



Advanced Computer Vision

Практический курс

Савельева Юлия Олеговна, 3 семестр, 01.10.2020

Ссылка на таблицу с баллами

В чате курса

Ссылка на материалы курса

https://github.com/IuliiaSaveleva/Advanced_Computer_Vision_course_students

Bitbucket

i.o.saveleva.kpfu@gmail.com IuliiaSaveleva

Text Recognition

Данные

Скачать Synthetic Word Dataset (MJSynth):

<https://www.robots.ox.ac.uk/~vgg/data/text/#sec-synth>

Распаковка архива долгая,
сделайте как можно раньше!

Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, Andrew Zisserman
Visual Geometry Group, Department of Engineering Science, University of Oxford, UK

1. OVERVIEW

Text recognition in natural scene images.



Contributions

- A synthetic data engine to generate unlimited training data.
- Three deep convolutional neural network (CNN) architectures for holistic image classification.
- A resulting set of state-of-the-art reading systems in language constrained and unconstrained scenarios.

2. SYNTHETIC DATA ENGINE

1. Font rendering GENERATOR generator

2. Border/shadow & colour GENERATOR generator

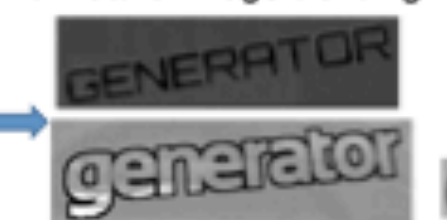


3. Composition GENERATOR generator

4. Projective distortion GENERATOR generator



5. Natural image blending GENERATOR generator



Existing scene text datasets are very small, and cover a small number of words.

Use a synthetic data engine to generate training samples.

Fonts selected from 1400 Google Fonts.

Projective distortion, elastic distortion, and noise applied.

Random crops of natural images alpha-blended with image-layers to generate texture and lighting.

Dataset Available!

- 9 million word images
- Covering 90k English words
- Download at:

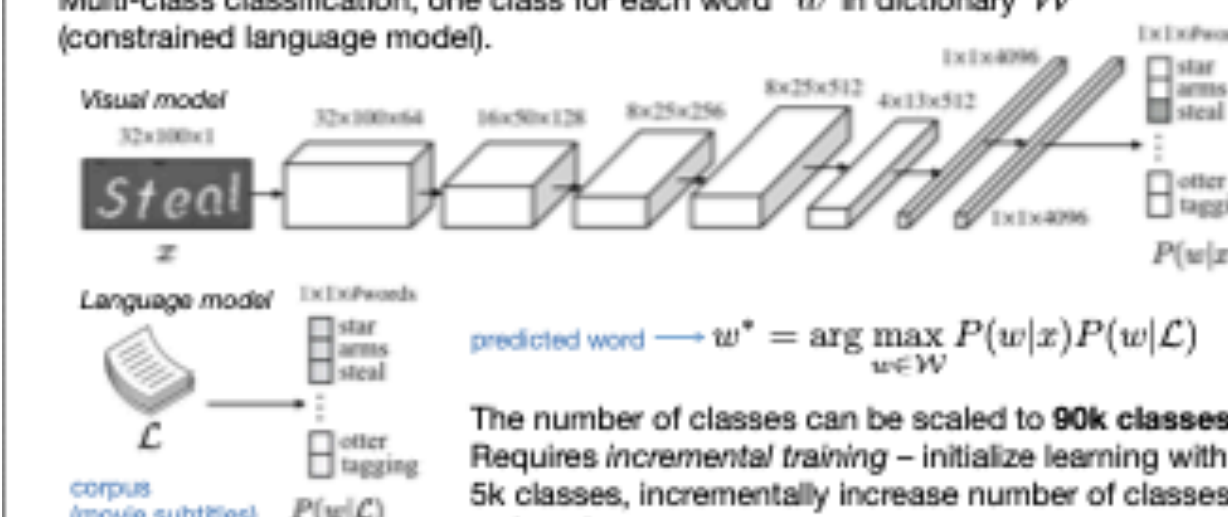
www.robots.ox.ac.uk/~vgg/data/text/



3. MODELS

DICTIONARY ENCODING (DICT)

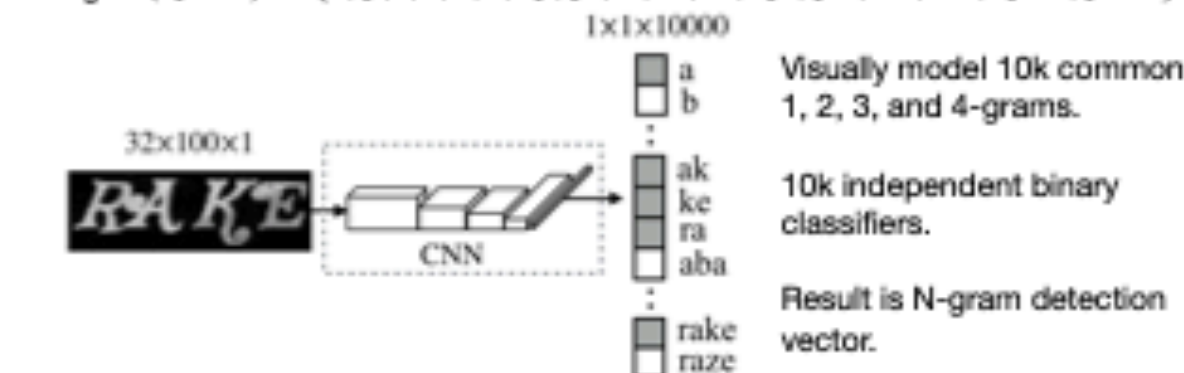
Multi-class classification, one class for each word w in dictionary W (constrained language model).



BAG OF N-GRAMS ENCODING (NGRAM)

Represent a string as a bag-of-N-grams.

E.g. $G(\text{spires}) = \{s, p, i, r, e, s, sp, pi, ir, re, es, spi, pir, ire, res, spire, pires\}$

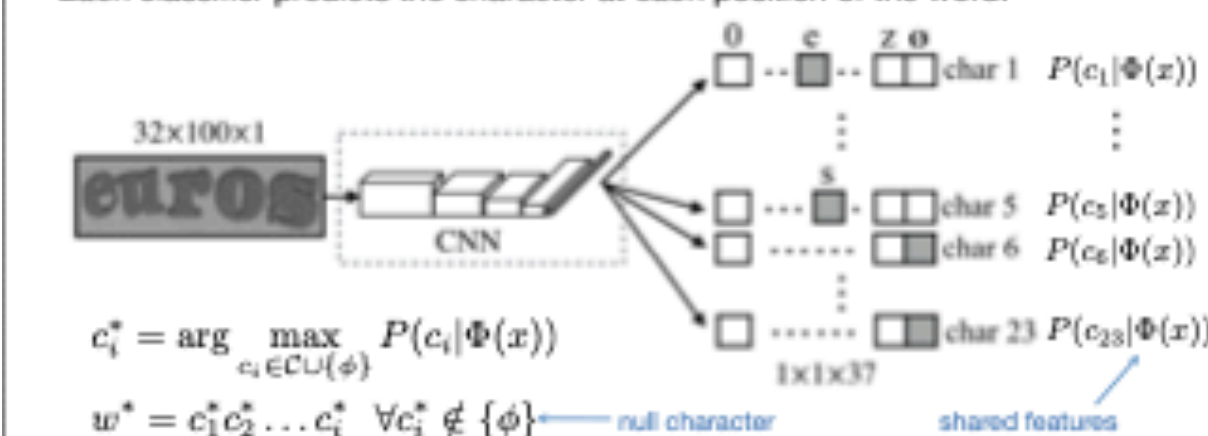


Two ways to recover words:

- Find nearest neighbour of output with ideal outputs of dictionary words.
- Train a linear SVM for each dictionary word, using training data outputs.

CHARACTER SEQUENCE ENCODING (CHAR)

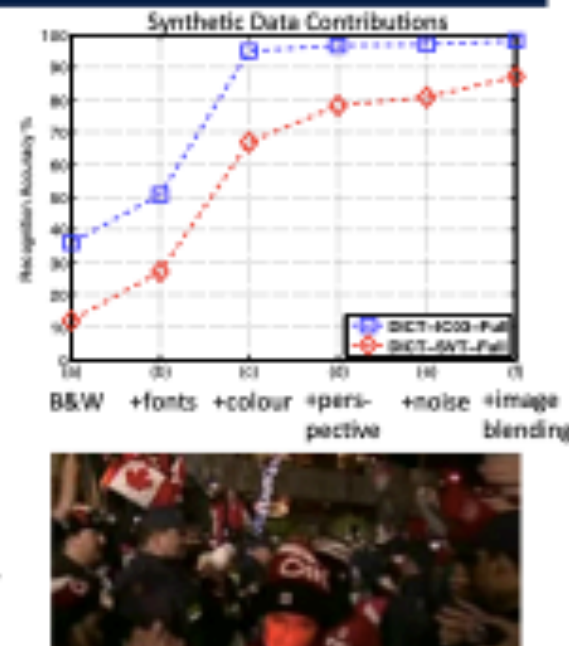
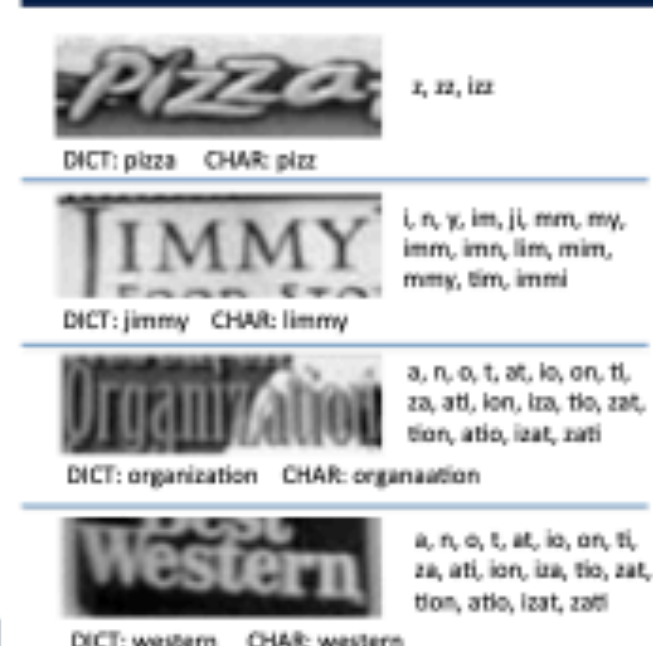
Single CNN with multiple independent classifiers, inspired by Goodfellow et al ICLR'14. Each classifier predicts the character at each position of the word.



No language model, suitable for unconstrained recognition.

4. EXPERIMENTAL SETUP

5. EVALUATION



Text Recognition

Препроцессинг

Реализовать:

Test (no augmentation):

1. Vertical resize (с сохранением пропорций изображения до высоты 32 пикселя)
2. Horizontal resize (в случае, если ширина превышает фиксированный максимальный размер, например, 500 пикселей)
3. Image whitening (посчитать среднее значение по каждому каналу на части обучающей выборки), а затем вычитать соответствующее среднее из каждого канала и делить на 255 (для каждого пикселя на изображении)

Train (augmentation):

1.
 - a) Random horizontal crop (с очень маленькими отступами от левого правого края)
 - b) Random horizontal resize (от 0.8 до 1.2 от исходной ширины, не забываем про ограничение из пункта 2)
 - c) Random Gaussian noise
- 3.

Text Recognition

Архитектура

Реализовать fully-convolutional
нейронную сеть:

[https://openaccess.thecvf.com/
content_ICCV_2017/papers/
Busta_Deep_TextSpotter_An_ICC
V_2017_paper.pdf](https://openaccess.thecvf.com/content_ICCV_2017/papers/Busta_Deep_TextSpotter_An_ICCV_2017_paper.pdf)

Type	Channels	Size/Stride	Dim/Act
input	C	-	$\overline{W} \times 32$
conv	32	3×3	leaky ReLU
conv	32	3×3	leaky ReLU
maxpool		$2 \times 2/2$	$\overline{W}/2 \times 16$
conv	64	3×3	leaky ReLU
BatchNorm			
recurrent conv	64	3×3	leaky ReLU
maxpool		$2 \times 2/2$	$\overline{W}/4 \times 8$
conv	128	3×3	leaky ReLU
BatchNorm			
recurrent conv	128	3×3	leaky ReLU
maxpool		$2 \times 2/2 \times 1$	$\overline{W}/4 \times 4$
conv	256	3×3	leaky ReLU
BatchNorm			
recurrent conv	256	3×3	leaky ReLU
maxpool		$2 \times 2/2 \times 1$	$\overline{W}/4 \times 2$
conv	512	3×2	leaky ReLU
conv	512	5×1	leaky ReLU
conv	$ \hat{\mathcal{A}} $	7×1	$\overline{W}/4 \times 1$
log softmax			

Table 1. Fully-Convolutional Network for Text Recognition

Text Recognition

Целевая функция

Реализовать CTC Loss:

https://www.cs.toronto.edu/~graves/icml_2006.pdf

Алгоритм вычисления **alpha** ->

Реализовать также вычисление **beta** по аналогии, только в обратном направлении

Algorithm 1: CTC Loss alpha computation

Data: $out_{m \times n}$ (result of softmax), where $m = \bar{W}/4, n = |\hat{A}|$,
 l (label encoded by alphabet),
 $bl=0$ (blank index)
begin
 $Loss = 0$
 $L = 2 \times len(l) + 1$
 $T = m$
 $a = zeros(T, L)$
 $a_0^0 = out_0^{bl}$
 $a_0^1 = out_0^{l_0}$
 $c = \sum_{i=0}^1 a_0^i$
 for $i := 0$ **to** 1 **do**
 $a_0^i = a_0^i / c$
 $Loss = Loss + c$
 for $t := 1$ **to** T **do**
 $s = \max(0, L - 2 \times (T - t))$
 $e = \min(2 \times t + 2, L)$
 for $s := 1$ **to** L **do**
 $i = (s - 1) / 2$
 $red = a_{t-1}^s$
 $blue = 0$
 if $s > 0$ **then**
 $blue = a_{t-1}^{s-1}$
 if $s \bmod 2 = 0$ **then**
 $a_t^s = (red + blue) \times out_t^{bl}$
 else if $s = 1$ **or** $l_i = l_{i-1}$ **then**
 $a_t^s = (red + blue) \times out_t^{l_i}$
 else
 $orange = a_{t-1}^{s-2}$
 $a_t^s = (red + blue + orange) \times out_t^{l_i}$
 $c = \sum_{i=s}^e a_t^i$
 for $i := s$ **to** e **do**
 $a_t^i = a_t^i / c$
 $Loss = Loss + c$

Дедлайн 15.10.2020 00:00