


DoubledMNIST

Developing an offline handwritten dataset

ACM Turing award 2019



Association for
Computing Machinery

Advancing Computing as a Science & Profession

[Digital Library](#) [CACM](#) [Queue](#) [TechNews](#) [Learning C](#)

[Join](#) [Volunteer](#)

[ABOUT ACM](#) [MEMBERSHIP](#) [PUBLICATIONS](#) [SPECIAL INTEREST GROUPS](#) [CONFERENCES](#) [CHAPTERS](#) [AWARDS](#) [EDUCATION](#) [PUBLIC POLICY](#) [GOV](#)

[Home](#) > [Media Center](#) > [ACM A.M. Turing Award 2018](#)

Fathers of the Deep Learning Revolution Receive ACM A.M. Turing Award

Bengio, Hinton and LeCun Ushered in Major Breakthroughs in Artificial Intelligence

New York, NY, March 27, 2019 – ACM, the Association for Computing Machinery, today named Yoshua Bengio, Geoffrey Hinton, and Yann LeCun recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing. Bengio is Professor at the University of Montreal and Scientific Director at Mila, Quebec’s Artificial Intelligence Institute; Hinton is VP and Engineering Fellow of Google, Chief Scientific Adviser of The Vector Institute, and University Professor Emeritus at the University of Toronto; and LeCun is Professor at New York University and VP and Chief AI Scientist, Facebook.

Working independently and together, Hinton, LeCun and Bengio developed conceptual foundations for the field, identified surprising phenomena through experiments, and contributed engineering advances that demonstrated the practical advantages of deep neural networks. In recent years, deep learning methods have been responsible for astonishing breakthroughs in computer vision, speech recognition, natural language processing, and robotics—among other applications.

Contact:
Jim Ormond
212-626-0505
ormond@hq.acm.org

[Printable PDF File](#)

Father of Deep Learning is also MNIST's father

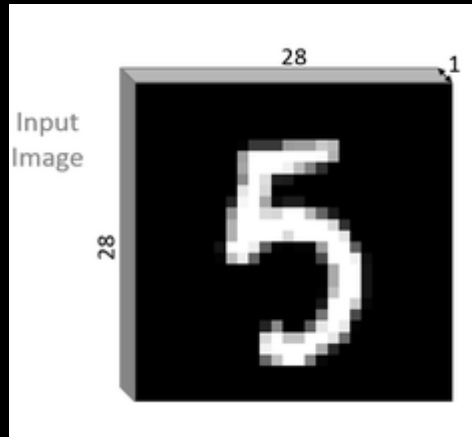
THE MNIST DATABASE

of handwritten digits

Yann LeCun, Courant Institute, NYU

Corinna Cortes, Google Labs, New York

Christopher J.C. Burges, Microsoft Research, Redmond



MNIST's grandfather?

- NIST Special Database 19, American National Institute of Standards and Technology
- Handprinted sample data
- Separate digit, upper and lower case, and free text fields
- ~3600 writers
- ~800,000 images with hand checked classification
- March 1995

NIST -> MNIST, NIST -> DoubledMNIST?

- 800k x (128, 128, 1) -> 70k x (28, 28, 1)
- 800k x (128, 128, 1) -> 140k x (56, 56, 1)
- Why to duplicate?
 - Teaching motivation
- How to duplicate?
 - Cousin influence (EMNIST)

Image Processing

- Crop image
 - Manually
- Add Noise
 - `scipy`
- Square Image
 - Manually
- Resize image
 - Image Interpolation
 - `cv2`

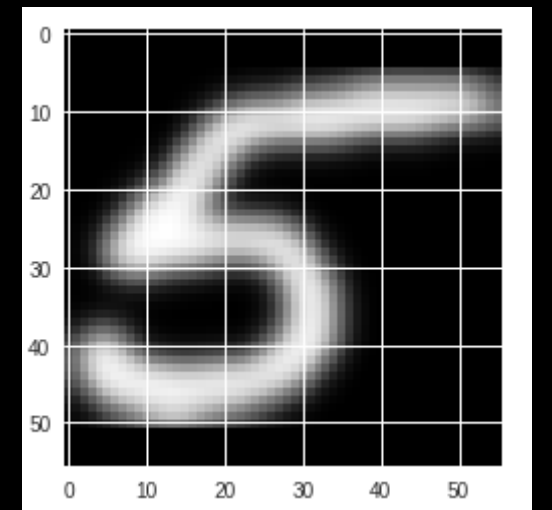
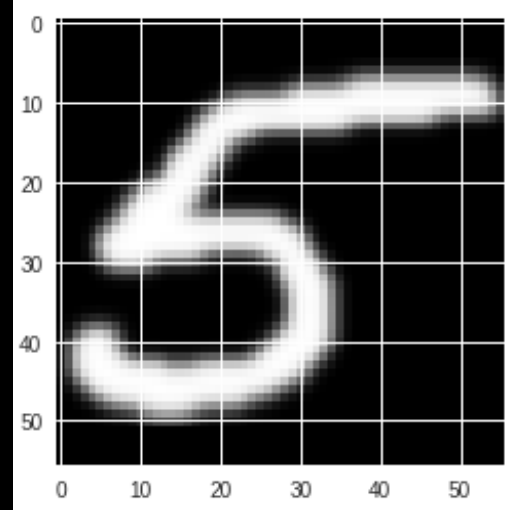
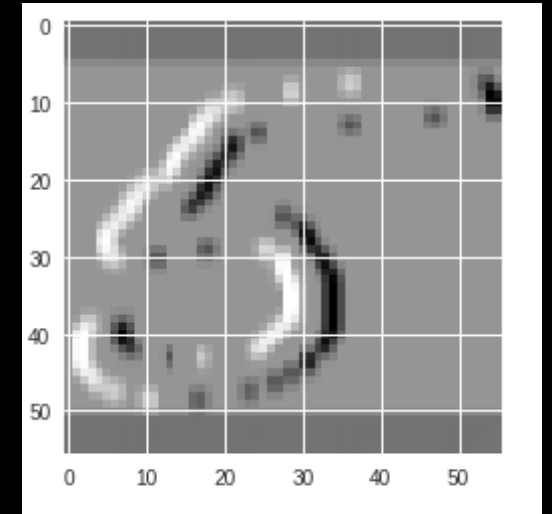
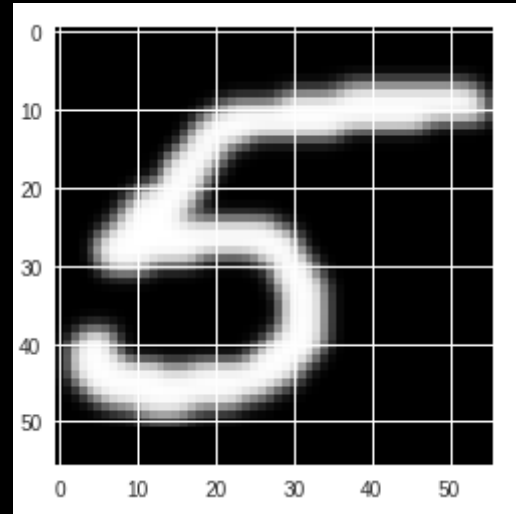
Stylish coding AKA Functional programming in Data Science

```
In [0]: import functools
def compose(*functions):
    return functools.reduce(lambda f, g: lambda x: f(g(x)), functions, lambda x: x)

def parse_image(interp, noise_function):
    """
    return image parser with desired interpolation and noise function
    """
    parser = compose(to8bit, functools.partial(resize_image, interp=interp), square_image, functools.partial(
        add_noise, noise_function=noise_function), crop_image, read_gray)
    return parser
```

Choose best params

- `interp = interp`
- `noise_function = noise_function`
- Gaussian + Bicubic
- Sobel + Bicubic ?
- Laplace + Bicubic
- $\text{Sigma} = 3$?



Evaluation heuristics

- KNeighbours Classifier
- 12k x 2k? (Lazy)
- ~95% / 77%?

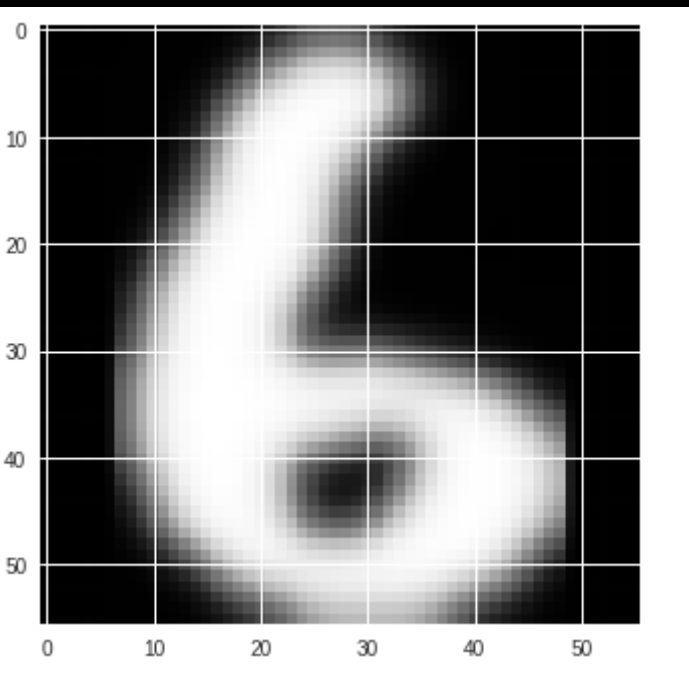
```
start: 0 score: 0.9695
start: 1 score: 0.956
start: 2 score: 0.963
start: 3 score: 0.9595
start: 4 score: 0.9475
start: 5 score: 0.9645
start: 6 score: 0.963
start: 7 score: 0.9665
start: 8 score: 0.957
start: 9 score: 0.9685
Avg: 0.9615
CPU times: user 40min 10s, sys: 11.2 s, total: 40min 22s
Wall time: 40min 22s
```

```
%%time
x_train, y_train, x_test, y_test = extract_dataset('./drive/My Drive/ni_sem/DoubledMNIST.zip')
```

```
CPU times: user 38.5 s, sys: 12.2 s, total: 50.8 s
Wall time: 51 s
```

```
x_train[0]
```

```
array([[255, 255, 254, ..., 255, 255, 255],
       [255, 255, 254, ..., 254, 255, 255],
       [254, 254, 254, ..., 254, 254, 254],
       ...,
       [248, 243, 233, ..., 176, 188, 197],
       [250, 245, 237, ..., 192, 202, 209],
       [251, 247, 240, ..., 205, 212, 218]], dtype=uint8)
```



Utils and Challenges

- `extract_dataset`
- Time complexity
- Scoring

Milan Čugurović

milan_cugurovic@matf.bg.ac.rs