The Challenge - Pathology Diagnosis

In the competition, participants are invited to use **MindSpore** as the AI training and inference framework, for developing trustworthy AI pathology diagnosis models that ensures ***privacy***, ***explainable*** and ***high accuracy***.

Quota	30 Teams
Team size	1 - 3 Members
Events	Workshops, Pitching, Award Ceremony
Competition Rounds	Qualification and Final Rounds

Organizers:

HUAWEI HKSTP 爱病理

Competition Timeline

1

Training Workshop

17/09

2

Model Training

MODELARTS + MINDSPORE

3

Model Evaluation

SUBMIT TO PORTAL

4

Final Submission

07/10

5

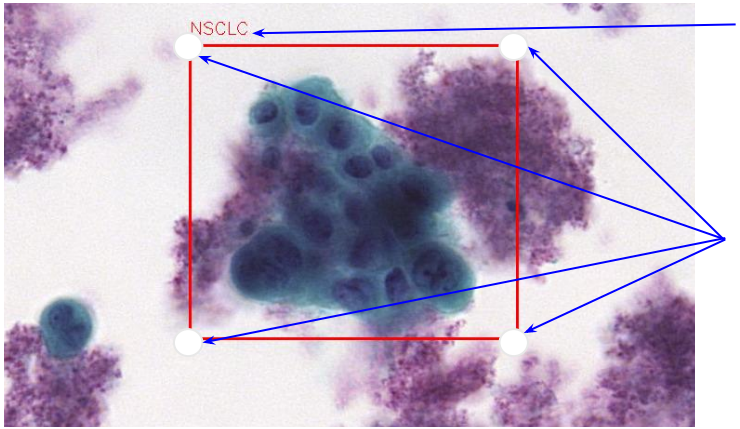
Final Pitching

15/10

Qualification Round - Problem Statement

Training a MindSpore AI model to **identify locations and classifications** of cancer cells in pathological images. The AI Models will **assist pathologists** in the diagnosis of **peripheral pulmonary diseases**.

This is a form of **multi-label object detection**.

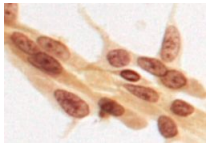
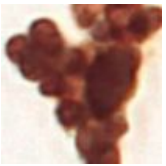
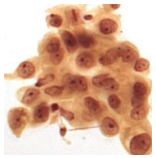


Classification:

- Class of diagnosed cell

Location:

- Bounding Boxes

Class	English Name	Example
SCC	*Squamous Cell Carcinoma*	
AC	*Adenocarcinoma*	
SCLC	*Small Cell Lung Cancer*	
NSCLC	*Non-Small Cell Lung Cancer*	

Qualification Round - Evaluation

$$\text{Classification Score} = \frac{1}{|M|}\sum_{i \in M} \text{FROC}_i$$

$$\text{AUC} = \frac{1}{M \times N} \sum_{i \in \text{positive class}} rank_i - \frac{M(1+M)}{2}$$

Accuracy

- The trained MindSpore AI models should ***accurately locate*** and ***classify*** cancer cells.

Explainability (Bonus)

- The trained MindSpore AI models should provides ***pixel level feature attribution*** as an explanation for the task!

$$\text{Model Score} = \text{Accuracy Score} * 0.8 + \text{Explainable Score} * 0.2$$

The *top 6 highest model score* teams are invited to enter the **FINAL ROUND**

Final Round & Ceremony

Pitching

Date **15 October 2021**

Venue **Charles K Kao Auditorium HKSTP**

Time 2 - 6pm

Agenda

- Welcoming Speech
- Solution Pitching
- On-site Evaluation
- Result Announcement

Charles K Kao Auditorium HKSTP

Award Ceremony

Event **Huawei Cloud Summit 2021**

Date **26 October 2021**

Venue TBD

Time Afternoon

Prizes!

Winning Teams Award	Prizes per Team
Model Score Winner	**HKD 70,000**
Model Score Runner-up	**HKD 50,000**
Pitching Score Winner	**HKD 50,000**
Pitching Score Runner-up	**HKD 35,000**
Special Price for Explainability	**HKD 2,000 (max: 30 winners)**

Register Now!

OFFICIAL WEBSITE

Sign Up!

DISCORD

Get Connected

GITHUB

Get Resources

RULE BOOK & GUIDELINES

Get Informed

MindSpore

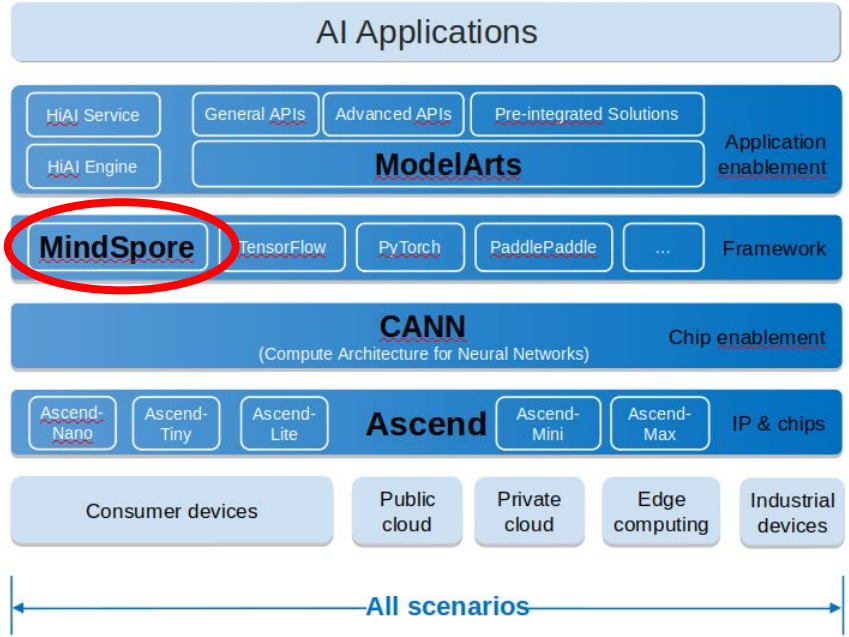
Introduction and Examples

What is MindSpore?

MindSpore

An **Open AI-framework** that supports the multi-processor architectures developed by Huawei.

It provides a unified APIs and end-to-end AI capabilities for AI model development, execution and deployment in all scenarios, including cloud, edge and devices.



DL Python Modules

	Dataset Preparation and Preprocessing	Network Construction and Training	Explainable XAI
MindSpore Modules 	`mindspore.dataset:` Complete solution equipped with vision and text operators.	`mindspore.nn:` network constructions `mindspore.ops:` common operators in neural networks `mindspore.model:` defining model, optimizers and loss function `mindspore.train:` provides common training utilities	**`mindspore .explainer:`** Provides methods to evaluate generate saliency maps/other explainable figures from inputs.
Tensorflow Pytorch Modules 	`tf.data` `torch.utils.data`	`tf.keras.Model` `torch.nn`	-

MindSpore Model Design and Training

```
import mindspore as ms
```

Dataset

```
dataset = ms.dataset.MnistDataset()

dataset.batch # batching data
dataset.map # preprocessing data
```

Network

```
class Net(ms.nn.Cell):
    def __init__(self):
        super(Net, self).__init__()
        self.flatten = ms.nn.Flatten()
        self.dense = ms.nn.Dense(1024, 10)

    def construct(self, x):
        x = self.flatten(x)
        x = self.dense(x)
        return x
net = Net()
```

Model

```
loss = ms.nn.SoftmaxCrossEntropyWithLogits()
optimizers = ms.nn.Adam(
  net.trainable_params(),
  learning_rate=0.01
)

model = ms.Model(
    net,
    loss,
    optimizers,
    metrics={"Accuracy": ms.nn.Accuracy()}
)

model.train(epoch=10, dataset)
```

AI APPS

Agenda

Part I - Beginner Tutorial

- MindSpore Dataset
- MindSpore Neural Network Design
- MindSpore Model Training

Part II - Intermediate Tutorial

- Training a YoloV3 model
- Using ModelArts, OBS and Moxing Framework
- Submission to Portal