

conv-paint: an easy to train interactive pixel classifier

Lucien Hinderling^{1,2,✉}, Guillaume Witz³, Ana Stojilkovic³, Maciej Dobrzyński¹, Mykhailo Vladymyrov³, Roman Schwob¹, Joël Frei¹, Benjamin Grädel^{1,2}, and Olivier Pertz^{1,✉}

¹Institute of Cell Biology, University of Bern, Baltzerstrasse 4, 3012 Bern, Switzerland

²Graduate School for Cellular and Biomedical Sciences, University of Bern, Baltzerstrasse 4, 3012 Bern, Switzerland

³Data Science Lab, University of Bern, Sidlerstrasse 5, 3012 Bern, Switzerland

We develop conv-paint, a fast and interactive pixel classification tool for multi dimensional images. A graphical user interface and extensive documentation with tutorials make the software accessible to use, while advanced users can easily script the software from python. As a napari plugin, conv-paint can be combined with other plugins into processing pipelines, which we demonstrate on two case studies with different data modalities.

image analysis | pixel classification | multi-dimensional data

Correspondence: lucien.hinderling@unibe.ch, olivier.pertz@unibe.ch

1 Introduction

At the start of many bio-image analysis pipelines lies a segmentation step. While deep learning methods offer high classification accuracy, they demand extensive data sets, annotated ground truth, and specific hardware for training which are all not always feasible to acquire. Even foundational models trained on a wider array of different data types generalize to new applications without retraining, are often not straight forward to use in basic research on novel and highly unique image data. Classical machine learning approaches with small interactive models, that learn from sparse annotations have thus been shown to be very useful (Ilastik, Trainable Weka, Qupath) [1–3]. Conv-paint builds on this work, striking a balance between labelling speed and accuracy. Trained in seconds, it enables an iterative workflow, allowing researchers to preview segmentation results and guide the model with new annotations. In this work we combine the power of deep learning models with quick to train machine learning models: We extract local image features using convolution filters from a pretrained convolutional neural network (CNN). Instead of training the neural network, we pass the features into random forrest classifier (fig. 1A). Compared to classical handcrafted filterbanks used in similar approaches, the filters of the CNN are computationally optimised to extract image features useful for image classification [4], while classical filterbanks are known to extract redundant information [5]. See fig. 3 for a visual comparison. This approach of using pretrained deep learning networks as feature extractors to use with classic machine learning algorithms has shown promising results on various image analysis tasks [6–8]. In this project we aim to integrate these insights, adapt them to the task of pixel classification, with the goal of creating a tool that is simple to use for

researchers without coding experience or expertise in machine learning. We implement a graphical user interface for napari (fig. 1C)[9], making conv-paint directly interoperable with a vast range of different plugins available on the napari hub, enabling complex image analysis workflows from within the same software environment. Additionally we provide APIs for more experienced users to easily control conv-paint programatically from python. The code is open source (BSD-3) and available as Python package, installable from the napari plugin hub¹, and runs on all common operating systems and computers with standard consumer hardware (good performance even without GPU). In this article we demonstrate two exemplary workflows combining conv-paint with other napari plugins to extract biological information from multi-dimensional imaging data. Installation instructions, documentation and video tutorials are available on the project website².

52 Implementation details

Image features are extracted using convolutional filters from a CNN with the VGG16 [10] architecture, pretrained on the image net dataset [11] on the task of general object classification. The intuition is that the first layers of a CNN capture local image features such as edges, texture or correlation between color channels (see fig. 3B) that generalize well to data outside the domain the CNN was trained on. We can transfer these already learnt functions without retraining the network. The features extracted with the convolutional filters are then passed into a random forest network, which can be trained much quicker than deep learning approaches, enabling the iterative approach of labelling, checking the results and adding annotations to steer the segmentation results. As the convolutional filters of the VGG16 network are 3x3 pixels, they only capture very local image information and longer range context is missed. We found that passing downsampled versions of the input image through the network and, then upscaling the resulting features to the original size and concatenating them with the features from the unscaled output (creating a featurized image pyramid), leads to better segmentation performance. The same holds for extracting, upscaling and appending features from deeper levels of the CNN, that

¹<https://www.napari-hub.org/plugins/napari-convpaint>

²<https://guiwitz.github.io/napari-convpaint/book/Landing.html>

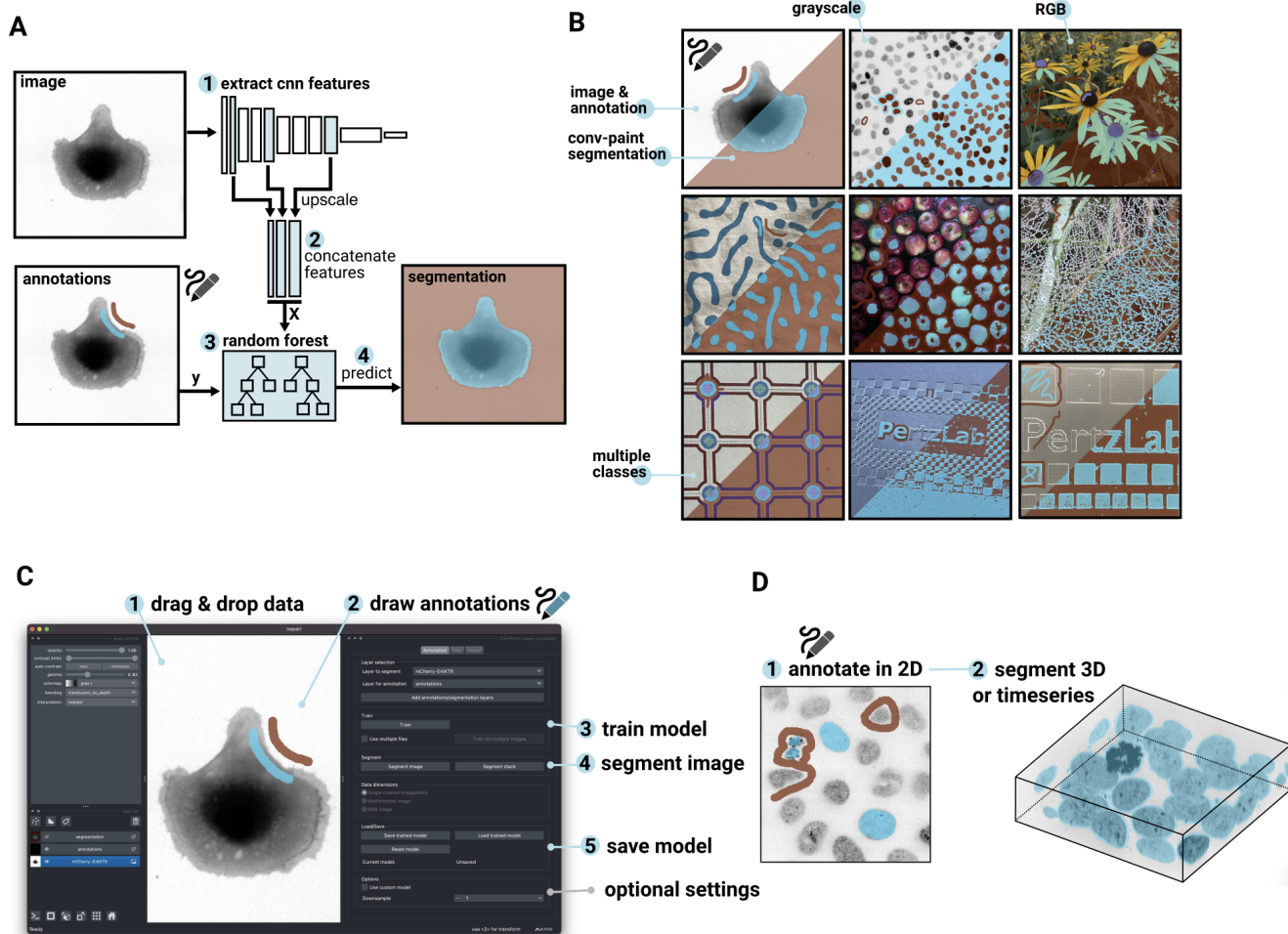


Fig. 1. A Scheme of the conv-paint architecture. Image features are extracted using convolutional filters, upsampled to the size of the original image, then concatenated. A random forest model is trained using sparse annotations to predict the class for each pixel from the extracted features. **B** conv-paint is capable of segmenting diverse types of images. Top left triangle shows the original image and user annotations, bottom right triangle shows the resulting segmentation. conv-paint automatically adapts to different numbers of input color channels and annotated classes. **C** Screenshot of the user interface implemented in napari. 1) napari natively supports a wide range of input formats, and many plugins are available to load unsupported data types. 2) Annotations can be drawn on top of the image using the napari labels layer. 3) The model can be trained with a single click. 4) Another click segments the image and displays the results in a labels layer. 5) The model can be saved for use on new images or to reproduce the segmentation at a later point. 6) Optional settings (custom models, normalisation) and training on multiple input images can be enabled for advanced users and are accessed in optional tabs. **D** Labels for conv-paint are drawn onto 2D slices, the classifier can then also be applied to stacks (3D images or timeseries).

hold more semantically rich information processed and pooled from the first layers. We provide a standard configuration of parameters that lead to a good tradeoff between speed and accuracy of segmentation, and a graphical user interface to chose combinations of VGG16 layers or even other CNN architectures and input scalings for advanced users. The source code is available on github³.

Working with multichannel data

New imaging technologies are capable of imaging an increasing number of biomarkers in the same sample (e.g. multiple fluorophores, cytometry by time of flight (CyTOF), imaging mass cytometry (IMC)), providing a wealth of information and new biological insights, but also posing additional challenges in data analysis, especially for interactive data exploration. Humans can only process a limited

number of color channels simultaneously, motivating the creation of computer assisted tools that can leverage this information. Conv-paint is capable of detecting correlations between an arbitrary number of color channels, as illustrated with artificial data in fig. 4. Fig. 2A-E shows an exemplary usecase on an imaging mass cytometry (ICM) dataset [12]. The data is loaded and browsed interactively with napari-ilmc [13]. Instead of exporting the data and running pixel classification in the software Ilastik as described previously [14], we can run pixel classification directly in napari with conv-paint. Two different structures of interest in the tissue are marked, and a classifier is trained on all 43 channels to find similar regions in the sample. Using the segmentation we calculate the log2 fold change of signal intensity, to interactively identify which markers are differentially expressed in one or the other class.

³<https://github.com/guiwitz/napari-convpaint>

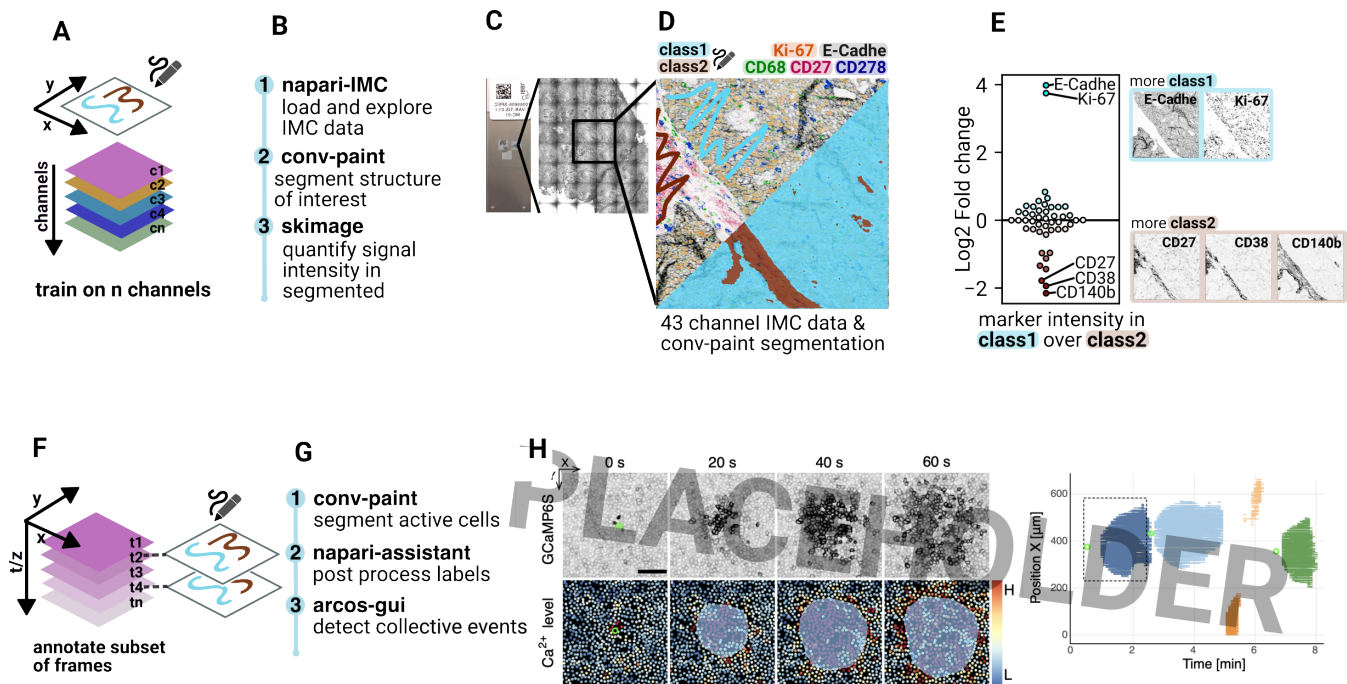


Fig. 2. **A** Labels are drawn in 2D, but the classification algorithm is capable of processing an arbitrary number of input color channels. **B** This is demonstrated on a workflow with multichannel IMC data. **C** IMC can be interactively explored at multiple resolutions with the napari-imc plugin. **D** On a preview of the highly dimensional data showing 5 of 43 channels of the dataset, a structure of interest is labeled, and conv-paint then uses intensity and structure information in all 43 channels to predict areas similar to the labels in the rest of the image (or on other images). **E** These class labels can be used interactively explore the dataset and get statistics for biological insight, e.g. plotting the composition of different structures: Here we calculate the fold change of signal intensity for all channels between the two classes and find markers that are differently expressed in the segmented regions. **F** conv-paint also natively supports 3D and time series data, where a classifier can be trained on multiple frames or z-slices. **G** This workflow shows combining conv-paint with the napari plugin arcoss-gui to detect and quantify collective signalling events in MDCK **H** Active cells cells are segmented with conv-paint. **I** arcoss-gui uses the binary activity map to find collective signalling events.

Working with time series and z-stacks

Biological phenomena often require additional imaging dimensions to the basic 2D image for adequate understanding, be it a time dimension to observe dynamics or an additional space dimension for 3D imaging. The image viewer napari has native support for fast and interactive visualization of timeseries and z-stacks. We leverage these capabilities to offer support for training on multiple frames or z-slices, and segmentation of the whole stack with a single click. napari's visualization capabilities facilitate interactive exploration, quality control, and iterative improvement of the results in 3D or time. The conv-paint output can then directly be used by other plugins in the napari ecosystem. In fig. 2F-I, we show detection of calcium signalling waves in a MDCK epithelial mono layer. Active cells and inactive cells are labeled with respective classes, and segmentation with conv-paint is applied to all frames of the movie. The binary output of active and inactive cells is then processed with ARCOS, a computational method to detect and quantify collective signalling [15].

Discussion

Conv-paint is a fast, lightweight and easy to use tool for pixel classification. It is capable of segmenting diverse types of images, and can be combined with other napari plugins to create complex image analysis workflows. We hope that conv-paint will be useful to a wide range of re-

searchers, and that the open source nature of the project will enable further development and adaptation to new use cases.

ACKNOWLEDGEMENTS

This work has been supported by the Chan Zuckerberg Initiative (CZI) grant NP2-0000000095 to LH and OP, Uniscientia fellowship (TODO) to OP, and Schweizerischer Nationalfonds (SNF) grant 310030_185376 to OP.

AUTHOR CONTRIBUTIONS

TODO LH, GW, AS, MD, MV, RS, and BG contributed to the development of the software. JF and LH acquired data. Figures were created by LH and MD. LH and OP wrote the manuscript. All authors read and approved the final manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Bibliography

- [1] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I. Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. Ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, September 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0582-9.
- [2] Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Kevin W Eliceiri, Johannes Schindelin, Albert Cardona, and H Sebastian Seung. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 33(15):2424–2426, 03 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx180.
- [3] Peter Bankhead, Maurice B Loughrey, José A Fernández, Yvonne Dombrowski, Darragh G McArt, Philip D Dunne, Stephen McQuaid, Ronan T Gray, Liam J Murray, Helen G Coleman, Jacqueline A James, Manuel Salto-Tellez, and Peter W Hamilton. QuPath: Open source software for digital pathology image analysis. *Sci. Rep.*, 7(1), December 2017.
- [4] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning convolutional feature hierarchies for visual recognition. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, page 1090–1098, Red Hook, NY, USA, 2010. Curran Associates Inc.
- [5] Francesco Bianconi and Antonio Fernández. Evaluation of the effects of gabor filter parameters on texture classification. *Pattern Recognition*, 40(12):3325–3335, 2007. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2007.04.023.

- [6] Wilson Bakasa and Serestina Viriri. Vgg16 feature extractor with extreme gradient boost classifier for pancreas cancer prediction. *Journal of Imaging*, 9(7):138, July 2023. ISSN 2313-433X. doi: 10.3390/jimaging9070138.
- [7] Deepak Kumar Swain. Combining vgg16 with random forest and capsule network for detecting multiple myeloma., December 2021. Available at <https://norma.ncirl.ie/5227/1/deepakkumarswain.pdf>.
- [8] Allou Koffi Franck Kouassi, Lin Pan, Xiao Wang, Zhangheng Wang, Alvin K. Mulashani, Paulo James, Mbarouk Shaame, Altaf Hussain, Hadi Hussain, and Edwin E. Nyakilla. Identification of karst cavities from 2d seismic wave impedance images based on gradient-boosting decision trees algorithms (gbd): Case of ordovician fracture-vuggy carbonate reservoir, tahe oilfield, tarim basin, china. *Energies*, 16(2), 2023. ISSN 1996-1073. doi: 10.3390/en16020643.
- [9] Jannis Ahlers, Daniel Althviz Moré, Oren Amsalem, Ashley Anderson, Grzegorz Bokota, Peter Boone, Jordão Bragantini, Genevieve Buckley, Alistair Burt, Matthias Bussonnier, Ahmet Can Solak, Clément Caporal, Draga Doncila Pop, Kira Evans, Jeremy Freeman, Lorenzo Gaiñas, Christoph Gohlke, Kabilar Gunalan, Hagai Har-Gil, Mark Harfouche, Kyle I. S. Harrington, Volker Hilsenstein, Katherine Hutchings, Talley Lambert, Jessy Lauer, Gregor Lichtner, Ziyang Liu, Lucy Liu, Alan Lowe, Luca Marconato, Sean Martin, Abigail McGovern, Lukasz Migas, Nadalyn Miller, Hector Muñoz, Jan-Hendrik Müller, Christopher Nauroth-Kreß, Juan Nunez-Iglesias, Constantin Pape, Kim Pevey, Gonzalo Peña-Castellanos, Andrea Pierré, Jaime Rodríguez-Guerra, David Ross, Loïc Royer, Craig T. Russell, Gabriel Selzer, Paul Smith, Peter Sobolewski, Konstantin Sofiuk, Nicholas Sofroniew, David Stansby, Andrew Sweet, Wouter-Michiel Vierdag, Pam Wadhwa, Melissa Weber Mendonça, Jonas Windhager, Philip Winston, and Kevin Yamauchi. napari: a multi-dimensional image viewer for python, 2023.
- [10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [12] Nils Eling and Jonas Windhager. Example imaging mass cytometry raw data. February 2022. doi: 10.5281/zenodo.5949116.
- [13] Jonas Windhager, Bernd Bodenmiller, and Nils Eling. An end-to-end workflow for multiplexed image processing and analysis. *bioRxiv*, 2021. doi: 10.1101/2021.11.12.468357.
- [14] Jonas Windhager, Vito Riccardo Tomaso Zanotelli, Daniel Schulz, Lasse Meyer, Michelle Daniel, Bernd Bodenmiller, and Nils Eling. An end-to-end workflow for multiplexed image processing and analysis. *Nature Protocols*, 18(11):3565–3613, October 2023. ISSN 1750-2799. doi: 10.1038/s41596-023-00881-0.
- [15] Paolo Armando Gagliardi, Benjamin Grädel, Marc-Antoine Jacques, Lucien Hinderling, Pascal Ender, Andrew R. Cohen, Gerald Kastberger, Olivier Pertz, and Maciej Dobrzyński. Automatic detection of spatio-temporal signaling patterns in cell collectives. *Journal of Cell Biology*, 222(10), July 2023. ISSN 1540-8140. doi: 10.1083/jcb.202207048.
- doi.org/10.5281/zenodo.5555575. The data for case study 2 is available on TODO

Code availability

The code is open source (BSD-3) and available on github [www.github.com/guiwitz/napari-convpaint](https://github.com/guiwitz/napari-convpaint), on PiPY <https://pypi.org/project/napari-convpaint/>, or can be installed from the napari plugin hub <https://www.napari-hub.org/plugins/napari-convpaint>.

Supplementary figures

Supplementary information

Methods

Case study 1: IMC multichannel data. IMC data was loaded using the napari-imc plugin [13] and napari was used to visualize the data. conv-paint was trained on one FOV (Fig 2 shows Patient 01, Panorama 02, Position 1-1). Skimage was used to extract the per channel statistics for the segmented regions. The log2 fold change of signal intensity was calculated using numpy and plotted with matplotlib. The code to recreate the figures is available on github TODO.

Case study 2: MDCK calcium waves. The movies showing calcium signaling waves are acquired from MDCK epithelial cells that stably express GCaMP6S – a GFP-based intracellular calcium sensor (imaging data courtesy of Yasuyuki Fujita). The movies were loaded into napari and segmented using conv-paint. The resulting binary masks were processed with ARCOS [15] to detect and quantify collective signalling events. The code to recreate the figures is available on github TODO.

Data availability

The data for case study 1 has been previously published [12] and is available on the Zenodo repository <https://>

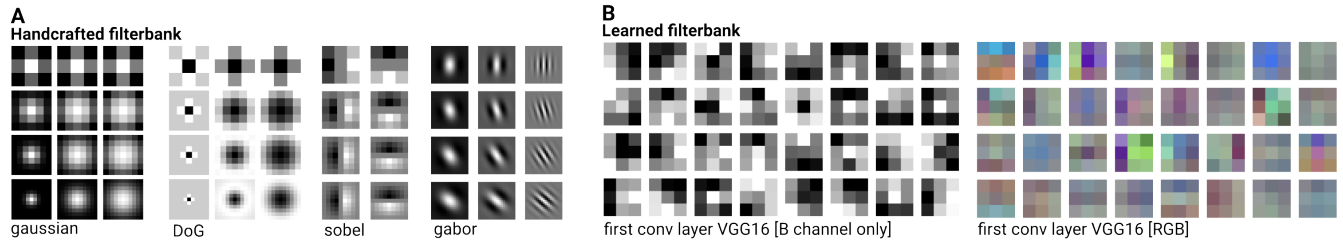


Fig. 3. **A** Filters classically used in handcrafted filterbanks. Here we show examples for filter kernels with different parameters for gaussians, difference of gaussians (DoG), sobel and gabor. **B** Filters extracted from the first convolution layer of a VGG16 network. The filters have shape 3x3x3, and are thus intrinsically capable of extracting correlations between color channels in RGB images.

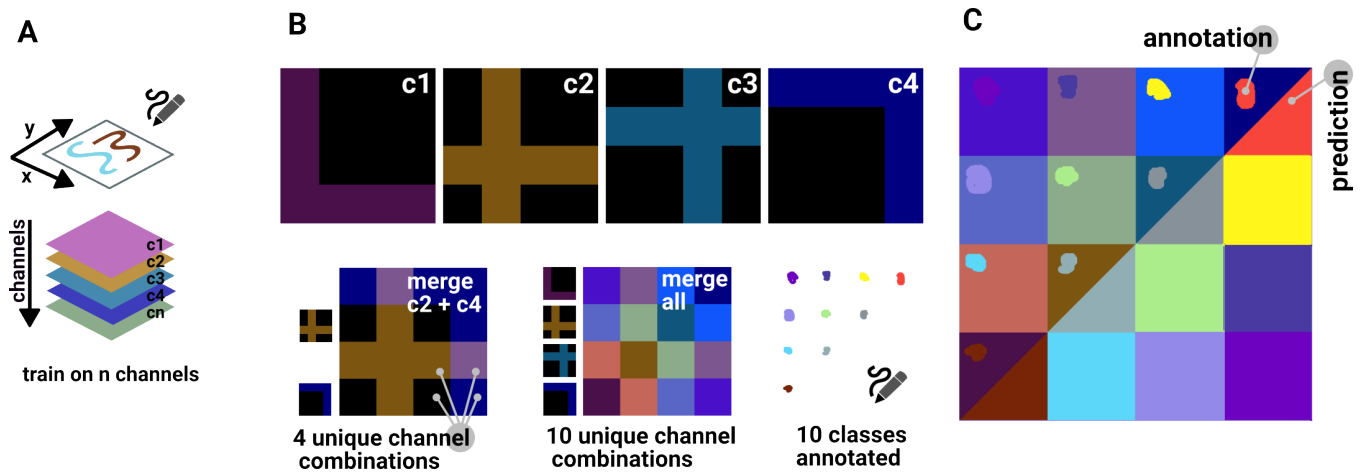


Fig. 4. **A** Labels are drawn in 2D, but the classification algorithm is capable of processing an arbitrary number of input color channels. **B** This is demonstrated on an artificial image with four channels, which when merged lead to 10 unique color combinations. These 10 colors are labeled with 10 class labels, that can only be reconstructed if the algorithm takes into account the interplay of the different channels. **C** Conv-paint correctly predicts the correct class label for pixels that were not labeled, with minor artefacts on boundaries between squares.