Supplementary Text

DETAILED DESCRIPTION OF THE MULTI-PLANAR U-NET MODEL

Overview

The MPUnet is an open-source segmentation system that learns to segment medical 3D scans with high performance without human expert involvement. The system has been previously validated on more than 10 non-knee segmentation tasks and obtained a top-5 position in the 2018 Medical Segmentation Decathlon despite being hyper-parameter search free – in the sense that the default settings have proven to give good results on variable medical image segmentation tasks – and data efficient (1). The MPUnet is simple and only requires a set of expert annotated scans to train on. The required number of training scans vary greatly depending on desired performance and task. For simpler tasks such as the segmentation of a healthy anatomical body, few (e.g., 20-30) samples may be sufficient, whereas hundreds of samples may be required to properly segment complex variable targets such as tumors.

Deep learning knowledge is not required to use the system and the open-source code includes example scripts for training and applying the MPUnet. It uses a fixed model topology and hyper-parameter setting, eliminating the need for computationally intensive model selection searches, which are often impractical in clinical research. The system performs segmentation of high quality, which is facilitated by a multi-planar training approach, which applies a statistically efficient 2D model simultaneously across multiple planes spanning the volume in order to utilize most of the information contained in volumetric scans (1, 2).

Segmentation Model

The MPUnet framework relies on a single 2D fully convolutional network (FCN) model fit to image slices sampled isotropically along multiple viewing planes through the image volume. Using 2D FCNs for 3D segmentation is a way to limit the number of trainable parameters, which increases the statistical efficiency and is therefore a common approach in medical image analysis where labeled data is scarce. The FCN model is based on the relatively new, but already prevalent U-net (3) architecture. Compared to the original version, the total parameter count is approximately 2 times larger (to a total of ~60 million learnable parameters) because the number of filters across all layers is increased. Nearest neighbor up-sampling operations

(4) are used in the up-sampling path followed by a standard convolution layer. Batch normalization (5) is employed between all convolution and up-convolution blocks.

Multi-Planar Training and Prediction

The 2D segmentation model is trained in a *multi-planar* fashion. The model is fit to 2D image slices sampled (isotropically) across a set of $V = 6$ planes angled to each other. Perslev et al. (2) found that segmentation performance increased with V, and $V = 6$ was chosen to balance computational load with performance. Bilinear and nearest-neighbor interpolation of the input scan is used to generate the image- and label map slices, respectively. During optimization, images from all these planes are fed to the (a priori plane-agnostic) model without additional information about the corresponding image plane, see **Figure 1**.

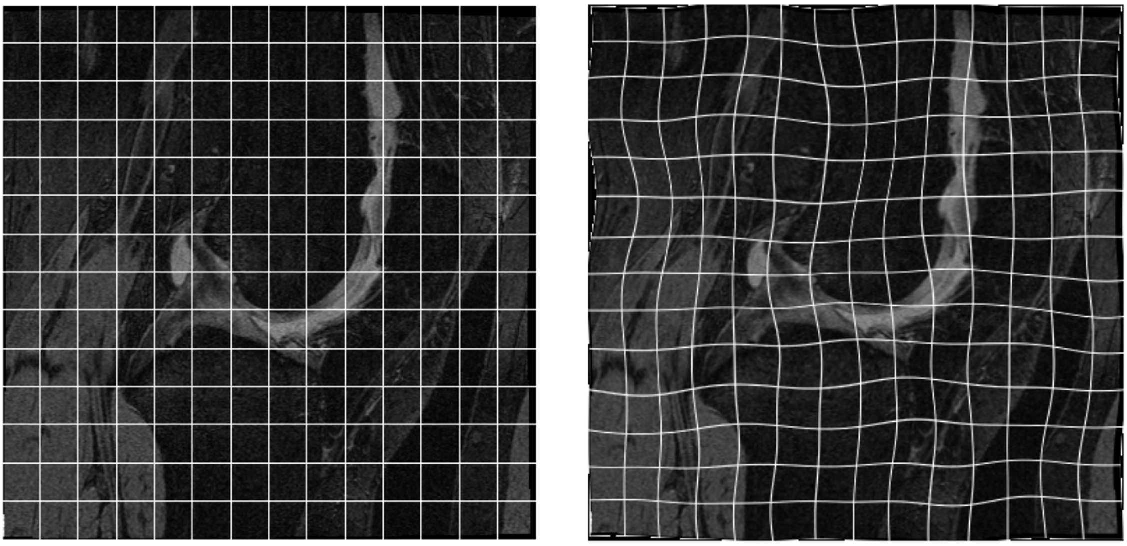
This training setup forces the model to learn to segment the medical target of interest as seen from multiple *views*. The amount of training data increases V times, but the different views of a volume are not independent of each other. In this way the extension of the training data resembles data augmentation (5), see below. When segmenting a new scan, the model first predicts along each plane in the isotropic scanner space creating a set of V full segmentation volumes for each input scan. The segmentation volumes are mapped to the voxel space by assigning to each voxel the nearest predicted label (as measured in isotropic space) from each of the V volumes. A final prediction score is computed for each voxel x as a weighted sum $z(x)_k = \sum_{n=1}^{V} W_{n,k} \cdot p_{n,x,k} + \beta_k$ where $z(x)_k$ is the score of class k at voxel x, $p_{n,x,k}$ is the probability predicted from view n that voxel x belongs to class k, and $W_{n,k}$ and β_k are parameters learned to maximize the overall segmentation performance. The scores $z(x)_k$ are mapped to probabilities using the softmax function, and the parameters $W_{n,k}$ and β_k are chosen to minimize the cross-entropy loss on the validation dataset.

The single neural network model plays the role of V experts in an ensemble method, which can be viewed as a form of *test time augmentation*. When combining the V segmentations, each class from each view has an individual trainable weight ($W_{n,k}$), which allows the model to exploit that some parts of the volume may be more easily segmented in one plane compared to others. The overall tendency of the ensemble to predict a given class can be tuned by the learned per-class bias parameters β_k. The approach is illustrated in **Figure 2**.

Augmentation and Class Balancing

To increase the robustness of the segmentation model, the MPUnet employs data augmentation on top of the multi-view images (6, 7). In general, data augmentation refers to artificially boosting the size of the training dataset by applying various transformations to the real training data (in this sense, the multi-planar approach can be interpreted as augmentation through rotating the input volume). Specifically, during training, images are sampled on-the-fly from across the cohort, and with a probability of $1/3$ random elastic deformations are applied to the sampled image (8), see **Supplementary Figure 1**. The elastic transformation is non-linear and may produce anatomically implausible images. However, the augmentation often leads to significantly improved performance by forcing the model to learn a more general representation of the structures to segment. This ultimately restricts the model from overfitting to the training data. Augmented images are weighted by a factor of $1/3$ when computing the loss to ensure that the optimization considers primarily true images. Both the probability with which augmentation is applied and the loss function weighting factor were selected based on experience and were not systematically varied.

The MPUnet uses the Adam optimizer (9) to minimize the standard cross-entropy loss function with no added regularization or class balancing terms. The ues of regularization was found unnecessary, as the combination of an efficient 2D model, multi-planar training and augmentation tends to reduce the overfitting to a negligible level. Instead of explicitly accounting for class imbalance in the loss function, each sampled batch is forced to contain a fraction of images displaying non-background compartments, and, when possible, it is further required that each batch contains a set of images that in union display all possible compartments. This scheme was empirically to effectively handle the class imbalance problem across many tasks.



Supplementary Figure 1: Visualization of the effect of random elastic deformations. (a) Input image. (b) Augmented image.

Hyperparameters

Both the deep learning model topology and the optimization parameters of the MPUnet were entirely fixed (based on Perslev et al. (2)) or estimated from the training data from non-intensive computations. For instance, the size of the sampled images input to MPUnet is heuristically selected based on the sizes and voxel resolutions of the input MRI volumes across the training set (see Perslev et al. (2) for details). Consequently, the framework will start training the model immediately without the need for running extensive hyperparameter cross-validation experiments, which are typically needed when a segmentation model is applied to a new dataset, task, or scanner modality. In particular, the complexity (i.e., number of free parameters) of deep neural networks usually requires tuning to ensure that the model neither over- nor underfits to the problem at hand. Based on extensive empirical evidence this is usually not required for the MPUnet, which is unlikely to overfit. The minimal hyperparameter optimization in MPUnet does not only make it easy to use, but it also reduces the risk of unintentional model overfitting.

REFERENCES

1. Antonelli M, Reinke A, Bakas S, et al.: The Medical Segmentation Decathlon. *arXiv* 2021; Preprint.

2. Perslev M, Dam EB, Pai A, Igel C: One Network to Segment Them All: A General, Lightweight System for Accurate 3D Medical Image Segmentation. In *Med Image Comput Comput Interv*. *Volume 11765*. Springer; 2019:30–38.

3. Ronneberger O, Fischer P, Brox T: U-net: Convolutional networks for biomedical image segmentation. In *Med Image Comput Comput Interv*. *Volume 9351*; 2015:234–241.

4. Odena A, Dumoulin V, Olah C: Deconvolution and Checkerboard Artifacts. *Distill* 2017; 1.

5. Ioffe S, Szegedy C: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd Int Conf Mach Learn*. *Volume 1*; 2015:448–456.

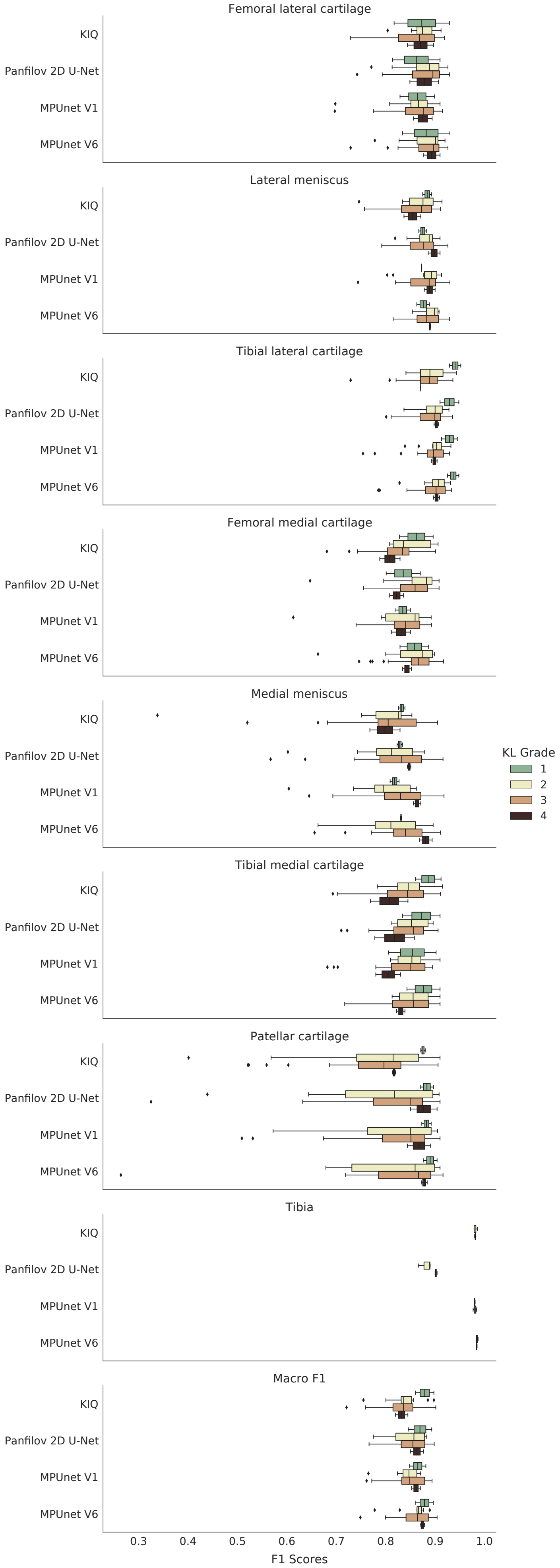
6. Shorten C, Khoshgoftaar TM: A survey on Image Data Augmentation for Deep Learning. *J Big Data* 2019.

7. Goodfellow I, Bengio Y, Courville A: Regularization for Deep Learning. In *Deep Learn*. MIT Press; 2016:224–270.

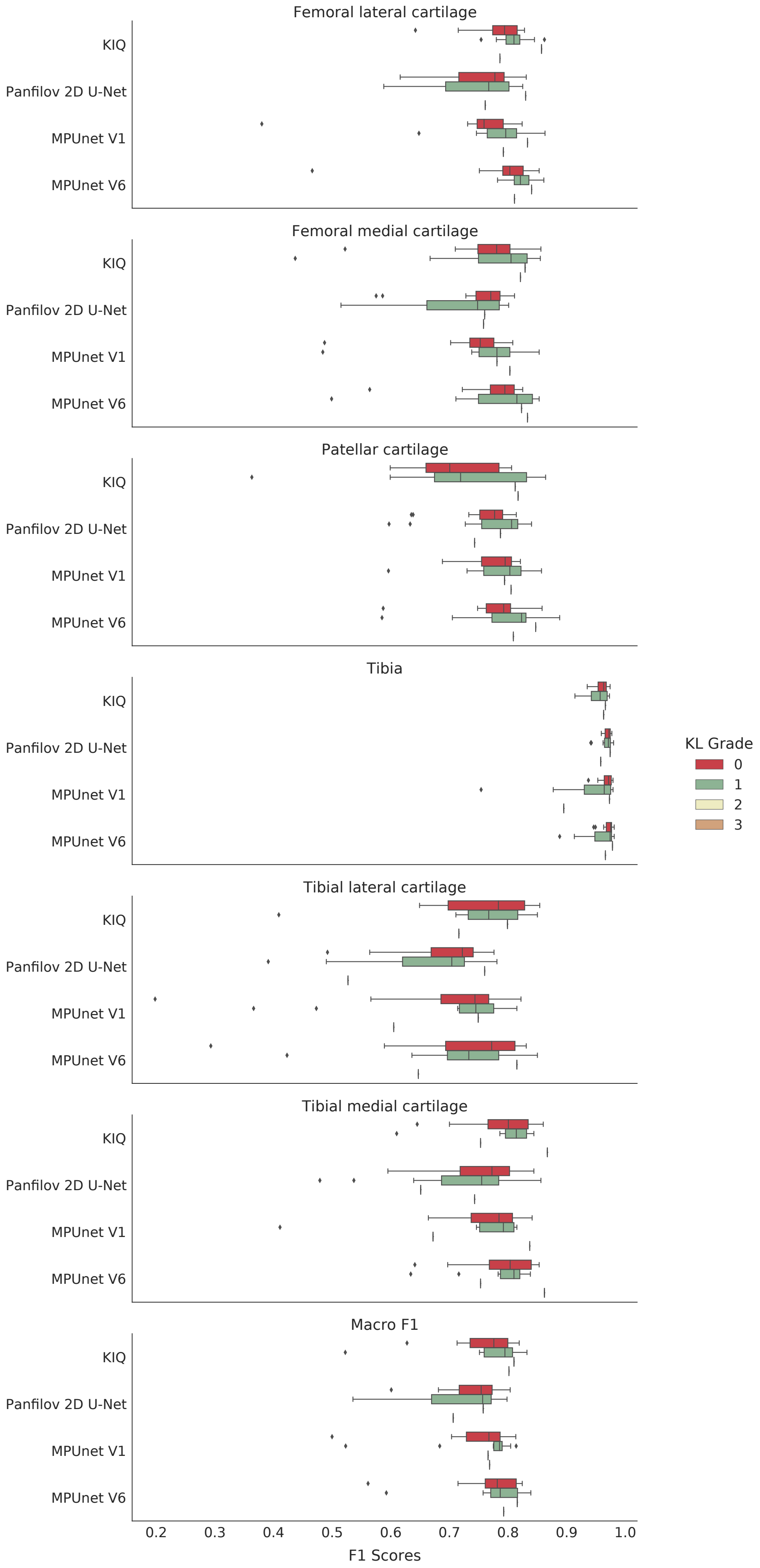
8. Simard PY, Steinkraus D, Platt JC: Best practices for convolutional neural networks applied to visual document analysis. In *Proc Int Conf Doc Anal Recognit*. Institute of Electrical and Electronics Engineers; 2003:958–963.

9. Kingma DP, Ba JL: Adam: A method for stochastic optimization. In *3rd Int Conf Learn Represent ICLR 2015*; 2015:arXiv:1412.6980.

Supplementary Figure 2: Box-plots showing the distribution of dice scores for the MPUnet, KIQ and the Panfilov 2D U-Net on the OAI dataset grouped according to the KL-grade score of the individual MRIs.



Supplementary Figure 3: Box-plots showing the distribution of dice scores for the MPUnet, KIQ and the Panfilov 2D U-Net on the PROOF dataset grouped according to the KL-grade score of the individual MRIs.


Femoral lateral cartilage
Femoral medial cartilage
Patellar cartilage
Tibia
Tibial lateral cartilage
Tibial medial cartilage
Macro F1
KIQ
Panfilov 2D U-Net
MPUnet V1
MPUnet V6
KL Grade
F1 Scores