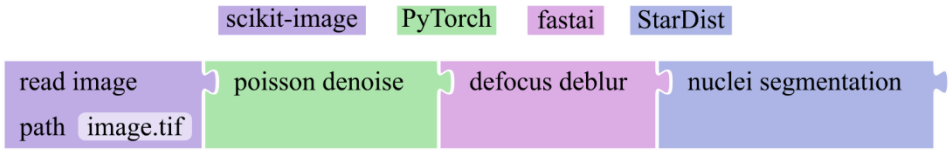
a)



b)

Process Overview	Pillow Image as Intermediate Representation	Our Method
input	pil_im: Pillow.Image.Image	im: im2im.Image

before operation	**if** pil_im.mode != 'L':
	  pil_im = pil_im.convert('L')
	torch_im = ToTensor()(pil_im)
	if torch.cuda.is_available():
	  torch_im = torch_im.to('cuda')
	torch_im = torch.unsqueeze(torch_im, 0)
		im = im2im(im, 'torch.gray')

operation	torch_im = poisson_denoise(torch_im)	torch_im = poisson_denoise(im.raw_im)
after operation	torch_im = torch_im.to('cpu') torch_im = torch.squeeze(torch_im) pil_im = ToPILImage()(torch_im)	im = im2im.Image(torch_im, 'torch.gray')

a)

A workflow with automatic image type conversion includes reducing Poisson noise with a PyTorch-based model, applying a FastAI-based model for defocus deblurring, and segmenting nuclei using the StarDist library [31]. *b) Comparison of the two implementation methods: using Pillow Image as an explicit intermediate format versus our approach. Our library creates minimal conversion code and eliminates the need to implement and maintain conversion code between the intermediate format and the specific input/output requirements of each operation.*