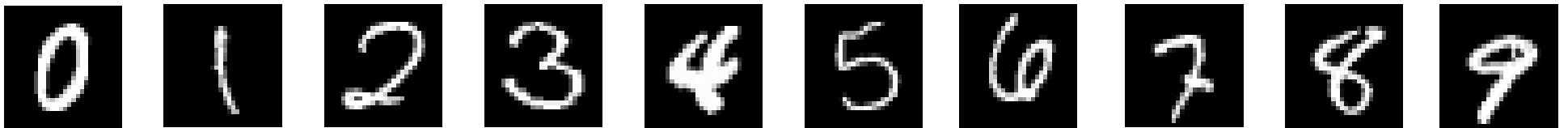


MNIST database

Алексей Пшеничный

MNIST database

MNIST -Mixed(Modified) National Institute of Standards and Technology handwritten digit database (~1998г.)



- в оттенках серого
- в рамке 28x28 пикселей
- 60,000 тренировочных
- 10,000 тестовых

Авторы:

Yann LeCun

Corinna Cortes

Christopher J.C. Burges

Как получена ? Почему mixed ?

NIST (20x20):

Тренировочная - сотрудники American Census Bureau

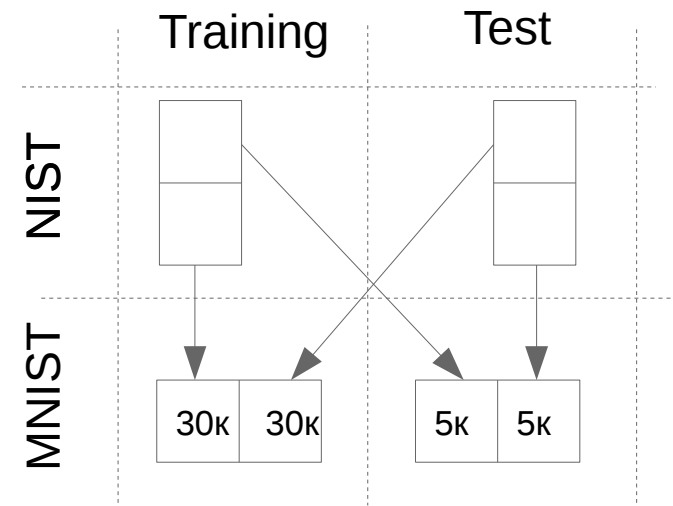
Тестовая - студенты American high school

MNIST (28x28):

- Масштабирование
- Центровка (центр масс)
- Перемешивание:

- Одна половина тестовой и половина тренировочной MNIST взята из **тренировочной** NIST

- Другая половина тестовой и половина тренировочной MNIST взята из **тестовой** NIST

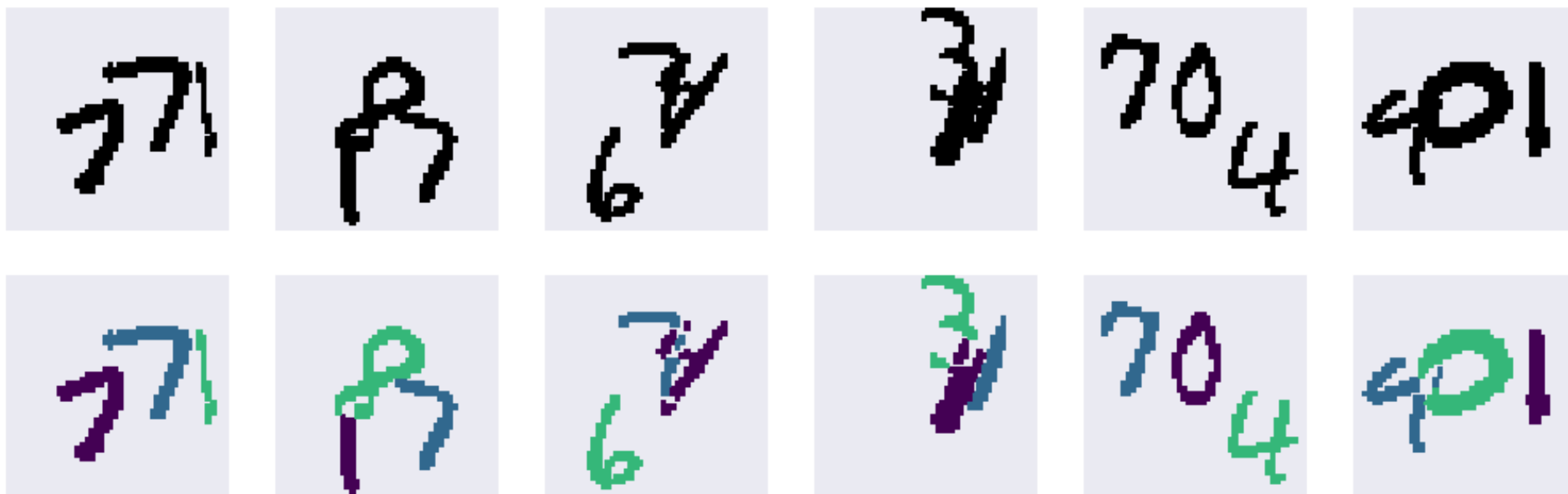


Хороший бенчмарк

Классификатор	Test error rate (%)	
	Использовался как есть	Вариации метода (min)
Linear Classifier	12	7.6
KNN(Euclidean)	5	0.63
Neural Network (FC)	4.7	0.35
Convolutional net	1.7	0.23
Boosted trees	1.53	0.87
SVM	1.4	0.56

Данные за 1998-2012 г.

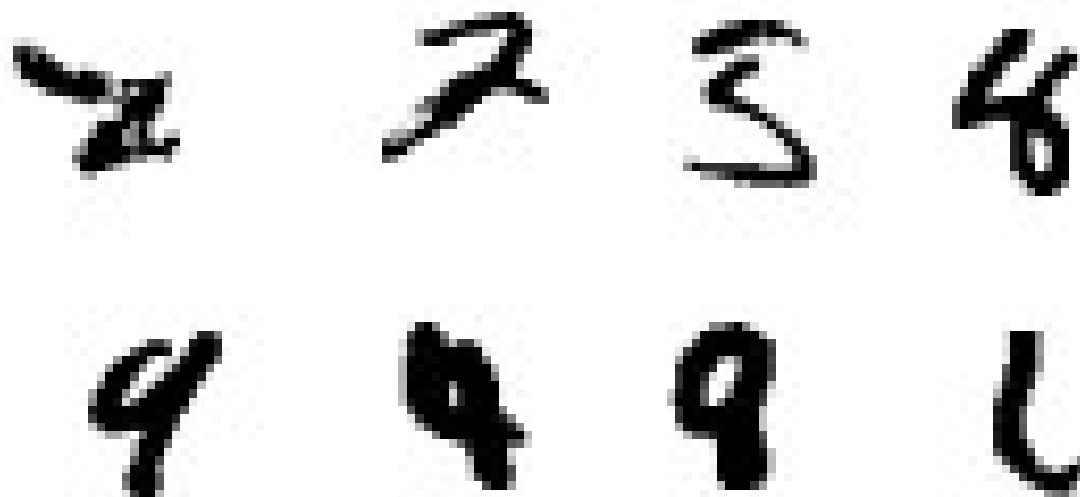
Multi-MNIST



“Superhuman performance”

- Человеческий уровень ~ **0.2%**
- **2012 - 0.23%** *“Multi-column Deep Neural Networks for Image Classification”*
- **2013 - 0.21%** *“Regularization of Neural Networks using DropConnect”*
- **2015 – 0.18%** *“A discriminative cascade CNN model for offline handwritten digit recognition”*

“Superhuman performance”



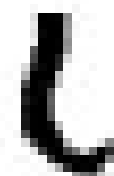
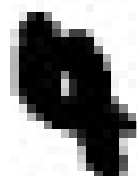
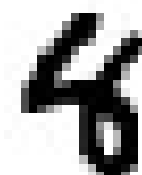
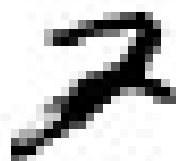
“Superhuman performance”

2

2

5

8



9

4

8

6

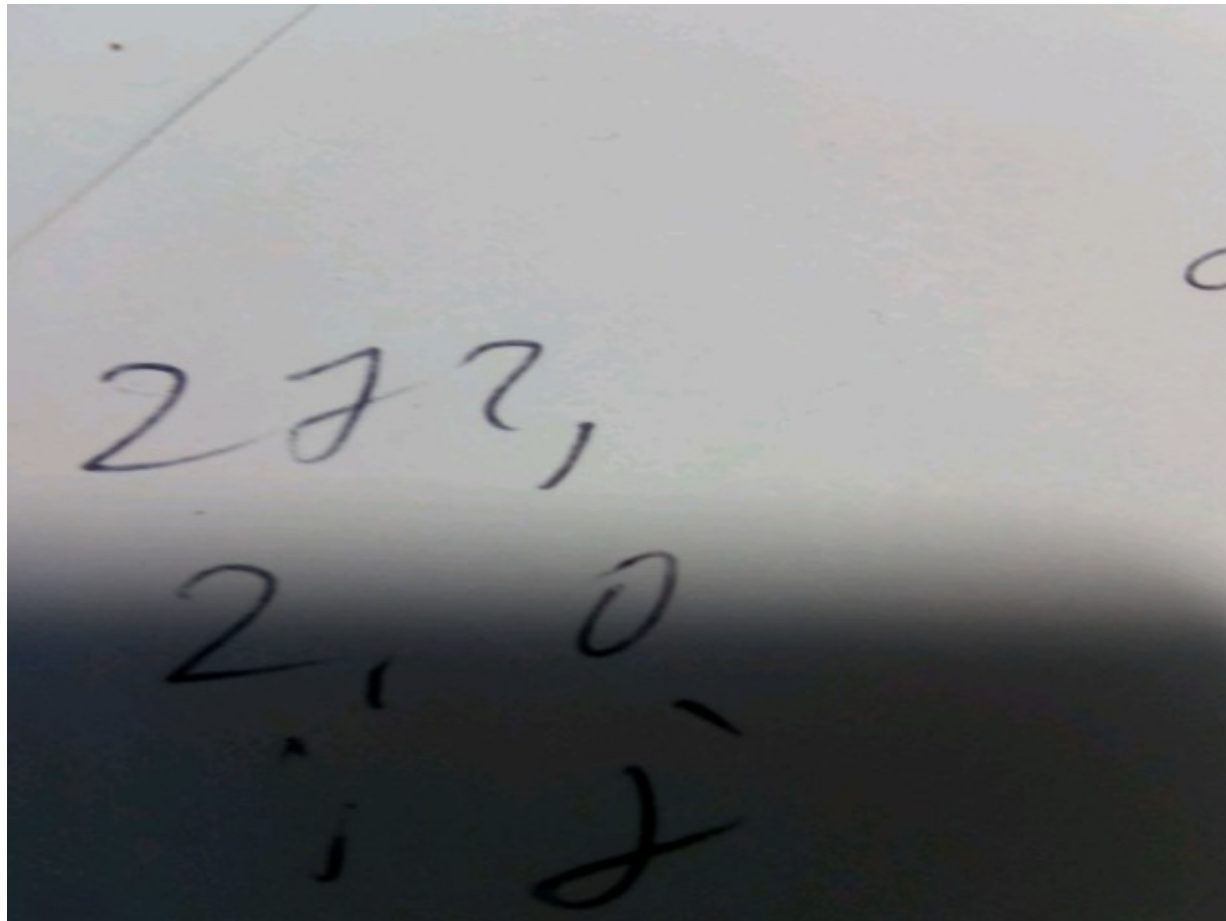
Как получить $\sim 0.00\%$?

- Использовать ответы \rightarrow 1 место на kaggle
- Использовать infimnist или affNIST
(и подглядеть в ответы) $\rightarrow 0.99943\%$
- “Оверфиттинг на валидации”

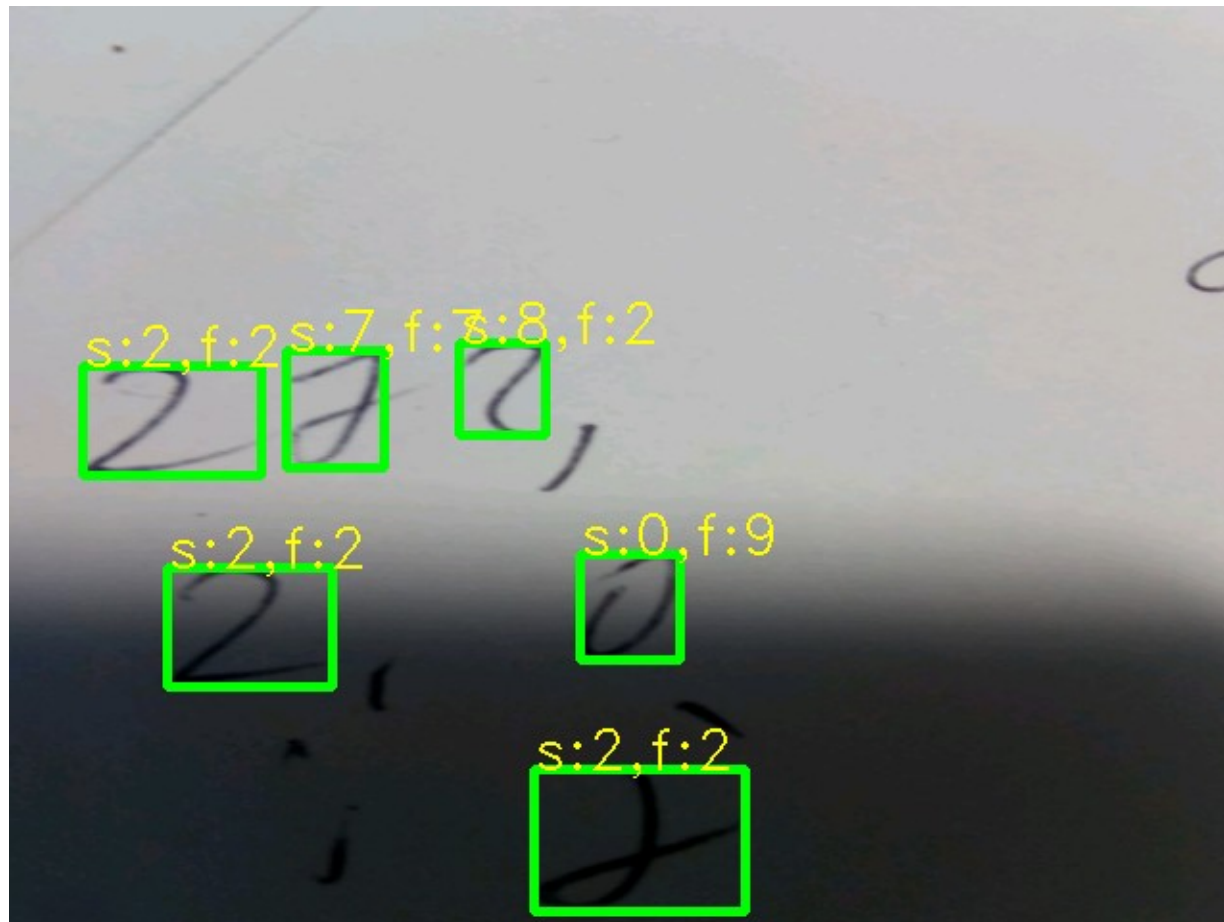
MNIST и OpenCV в действии

- **Прототип** <http://178.162.71.245:9000/>
(нормально работает в Firefox)
- **Фичи** – HOG (Histogram of Oriented Gradients)
 - окно 14X14, направлений 9
 - вектор из 784 стал 36
- **Классификаторы:**
 1. **SVM** – Error: **12.0 %**
 2. **RandomForest** – Error: **7.8 %**
- **Алгоритм:**
blur → adaptiveThreshold → findContours → фильтрация → классификация

MNIST и OpenCV в действии



MNIST и OpenCV в действии



Полезные ссылки

- **MNIST database**

<http://yann.lecun.com/exdb/mnist/>

- **Multi-MNIST**

<https://github.com/Qwlouse/Binding/blob/master/Datasets/Multi-MNIST.ipynb>

- **MNIST top error rate**

http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

- **Пример**

https://github.com/izhleba/mnist_opencv