Electron microscopy (3D) – image processing and analysis

František Kitzberger

Biology Centre CAS, České Budějovice

Faculty of Science, Charles University in Prague

frantisek.kitzberger@paru.cas.cz



How to process data from the EM? – Lecture outline



What if the dataset is way too big for manuAl segmentation?

Software

FIJI – Fiji is just ImageJ



ImageJ

Java-based image processing software

Open architecture - lot of plugins to instal to do what you need

• Did not find the plugin? Learn to code and write one yourself – you can even share it

Recordable macros – even in the human legible protocol (not only the code but it explains you what you did, when you recorded the macro long time ago)

Image filtering, bit-depth transformation, Stitching, alignment, segmentation

Too many choices – you need to know what do you need



FIJI – ImageJ with batteries included (lot of plugins are already installed)

MIB – Microscopy Image Browser

- Matlab based software developed by Ilya Belevich (Laboratory of Eija Jokitalo, EMBI Helsinki)
 - Matlab version/standalone version (does not require a Matlab installed)

Software

- Freeware, constantly developed and updated
- Functions for data processing
 - Alignment→filtering→segmentation→deep learning→basic model rendering
- Most processes are dependent on RAM usually 2.5x RAM than is your dataset
 - For data from 3D-SEM (tens to hundreds of Gb) you need powerfull workstation or a cluster

Raw data/dataset

- 8bit or 16bit image, usually .tif
 - Raw data formats:
- From 3D-SEM images, images to be stitched or stitched images (big map)

MAGIC







Algorithm based combining multiple images into a big high-resolution image



Neighboring tiles are "sewn" together and they are aligned within the stack

Stitching







Imaging software (e.g. MAPS) provide some sort of own stitching

- Works well on good samples (brain or nervous tissues, or samples with a lot of structures)





Trypanosomas

MANY STRUCTURES = GOOD AUTOMATIC STITCHING

Stitching





Empty resin = problems for the automatic stitching

MAGIC







Software works differently than a human, we are looking for a trends, software matches points in the overlapping area.

Stitching – how does it work?

Software works differently than a human, we are looking for a trends, software matches

points in the overlapping area.



Software steps – Dataset

How it works (in Image J – TrakEM)

A) Translation





only displacements in X,Y. translations plus rotation

C) simillarity





translation, rotation and isotropic scaling (that is, it preserves image aspect ratio.) free affine transform, which amounts to translation, rotation, scaling, and shear







Alignment – put the disordered stacked (stitched) images to aligned state





Stack of EM images that comes from the microscope is never aligned –Beam jumps etc.

Processing

Alignment – put the disordered stacked (stitched) images to aligned state

Various algorithms to align the images – each has specific use (MIB)

		Current dataset		
Filename: C:\ SBEM	Trypanosoma.tif	Filename: C:\ SBEM_	Trypanosoma.tif	
Diminsions, px: 171 x	813 x 887	Diminsions, px: 171 x 8	13 x 887	
	019 x 0.014019 x 0.030000	Pixel size, um: 0.0140		30000
lign		Align		
Mode	Algorithm: Drift correction	Mode	Algorithm:	Automatic feature-based
 Current dataset 	HDD mode (process non-loaded files)	 Current dataset 		Drift correction
Two stacks		O Two stacks		Template matching
U Two stacks	Correlate with: Previous slice Step: 1	U TWO SIACKS	Correlate with:	Automatic feature-based
Use the Drift correction	Color channel: ColCh 1 V use intensity gradient	Use automatic feature	Color channel:	AMST: median-smoothed template Single landmark point
mode for small shifts or comparably sized		detection to align slices	Type:	Landmarks, multi points
images			Type.	Three landmark points
	Mode: cropped \checkmark 2 (min: 6 \checkmark		Mode:	Color channels, multi points
	Feature detector: Blobs: Speeded-Up Rob Y Preview		Feature detector:	Blobs: Speeded-Up Rob Y Preview
	Median size: 15 vise parallel computing		Modian sizo:	15 vise parallel computing
	Median size: 15 Use parallel computing		Median size:	15 vse parallel computing
	Background color		Background color	
			Background color	
ation	Background color		Background color	
ptions	Background color White Black Mean Custom 255	Options	Background color	Black Mean Custom 255
ptions Subarea Full image	Background color White Black Mean Custom 255	Options Subarea Full image	Background color	
Subarea Full image	Background color White Black Mean Custom 255	Subarea Full image	Background color White	Black Mean Custom 255
Subarea Full image	Background color White Black Mean Custom 255 Save shifts to a file C:SBEM_Trypanosoma_align.coefXY	Subarea Full image	Background color	Black Mean Custom 255 Save shifts to a file SBEM_Trypanosoma_align.coefXY
Subarea Full image minX: 43 minY: 203	Background color White Black Mean Custom 255 Save shifts to a file C:SBEM_Trypanosoma_align.coefXY	Subarea Full image	Background color	Black Mean Custom 255

Straightforward and robust stack alignment More individual parameters to set (if you know what they do...)

1/50 z:0.0 inches (50%) Untitled 0 66.72x72.45x	Register stack slices: SIFT parameters ×	Register stack slices: Geometric Conse 🗙	Register stack slices: Alignment para ×
Tel egins Anotations Live file Occiv Labels Live file Occiv Roy Constants Packes Profiles Z egace Consultations Units (Constants) Packes Profiles Z egace Consultations (Constants) Constants (Constants) Constants (Constants) Constants (Constants) Constants (Constants) Constants (Constants)	Scale Invariant Interest Point Detector: Initial gaussian blur: 1000 px steps per scale octave: 3 px minimum Image size: 64 px maximum Image size: 600 px Feature Descriptor: feature descriptor size: 8 feature descriptor orientation bins: 8 closestitext closest ratio: 9.92	maximal alignment error: 100.00 px minimal inlier ratio: 0.20 minimal number of inliers: 7 expected transformation: Rigid • fignore constant background tolerance: 0.50 px OK Cancel	desired transformation : Rtpid correspondence weight : 1.00 regularize Optimization: maximal iterations : 2000 maximal plateauwidth : 200 f fitter outliers mean factor : 3.00
-	OK Cancel	Platch → Diplicatel Cancel Color Color Lock Agentatick slices Beeten Deleten Beeten Beeten Lock Agen stack slices Transform Agen multi-kayer mosaic Luk Agen multi-kayer mosaic Lik Agen multi-kayer mosaic Script Moritige mailtinge in this kayer Image Agent mosaic Dial Agent mosaic slaver setter	OK Cancel

Can do in parallel stitching of multiple tiles

Cannot do the stitching

Alignment – put the disordered stacked (stitched) images to aligned state

XZ

YΖ

From this:



XY

To that:



XY



ΧZ

Processing

Pixel binning

- Process of combining adjacent pixels in the image by summing or averaging their values into single pixels.
 - Binning 2x2 merge array of 4 pixels into one

Normal



2x2 binning



4x4 binning









Pixel binning



2x2 binning



4x4 binning







- The binning decreases number of pixels in the image thus decreases the image size, decreases noise and increases contrast ⁽²⁾
- BUT also decreases image resolution we lose some image information [®]
- For huge datasets it is inevitable as too big data are difficult to work with (long calculation time, not enough RAM/GPU...)

Processing

Image filtering

۲ می ا
$\bullet \bullet \bullet \bullet$

Image enhancement and modification.

Smoothing, sharpening, edge enhancement, noise removal





ALWAYS STORE THE UNFILTERED DATA FOR **PUBLICATION!!!**

The altered ones can be used to improve the segmentation - models can be then applied to original data

4	Image filters		—	
F	Filter selection			
	DISK	 ction and the "average" pr 	redefined filter from <u>fspec</u>	ial
F	Frangi Gaussian Gradient LoG Mode Motion Prewitt Range SaltAndPepper Sobel	Color channel: All Material index: 1 Filteri	■ 3D ■ use paral	
ſ	Std Help Preview	Auto preview Fitler	image 🔻 Fi	ilter

Image filtering – examples









Processing



"Fast(est)" quantification for big datasets

- Stereology Random systematic sampling method for obtaining quantitative information about 3D material from measurements done on 2D planar sections
 - Grid based
 - We use simplified version (Cavalieri's principle) just for the statistical purposes
 - True stereology is much more complicated

How to find out the volume of the object



Sink it in the water and find out how much the water level rised.



Cavalieri's principle

Cut the object into equally thick pieces Put a grid of known size on the cuts Count the points (we know the area represented by one point – grid constant) Multiply the point count with the grid constant and with the cut thickness

- "Fast(est)" quantification for big datasets
 - Start with an image of known pixel size (defined at the acquisition)
 - Apply a grid with known proportions
 - Assign each cross-section to a "material"



Stereology

- "Fast(est)" quantification for big datasets
 - Start with an image of known pixel size (defined at the acquisition)
 - Apply a grid with known proportions
 - Assign each cross-section to a "material"
 - Count numer of dots of each material in the image
 - Repeat for all images in the stack (or every Nth)

	ervtrocvte" sta	ats —		
	Pi 💁 "ervtrocyte'	stats	— П X	
		vte" stats		X
The second second		rocyte" stats		
		envtrocyte" stats		
	Pa 🗹	"ervtrocvte" stat		
	P		stats —	
	State State	-Parameters	Chana	Mada
		Objects erytrocyte	Shape	Mode
	ALL ALL STA			Object
	D ANY Y	3D, Stack	✓ O 3D objects	Intensity
	Real Car	Area	 First cold 	r: ColCh 1 🗸
		4/6 connectivity	✓ Second cold	
	1 Anna Martin	pixels ~	Multiple	Define properties
	S. PARA	Statistics		
	12	Object	d Value SliceNo	o TimePnt
	Constant of the		37 9636	1 1
	1BALLO		43 9633	1 1
	1 AMARTON ON		39 9626	1 1
			33 9622	1 1
			44 9618	1 1
		34	38 9612	1 1
	1-1-1-1-1	35	36 9608	1 1
	C. C. S. S. S.	9	40 9608	1 1
	Carl Marcia		42 9606	1 1
	CC TO		31 9600	1 1
No State	A State		35 9600 32 9594	1 1
			52 9594 18 9531	1 1
120000	and the state		14 9522	1 1
			45 8220	1 1
			24 7765	1 1
	1 A BAR	45	25 7759	1 1 🗸
	and the state of the			
a start		Histogram		
Sec.		8		
		4		
		2		
		8000	9000 10000	11000
		1		

"Fast(est)" quantification for big datasets

- Start with an image of known pixel size
- Apply a grid with known proportions
- Assign each cross-section to a "material"
- Count numer of dots of each material in the image
- Repeat for all images in the stack (or every Nth)
- Calculate "comparisonal" stereology

		ľ						2 DI	PI							5 C	OPI					voxelsize	2x2x5um								
	Mutation		NC		9	951		LP			Brit			951		LP				Brit						N	c	21	DPI	S D	PI
	1	points	%		points	%	points	5 %	%	points	%	%	points	%	%	points	%	96	points	%	%					points	%	point	%	points	%
3	Pneul	521	8	16	596	11,2 19	1018	9,9	19,9	594	7,3	17,4	317	6,0	15,5	405	8,0	10,4	1582	7,9	19,6			rtes	Pneu Lok	521	16	2208	19,0	2304	16,5
Ĕ	Pneuldam	55	1	2	14	0,3 0	28	0,3	0,5	32	0,4	0,9	109	2,1	5,3	71	1,4	1,8	16	0,1	0,2			Loc l	Pneu I dam	55	2	74	0,6	196	1,4
le	Pneu II	372	6	12	227	4,3 7	408	4,0	8,0	412		12,1	231	4,4		521	10,3	13,4	1108	5,5	13,7			feur	Pneull ok	372	12	1047	9,0	1860	13,3
<u> </u>	Pneu II dam	0	0	0	16	0,3 1		0,0	0,0	27	0,3	0,8	0	0,0	0,0	84	1,7	2,2	0		0,0			P,	Pneull dam	0	0	43	0,4	84	0,6
	Endoth. cell	1052		33	744	14,0 24	1267	12,4		655	8,1	19,2	439		21,4	522	10,3	13,4	1807	9,0	22,4				Endoth. Cell	1052	33	2666	23,0	2768	19,8
	Endoth. cell dam	18 271	0	1	154 203	2,9 5 3,8 7	96 254	0,9	1,9	123 131	1,5 1,6	3,6 3,8	42 201	0,8 3,8	2,1 9,8	86 130	1,7 2,6	2,2 3,3	62 534	0,3 2,7	0,8 6,6				Endoth.cell dam Erv	18 271	9	373 588	3,2	190 865	1,4 6,2
>	Ery Immature ery	0	ō	0	8	0,2 0	254	2,5 0,0	5,0	0	0.0	5,0	201	0,0	0,0	150	2,0	5,5	0	2,1	0,0					0	0	900	0,1	0	0,0
Ilar		_	-	ž	-	0,2 0				-	-,-		-	1.1	1				-	~ ~				ŝ	Immature ery	39	-	130	1,1	113	0,0
E.	Thrombocyte	39	1	1	55	1,0 2	65	0,6	1,3	10	0,1	0,3	19	0,4	0,9	21	0,4	0,5	73	0,4	0,9			<u>G</u>	Thrombocyte		1				
0	Immune cells	13 0	0	0	147	2,8 5	14	0,1	0,3	134 0	1,6	3,9	46	0,9	2,2	173	3,4	4,4	194	1,0	2,4				Immune cells	13	0	295 41	2,5	413	2,9
	Neutrophil Eosinophil	3	0	0	23	0,4 1	18	0,2	0,4	ŏ	0,0 0,0	0,0 0,0	1	0,0 0,0	0,0 0,0	89	1,8 0,0	2,3 0,0	38 11	0,2 0,1	0,5 0,1				Neutrophil Eosinophil	3	0	41	0,4	128	0,9
	Capillary lumen	162	2	5	53	1.0 2	260	2.5	5.1	69	0.8	2.0	58		2.8	31	0.6	0,8	181	0.9	2.2				Capillary lumen	162	5	382	3.3	270	1.9
	Lipofibroblasts	555	8	18	313	5,9 10	897	8,8	17,5	706	8,7	20,7	395	7,5	19,3	465	9,2	11,9	1287	6,4	16,0				Lipofibroblasts	555	18	1916	16,5	2147	15,3
	Lipofibroblasts dam	0	0	0	37	0.7 1	0	0,0	0,0	2	0,0	0,1	7	0,1	0,3	0	0,0	0,0	0		0,0				Lipofibroblasts da	0	0	39	0,3	7	0,0
EGM	Fibroblasts	0	0	0	0	0 0	0	0,0	-,-	0	0,0	0,0	0	0,0	0	0	0	0	27	0,1	0,3			~	Fibroblasts	0	0	0	0,0	27	0,2
ų	LD	0	0	0	0	0,0 0	0	0,0	0,0	8	0,1	0,2	0	0,0	0,0	27	0,5	0,7	52	0,3	0,6			-EC	LD	0	0	8	0,1	79	0,6
Ebr	ECM	27	0	1	122	2,3 4	77	0,8	1,5	64	0,8	1,9	51	1,0	2,5	90	1,8	2,3	240	1,2	3,0			- fil	ECM	27	1	263	2,3	381	2,7
	Dense Fibers	7	0	0	49	0,9 2	61	0,6	1,2	17	0,2	0,5	10	0,2	0,5	43	0,9	1,1	68	0,3	0,8				Dense Fibers	7	0	127	1,1	121	0,9
	Collagen Fibers	0	0	0	80	1,5 3	77	0,8	1,5	27	0,3	0,8	22	0,4	1,1	77	1,5	2,0	213	1,1	2,6				Collagen Fibers	0	0	184	1,6	312	2,2
	Macrophage	39	1	1	43	0,8 1	430	4,2	8,4	141	1,7	4,1	10	0,2	0,5	212	4,2	5,4	375	1,9	4,7				Macrophage	39	1	614	5,3	597	4,3
e	Macrophage cryst.	0	0	0	0	0,0 0	0	0,0	0,0	0	0,0	0,0	0	0,0	0,0	467	9,3	12,0	0		0,0				Macrophage	0	0	0	0,0	467	3,3
ă	Neutrophil free	0	0	0	44	0,8 1	0	0,0	0,0	17	0,2	0,5	0	0,0	0,0	0	0,0	0,0	0	0,0	0,0			Ded.	Neutrophil free	0	0	61	0,5	0	0,0
E .	Trombo free	0	0	0	8	0,2 0	0			0	0,0	0,0	0	0,0	0,0	0	0,0	0,0	0	0,0				lar.	Trombo free	0	0	8	0,1	0	0,0
veol	Necrotic unknown	22	0	1	60	1,1 2	95	0,9	1,9	220	2,7	6,4	38	0,7	1,9	314	6,2	8,1	99	0,5	1,2			Alveo	Necrotic unknown	22	1	375	3,2	451	3,2
R	Erythrocyte	2	0	0	22	0,0 1	0	0,0	0,0	0	0,0	0,0	0	0,0	0,0	0	0,0	0,0	15	0,1	0,2				Erythrocyte	2	0	22	0,2	15	0,1
	Empty	3509	53		2269	42,5	5115	50,0		4719	58,0		3198	61,0		1154	22,9		11952	59,7					Empty	3509		*****		16304	
	Unknown	4	0	0	46	0,9 2	55	0,5	1,1	27	0,3	0,8	52	1,0	2,5	65	1,3	1,7	79	0,4	1,0				Unknown	4	0	128	1,1	196	1,4
	Sample dam	0	0	0		0		0,0		0	0,0								0						Sample dam	0	0	0	0,0	0	0,0
	total points	6671			5333		10235	5		8135			5246			5047			20013						total points	6671		*****		30306	
	total bez empty	3162	100	****	3064	99,6 ##	\$ 5120	100,0		3416	*****	100,0	2048	100,0	*****	3893	#####	100,0	8061	100,0	*****				total points without empty	3162					
Lo	/total fibroblas	0			0	_	0			1,13			0			5,806451613	3		3,96												

Relative area of interest(%) =
$$\frac{\text{number of poins of material}}{\text{total number of points}} * 100\%$$

Absolute area of interest = relative area of interest * image size * pixel size

Stereology

"Fast(est)" quantification for big datasets

- Start with an image of known pixel size
- Apply a grid with known proportions
- Assign each cross-section to a "material"
- Count numer of dots of each material in the image
- Repeat for all images in the stack (or every Nth)
- Calculate "comparisonal" stereology
- Extrapolate the data...



Segmentation – what is a segmentation?

- Process of partitioning of digital image into multiple segments/regions/objects
 – giving every pixel a label
 - Pixels with certain characteristics share a label
 - Goal: Simplify the image in order to facilitate reconstruction and analysis



https://upload.wikimedia.org/wikipedia/commons/d/d4/Image_segmentation.png

Brief explanation of how the segmentation works

- In layers (similar to Photoshop):
 - Image Base layer original data which we use to selet objects of interest on
 - Model Second Layer digital representation of object of interest in the image (different objects – different materials)
 - Mask Third Layer separate layer for selecting region of interest

Most basic/manual segmentation tools Brush – Manual painting of regions of interest – similar to Microsoft Paint

 B/W threshold – selects pixels of corresponding parameters – in whole image/stack or in the selected area (mask/material)

 Membrane ClickTracker – tool for following lines along the same or very similar pixel intensity

They still have their uses but for big datasets are slow – we need something more powerful!







Superpixel clustering used for semiautomatic segmentation











Superpixels

- Compact and coherent regions obtained by grouping pixels together based on their similarities in color, texture, and spatial proximity.
- Higher-level representation of an image by reducing the number of pixels while preserving important image structures.
 - Each superpixel represents a region of similar characteristics.
 - Less pixels→less datapoints→faster segmentation
 - Boundary adherence
 →more precision than
 pixel-based segmentation



Zhen Yu, et al 2021



https://forum.image.sc/t/need-to-find-out-the-order-in-whichindividual-superpixels-are-numbered-on-the-image/66535



https://www.analyticsvidhya.com/blog/2021/05/image-segmentation-with-felzenszwalbs-algorithm/



(a)





https://imagej.net/imaging/watershed

What to do with the superpixels? - Manual and semi automatic segmentation

Manual pixel brush vs superpixel brush

Manual brush segmentation - lengthy, tedious, imprecise

Superpixel brush segmentation

- with propper clustering
- fast, precise

- there is no sigle optimal clustering that works perfectly for everything



What to do with the superpixels? - Manual and semi automatic segmentation

Graphcut semiautomatic segmentation

- Pre-calculates the superpixels/supervoxels in whole image/stack – oversegmentation
- Graph representation of an image is created
 - Superpixel is represented as node in the graph, coneced by edges to another superpixel node
 - The weight of each edge is dependent on the dissimilarity of neighboring superpixels more dissimilarity less weight
- Cuts the edges in order to separate image into background and desired object (segments)
 - The cut tries to find optimal path in with minimal energy cost and/maximizing desired criteria, such as boundary adherence or pixel intensity coherence

With a good data, you can have segmentation of individual organelles/image parts in couple of clicks Most commonly works in binary



Segmentation pipeline with MIB Graphcut




What to do next?

Statstics

- Lengths, areas, surfces, volumes, counts
- Not visual
- Scientifically important data

Model rendering

- Nice visual representation of your laborious work
 - And to have better image what actually goes on
- Good to show to the public
- Not so scientifically important

Model Rendering

- The segmenting software usually have way to visualize what you segmented or you can use some other software
 - MIB VolumeViewer, Imaris viewer (free), ImageVis3D, AMIRA Avizo

Object visualization in Amira orthoslice

- No segmentation is necessary
- Simply look through the stack on all 3 axes



Model Rendering in Amira - volren

- Simplest and fastest visualization, if the dataset has good contrast
- No segmentation is necessary



Set the minimal values, maximal values, and opacity to visualize objects of interest

Model Rendering in Amira – volume rendering/ voxelized rendering

• Already requires segmented sample



Volume rendering

Voxelized rendering



Model Rendering in Amira – Surface generation

- Already requires segmented sample
- Bit more complex, but more variable
 - Allows smoothing
 - Each material can be visualized individually – much more responsive transparency
 - More laborious each material is set individually



Model Rendering in Amira – Surface generation difference between volume and surface



Volume – cut

Data visualization

Surface – cut

What if the dataset is much bigger?

Segment Anything Model (META)

Artificial Intelligence The theory and development of computer systems able to perform tasks normally requiring human intelligence Ilastik, Arivis Machine Learning Gives computers "the ability to learn without being explicitly programmed" **Deep Learning** Machine learning algorithms with brain-like logical structure of algorithms called artificial neural networks LEVITY Deep MIB, Apeer, Amira

DeepMIB – actual neural network segmentation – GPU Based

- Manually segmented part of a dataset serves a textbook
 - The network learns about particular type of data and can be used on similar ones
 - The images of the stack are split with the corresponding labels into 2 files
 - Training
 - Validation
 - The software then trains on the training data and checks how it is doing on the validation data
 - From tens of minutes to days (depending on the complexity of segmentation, size of the dataset, GPU
- When the training is finished the learning is tested on "unseen" part of the dataset
- If the predictions are good predict the rest of the dataset

DeepMIB neural network segmentaion – tick gut

- 23450px x 21750px x 558sections
- Resolution 15nmx15nm*120nm
- (351.8μm x 326.3μm x 156.2μm=0.018mm³)
- 265 Gb
- Cooperation with scientific group (BC CAS): J. Perner



DeepMIB actual neural network segmentaion

Help

Part of the dataset to be segmented: 6750px x 4300px x 100 sections



Facilitated segmentation with semi automatic tools using the superpixel clustering, thresholding...



Roughly 10 days of work

	💰 DeepMIB			- 🗆 X
	Network			
	Workflow 2			WIDIA GeForce RTX 3 🔻 ?
		eepLabV3 Resnet18		
	Network filename:	C:\Users\kitzb\	Desktop\deep\myLove	elyNetwork.mibDeep
	Directories and Prepro	cessing Train	Predict Option	s
	Network design			
	Input patch size 6	4 64 1 1	Filters	32 🔯 Input layer
	Padding s		Filter size	3 use ImageNet weights
	Number of classes	2	Activation layer	
	Encoder depth	4	Segmentation layer	dicePixelCustomCla
				Check network
	Augmentation design			
	augmentation	🔯 2D 🔯	20	Preview 🔯
	augmentation		00	
	Training process desig			
	Patches per image	32 🔯 Traini	ng	Save checkpoint networks
	Mini Batch Size	1		 Export training plots
	Random seed	2		
	Beta version			Transfer learning
	Downsampling facto	pr 2		
	Help	Prepr	rocess Tr	ain Predict
		LT.		
	- 0	×		
GPU: 1. NVIDI	A GeForce RTX 3 *	2		
* 0P0. 1. NVID5	NOLI SILE RIA S *			
	twork.mibDeep			
Predict Options				
ining extension AM	Bio Index	1		
extension AM	Bio Index	1		
extension (MODEL V	Number of classes 2			
extension MASK V				
	iges for validation 0.2	15		
Frantise of in-				
		-		
Rando	m generator seed	2		
Rando 6	m generator seed			
Rando				
Rando	m generator seed			

2D augmentation settings				- 0	
	Each augmentation is defined with 2 or 3 values, the augmentation. The resulting augmented patch may	e first value(s) define variatio be made from a cocktail of	n of the augmentation filter, multiple augmentations	, while the last value specify probability of triggering the	
	Fraction: fraction of images to be augmented [0-1, def=0.9]	RandScale: random scaling, range (0-lnf), def=[1 1.1 0.05], off=[1 1 0.05] 1 1.1 0.05		HueJitter: jitter of Hue using a random value in the range (-1 1), def=[-0.03 0.03 0.05], off=[0 0 0.05]	
	FillValue: fill value for out-of-bounds points when rotating [0-Inf, def=255]	RandXScale: random X s	caling, range (0-Inf),	-0.03 0.03 0.05	
	when rotating [0-Inf, def=255] 255	RandXScale: random X scaling, range (0-lnf), def=[1 1.1 0.05], off=[1 1 0.05] 1 1.1 0.05		Saturation Jitter: jitter of Saturation using a random value in the range (-1 1), def=[-0.05 0.05 0.05], off=[0 0 0.05]	
	RandXReflection: random left-right reflections, def=[1 0.05], off=[0 0.05]	RandYScale: random Y def=[1 1.1 0.05], off=[1 1	scaling, range (0-Inf), 0.051	-0.05 0.05 0.05	
	1 0.05 1 1.1 0.0		0.05]	BrightnessJitter: jitter of Brightness using a random value in the range (-1 1), def=[-0.1 0.1	
	RandYReflection: random top-bottom reflection, def=[1 0.05], off=[0 0.05]	RandXShear: horizontal random shear, in degrees in the range (-90 90), def=[-10 10 0.05], off=[0 0 0.05]		0.05], off=[0 0 0.05] -0.1 0.1 0.05	
	1 0.05	-10 10 0.05		ContrastJitter: jitter of Contrast using a	
	Rotation90: rotation to 90 or 270 degrees, def=[1 0.05] RandYShear: vertii degrees in the range 0.05] 1 0.05 0.05		dom shear, in 90), def=[-10 10	random value in the range (0 Inf), def=[0.9 1.1 0.05], off=[1 1 0.05]	
				ImageBlur; allow Gaussian blur defined as	
	left-right reflection, def=[1 0.05], off=[0 0.05]	-10 10 0.05 GaussianNoise: add Gaussian noise using a		sigma in the range (0 lnf), def=[0 0.5 0.05], off=[0 0 0.05]	
	1 0.05 RandRotation: random rotations, in degrees	random variance in range (0 Inf), def=[0. 005 0.05], off=[0 0 0.05]		0 0.5 0.05	
	from -90 to 90, def=[-10 10 0.05], off=[0 0 0.05]	0 0.005 0.05			
	-10 10 0.05	PoissonNoise: allow Pois 0.05], off=[0 0.05]	sson noise, def=[1		
		1 0.05			
Help			ок	Cancel	
承 Training settings				:	
	ackerName, acker for training actuart		Manantum Tagda	n colul contribution of the personator	
	solverName, solver for training network adam	·	update step of the iteration of sgdm [n only] contribution of the parameter previous iteration to the current [0.9]	
	MaxEpochs, maximum number of epochs to use for training [30]		0.9		
	50		Decay rate of gradient moving average [adam only], a non-negative scalar less than 1 [0.9]		
	Shuffle, options for data shuffling [once	e]	0.9		
	every-epoch	~	Decay rate of squared gradient moving average for the Adam and RMSProp solvers [0.999 Adam, 0.9 PMSProp]		
	InitialLearnRate, used for training; The 0.01 for the "sgdm" solver and 0.001 fo	e default value is or the "rmsprop"			
	and "adam" solvers. If the learning rate then training takes a long time. If the learning	e is too low,	0.9		
	too high, then training might reach a suboptimal result or diverge		ValidationFrequency, number of validations per Epoch [2]		
	0.0005		2		
	LearnRateSchedule, option for droppin during training [none]	g learning rate	Patience of validation stopping of network training, the number of times that the loss on the validation set can be larger than or equal to the previously smallest		
	piecewise		Inf		
	LearnRateDropPeriod, [piecewise only] number of epochs for dropping the learning rate [10]		Plots, plots to display during network training		
	10		training-progress		
	LearnRateDropFactor, [piecewise only] factor for dropping the learning rate, should be between 0 and 1 ro 11		OutputNetwork, type of the returned network (R2021b)		
	[0.1]		last-iteration	~	
	0.1 L2Regularization, factor for L2 regularization (weight		Frequency of savir (R2022a)	ng checkpoint networks in epochs	
	decay) [0.0001]		1		
	0.0001				
Help			ок	Cancel	

DeepMIB actual neural network segmentaion

Part of the dataset to be segmented: 6750px x 4300px x 100 sections



Facilitated segmentation with semi automatic tools using the superpixel clustering, thresholding...





Roughly 10 days of work

Roughly 4 days of network learning

prediction

Ъ

couple of hour

and

Okay, some more time for correction of

prediction (2-3days)



TAKEAWAY MESSAGE: The segmentation of dataset of this size just by basic tools would take years. With help of semiautomatic tools and DeepLearning this was done in a few weeks.

DATASET size: 23450px x 21750px x 558sections 351.8µm x 326.3µm x 156.2µm

DeepMIB segmented datase

AMIRA rendered DeepMIB segm

model from the

segmented dataset

Deep learning module in AMIRA

• Straightforward – but requires to buy an expensive package to even more expensive software



Machine learning – Ilastik and Arivis

- "Manual segmentation" for machine learning can be more like an interactive lecture than a textbook
 - You are selecting pixels that are of structures of interest and the program tries to reproduce what you do in the rest of image
 - And you correct it





Future look of segmentation Using Al

• Implemetation of artificial inteligence into segmentation processes

• SAM from META





Constantin Pape @cppape

Introducing #SegmentAnything for microscopy, our napari based tools for interactive microscopy annotation: github.com/computational-... Pressit tweet



Still with me? Good Job!

Thank you for your attention!

Laboratory of Electron Microscopy Institute of Parasitology, BC CAS

Marie Vancová Jana Nebesářová Petra Masařová Martina Tesařová Jana Kopecká Jan Langhans Jiří Vaněček Tomáš Bílý Zdeno Gardian Eva Ďurinová Andrea Hatmanstorfer Valentina Hawlicek Ana-Maria Coroianu And least I forget: Jiří Týč



Laboratory of Molecular Biology of Ticks

Institute of Parasitology, BC CAS Jan Perner Veronika Urbanová





University of South Bohemia in České Budějovice Faculty of Science



