

Estimation of Breast Cancer Density in Hematoxilin-Eosin Slides Using a Convolution Deep Neural Network

This project is aimed at a gross estimation of the percentage of breast cancer cells in a Hematoxillin-Eosin(H&E) slide prepared from samples breast cancer patients. In any patient diagnosed with cancer and requiring a surgical procedure, the tumor is sent to the pathology lab and will be analyzed via microscopy for staging and qualification of the type of cancer and how aggressive the cancer is. This evaluation focuses on the preparation of H&E slides from the tumor which will examine the tumor under the microscope.

The goal of the project is to estimate the cancer cellularity for assessment of the tumor burden in these patients. The main task is to develop an automated method for analyzing histology patches extracted from whole slide images and assign a score reflecting cancer cellularity in each. This is done by expert pathologists who have to identify and quantify the cancer burden. Furthermore, reproducibility of cancer cellularity scores is a concern in current practice.

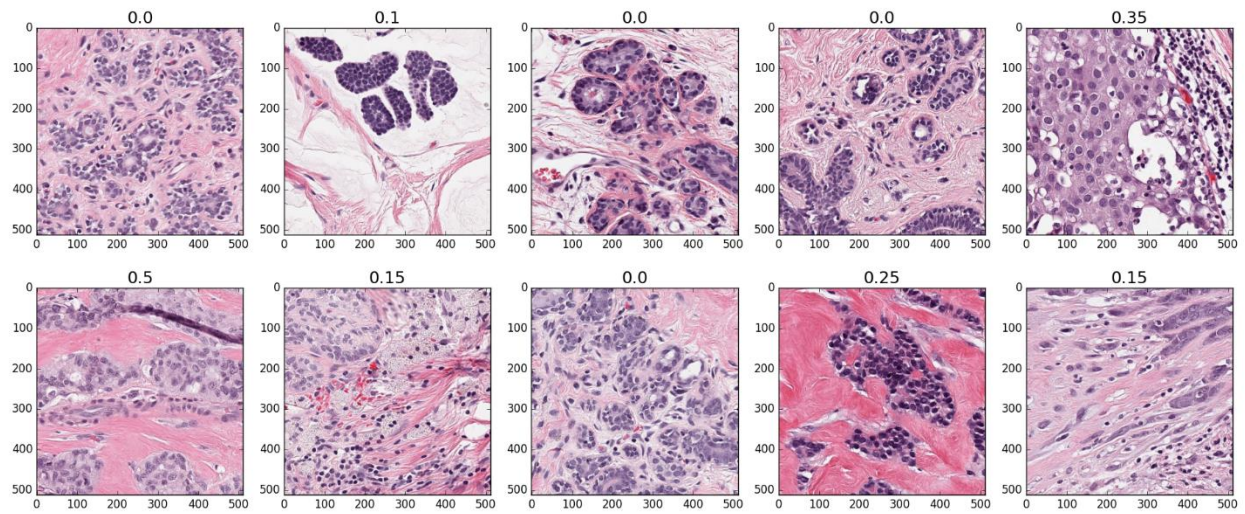
This report will summarize the data as well as the environment and experimental setup for achieving this goal.

Data

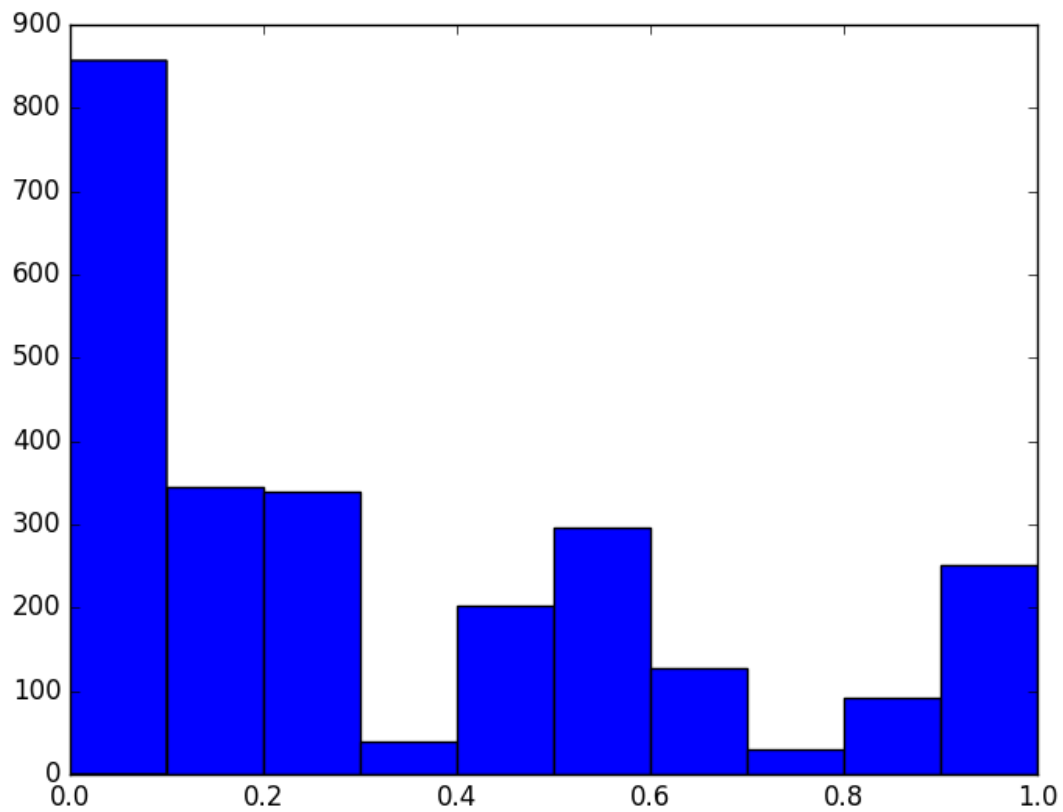
The data was taken from the international society for optics and photonics (SPIE), along with the American Association of Physicists in Medicine (AAPM), and the National Cancer Institute (NCI), "Grand Challenge" for 2019. This challenge focused on breast cancer pathology quantification and automation.

The data is a set of 2570 patches from 64 H&E slides presented in 512*512 size, in a tif format. The patches are graded by expert pathologists in terms of their cellularity from 0 to 1 and these represent the label for each image. The labels are presented in a separate csv file.

An exploratory data analysis was carried. A random sample of 10 images was displayed along with their labels to have an idea of the data.



To understand the distribution of the labels a similar code was run to have the histogram of the labels displayed and see the distribution;



Some imbalance of the labels is noted. This is a left skewed distribution favoring the absence of cancer cells within an area. This may create a slight bias favoring the lower cellularity scores.

Training Algorithm

PyTorch was used as the framework to implement a deep convolutional neural network to conduct a regression analysis to train and then predict the cellularity of the data. The initial network was a shallow network with two convolution networks as well as two pooling networks, two ReLUs and a final Linear layer as the output layer.

A hand crafted 9 layer residual network was then created using a similar structure to above. A densely connected torchvision model was then used (densenet 121) was also tried. Finally a handcrafted dilated CNN model was created with 13 layers to account for the lower size of the dataset and used.

Given that most of these networks have been published with classification problems, the dataset was modified to allow for a classification network to be run as well. The data was divided into a 0/1 labeling with 0 indicating cellularity of less than 50% and 1 indicating above that. A Keras framework was used to study a subset of the data and see if the model would work.

Experimental Setup

The framework will be PyTorch. Mini batching will be used. Given the intense requirement of memory for the project, initially a mini batch size of 64 was used. This was subsequently decreased depending on the model used given memory was not bountiful.

The loss functions chosen were an MSE loss and KLDiv. These are the functions which are well described in the literature to be best suited for a regression problem. Each model was run with each of the loss function and the performance compared.

The network evaluation has to be with an RMSE calculation given that this is a regression problem. The predicted and actual labels were compared with that. An attempt was made to use a predicted probability analysis as well between the concordant and discordant pairs, however, it was unsuccessful.

For all the models a constant learning rate was used at 0.001. No comparison was done between different learning rates and different models.

Results

A summary of the RMSE obtained with each model under different circumstances is summarized below;

Model	Loss function	Epochs	RMSE
CNN - 8	MSE	5	0.23
CNN - 8	KLDiv	5	0.21
CNN - 8	MSE	10	0.01
Resnet - 9	MSE	5	0.01
Resnet - 9	KLDiv	5	0.47
DilatedCNN-13	MSE	5	0.2
Densenet - 121	Cross entropy	5	0.26

The loss functions were plotted as well and are summarized below in the different pics

Fig3:Loss function for CNN8-MSE

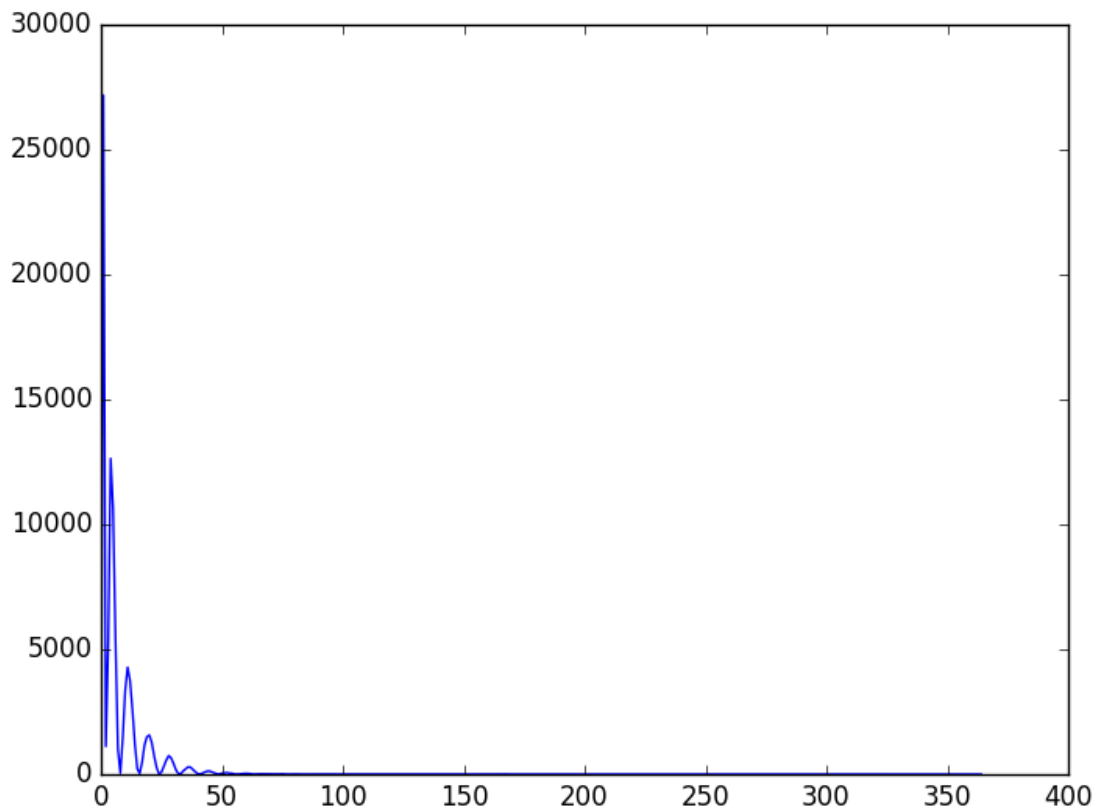


Fig4: Loss function for CNN8 – KLDiv

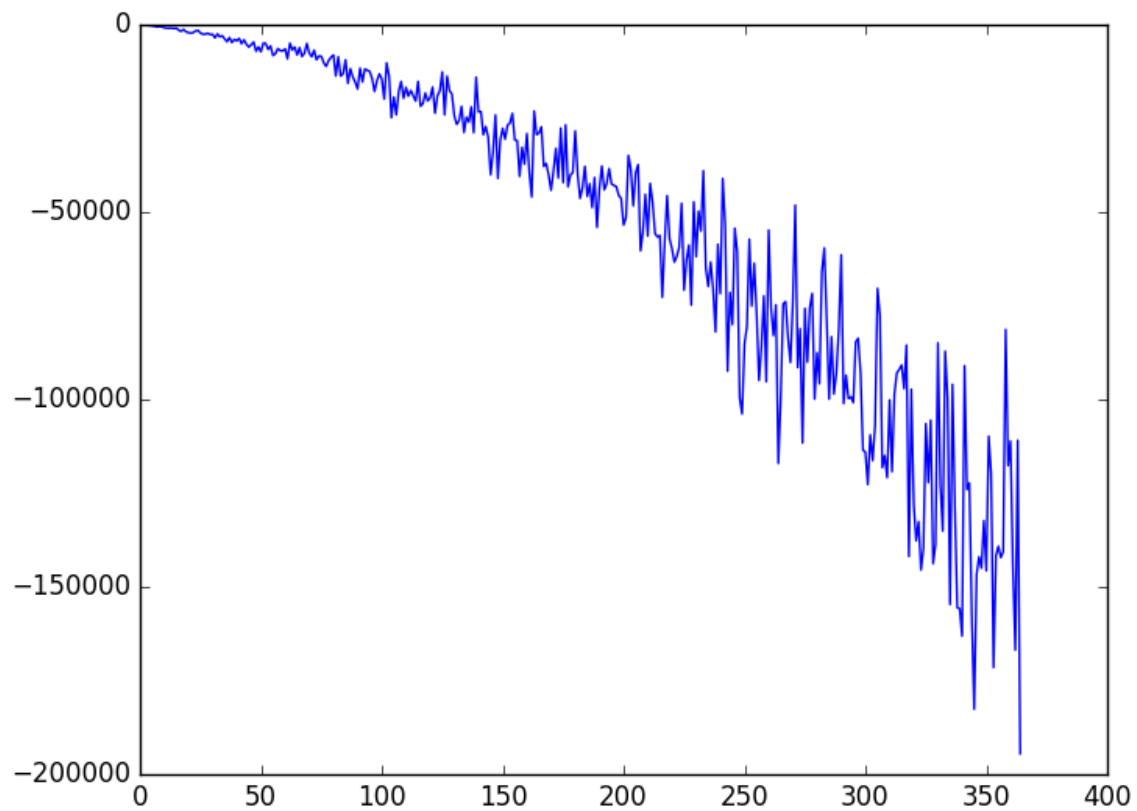


Fig5: Loss Function for Resnet9 - MSE

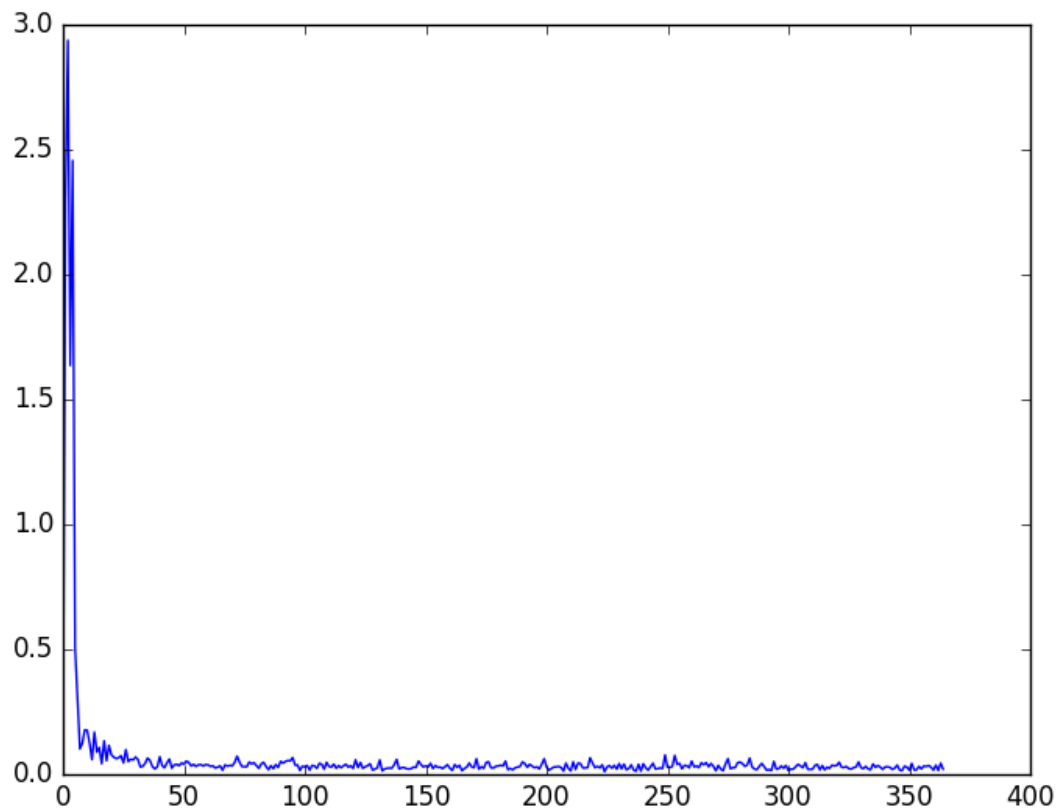


Fig6: Loss Function for Resnet9 – KLDiv

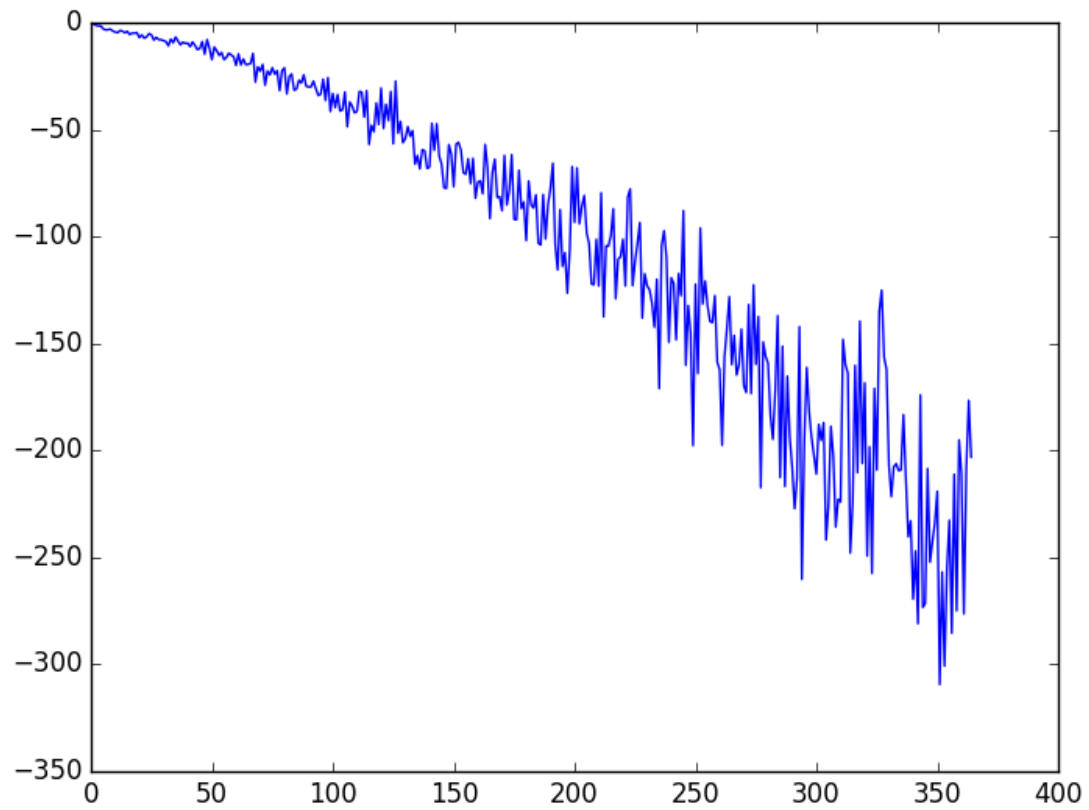


Fig7: Loss function for DilatedCNN – MSE

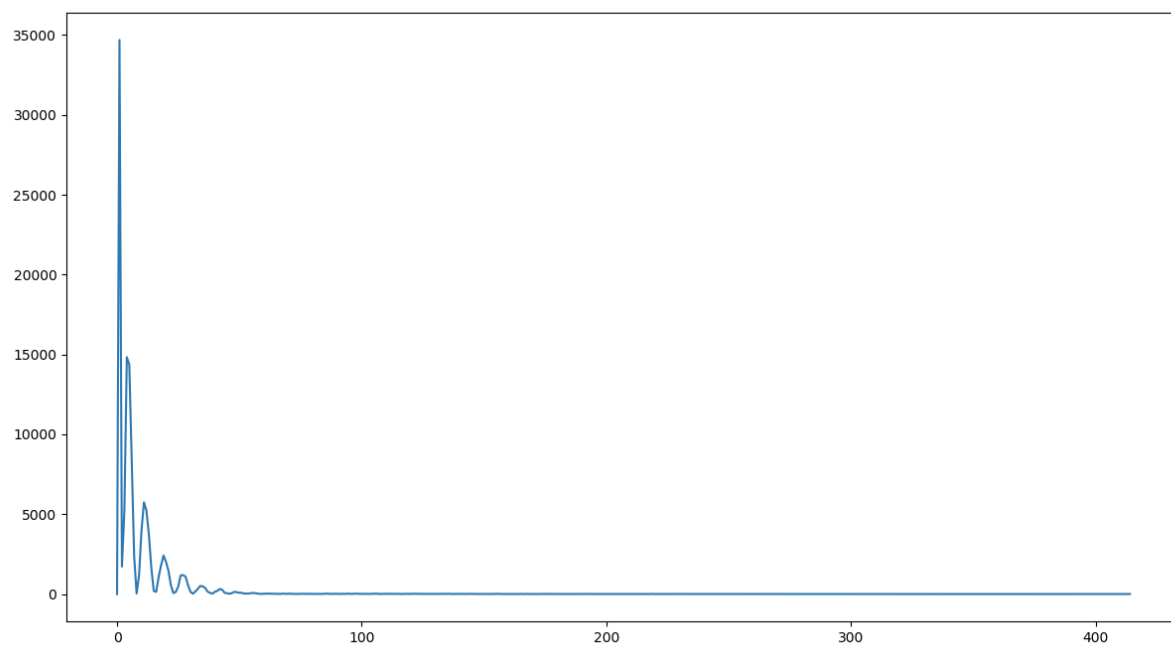
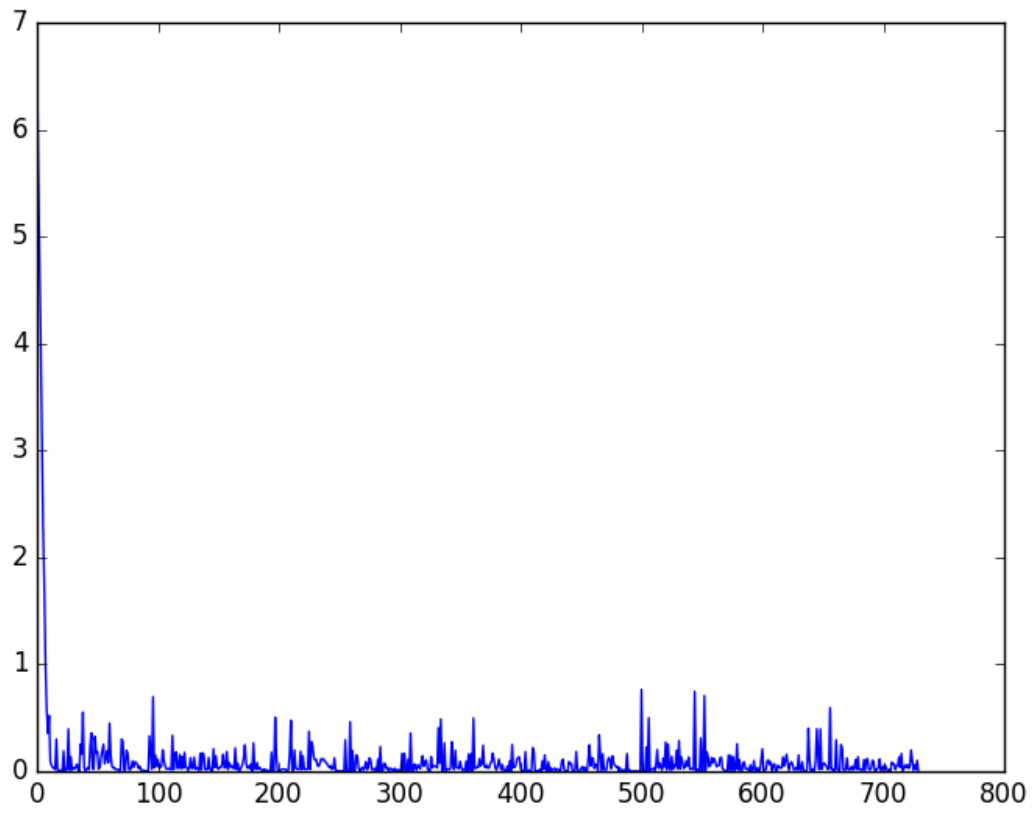


Fig8: Loss function for densenet121



References

http://spiechallenges.cloudapp.net/competitions/14#learn_the_details-overview

<https://www.kaggle.com/sermakarevich/complete-handcrafted-pipeline-in-pytorch-resnet9/data?scriptVersionId=10694803>

<https://www.kaggle.com/artgor/simple-eda-and-model-in-pytorch>

<https://www.kaggle.com/soumya044/histopathologic-cancer-detection>

<https://www.kaggle.com/c/histopathologic-cancer-detection/kernels>

<https://www.kaggle.com/eiffelwong1/basic-cnn-for-cancer-detection-pytorch>

<http://cs231n.stanford.edu/reports/2017/pdfs/203.pdf>

Google.com, stackskills