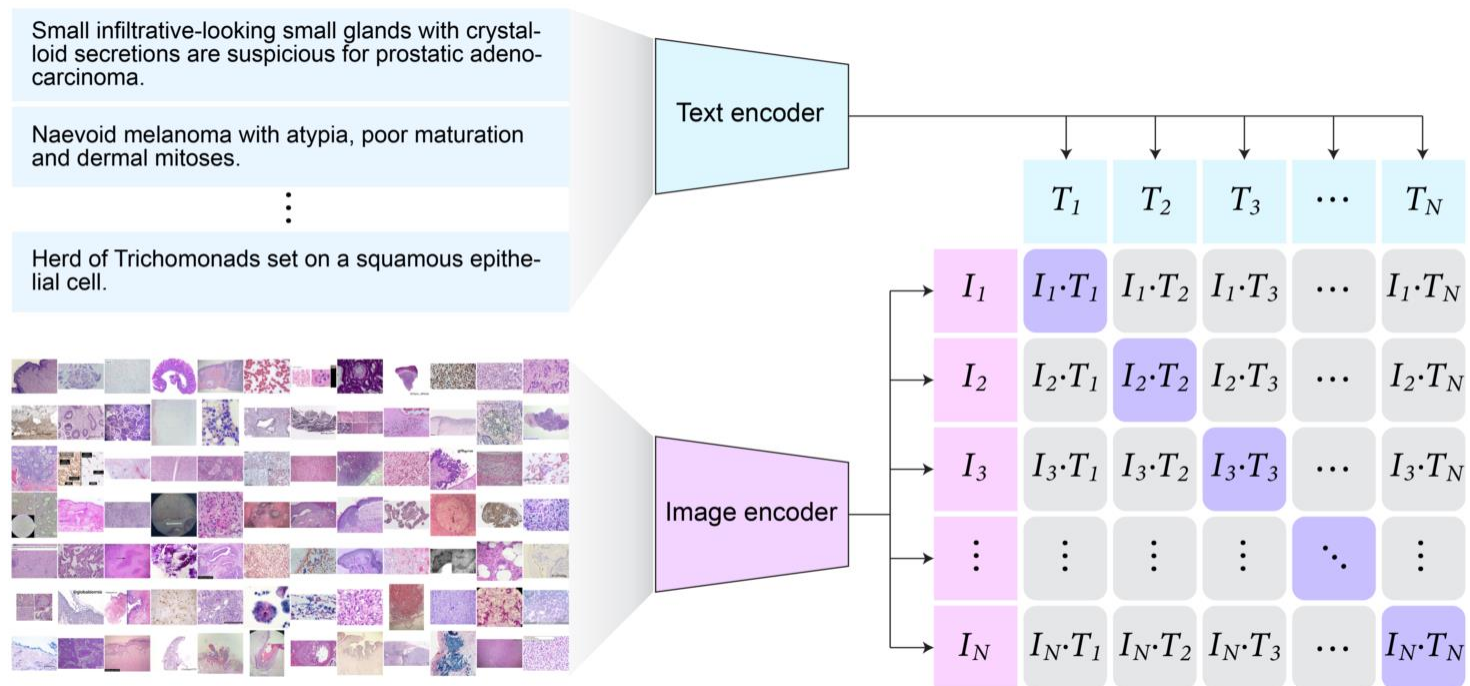
Usually applied in ViTs to k,q,v
layers of MultiHeadAttention



Add and tune a
residual connection
with low-rank weights.
Important!
Note B=0 when t=0

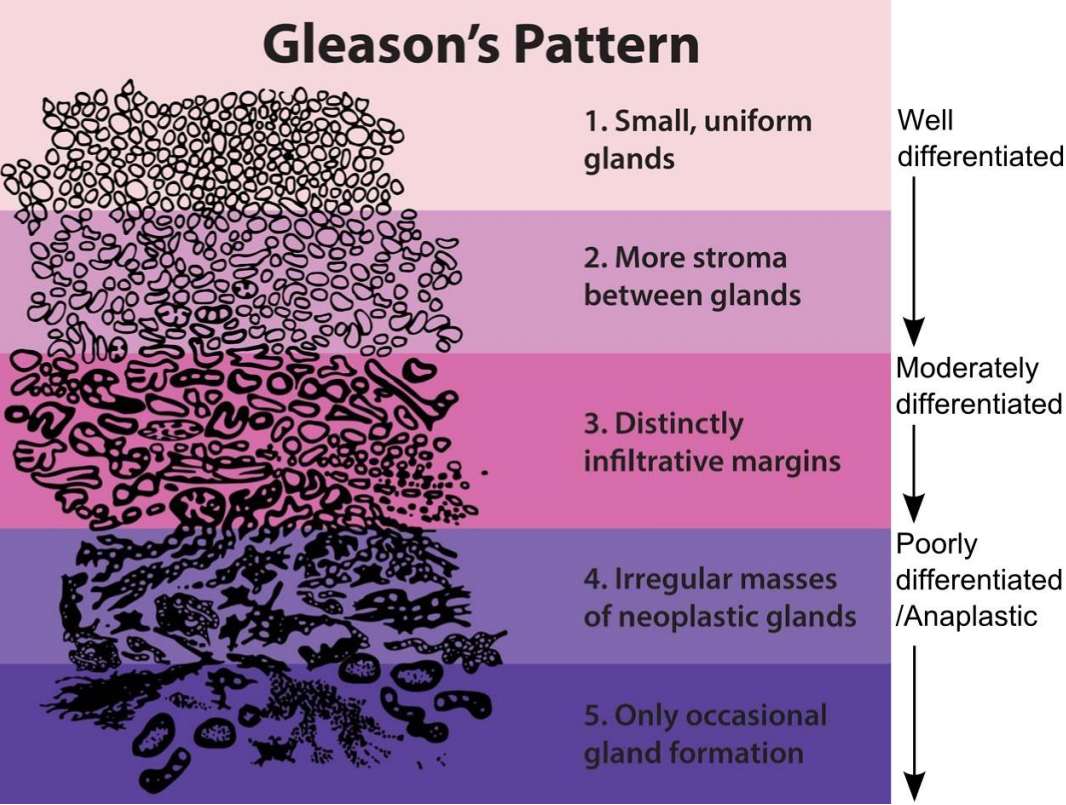
MEDICAL VLMs - PLIP

Twitter data!

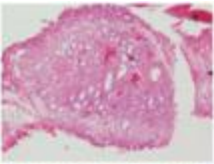
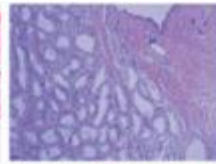
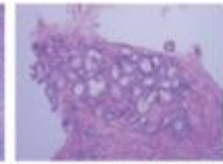
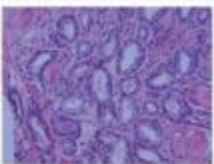
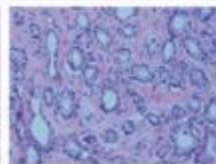
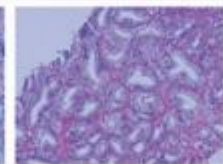

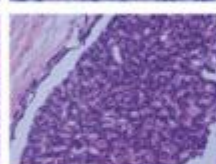
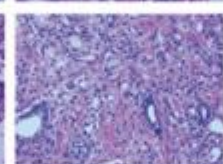
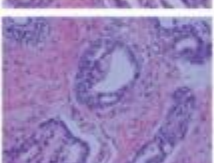

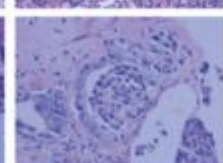
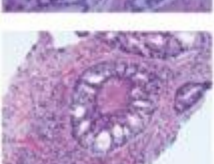
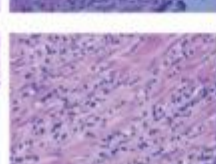

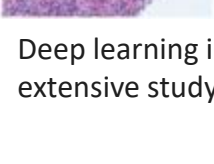
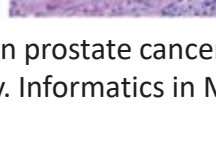
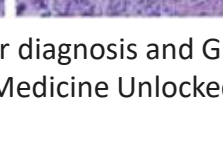


Specialized on histology images

APPLICATION: GLEASON GRADING



Gleason grading system - Wikipedia

			Gleason patterns 1-3 distinct, discrete, individual glands	Gleason score ≤ 6	Grade group I
					
			Gleason pattern 4 fused, cribriform, or poorly-formed glands, or glomerulation	Gleason score 3+4=7	Grade group II
				Gleason score 4+3=7	Grade group III
				Gleason score 4+4=8 3+5=8 5+3=8	Grade group IV
			Gleason pattern 5 comedo necrosis, cords, sheets, solid nests, single cells	Gleason score 4+5=9 5+4=9 5+5=10	Grade group V

Deep learning in prostate cancer diagnosis and Gleason grading in histopathology images: An extensive study. Informatics in Medicine Unlocked.

Thanks!