

# Loading Packages

```
In [1]: import os
import torch
from utils.visualize import visualize_images_from_classes
from utils.datasets import returnDataLoader
from utils.model_execution import runFMA, runDLA
from utils.analysis import print_analysis
```

/Users/alex/programming/python/HistoART/.venv/lib/python3.13/site-packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See [https://ipywidgets.readthedocs.io/en/stable/user\\_install.html](https://ipywidgets.readthedocs.io/en/stable/user_install.html)

```
from .autonotebook import tqdm as notebook_tqdm
```

```
In [2]: if torch.cuda.is_available():
device = torch.device("cuda")
elif torch.xpu.is_available():
device = torch.device("xpu")
elif torch.mps.is_available():
device = torch.device("mps")
else:
device = torch.device("cpu")

print(f"Using device: {device}")
```

Using device: mps

# Load Patches

```
In [11]: while True:
data_dir = input("Please enter the dataset directory, for example:

if not os.path.isdir(data_dir):
print(f"Directory '{data_dir}' does not exist. Please try again")
continue
print("\nDataset directory confirmed.\n")

break

classes = []
artifact_free = input("Please enter the name of the artifact free
artifact_free_path = os.path.join(data_dir, artifact_free)

if artifact_free and os.path.isdir(artifact_free_path):
classes.append(artifact_free)
print(f"Folder '{artifact_free_path}' exists and was added as arti
```

```
else:
    if artifact_free:
        print(f"Folder '{artifact_free_path}' does not exist. Please c

artifact      = input("Please enter the name of the artifact folder (e
artifact_path = os.path.join(data_dir, artifact)

if artifact and os.path.isdir(artifact_path):
    classes.append(artifact)
    print(f"Folder '{artifact_path}' exists and was added as artifact.
else:
    if artifact:
        print(f"Folder '{artifact_path}' does not exist. Please check

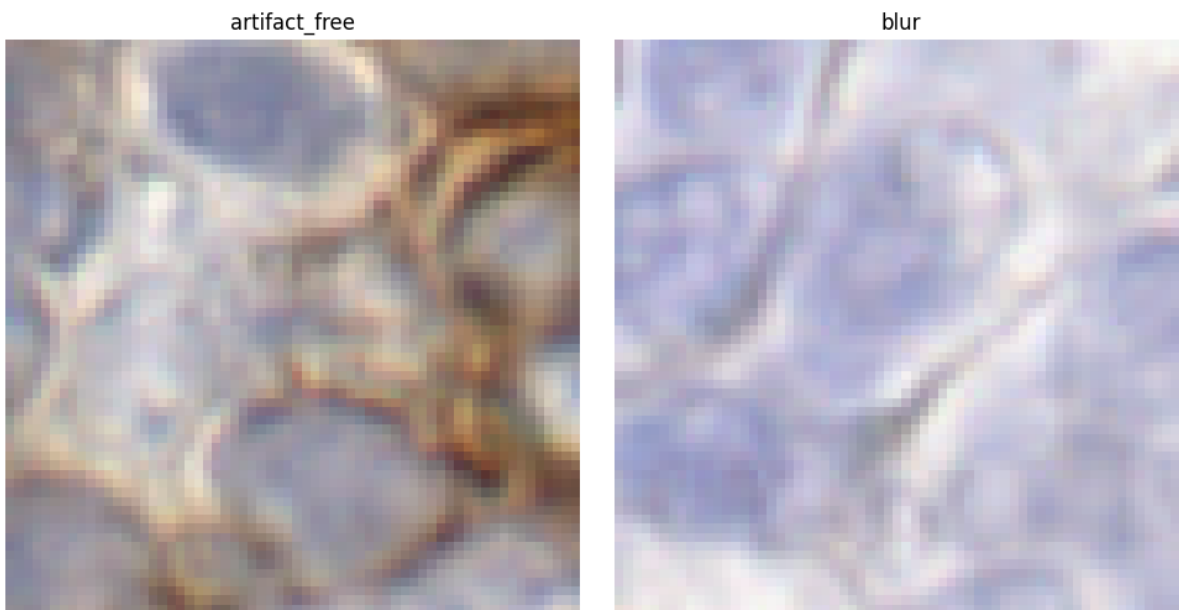
if classes:
    print("\nDataset loaded successfully.")
else:
    print("\nNo valid class folders were provided. Defaulting to sampl
    classes = ['artifact_free', 'blur']
```

Dataset directory confirmed.

No valid class folders were provided. Defaulting to sample classes.

## Random visualization of different classes

In [12]: visualize\_images\_from\_classes(data\_dir, classes)



## Prepare data for classification

```
In [13]: dataloader = returnDataLoader(data_dir, classes)
         print("Total dataset size (samples): ", len(dataloader.dataset))
```

Total dataset size (samples): 204000

## Classify images with different models

### FMA

```
In [15]: runFMA(dataloader, device, './models/fma_binary.pth')
```

Epoch 1 [test]: 100%|██████████| 12750/12750 [2:41:07<00:00, 1.32it/s]

### DLA

```
In [16]: runDLA(dataloader, device, './models/dla_binary.pth')
```

/Users/alex/programming/python/HistoART/.venv/lib/python3.13/site-packages/torchvision/models/\_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.

warnings.warn(

/Users/alex/programming/python/HistoART/.venv/lib/python3.13/site-packages/torchvision/models/\_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=ResNet50\_Weights.IMAGENET1K\_V1`. You can also use `weights=ResNet50\_Weights.DEFAULT` to get the most up-to-date weights.

warnings.warn(msg)

Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to /Users/alex/.cache/torch/hub/checkpoints/resnet50-0676ba61.pth

100.0%

Epoch 1 [test]: 100%|██████████| 12750/12750 [22:35<00:00, 9.41it/s]

### KBA

## Analysis

```
In [3]: print_analysis('./results/fma_results.csv')
```

Percentage of predicted Artifact: 100.00%

Percentage of predicted Artifact Free: 0.00%

```
In [4]: print_analysis('./results/dla_results.csv')
```

Percentage of predicted Artifact: 95.51%  
Percentage of predicted Artifact Free: 4.49%

In [ ]:

