

Reusing existing trained AI models in WIPP

Antoine Meyer, Mylene Simon, Timothee Kheyrkhah, Peter Bajcsy

1 Background

- AI model cards
- Problem statement
- Related work
- No common APIs
- Our approach
- Plugin concept

2 WIPP plugins for AI models reuse

- Public AI repositories
- Unlock these repositories
- Evaluate results

3 Benchmarks

- 2-steps workflow
- Data 'cell boundary'
- Accuracy for 'cell boundary'
- Data 'nuclei segmentation'
- Accuracy for 'nuclei segmentation'

4 Conclusion

- New WIPP plugins
- Fine-tuning plugin

Goal: Automatic documentation for AI models trained inside WIPP

Work: Make AI model card proposal and development for integration

```
1      public class AiModelCard {
2          private String          version;
3          private String          name;
4          private Date            creationDate;
5          private String          framework;
6          private Map<String, String> trainingData;
7          private Map<String, String> trainingParameters;
8          [8 additional fields]
9      }
```

Feedback: Not relevant, AI users test directly, they don't read the documentation

Problem

- ▶ Lower the barriers for reusing trained AI models available in AI models repositories

Challenges

- ▶ Scattered models in a lot of different repositories and stored in many formats
- ▶ It is not trivial to setup and execute AI models for non-technical researchers
- ▶ Re-training models from scratch take a lot of time and money

Motivations

- ▶ Accelerate applications of AI models to scientific problems
- ▶ Save computation time by reusing pre-trained models from diverse repositories

Already existing APIs or tools to download, inference (and train) AI models

- ▶ Transformers (Hugging Face API)*
- ▶ Cellpose API**
- ▶ BioImage.IO Core (Python libraries)
- ▶ SAM2 repository***

*<https://aclanthology.org/2020.emnlp-demos.6/>

**<https://www.biorxiv.org/content/10.1101/2020.02.02.931238v1>

***<https://arxiv.org/abs/2408.00714>

Background

No common APIs

No common APIs/CLIs/no standard for 'download API'

```
1      # BioImage.IO
2      model = load_description(args.model)
3
4      # Hugging Face
5      pipe = pipeline(task="mask-generation", model=args.model,
6                      points_per_batch=32, device=device)
7
8      # SAM2
9      mask_generator = SAM2AutomaticMaskGenerator.from_pretrained(
10         args.model, device=device)
```

```
1      # Cellpose
2      python -m cellpose --pretrained_model cyto3
```

Same problem with 'inference API', 'train API', etc

Our approach

- ▶ Using existing API of AI repositories
- ▶ Leverage WIPP
- ▶ Use AI model card

Contributions

- ▶ Analyze API of AI repositories
- ▶ Implement them for containerized plugin that run in WIPP without learning the API
- ▶ Trained/Retrained AI model will comes with proper AI model card

Web Image Processing Pipelines

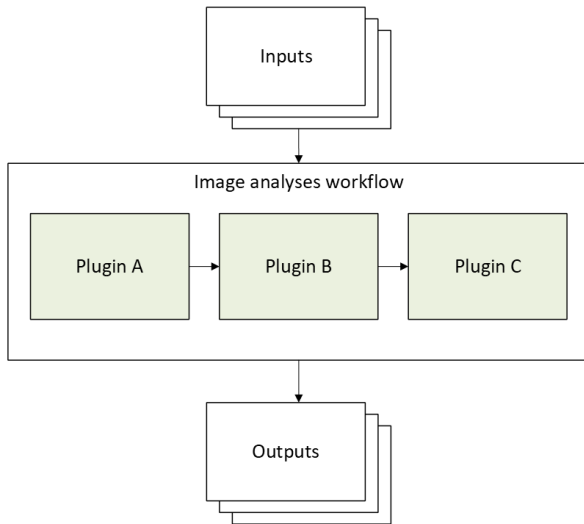
- ▶ Purpose: Measurements based on terabyte-sized images
- ▶ Goal: Lower the barrier to execute image analyses

WIPP workflow

- ▶ Sequence of plugins

WIPP plugin

- ▶ Piece of code taking inputs/outputs and executing code



- 2 WIPP plugins for AI models reuse
 - Public AI repositories
 - Unlock these repositories
 - Evaluate results

Many public AI models on lots of public AI repositories

AI repositories	Image classification models	Segmentation + MG* models
Hugging Face	15,593	1,160 + 176
BioImage.IO	1	4 + 32
Cellpose	×	21
SAM2	×	8
PyTorch Hub	20	5

Table: Number of models per repository

*MG: Mask Generation

WIPP plugins for AI models reuse

Unlock these repositories

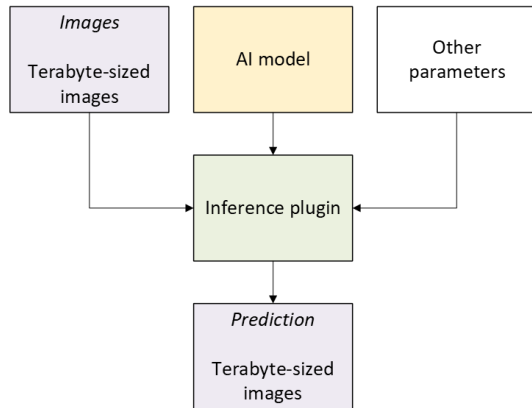
Goal: Access external AI models in WIPP

Work:

Create one plugin per public repository

Question:

How to assess the relevance of results?



WIPP plugins for AI models reuse

Evaluate results

Goals

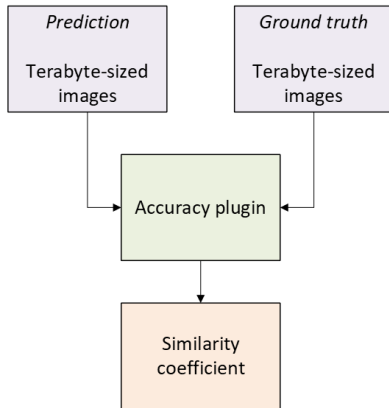
- ▶ Measure accuracy of external AI results
- ▶ Select the most accurate/fastest one

Work:

Plugin to compute the Dice-Sørensen coefficient*

$$Dice = \frac{2 * TP}{2 * TP + FP + FN}$$

*Statistic used to gauge the similarity of two images



3 Benchmarks

- 2-steps workflow
- Data 'cell boundary'
- Accuracy for 'cell boundary'
- Data 'nuclei segmentation'
- Accuracy for 'nuclei segmentation'

Benchmarks

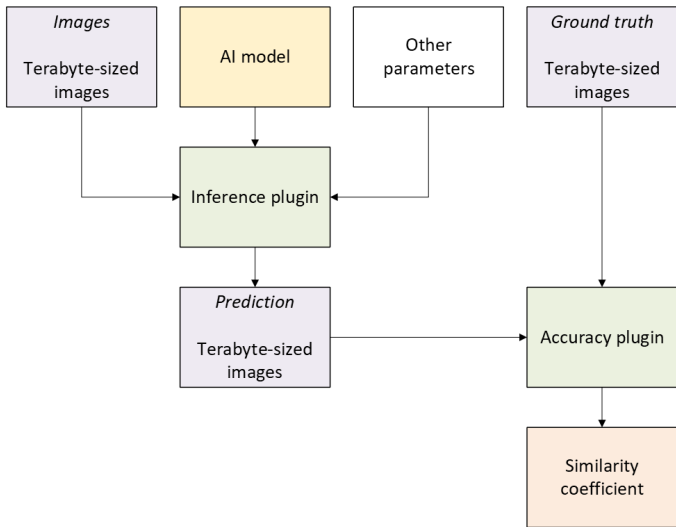
2-steps workflow

WIPP 2-steps workflow

- 1 Inference the model
- 2 Compute the accuracy

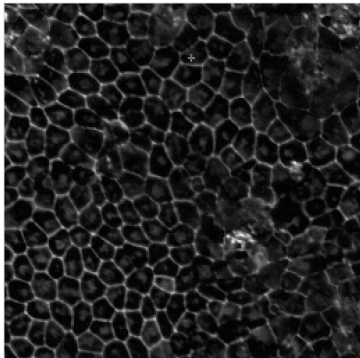
Hardware

- ▶ AMD Ryzen 9 3950X
16-Core Processor
- ▶ NVIDIA GeForce RTX
3090
- ▶ 64G RAM



Name: Retinal Pigment Epithelium

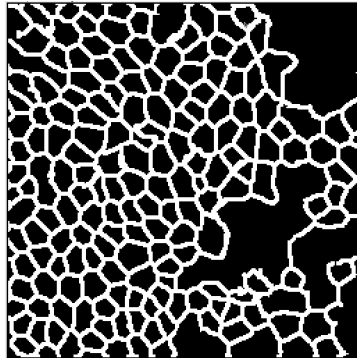
Number: 1032 images (822/210)



Image

Type: Cell microscopy

Size: 256x256



Mask

Source: <https://doi.org/doi:10.18434/T4/1503229>

Task: Segments cell edges

Repository	Model	Accuracy	Time per image
WIPP	unet-cnn*	95.11% \pm 0.78%	8.56s + 2.35s
Biolmage.IO	10.5281/zenodo.5869899	89.30% \pm 0.84%	0.31s
Hugging Face	facebook/sam-vit-huge	86.01% \pm 2.50%	4.16s
SAM2	facebook/sam2.1-hiera-large	80.18% \pm 5.02%	2.0s
Cellpose	cyto3	78.51% \pm 2.35%	1.08s

Table: Models accuracy after inference on data 'cell boundary'

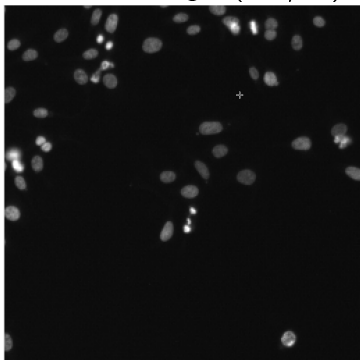
*Trained with data, then inferenced (in WIPP)

Benchmarks

Data 'nuclei segmentation'

Name: Data Science Bowl 2018

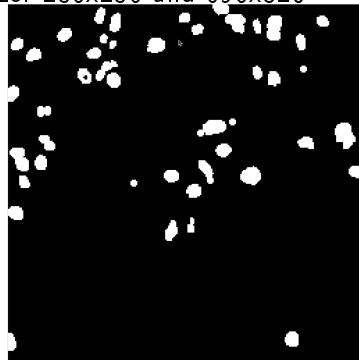
Number: 497 images (447/50)



Image

Type: Cell microscopy

Size: 256x256 and 696x520



Mask

Source: <https://bbbc.broadinstitute.org/BBBC038/>

Benchmarks

Accuracy for 'nuclei segmentation'

Task: Segments nuclei of cells

Repository	Model	Accuracy
Biolmage.IO	10.5281/zenodo.5764892	93.73% \pm 3.98%
WIPP	Stardist 2D paper DSB 2018*	90.67% \pm 4.42%
Cellpose	cyto3	82.31% \pm 17.25%
Cellpose	nuclei	81.00% \pm 21.00%
SAM2	facebook/sam2.1-hiera-small	48.18% \pm 32.41%
Biolmage.IO	10.5281/zenodo.5869899	29.47% \pm 8.32%
Hugging Face	facebook/sam-vit-huge	21.63% \pm 15.37%

Table: Models accuracy after inference on data 'nuclei segmentation'

*Trained with data, then inferenced (in WIPP)

4 Conclusion

- New WIPP plugins
- Fine-tuning plugin

Inference plugins

- ▶ wipp/wipp-huggingface-maskgeneration-inference
- ▶ wipp/wipp-bioimage-io-inference-plugin
- ▶ wipp/wipp-sam2-inference-plugin
- ▶ wipp/wipp-cellpose-inference-plugin

Accuracy plugin

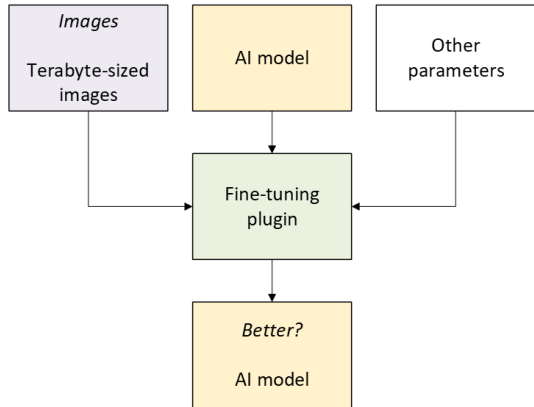
- ▶ wipp/wipp-dice-segmentationaccuracy

Goal

- ▶ Improve already existing AI models

Work

- ▶ Identify if models can be retrain/fine-tune
- ▶ Plugin to fine-tune an AI model



Thanks !

WIPP enhancements

CSV viewer

Issue: Impossible to view CSV content without downloading

Solution: Content directly in the user interface

CSV Collection files

Total size: 11.0 kB

Name ↑↓	Size ↑↓	Actions
dice-coef-global.csv	150.0 B	Preview Download Delete
dice-coef-per-image.csv	10.9 kB	Preview Download Delete

« < 1 > » 10 ▾

CSV file content

Total number of lines: 163 | Total number of columns: 4

IMAGE-NAME	DICE-COEFFICIENT	TRUE POSITIVE	TRUE NEGATIVE
P1-W3-TOM_E02_F006_DNA_RPE0000.ome.tif	0.0917542213402659	8947	1191610
P1-W3-TOM_E02_F002_DNA_RPE0012.ome.tif	0.8045320415876903	539699	575736
P1-W3-TOM_E02_F005_DNA_RPE0025.ome.tif	0.12852281362540063	11670	1207752
P1-W3-TOM_E02_F001_DNA_RPE0021.ome.tif	0.3706311203140503	21479	1283258
P1-W3-TOM_E02_F002_DNA_RPE0010.ome.tif	0.8228968069924416	552690	587095
P1-W3-TOM_E02_F004_DNA_RPE0005.ome.tif	0.6573651117015008	281701	802324
P1-W3-TOM_E02_F005_DNA_RPE0017.ome.tif	0.7077754109348144	266642	890861
P1-W3-TOM_E02_F004_DNA_RPE0011.ome.tif	0.7826014546687202	530999	551673

« < 1 2 3 4 5 > »

WIPP enhancements

Multi-dialogs workflow

Issue: Difficult to navigate between different plugins in a workflow

Solution: Allows user to open one modal per plugin

The screenshot displays the WIPP interface with a workflow graph and two modal dialogs. The workflow graph shows three nodes: 'dfd-test' (blue), 'dfd-test-2' (purple), and 'dfd-dfsdfsdfsdfsdfs' (red). The 'dfd-test' modal is open, showing details for a job named 'dfd-test' created on Feb 6, 2025, by 'wipp-user'. The 'dfd-dfsdfsdfsdfsdfs' modal is also open, showing details for a job named 'dfd-dfsdfsdfsdfsdfs' created on Feb 6, 2025, by 'wipp-user'. Both modals show the workflow name 'dfd' and the plugin 'WIPP Resnet50 Regression Inference Plugin:2.0.0'. The 'dfd-test' modal also shows the computed version '3.2.0-SNAPSHOT'. The 'dfd-dfsdfsdfsdfsdfs' modal shows the computed version '3.2.0-SNAPSHOT'. The 'dfd-test' modal also shows inputs: 'image_folder' (test) and 'saved_model_filepath' (66e987f5377e7b28636ce973). The 'dfd-dfsdfsdfsdfsdfs' modal shows inputs: 'image_folder' (test) and 'saved_model_filepath' (66e84c0b84d9b674549d46dd). The 'dfd-test' modal also shows outputs: 'csvCollection'.

dfd [Share] [Refresh]

— Details

Name: dfd
Creation date: Feb 6, 2025
Owner: wipp-user
Status: CREATED

[Submit workflow](#)

Graph Orientation: Left to right

Job detail [Close]

Name:	dfd-test
Creation date:	Feb 6, 2025
Status:	CREATED
Owner:	wipp-user
Workflow Name:	dfd
Plugin:	WIPP Resnet50 Regression Inference Plugin:2.0.0
Computed in version:	3.2.0-SNAPSHOT

Inputs ▾

image_folder	test
saved_model_filepath	66e987f5377e7b28636ce973

Outputs ▾

csvCollection	
---------------	--

Job detail [Close]

Name:	dfd-dfsdfsdfsdfsdfs
Creation date:	Feb 6, 2025
Status:	CREATED
Owner:	wipp-user
Workflow Name:	dfd
Plugin:	WIPP Resnet50 Regression Inference Plugin:2.0.0
Computed in version:	3.2.0-SNAPSHOT

Inputs ▾

image_folder	test
saved_model_filepath	66e84c0b84d9b674549d46dd

Outputs ▾

csvCollection	
---------------	--