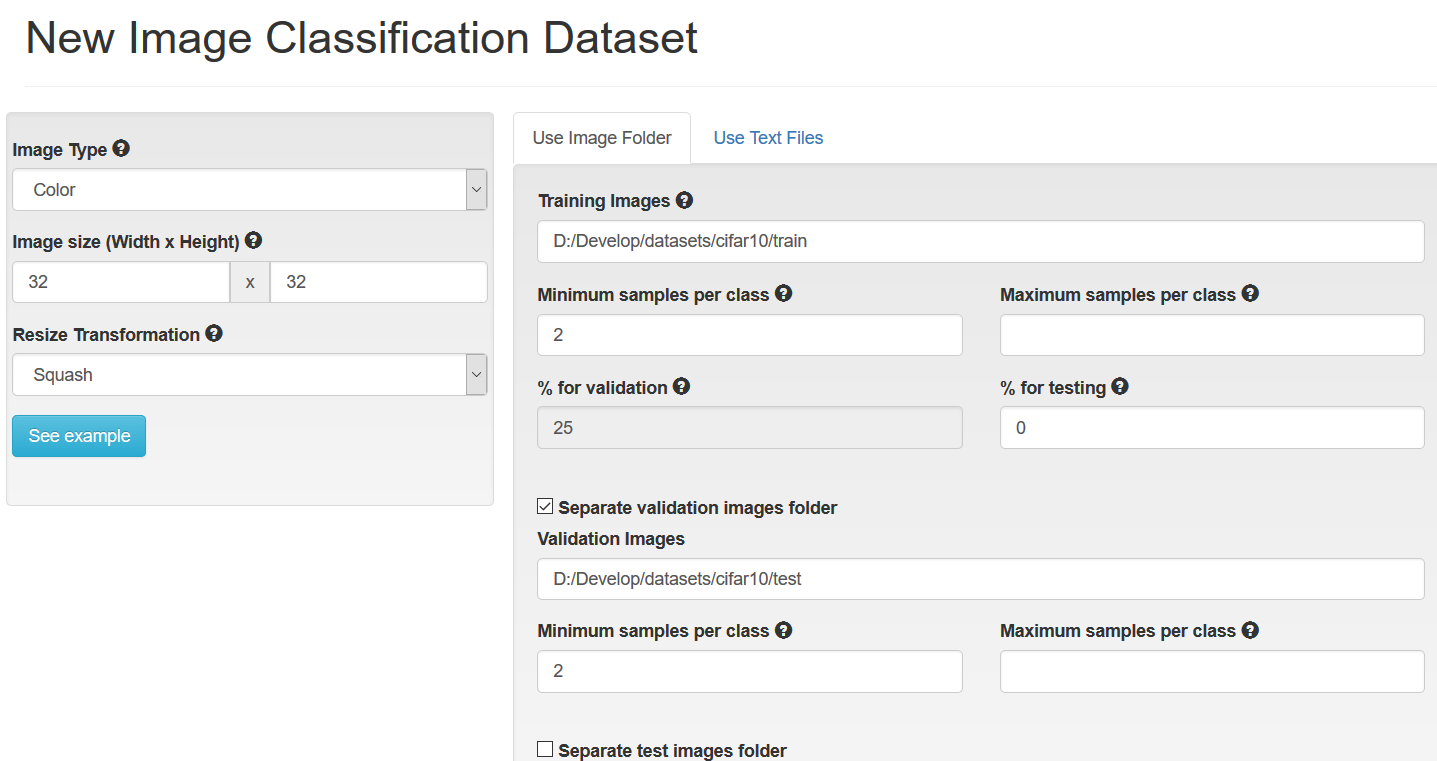
В этом задании вы научитесь работать с Caffe и NVIDIA DIGITS и научитесь тренировать нейронные сети для задач классификации.

**Задание**: обучите нейронную сеть для классификации изображений датасета CIFAR10.

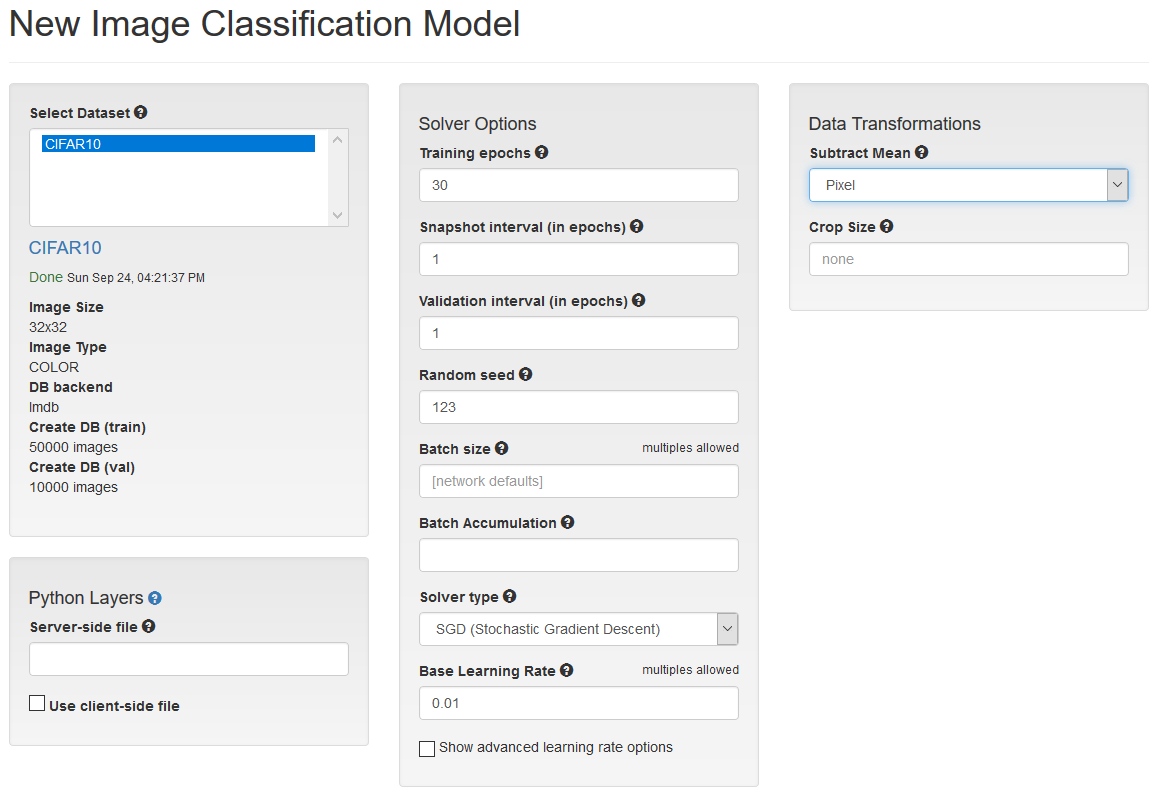
**Цель**: получите точность на тестовом наборе более 70%. На данный момент рекордная точность составляет около 96%, дерзайте.

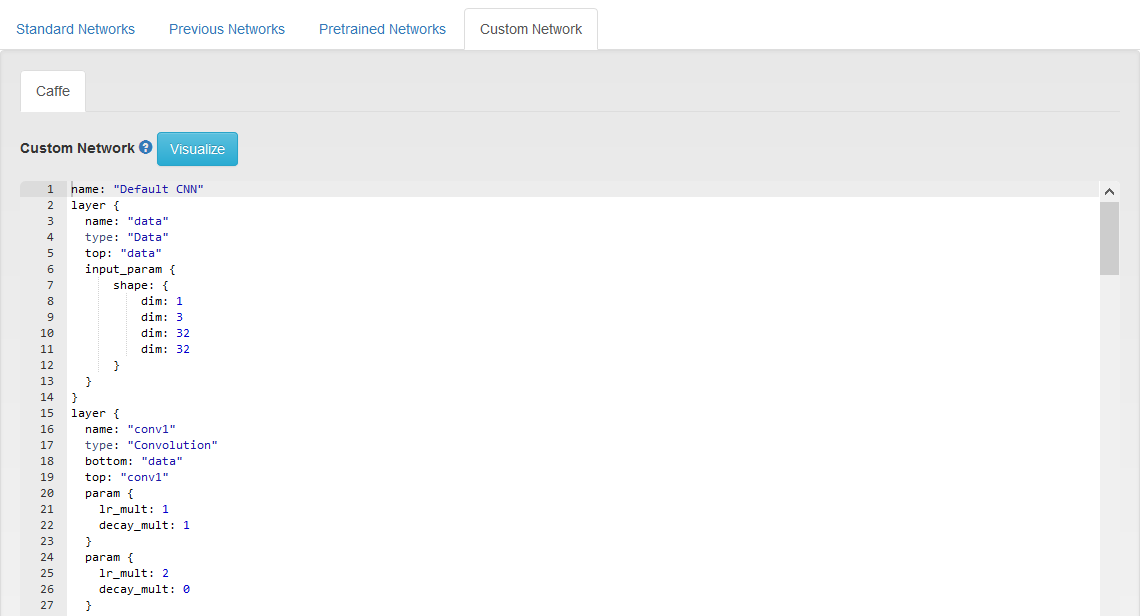
Для выполнения задания используйте caffe + NVIDIA DIGITS.

1. Зайдите в NVIDIA DIGITS  
   <http://localhost:5000/>
2. Создайте классификационный датасет CIFAR10  
   CIFAR10 лежит здесь  
   <https://github.com/dyashuni/HSE_2017/tree/master/Datasets/cifar10>  
   Datasets –> Images –> Classification  
   Примеры настроек



1. Настройте параметры обучения, создайте нейронную сеть и запустите тренировку  
   Models –> Images –> Classification  
   Примеры настроек





Пример текста в prototxt формате, который вы можете поместить во вкладку Custom Network в NVIDIA DIGITS

*name: "Default CNN"*

*layer {*

*name: "data"*

*type: "Data"*

*top: "data"*

*input\_param {*

*shape: {*

*dim: 1*

*dim: 3*

*dim: 32*

*dim: 32*

*}*

*}*

*}*

*<Ваша нейронная сеть в формате prototxt здесь, входной слой - data, выходной слой - fc8. Вы можете задавать любые названия слоев. Эти названия приведены для примера.>*

*layer {*

*name: "accuracy"*

*type: "Accuracy"*

*bottom: "fc8"*

*bottom: "label"*

*top: "accuracy"*

*include { stage: "val" }*

*}*

*layer {*

*name: "loss"*

*type: "SoftmaxWithLoss"*

*bottom: "fc8"*

*bottom: "label"*

*top: "loss"*

*exclude { stage: "deploy" }*

*}*

*layer {*

*name: "softmax"*

*type: "Softmax"*

*bottom: "fc8"*

*top: "softmax"*

*include { stage: "deploy" }*

*}*

**Рекомендации.**

Архитектуры нейронных сетей, которые стоит попробовать

[conv-relu-pool]xN -> [fully connected]xM -> [softmax]  
[conv-relu-conv-relu-pool]xN -> [fully connected]xM -> [softmax]  
[conv-batchnorm-relu]xN -> [fully connected]xM -> [softmax]

Для визуализации архитектур нейронных сетей используйте netscope  
<https://dgschwend.github.io/netscope/quickstart.html>

Попробуйте разные стратегии апдейта градиентов: SGD, Adam.

Для уменьшения переобучения можете добавить слой dropout, аугментацию изображений (в caffe практически отсутствуют инструменты для аугментации изображений, поэтому используйте слой Data с параметром *transform\_param {mirror: true}* в фазе обучения и *transform\_param {mirror: false}* в фазе валидации)

**Бонус.** Сделайте что-нибудь дополнительное в NVIDIA DIGITS и получите бонусные баллы.

**Форма сдачи задания**: устно расскажите о том, что было сделано, какие результаты получились. Какие архитектуры нейронных сетей сработали хорошо, какие плохо? Сделайте небольшую презентацию, с описанием результатов, или покажите демо на ноутбуке.