

Изображения

Семинар DMIA 2016

Гущин Александр

Selective phase of the Data Science Game 2016

Welcome to the online phase of the Data Science Game 2016 !

For this phase, you will be given a dataset of satellite images of roofs. You will have to predict the orientation of the roofs into 4 different categories.

For instance:

- Category 1: North-South orientation



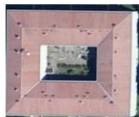
- Category 2: East-West orientation



- Category 3: Flat roof



- Category 4: Other



Аугментация изображений с помощью поворотов: Цель - увеличение размера выборки

1. Картинка класса 1 переходит в класс 2 и наоборот
2. Картинки из класса 3 и 4 остаются в том же классе

Аугментация ответов: Цель - улучшение ответов на тесте

1. Усреднение вероятностей принадлежать 3 классу для всех поворотов картинки (аналогично 4)
2. $\text{Вероятность_принадлежать_классу_1} == 1 - \text{вероятность_для_повернутой_картинки_принадлежать_классу_2}$

State Farm Distracted Driver Detection



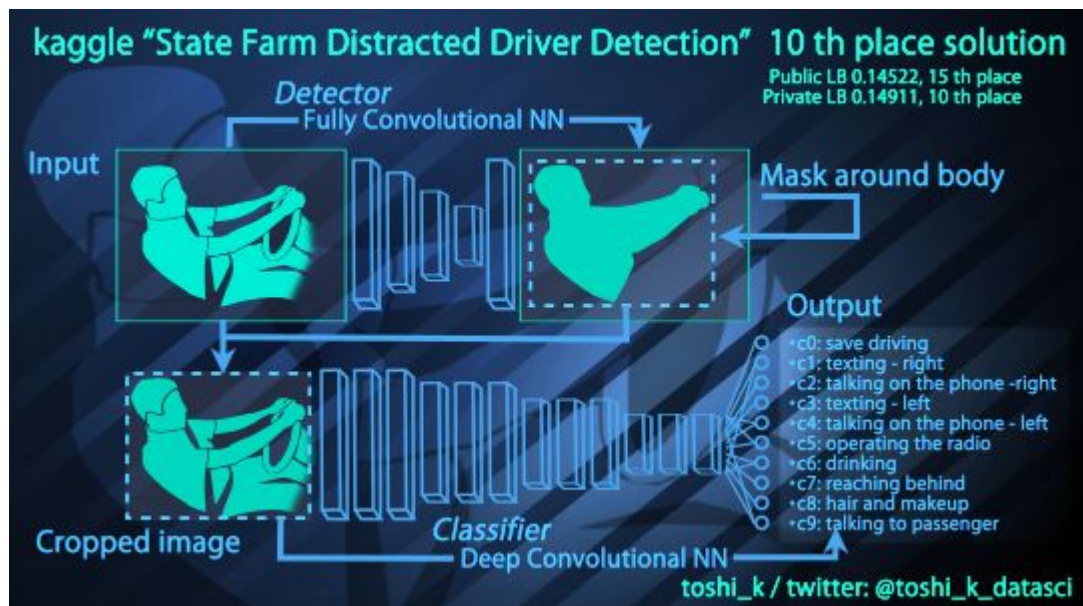
The 10 classes to predict are:

- c0: safe driving
- c1: texting - right
- c2: talking on the phone - right
- c3: texting - left
- c4: talking on the phone - left
- c5: operating the radio
- c6: drinking
- c7: reaching behind
- c8: hair and makeup
- c9: talking to passenger

Другие аугментации

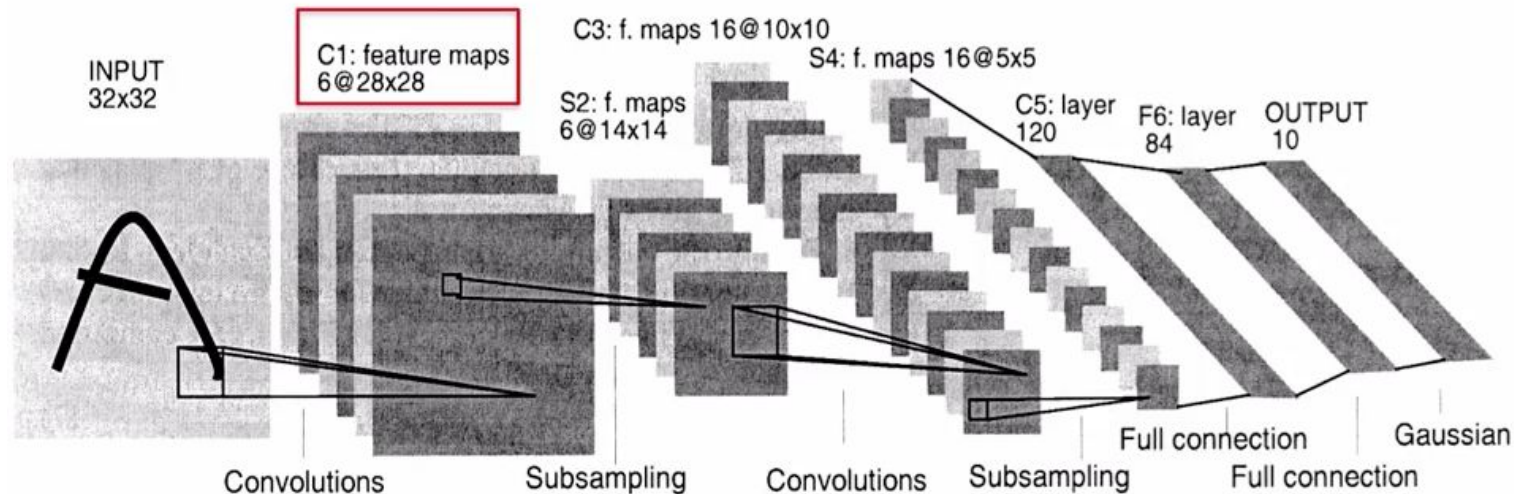


Другие аугментации



Что делать, если изображений в обучающей выборке мало?
Можно файнтюнить (дообучать) уже натренированные сети

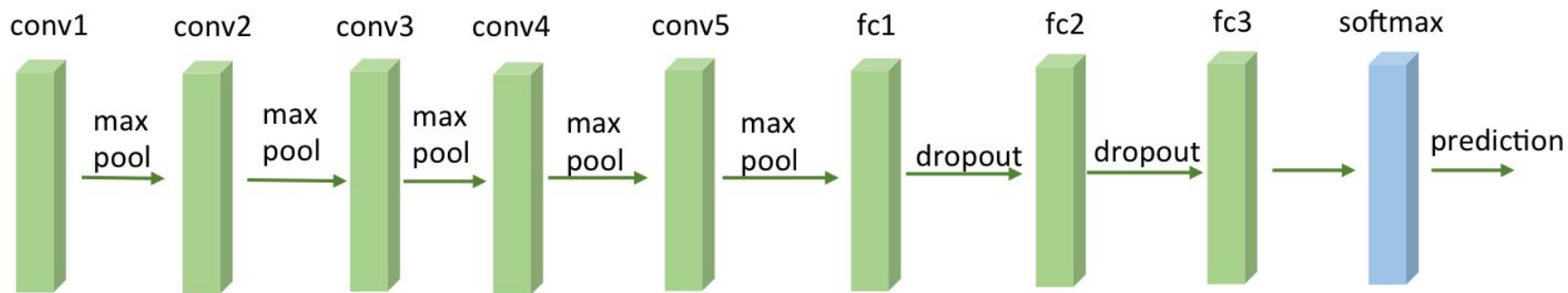
The architecture of LeNet5



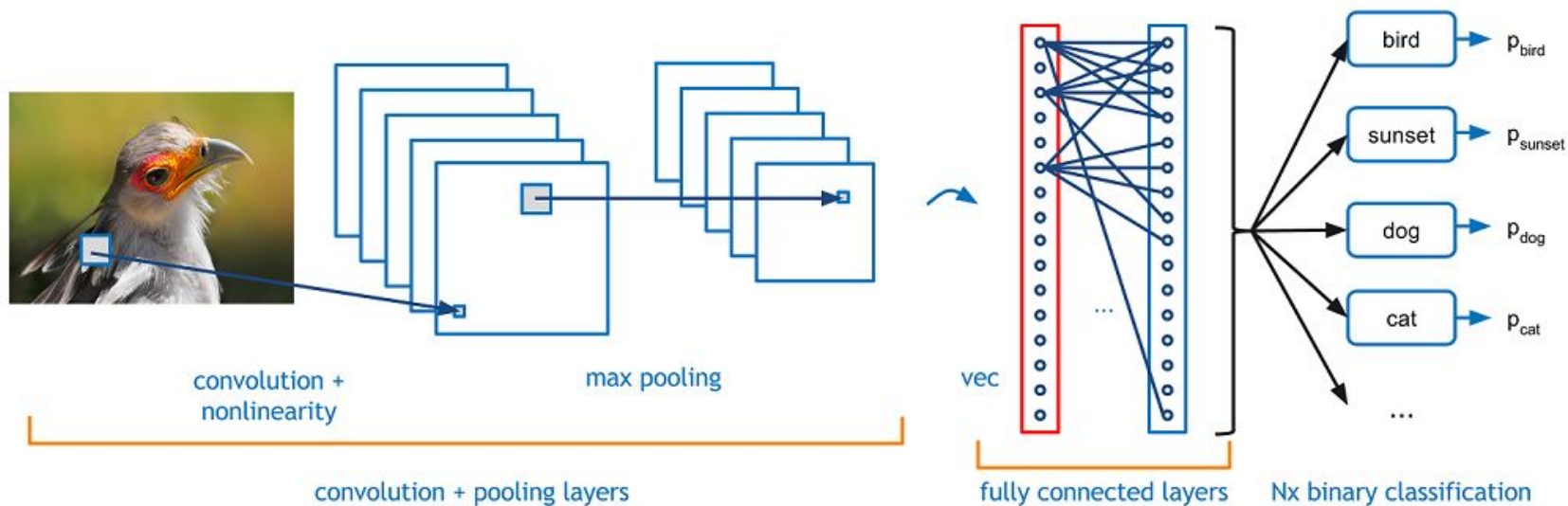
Finetuning

Часто лучше всего заменить последний слой

each conv includes 3 convolutional layers

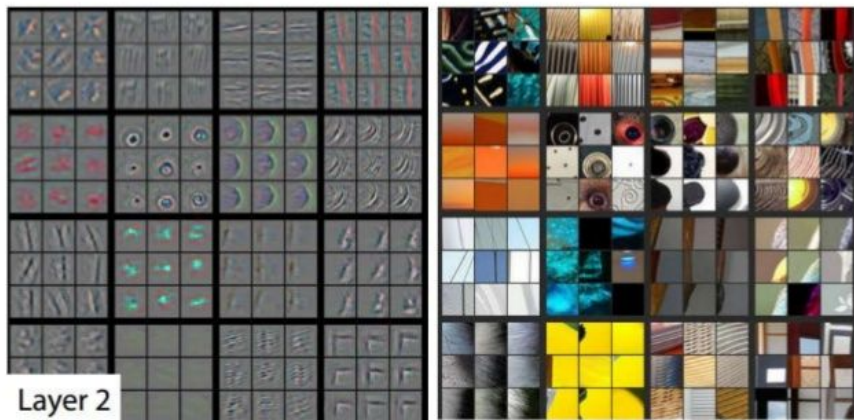


Необязательно переобучать сеть даже для картинок другого размера



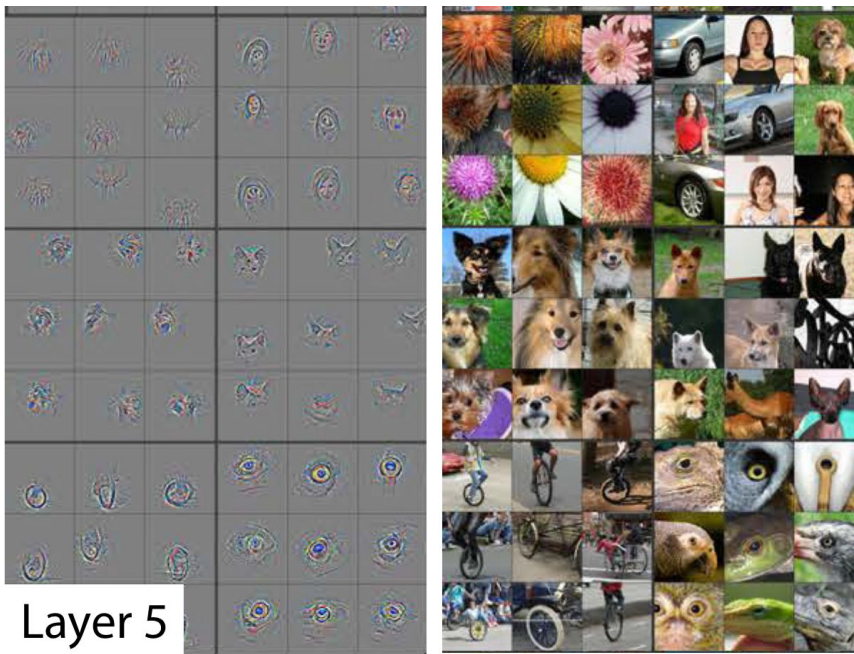
Сверточные слои

Layer 2



Layer 2

Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]



Layer 5

Инструмент: Keras

Плюсы:

1. легко установить через pip
2. легко разобраться и запустить

Минусы:

1. Почти нет предобученных моделей
2. нужно устанавливать драйвера для видеокарты самому

Удобно использовать, чтобы учить сети с нуля

```
42
43 model = Sequential()
44
45 model.add(Convolution2D(32, 3, 3, border_mode='same',
46                          input_shape=X_train.shape[1:]))
47 model.add(Activation('relu'))
48 model.add(Convolution2D(32, 3, 3))
49 model.add(Activation('relu'))
50 model.add(MaxPooling2D(pool_size=(2, 2)))
51 model.add(Dropout(0.25))
52
53 model.add(Convolution2D(64, 3, 3, border_mode='same'))
54 model.add(Activation('relu'))
55 model.add(Convolution2D(64, 3, 3))
56 model.add(Activation('relu'))
57 model.add(MaxPooling2D(pool_size=(2, 2)))
58 model.add(Dropout(0.25))
59
60 model.add(Flatten())
61 model.add(Dense(512))
62 model.add(Activation('relu'))
63 model.add(Dropout(0.5))
64 model.add(Dense(nb_classes))
65 model.add(Activation('softmax'))
66
67 # let's train the model using SGD + momentum (how original).
68 sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
69 model.compile(loss='categorical_crossentropy',
70               optimizer=sgd,
71               metrics=['accuracy'])
72
```

Инструмент: Caffe

Написан на C++, препроцессинг и обучение запускается из консоли

Плюсы:

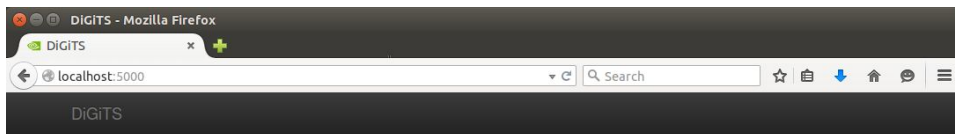
1. много предобученных сетей
2. есть Nvidia Digits
3. одной командой можно поставить драйвера для видеокарты + caffe + digits (на linux)

Минусы:

1. нужно заморачиваться с файлами архитектуры сети
2. (иногда) с упаковкой картинок в db, с подсчётом средней картинки

Удобно использовать, чтобы файнтюнить предобученные сетки или извлекать дескрипторы

Nvidia DIGITS



Home

Datasets

New Dataset
Images ▾

In progress

[dataset_imagenet@256x256](#)
Submitted: 05:29:57 PM (53 seconds ago)
Status: Running

Delete

Completed

[dataset_minst_10k@256x256](#)
Submitted: 05:21:27 PM
Status: Done after 31 seconds

Delete

[voc_cropped@256x256](#)
Submitted: 05:14:31 PM
Status: Done after 2 minutes, 26 seconds

Delete

Models

New Model
Images ▾

In progress

[model_mnist10k](#)
Submitted: 05:30:48 PM (2 seconds ago)
Status: Running

Delete

[LeNet_model_voc_cropped@256x256](#)
Submitted: 05:18:43 PM (12 minutes, 7 seconds ago)
Status: Running

Delete

Completed

model_AlexNet@256x256_voc_cropped@256x256 - Mozilla Firefox

model_AlexNet@25...
localhost:5000/models/20150312-082815-73c3

DIGITS Image Classification Model

Solver

[solver.prototxt](#)
[Network \(train/val\)](#)
[train_val.prototxt](#)
[Network \(deploy\)](#)
[deploy.prototxt](#)

Dataset

[voc_cropped@256x256](#)
Done Wed Mar 11, 05:16:57 PM
Image Size
256x256
Image Type
COLOR
Create DB (train)
26759 images
Create DB (val)
8917 images

Job Status Running

- Initialized at 08:28:15 AM (1 second)
- Running at 08:28:16 AM

Train Caffe Model Running ▾

37%

Estimated time remaining: 30 minutes, 31 seconds

- Initialized at 08:28:15 AM (1 second)
- Running at 08:28:16 AM

Epoch	Loss (train)	Loss (val)	Accuracy (val)
0.0	3.5	3.0	0
1.0	2.5	2.0	40
2.0	2.0	1.8	55
3.0	1.8	1.6	60
4.0	1.6	1.5	62
5.0	1.5	1.4	63
6.0	1.4	1.3	64
7.0	1.3	1.2	65
8.0	1.2	1.1	65
9.0	1.1	1.0	65
10.0	1.0	1.0	65

Аугментации

Аугментации можно делать “на лету”

1. В Nvidia Digits - добавить слой, предобрабатывающий данные (на питоне)
<https://github.com/NVIDIA/DIGITS/tree/master/examples/python-layer>
2. В Keras - делать модификации прямо в коде

Ссылки

Гайд для начинающих по сверточным сетям

<https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>

Classifying plankton with deep neural networks <http://benanne.github.io/2015/03/17/plankton.html>

Стартовый пример для caffe - файнтюнинг и извлечение дескрипторов

<https://github.com/BVLC/caffe/blob/master/examples/00-classification.ipynb>

Стартовый пример для keras - Cifar-10: https://github.com/fchollet/keras/blob/master/examples/cifar10_cnn.py

Nvidia DIGITS <https://github.com/NVIDIA/DIGITS>

Caffe model Zoo <https://github.com/BVLC/caffe/wiki/Model-Zoo>

State Farm Distracted Driver Detection <https://habrahabr.ru/post/307078/>

<https://www.kaggle.com/c/state-farm-distracted-driver-detection/forums/t/22666/10th-place-solution>