Приложение 2

к договору №

от\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| СОГЛАСОВАНО | УТВЕРЖДАЮ |
| Сторона ЗАКАЗЧИКА  Старостин Н. В.  «04» декабря 2018 г. | Сторона ИСПОЛНИТЕЛЯ  Губарев С. Ю.  «04» декабря 2018 г. |

**Руководство оператора**

**на опытно-конструкторскую разработку**

Построение и обучение нейронной сети для идентификации фокусного расстояния по серии изображений.

**(Шифр ПО «**SmartGetDistance**»)**

2018 г

Оглавление

[1. Введение 3](#_Toc529372112)

[1.1. Область применения 3](#_Toc529372113)

[1.2. Краткое описание возможностей 3](#_Toc529372114)

[1.3. Уровень подготовки пользователя 3](#_Toc529372115)

[1.4. Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю 3](#_Toc529372116)

[2. Назначение и условия применения «SmartGetDistance» 3](#_Toc529372117)

[3. Подготовка и работа с программой 4](#_Toc529372118)

[3.1. Состав и содержание дистрибутивного носителя данных 4](#_Toc529372119)

[3.2. Порядок подготовки данных для нейронной сети 4](#_Toc529372120)

[3.3. Порядок работы с нейронной сетью 4](#_Toc529372121)

[Формирование модели 11](#_Toc529372122)

[**Тестирование модели** 18](#_Toc529372123)

[5. Рекомендации по освоению 20](#_Toc529372124)

# 1. Введение

## 1.1. Область применения

Опытно-конструкторская разработка: «Построение и обучение нейронной сети для идентификации фокусного расстояния по серии изображений. ПО «SmartGetDistance»».

## 1.2. Краткое описание возможностей

Объектом автоматизации является процесс построения и обучения нейронной сети для определения расстояния между фокусом оптической системы и поверхностью рельефа по серии изображений. ПО «SmartGetDistance» должно обеспечивать нахождение данного расстояния.

## 1.3. Уровень подготовки пользователя

Пользователь ПО «SmartGetDistance» должен иметь опыт работы с ОС MS Windows (XP/Windows 7/ Windows 8/8.1 /Windows 10), навык работы с командной строкой, а также обладать следующими знаниями:

* знать соответствующую предметную область;
* допустимые параметры входных данных

## 1.4. Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

* руководство системного программиста;
* руководство оператора;
* программа и методика испытаний.

## 2. Назначение и условия применения «SmartGetDistance»

ПО « SmartGetDistance» предназначено для определения расстояния между фокусом оптической системы и поверхностью рельефа по серии изображений, полученного микросъёмкой с малой глубиной резкости.

Результатом решения является ответ нейронной сети по удаленности фокуса от поверхности.

## 3. Подготовка и работа с программой

## 3.1. Состав и содержание дистрибутивного носителя данных

Для работы с «SmartGetDistance» необходимо следующее программное обеспечение:

* Командная строка «cmd.exe» (входит в состав операционной системы Windows);
* Развернутая и настроенная нейронная сеть NVIDIA DIGITS.
* Программка для подготовки файлов.
* Файлы...

Get3DModel.exe устанавливается автоматически при первом обращении пользователя. Архив Get3DModel.rar содержит следующие папки:

* Debug;
* Test\_3x3;
* Testing.bat;

Серия изображений пронумерованы (в имени файла с расширением .png) в порядке возрастания. Также там должен находиться файл конфигурации и в нем должном образом быть указаны характеристики оптики:

## 3.2. Порядок подготовки данных для нейронной сети

Для работы с программой необходимо запустить Installer.bat. После чего произойдет автоматическая разархивация в директорию С:\Get3DModel. Путь для разархивации можно поменять, открыв Installer.bat, при помощи блокнота.

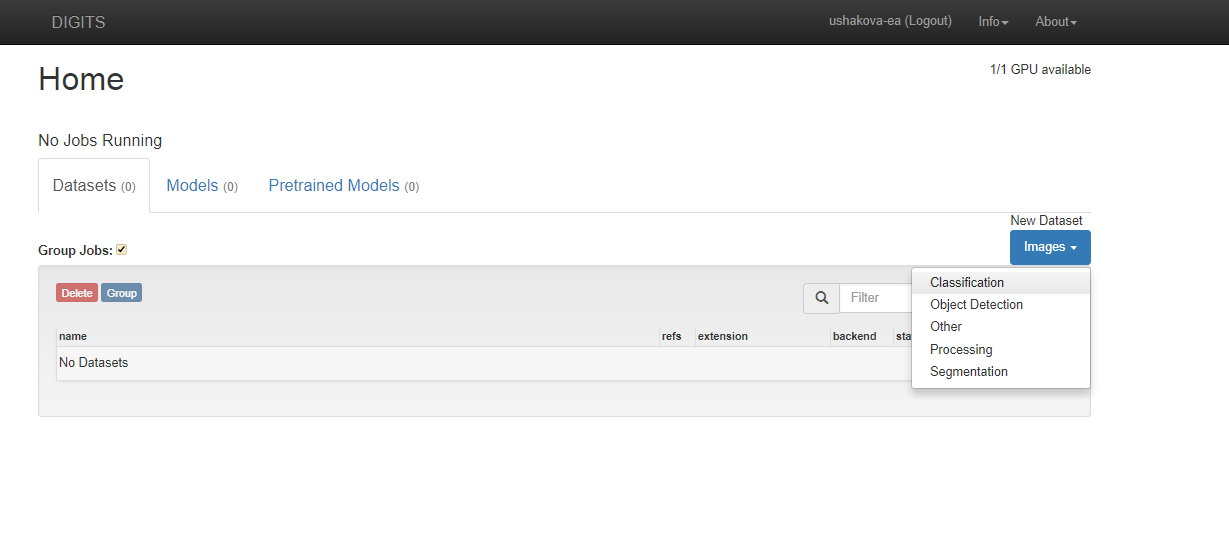
После успешного выполнения, необходимо нажать любую клавишу на клавиатуре для завершения работы bat файла.

Запустите NVIDIA DIGITS из директории проекта NVIDIA DIGITS  
python –m digits –p 5001

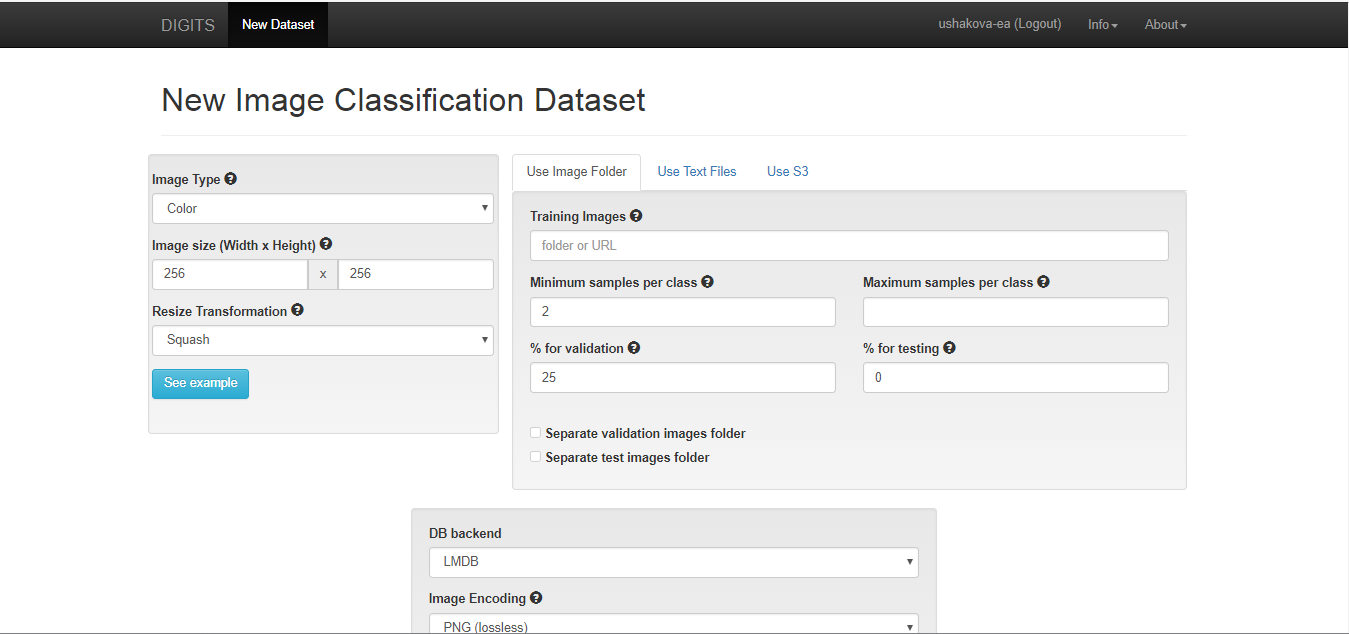
В браузере откройте станицу по адресу localhost::5001

## 3.3. Порядок работы с нейронной сетью

1. Для формирования данных для решения задачи классификации необходимо открыть вкладку Datasets и выбрать Classification.



1. Далее откроется форма, в ней выбрать вкладку Use Image Folder

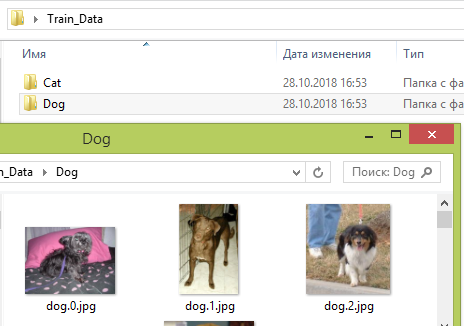


1. В форме присутствуют следующие параметры:

* Image Type – тип изображения (черно-белое или цветное)
* Image size (Width x Height) – Размер изображений (все изображения будут приведены к одномц размеру)
* Resize Transformation – опции для изменения соотношения сторон
* Training Images - папка, содержащая вложенные папки с изображениями.
* Minimum samples per class – минимальное количество изображений для каждого класса (можно оставить пустым – параметр будет проигнорирован)
* Maximum samples per class - максимальное количество изображений для каждого класса (можно оставить пустым – параметр будет проигнорирован)
* % for validation - процент изображений для валидации из обучающего набора
* % for testing – процент тестовых изображений из обучающего набора
* Separate validation images folder – отдельная папка с изображениями для валидации
* Separate test images folder – отдельная папка с тестовыми изображениями
* Image Encoding – тип изображений (все изоражения приводятся к определенному типу)
* Group Name – название группы
* Dataset Name – название набора

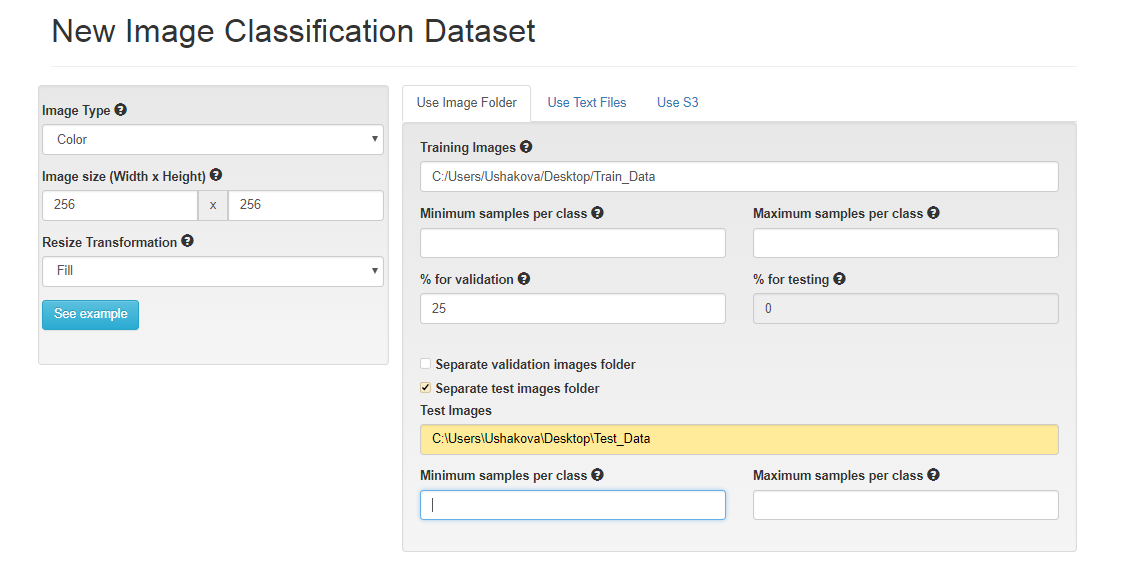
1. Для обучения нейронной сети потребуется DataSet. Для этого необходимо сформировать: трейновую, тестовую и валиационную часть.

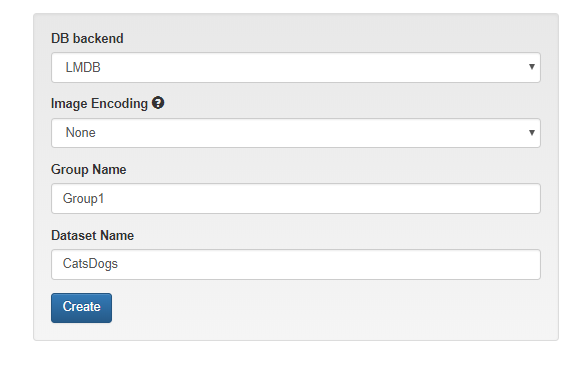
* Изображения для обучения нейронной сети нужно распределить по папкам, название папки – желаемая метка класса (label): Cats и Dogs. В каждой папке находятся изображения членов класса. Полученные папки вложить в одну папку – Train\_Data.



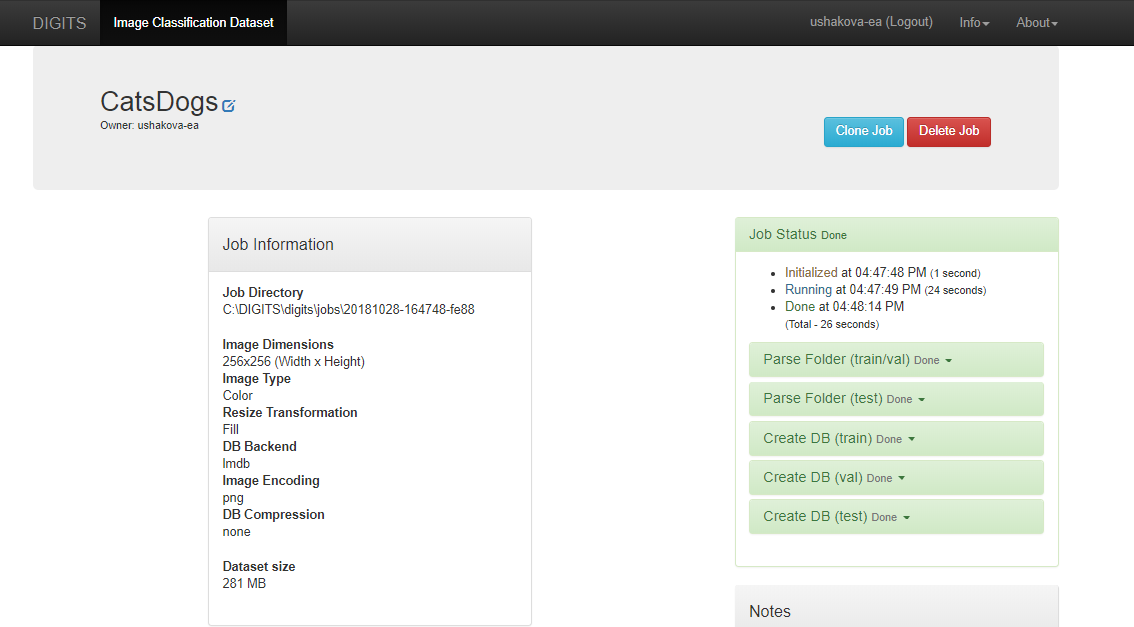
* При необходимости тестовые и валидационные изображения распределяются аналогичным образом в папки – Test\_Data, Val\_Data либо просто указывыется процент от Train\_Data

1. Далее необходимо заполнить поля в форме. Пример заполнения:



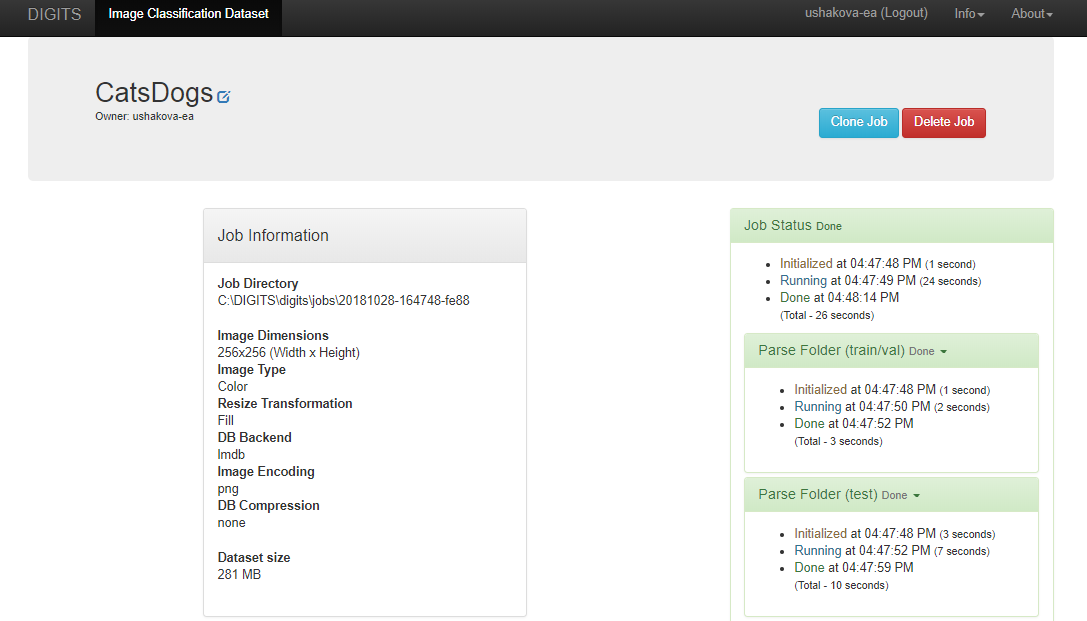


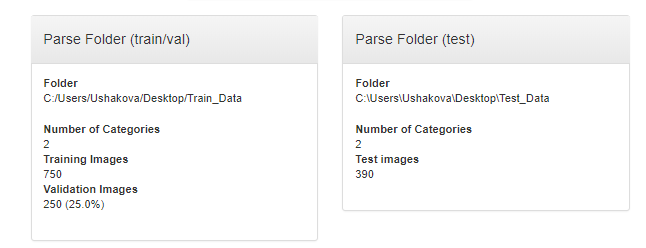
1. Нажать Create → Открывается форма с прогресс барами выполнения каждого из этапов → Об успешном завершении информирует положительный статус.

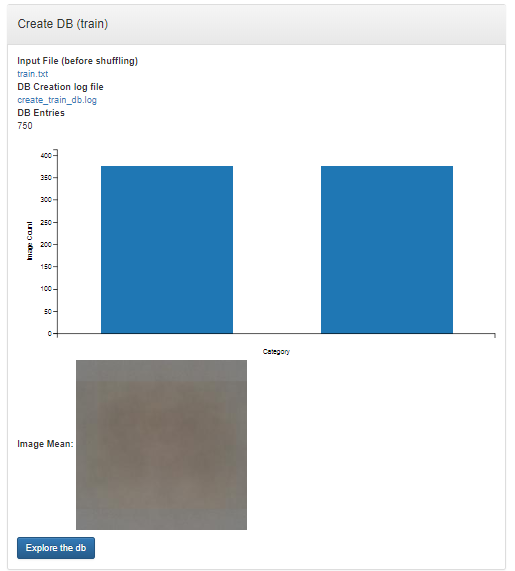


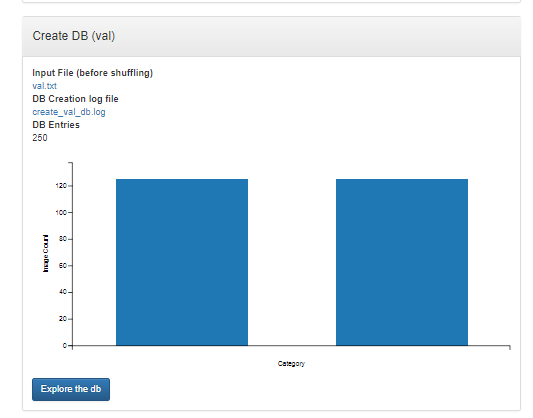
В результате можно увидеть:

* Время, затраченное на парсинг и создание данных.
* Число классов
* Число обучающих, валидационных и тествых изображений
* Среднее изображение классов
* Изображения каждого из наборов

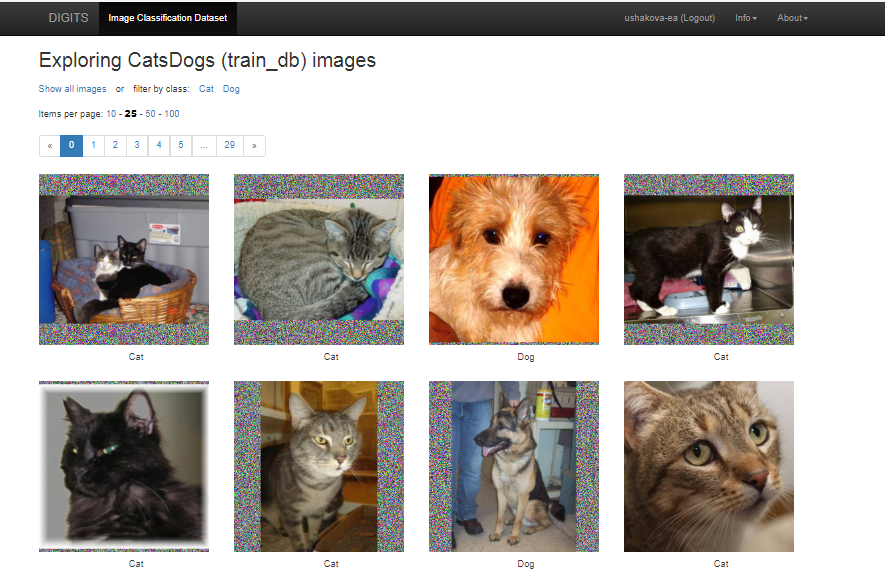




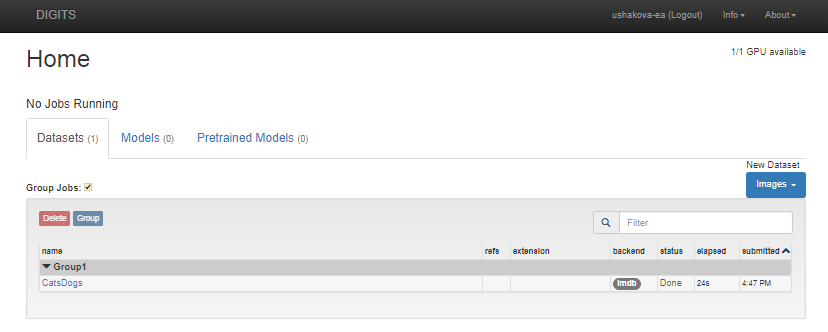






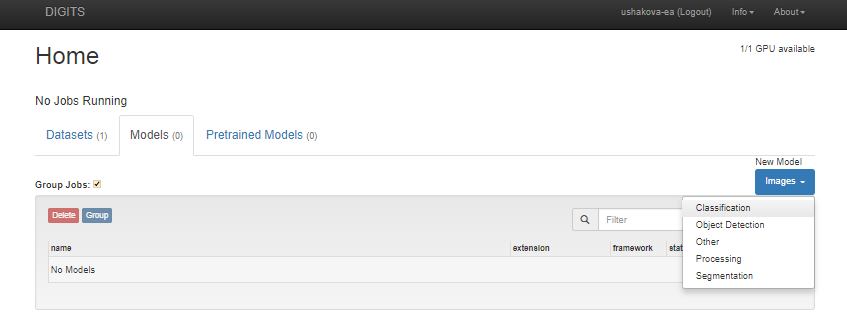


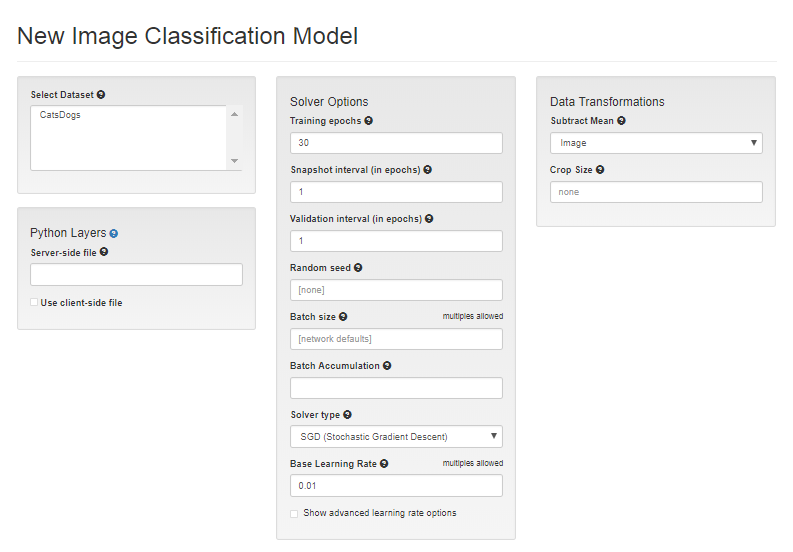
1. Сформированный DataSet будет сохранен во вкладке DataSets

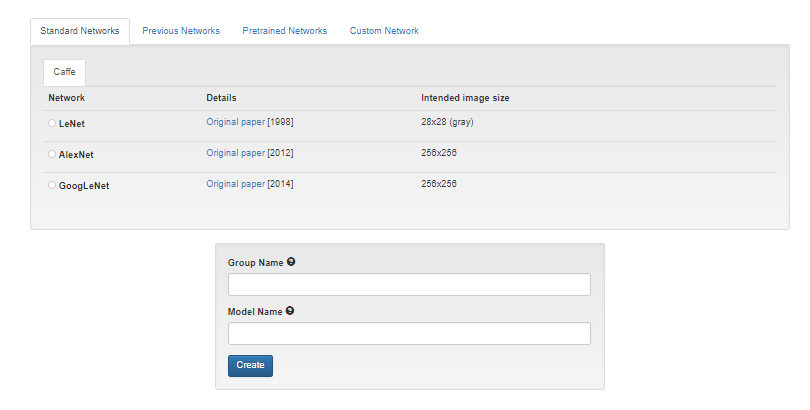


## Формирование модели

1. Для формирования модели для решения задачи классификации необходимо открыть вкладку Models и выбрать Classification.



1. Далее откроется форма



1. В форме присутствуют следующие параметры:

* Select Dataset - набор данных для модели

Слои Python:

* Server-side file – файл на сервере, содержащий определение слоев (можно использовать файл на стороне клиента)

Настройки для решения:

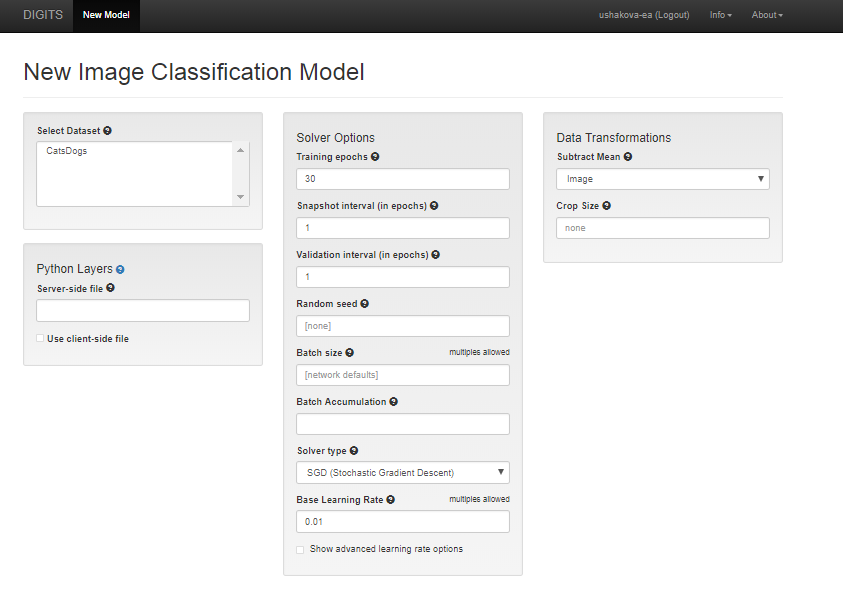
* Training epochs – количество эпох для обучения
* Snapshot interval (in epochs) – количество эпох обучения между съемками
* Validation interval (in epochs) - количество эпох обучения между прохождением через один проход валидации
* Random seed - случайное начальное значение
* Batch size – размер батча (сколько изображений сразу обрабатывается)
* Batch Accumulation – накапление градиентов по нескольким пакетам (полезно, когда вам нужен больший размер пакета для обучения, но он не помещается в память)
* Solver type – стратегия изменения весов нейронной сети
* Base Learning Rate – скорость обучения сети (принимает список через запятую)

Дополнительные параметры скорости обучения:

* Policy – правило
* Step Size - шаг
* Gamma - параметр

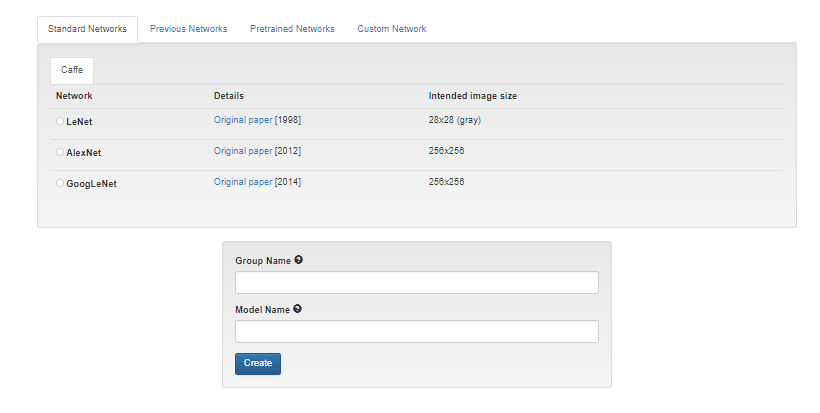
Преобазование данных:

* Subtract Mean – среднее избражение (вычитание среднего изображения из каждого изображения)
* Crop Size – размер кадра

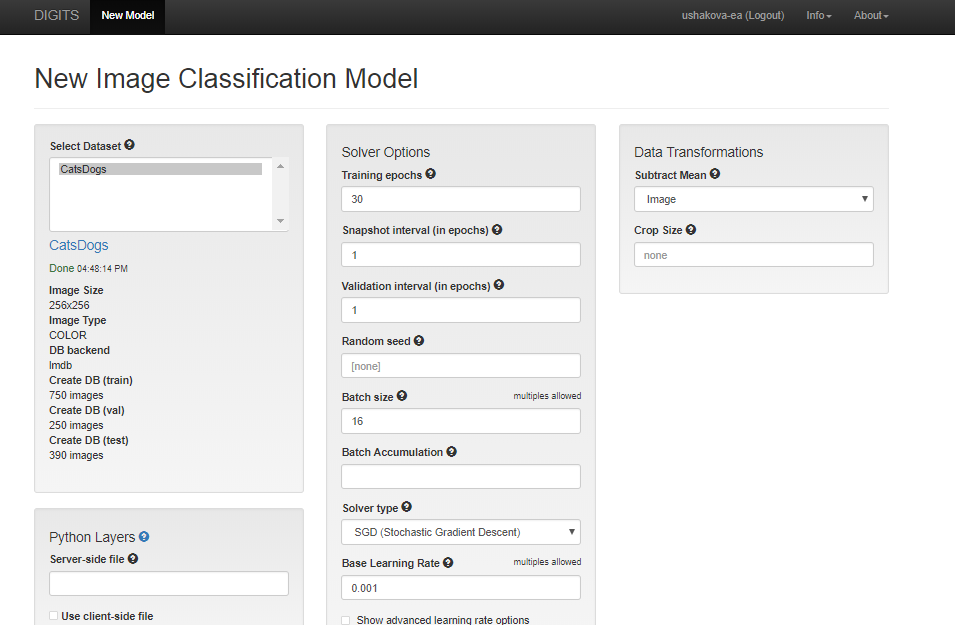


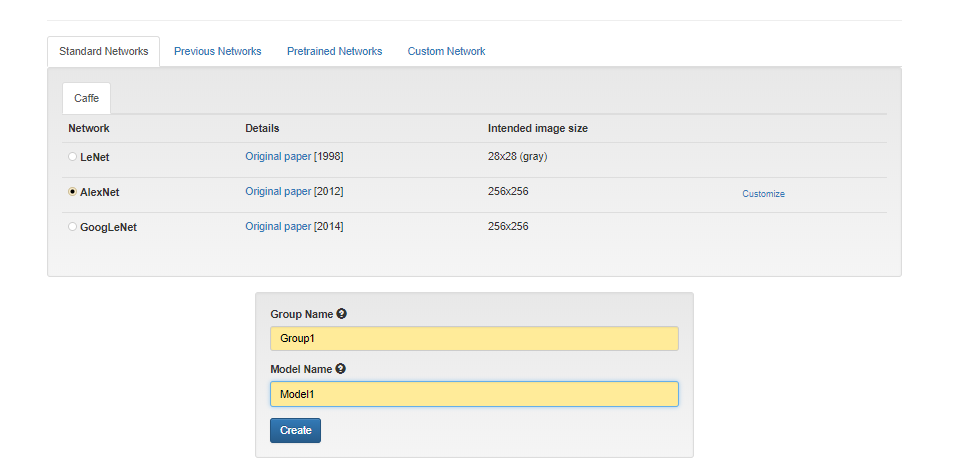
1. Для построения модели можно выбрать:

* одну из трех стандартных сетей **LeNet**, **AlexNet, GoogLeNet**
* ранее сформированную сеть или
* задать свою сеть

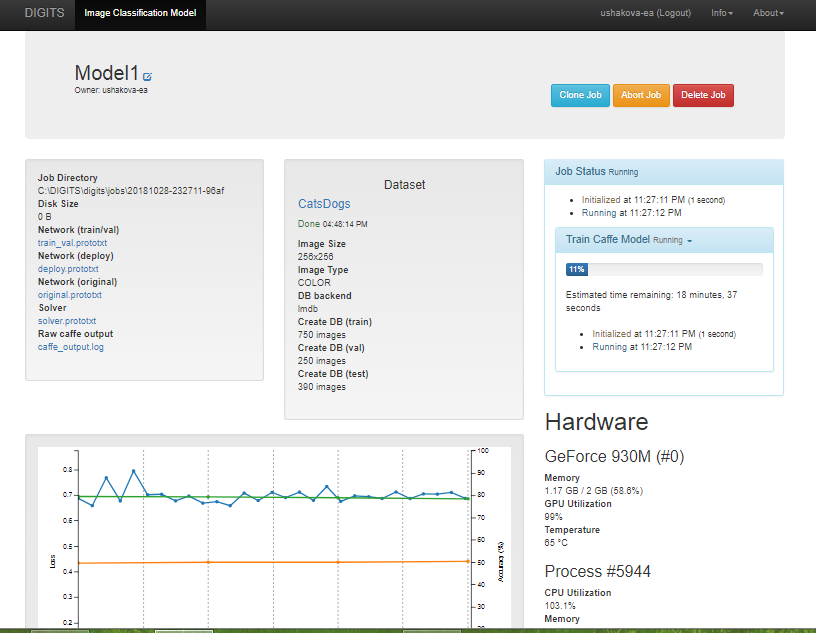


1. Далее необходимо заполнить поля в форме. Пример заполнения:



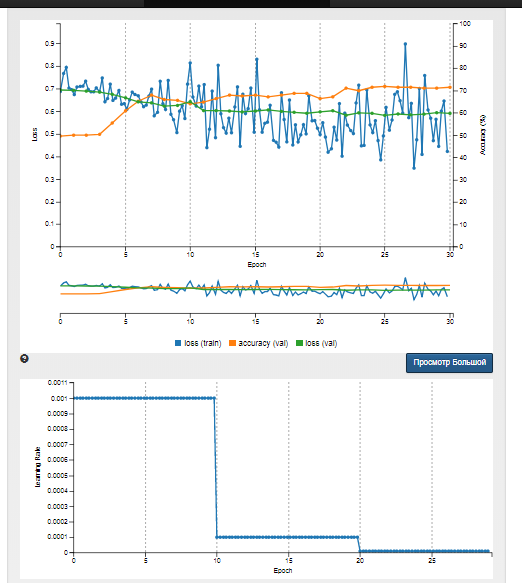


1. Нажать Create → Открывается форма с прогресс баром выполнения → Об успешном завершении информирует положительный статус - Done.

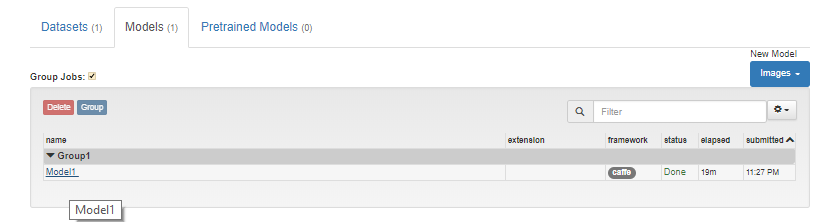


В результате можно увидеть:

* Время, затраченное на формирование модели
* Количество использованной памяти на видеокарте и процессоре
* Ошибку на трейновом наборе
* Ошибку на тестовом наборе
* Точность валидации
* Скорость обучения



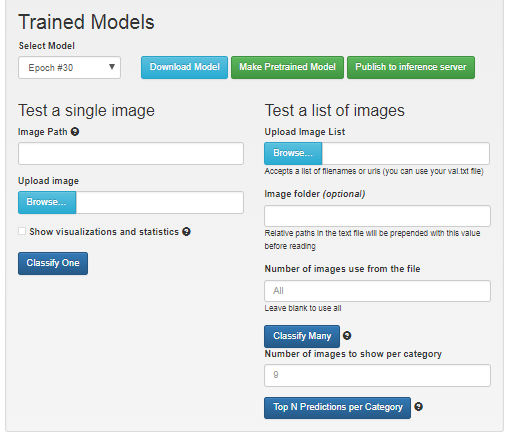
1. Модель будет сохранена во вкладке Models



## **Тестирование модели**

Существует возможность проверить:

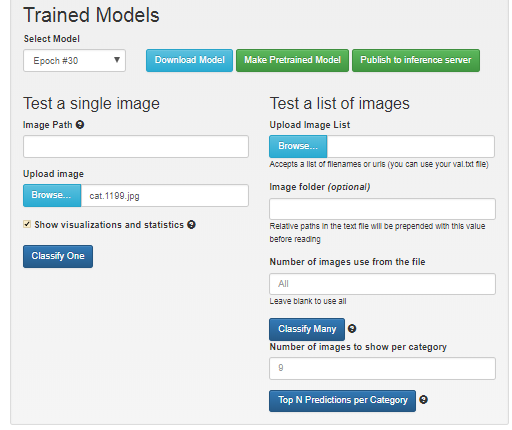
* Список изображений
* Одно изображение



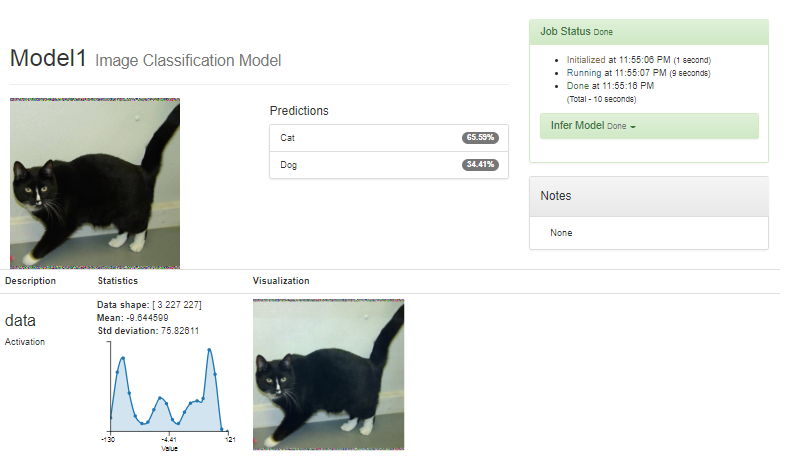
Для тестирования одного изображения

* Загрузить изображение
* Нажать Classify One

Например:



В результате можно увидеть:

* Процент принадлежности к тому или иному классу
* Статистику и визуализацию по нормализации, пулингу и сверточным слоям
* График функции ошибки
* Количесто изученных параметров

## 5. Рекомендации по освоению

Рекомендуемая литература:

1. Уильям Р. Командная строка Microsoft Windows. Справочник администратора (2004)