CDeep3M-Preview: Online segmentation using the deep neural network model zoo

Authors: Matthias G Haberl[1], Willy Wong[1], Sean Penticoff[1], Jihyeon Je[1], Matthew Madany[1], Adrian Borchardt[1], Daniela Boassa[1], Steven T Peltier[1], Mark H Ellisman[1]

[1] National Center for Microscopy and Imaging Research, School of Medicine, University of California San Diego, Biomedical Science Building, 9500 Gilman Drive, La Jolla

Abstract

Sharing deep neural networks and testing the performance of trained networks typically involves a major initial commitment towards one algorithm, before knowing how the network will perform on a different dataset. Here we release a free online tool, CDeep3M-Preview, that allows end-users to rapidly test the performance of any of the pre-trained neural network models hosted on the CIL-CDeep3M modelzoo. This feature makes part of a set of complementary strategies we employ to facilitate sharing, increase reproducibility and enable quicker insights into biology. Namely we: (1) provide CDeep3M deep learning image segmentation software through cloud applications (Colab and AWS) and containerized installations (Docker and Singularity) (2) co-hosting trained deep neural networks with the relevant microscopy images and (3) providing a CDeep3M-Preview feature, enabling quick tests of trained networks on user provided test data or any of the publicly hosted large datasets. The CDeep3M-modelzoo and the cellimagelibrary.org are open for contributions of both, trained models as well as image datasets by the community and all services are free of charge.

Main

New deep neural networks are developed rapidly and a startling number of trained models are available online for a wide range of image enhancement and analysis tasks (see [1] for a recent review). Since training new models is however expensive and typically requires laboriously expert annotated training data, innovations in sharing trained models effectively are critical to reduce time and cost in research. Model zoos and GitHub repositories with different networks and/or trained models are currently the most common way to share models, but are fairly disparate from the typical workflow or processing pipelines of biomedical labs. Passively hosted model zoos do not offer an immediate entry point to evaluate the performance of a neural network or a trained model, instead require to go first through complex installations - usually on high performance systems - before being able to know if the network will be useful for the specific question at hand. The computations to test a deep neural network typically require installation of several requisites on a high-performance GPU-equipped system and familiarization with the individual processing routines and configurations employed. Therefore, the use of model zoos has not been able to eliminate a major time commitment required to reproduce results. Even recent developments of more user-friendly solutions for running deep neural networks for image segmentation[2–4] on local or cloud resources do require a time commitment of researchers at different levels to: set up, configure and troubleshoot then familiarize themselves with software settings and testing parameters and their influence on performance. Further training and testing different models on several small or individual very large imaging datasets can be cumbersome. As a result, many cutting-edge developments for image analysis with deep learning are still not used by a large portion of the biomedical imaging community.

To facilitate sharing, increase reproducibility and enable quicker insights into biology we employ a set of innovative complementary strategies: (1) we recently released a deep neural network platform, CDeep3M[4], which circumvents installation issues and hardware requirements for end-users. We now provide a docker container of CDeep3M2 as well as a Google Colab installer with GUI (2) we are hosting CDeep3M pre-trained models in a public database (modelzoo), on cellimagelibrary.org (CIL) that also hosts relevant large microscopy datasets[5] and (3) we are now releasing an online CDeep3M-Preview feature, offering instantaneous testing of any trained neural network that is hosted on the CIL database. This allows users to 'test drive'

62 CDeep3M models within minutes on either their own data or on any region of interest

63 on a large number of publicly hosted imaging data to determine if a trained model of

64 interest performs well for their purpose and/or dataset (**Figure 1a-g, Supplementary**

65 **Figure 1a-b**). All results are displayed through a web interface, accessible to download

66 and can be shared with a link (**Figure 1e-g, Supplementary Figure 1c**). Users are

67 then guided through different options how to run the same model on a larger scale or

68 re-train the model with their own data using one of our distributions (**Figure 1h**). At the

69 same time the CDeep3M model uploader further provides users with a way to share

70 their trained models with the community in the modelzoo in a common place in a fully

71 functional and testable state (**Figure 2a**).

72   Data processing with state-of-the-art deep neural networks requires high-end

73 hardware with GPUs with sufficient vRAM. Online processing using deep neural

74 networks for many users, as provided by CDeep3M-Preview, is limited by hardware

75 availability equipped with high-end GPUs. We therefore implemented a scheduling

76 system outsourcing the processing of the preview to the infrastructure of the Pacific

77 Research Pipeline (PRP), a decentralized computing cluster with GPUs and storage

78 nodes. The PRP cluster is managed by Kubernetes, using containerized applications

79 to rapidly deploy computing jobs to available hardware. Our installation for the

80 CDeep3M-Preview is containerized as a Docker image on the PRP cluster, and is

81 streamed to available nodes allowing for near instantaneous start-up times (within

82 seconds). The imaging data and pre-trained models are sent from CIL and the

83 commands to initiate segmentation and the subsequent overlay with the segmentation

84 is submitted. Using a next-generation decentralized computer cluster, rather than

85 running the backend processing on workstation/s, provides scalability at times of high

86 demand. In practice, this means that the end users can test whether a trained model

87 performs on their dataset in less than 5min total, without programming knowledge and

88 without any hardware requirements (**Figure 1**).

89   Once the user has identified a pretrained CDeep3M model and tested settings

90 which perform well for a dataset of interest (**Figure 1e**), several routes are made

91 available that provide quick implementations for applying the pre-trained network and

92 the settings on the complete large-scale dataset. To this end we maintain several

93 preconfigured installers for the use on local hardware, on supercomputer clusters or

94 on cloud providers. Following pre-configured CDeep3M installers are provided: (i)

95 Docker container (ii) Singularity recipe (iii) AWS cloudformation template (iv) Colab

96 Notebook (**Figure 1h**). Detailed descriptions for configurations of each of those
97 solutions are available with the links provided. Without much effort or the obstacle of
98 requiring their own GPUs or funding for high-performance hardware the users can now
99 also take advantage of the free GPUs provided Google Colab, with the CDeep3M-
100 colab installer and graphical user interface (**Supplementary Figure 2**).
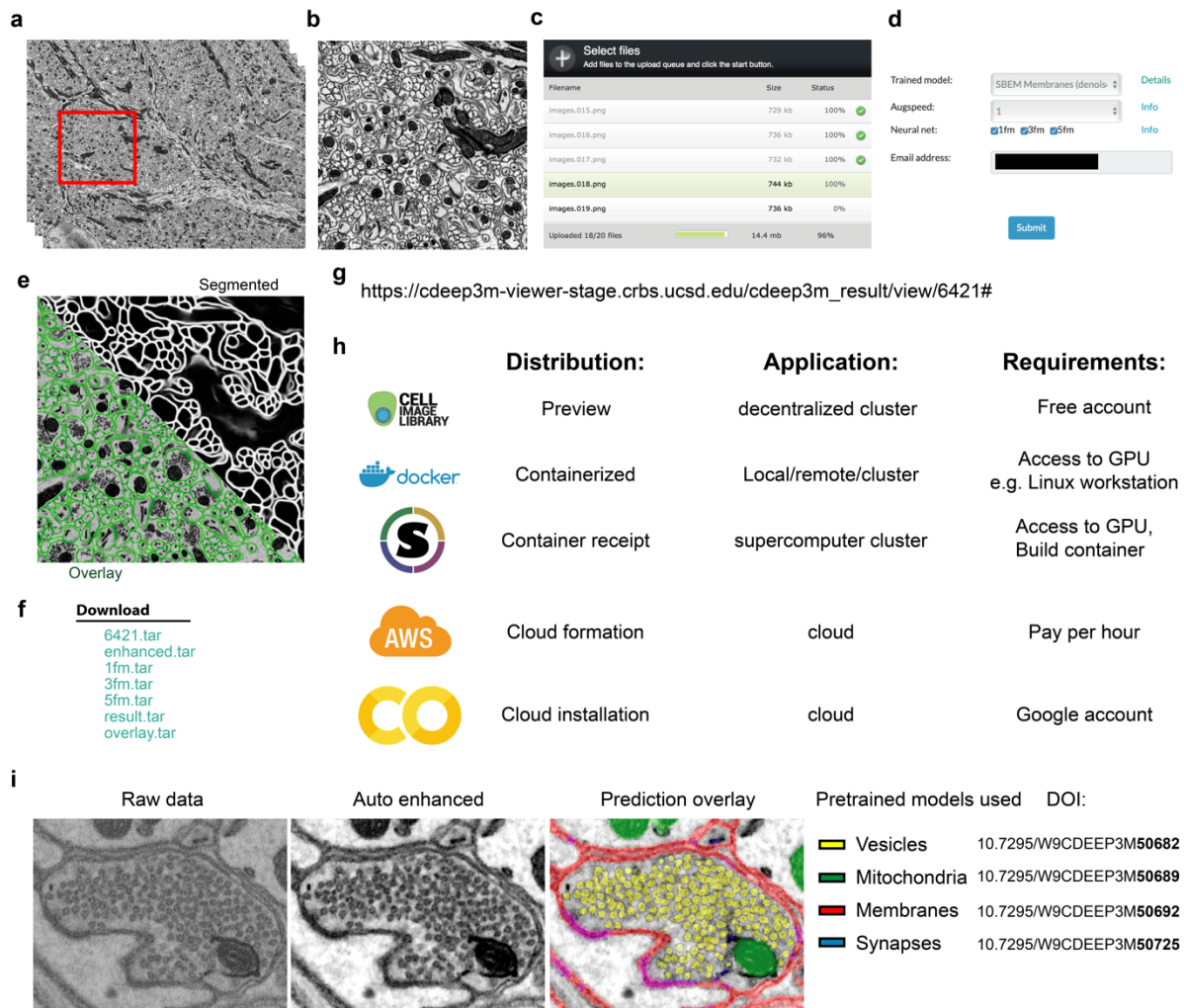
101
102 **Figure 1. Application of CDeep3M, using the preview function.** (a-b) Selection of a region of interest
103 (ROI) from a large dataset to run CDeep3M-Preview. (c) Upload ROI data through web interface
104 (https://cdeep3m-stage.crbs.ucsd.edu/cdeep3m). (d) Chose trained model from CDeep3M modelzoo
105 and select parameters to perform preview. (e) Results are accessible through a web interface (here
106 shown segmented and overlay) and (f) data is accessible for download. (g) Through the web interface
107 the link can easily be shared with collaborators. (h) CDeep3M is available in multiple distributions to
108 facilitate access for many groups, for quick entry points (CDeep3M-Preview, Docker, AWS, Colab),
109 small scale tests (Preview, Colab) as well as large scale data science projects (Docker, Singularity,
110 AWS). The end-user can then either apply the trained model to the large dataset or re-train the model
111 with specific training data through one of those distributions. (i) Multiple pre-trained models are available
112 on the CDeep3M-modelzoo and were combined in this example (without re-training the network for this

113 dataset) to segment the cellular constituents of synapses. Auto-enhancement is performed in
114 CDeep3M2 providing generalizable models.
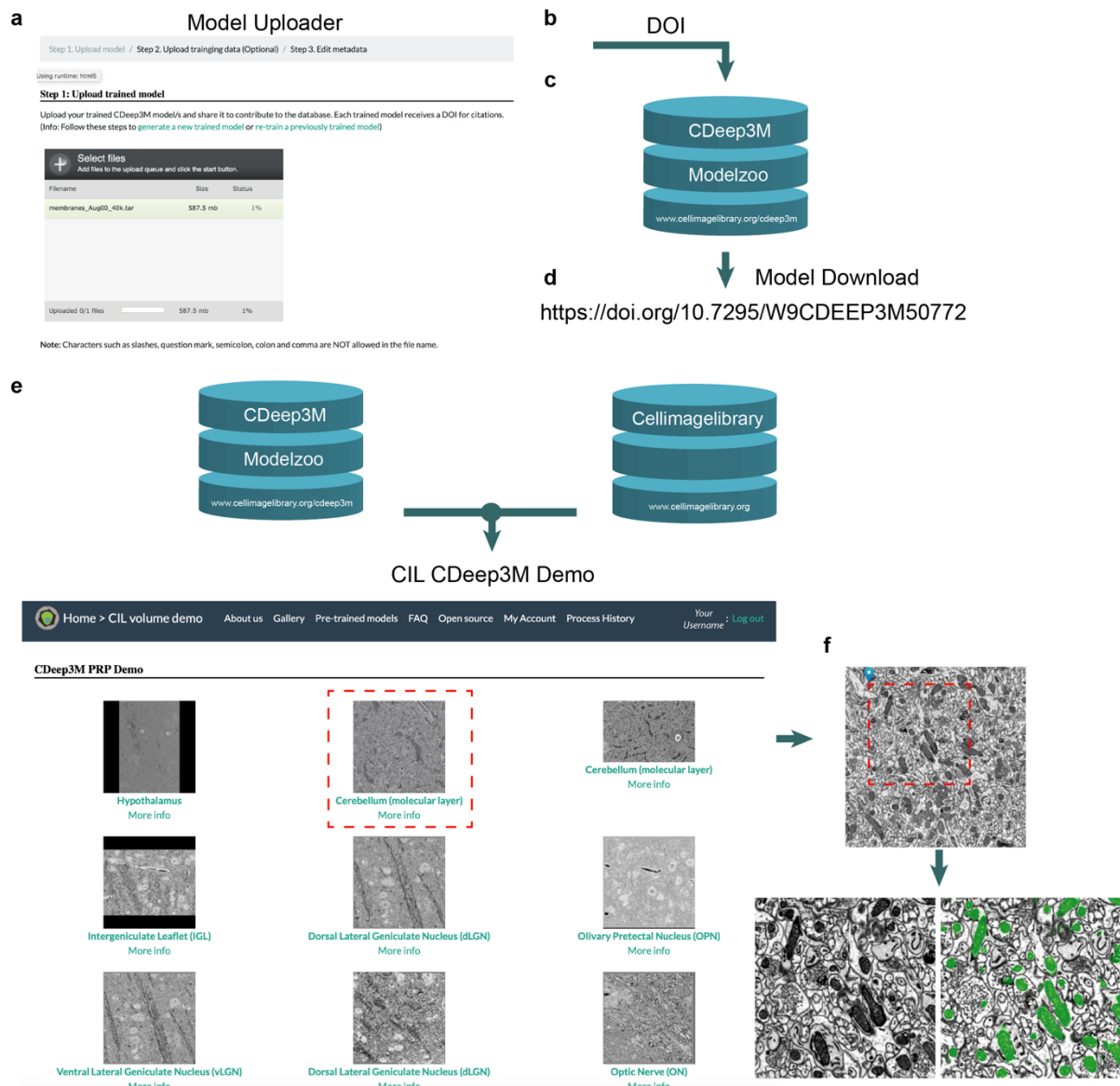
Figure 2. CDeep3M model uploader, database and demo function. (a) Community contributions to the CDeep3M-modelzoo are facilitated through a web interface to upload and add metainformation to trained models. (b) Each trained model receives a citable DOI and (c) is added to the CDeep3M database, and becomes therefore accessible for the Preview function and (d) can be easily downloaded. (e) The cellimagelibrary hosts many large-scale imaging datasets to which any of the trained networks in the modelzoo can now be applied through the CDeep3M-demo function in an image browser. (f) Example using a broadly trained CDeep3M model on a dataset for which it was no trained (before any transfer learning is applied). Results can be easily shared through using the specific job ID and the web interface. Results from (f) are at: https://cdeep3m-viewer.crbs.ucsd.edu/cdeep3m_result/view/6447

126   It is advantageous to co-host trained models and imaging data on the same
127 platform and maximize cross-linking between the two and facilitate testing across
128 several large datasets for generalizability. With the new extensions to the CDeep3M-

129   Modelzoo the users can upload their trained models to the CIL repository (**Figure 2a-**

130   **d**), in order to share them or to apply them through the preview function on one of the

131   imaging datasets. In addition, metadata about the available trained models is stored in

132   the database, such as the targeted cell component, staining procedures, imaging

133   modality and voxel dimensions. When releasing a trained model on the CIL database

134   to the public, it will generate a citable Digital Object Identifier (DOI) (**Figure 2b, 2d**).

135   The DOI is a persistent identifier used to identify objects uniquely, standardized by the

136   International Organization for Standardization (ISO). To help other groups unlock

137   valuable large scale data we are providing the CDeep3M-Demo, facilitating to test pre-

138   trained models on areas of interest on the large scale datasets available on

139   http://cellimagelibrary.org/ (CIL; **Figure 2e**). We co-host trained CDeep3M models on

140   the CIL providing us with the infrastructure already in place for data storage, metadata

141   organization and large-scale image visualization. CIL is open-source software

142   providing storage and user interfaces to deliver a publicly searchable database of

143   microscopy images and metadata to facilitate data sharing and reuse. The CIL data

144   input form allows end-users to submit images to the CIL data repository and annotate

145   the images with the ontology markup.

146

147   Together with the online preview function we release an upgrade to CDeep3M2,

148   which provides additional functionalities and reduced runtimes. The new version of

149   CDeep3M is backwards compatible, so that all previously trained models can still be

150   applied and used for transfer learning with the new release. Importantly, we

151   incorporated enhanced image augmentation strategies, that can easily be configured,

152   to facilitate the training of more broadly tuned models. In the enhanced training

153   augmentation pipeline, the images are first processed through the sixteen rotations

154   and inversions (x/y and z) before each stack will go through a set of additional

155   secondary and/or tertiary augmentations. The augmentations are performed as

156   follows: primary augmentations consisting of rotations and inversions (x: left/right, y:

157   top/down, z: forward/reverse) are always performed, secondary augmentations

158   consisting of image filters (contrasting, sharpening, blurring, total variation denoising,

159   introduction of uniform noise, histogram equalization, skewing, elastic distortion; and

160   tertiary augmentations, resizing the images. Secondary and tertiary augmentation

161   strengths are determined by the user with a scaling factor between 0 (no additional

162   augmentation) and 10 (strong augmentation). The combination of augmentation

163 strength can be customized for each dataset depending on the purpose of the training
164 (fine tuning of model for one individual dataset or broad training for generalizable
165 model). Furthermore, users can now easily provide multiple training datasets that will
166 be used during training, which facilitates generating broadly applicable trained
167 networks.

168   Applying the CDeep3M-Preview and Demo functions on the cellimagelibrary
169 serves us as an extreme test case scenario at unprecedented scale to test and improve
170 how well a trained deep neural network performs on previously unseen data
171 (generalizability). The available datasets are stored at various imaging conditions
172 (pixel- and voxelsize), ranging from 8bit unsigned integer to 32bit signed integer, with
173 different levels of noise, staining intensities and imaging conditions, acquired from
174 different tissue types. These constraints are very typical for different biological sample
175 preparation and imaging. Rather than performing training for each individual dataset,
176 we focused on training and providing more generalizing models, and stabilize their
177 performance through improved image pre-processing, which will reduce the
178 requirements to re-train the models. Mainly we automatized the following steps: image
179 conversion to 8bit, with simultaneous clipping of outlier pixels, readjustment of contrast
180 and a total variation denoising (**Figure 1i**). Overall, we note that these implementations
181 improved the generalizability of trained models making them broadly applicable to
182 many more datasets, since this reduces most of the extreme variations in signal-to-
183 noise levels and contrast between different SBEM datasets.

184   Altogether, the CDeep3M-Preview and new set of tools provided here gives
185 biomedical researchers immediate access to experiment with AI for image
186 segmentation and the ability to test different trained models near-instantaneously. A
187 similar quick entry approach has been taken recently with DeepCell, which allows
188 users to track cells in their own live cell imaging data with deep learning[6]. On the
189 CDeep3M modelzoo a broad range of pre-trained models, trained on segmentation
190 tasks for electron microscopy, X-ray microCT and light microscopy data are readily
191 available. Furthermore, by taking advantage of an emerging cyberinfrastructure of
192 decentralized compute cluster, the preview is scalable to account for the high demand
193 of many users. This approach can serve as an entry point for community members
194 with no experience in deep or machine learning to become familiar with the technology
195 and experiment with the effect of different parameter settings and will contribute to
196 democratize deep learning in the bioimaging community while allowing them to scale

197 their use case afterwards quickly to extremely large datasets through the CDeep3M

198 built in processing pipelines.

199

References

201 1. Moen, E. *et al.* Deep learning for cellular image analysis. *Nature Methods* **16**, 1233–

202   1246 (2019).

203 2. Falk, T. *et al.* U-Net: deep learning for cell counting, detection, and morphometry.

204   *Nature Methods* **16**, 67–70 (2019).

205 3. McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology.

206   *PLOS Biology* **16**, e2005970 (2018).

207 4. Haberl, M. G. *et al.* CDeep3M—Plug-and-Play cloud-based deep learning for image

208   segmentation. *Nature Methods* **15**, 677–680 (2018).

209 5. Orloff, D. N., Iwasa, J. H., Martone, M. E., Ellisman, M. H. & Kane, C. M. The cell:

210   an image library-CCDB: a curated repository of microscopy data. *Nucleic Acids*

211   *Research* **41**, D1241–D1250 (2012).

212 6. Moen, E. *et al.* Accurate cell tracking and lineage construction in live-cell imaging

213   experiments with deep learning. *bioRxiv* 803205 (2019) doi:10.1101/803205.

214

Acknowledgements

222

Author contributions

224 M.G.H. performed computational experiments. M.G.H. and J.J. wrote CDeep3M

225 upgrade. W.W. implemented CDeep3M-Preview and CDeep3M-Demo interface on

226 CIL and model zoo uploader and database. S.P. implemented preview backend
227 scheduler. A.B. maintains singularity recipe and AWS cloud formation template.
228 M.G.H. wrote colab implementation with GUI. M.G.H. and M.M trained models hosted
229 on modelzoo. D.B. acquired SBEM data. M.G.H., S.T.P. and M.H.E. supervised
230 project. M.G.H. wrote manuscript with input from all authors.

231

Competing interests

233 The authors declare no competing interests.

234

Methods

236 **CIL backend system.** Under the CIL system, the metadata will be converted into
237 JSON format and stored in a NoSQL datastore. The CIL utilizes Elasticsearch as its
238 JSON search engine since it provides a distributed, multitenant-capable full-text
239 search engine with an HTTP web interface. When JSON data is imported into the
240 Elasticsearch datastore, all of the data fields are automatically indexed and
241 immediately searchable using the built-in web-service. These built-in functions in
242 Elasticsearch are crucial for software development because it saves tremendous
243 development time otherwise spent building data models and backend services.

244

245 **Backend operations Scheduling system.** At the core of the scheduling system for
246 the CDeep3M-Preview and Demo functions is the beanstalkd queue
247 (https://beanstalkd.github.io/). Beanstalkd is a robust multichannel FIFO queuing
248 system. A single queue (or tube) contains the jobs in the order they are submitted.
249 Each job consists of the UID of the requestor, a description of the job to be run and its
250 arguments, and an authentication token with a timestamp. A job can be in one of 4
251 states, ready, reserved, delayed, or buried. A costum written Perl based web API
252 (stalker_web) is used as an abstraction layer for the interaction with beanstalkd. A job
253 is submitted to the system via a client; the client does a few basic sanity checks against
254 the submitted job, secures a token and submits the job to the queue via stalker_web
255 and if the submission is successful is returned the job id in the tube. Once in the tube
256 the job is in the ready state.
257 On the processing side there is a worker that periodically checks the tube for
258 ready jobs, if a job is found, it's parsed and if the worker has the capabilities needed
259 to run the job it verifies the token then reserves the job and then places it in the delayed

260 state for an amount of time equal to the expected runtime (ERT). This removes the job

261 from the ready state so no other workers will see the job. Since these are typically

262 longer running jobs and are designed to run on geographically disparate hardware in

263 an ad libitum fashion the delay is set to remove the need for the constant

264 communication between worker and queue required by the reserved state.

265   The worker then sets the environment, downloads any data or models it needs,

266 and proceeds to process the job as the UID of the submitter, collecting the output of

267 the commands into a log. A separate watchdog process is started that will check every

268 90% of ERT if the worker process is still running and if it is, try to reset the delay on

269 that job to ERT until successful or the parent process exits. Once the commands finish,

270 the worker deletes the job from the queue and the log is published to a separate tube

271 named with the job id of the original job.

272   The worker sends the output data back to the requestor via an API call. If the

273 worker should die or communication between the worker and the queue goes down

274 the delay time will eventually run out and the job will go back into ready state for

275 another worker to pick up and process.

276

277 **PRP platform.** CDeep3M-preview is running as a Kubernetes Pod on the Pacific

278 Research Platform (PRP) and configured to use gpu-pods equipped with GPUs with

279 least 11GB vRAM. PRP is spanning over 20 universities and institutions, all

280 connected by dedicated optical light-paths at speeds of 10-100gb/s. A list of currently

281 available PRP resources can be found at: https://ucsd-

282 prp.gitlab.io/userdocs/running/gpu-pods/.

283

284 **CDeep3M2 data augmentations.** Data augmentations are used to avoid overfitting to

285 training data and intended to increase generalizability of trained models. To facilitate

286 regulating which data augmentation strategies are used we chose to separate

287 augmentation strategies into following three categories:

288 *Primary augmentations* refer to augmentations that only change the image

289 orientations, such as data rotations and flipping in x, y and z directions. Since those

290 leave the data unaltered they are always performed, on each training dataset, to

291 generate 16 variations of the same data.

292 *Secondary augmentations* are data augmentations which alter the noise level, the

293 brightness or the contrast of the images. Secondary augmentations are performed if a

294 setting between 1 (weak) to 10 (strong) is chosen. Following operations are used in

295 secondary augmentations: increase and lowering of the image contrast; sharpening

296 and gaussian blurring of the image stack; total variation using the Chambolle and/or

297 Bregman filter to denoise the image; adding of random binary gaussian noise;

298 normalization of the image by performing histogram equalization; random skewing of

299 the image stack to four different directions (upper left, upper right, lower left, lower

300 right) while maintaining the image's aspect ratio; elastic distortions across the image

301 stack in which the first and the last images of the stack are distorted by a randomly

302 generated gaussian vector field while the images in between are distorted by the

303 interpolated field in between the two.

304 *Tertiary Augmentation* are performed if a setting between 1 (weak) to 10 (strong) is

305 chosen. During the tertiary augmentations images are resized, according to a strength

306 selected by the end user (values 1-10). Data is resized using upscaling alternating with

307 downscaling, to broaden the networks capabilities to recognize the same object at a

308 different pixelsize.

309

Code availability

311 All code is open access. The CDeep3M2 Docker container can be pulled directly from

312 Docker-Hub at https://hub.docker.com/r/ncmir/cdeep3m or simply running 'docker pull

313 ncmir/cdeep3m:latest'. The Google Colab Jupyter Notebooks with graphical user-

314 interface (GUI) are available on GitHub at https://github.com/haberlmatt/cdeep3m-

315 colab. The CDeep3M2 AWS cloudformation template is available here. CDeep3M2

316 source code is available on GitHub https://github.com/CRBS/cdeep3m2. The

317 singularity image is available at: http://cellimagelibrary.org/cdeep3m/singularity.