

- Contact details
 - Satoshi Kondo (satoshi.kondo.jp@gmail.com)
- Name of the algorithm.
 - VGG fine-tuning
- Give the overall structure of the algorithm and briefly describe each step in this structure.

We use pre-trained convolutional neural network (CNN), which is trained with ImageNet dataset. We use VGG16 [1] as our base network. We change the last fully connected layer, i.e., fc8, to have twenty-one output neurons which are likelihood of presenting the tools in each image. The last two fully connected layers, i.e., fc7 and fc8 are fine-tuned using the training dataset.

The training method is as follows. The input images are resized to 288 by 288 pixels. The mini-batch size is 48 and we run 80 epochs in the training. We use stochastic gradient descent for the optimization. The hyper-parameters in the optimization are that the learning rate is 0.001 and the momentum is 0.9. We use sigmoid cross entropy for the loss function. We apply weights for the loss function that one for non-present (0) and thirty for present (1). `ImageData` augmentation is not applied in the training and the test phases.

[1] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556.

- Does your system use additional training data? If so, describe the characteristics of that training data.
 - No
- Do predictions for a given frame solely rely on that frame? Or do they also rely on previous frames? on following frames?
 - Predictions are made frame by frame.
- Give computation times (number of frames processed per second).
 -
- If the algorithm has been tested on other datasets, you could consider including those results.
 - No