Optimization of the image congealing process for handwritten Chinese character recognition

Vincent Giraud, December 17th 2018

Chinese is used by 25 % of the world population

The simplified set contains 3755 characters, while the traditional one contains 6763 of them. Character recognition in this language is thus harder than it is with the Latin alphabet.

Chinese is used by 25 % of the world population

The simplified set contains 3755 characters, while the traditional one contains 6763 of them. Character recognition in this language is thus harder than it is with the Latin alphabet.

备 奋

王 玉

辨 辩 辫

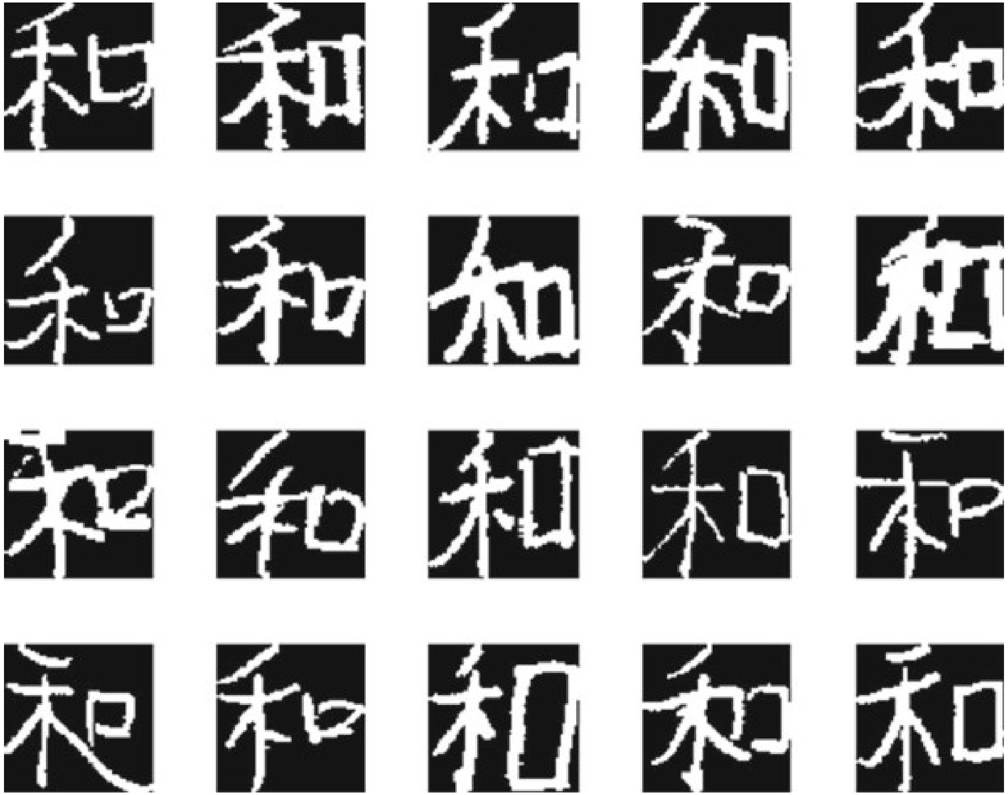
Feature extraction is often used to do handwritten Chinese character recognition.

However, for such a wide character set, it can produce large and complex databases.

In this project, a method without feature extraction has been implemented.

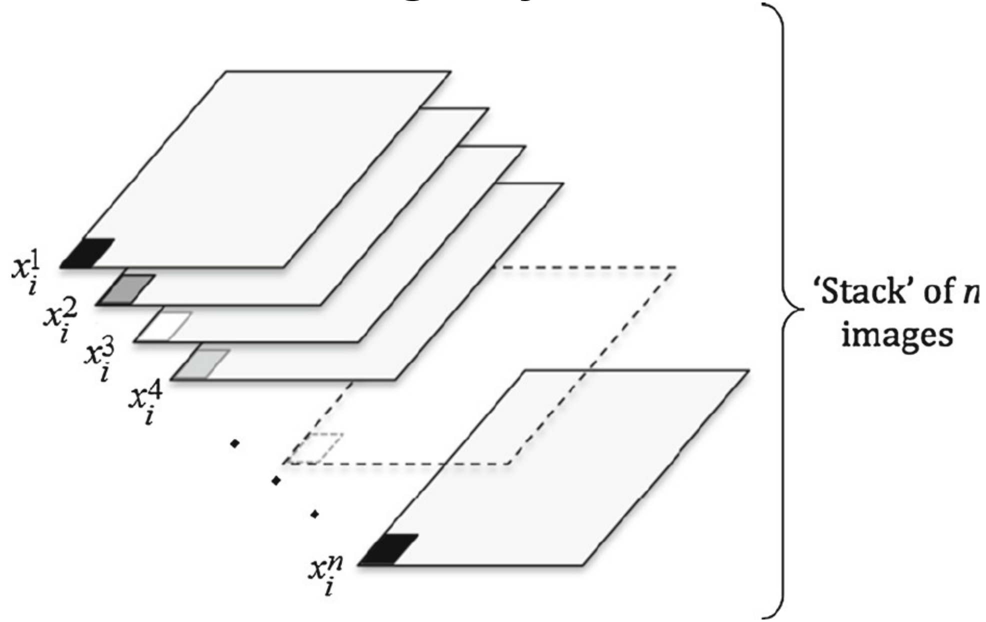
Picture alignment
(or image congealing)

During the training, we need to make models that takes into account the different ways of writing a symbol



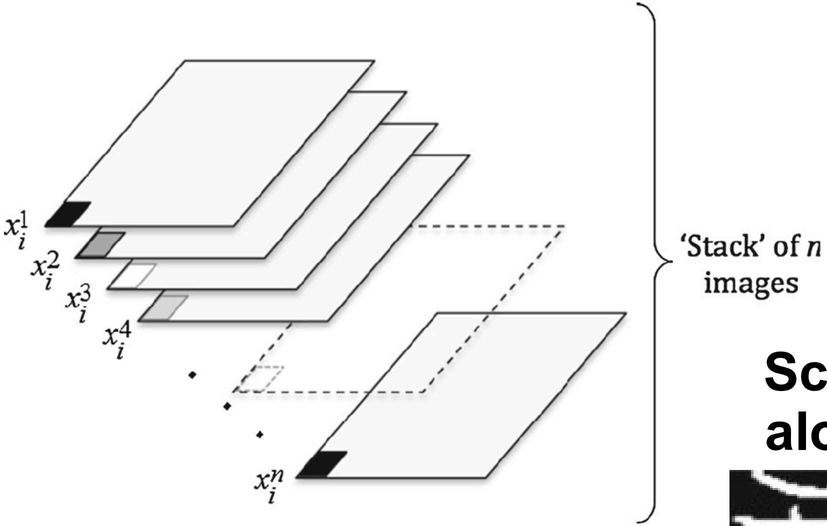
Picture alignment (or image congealing)

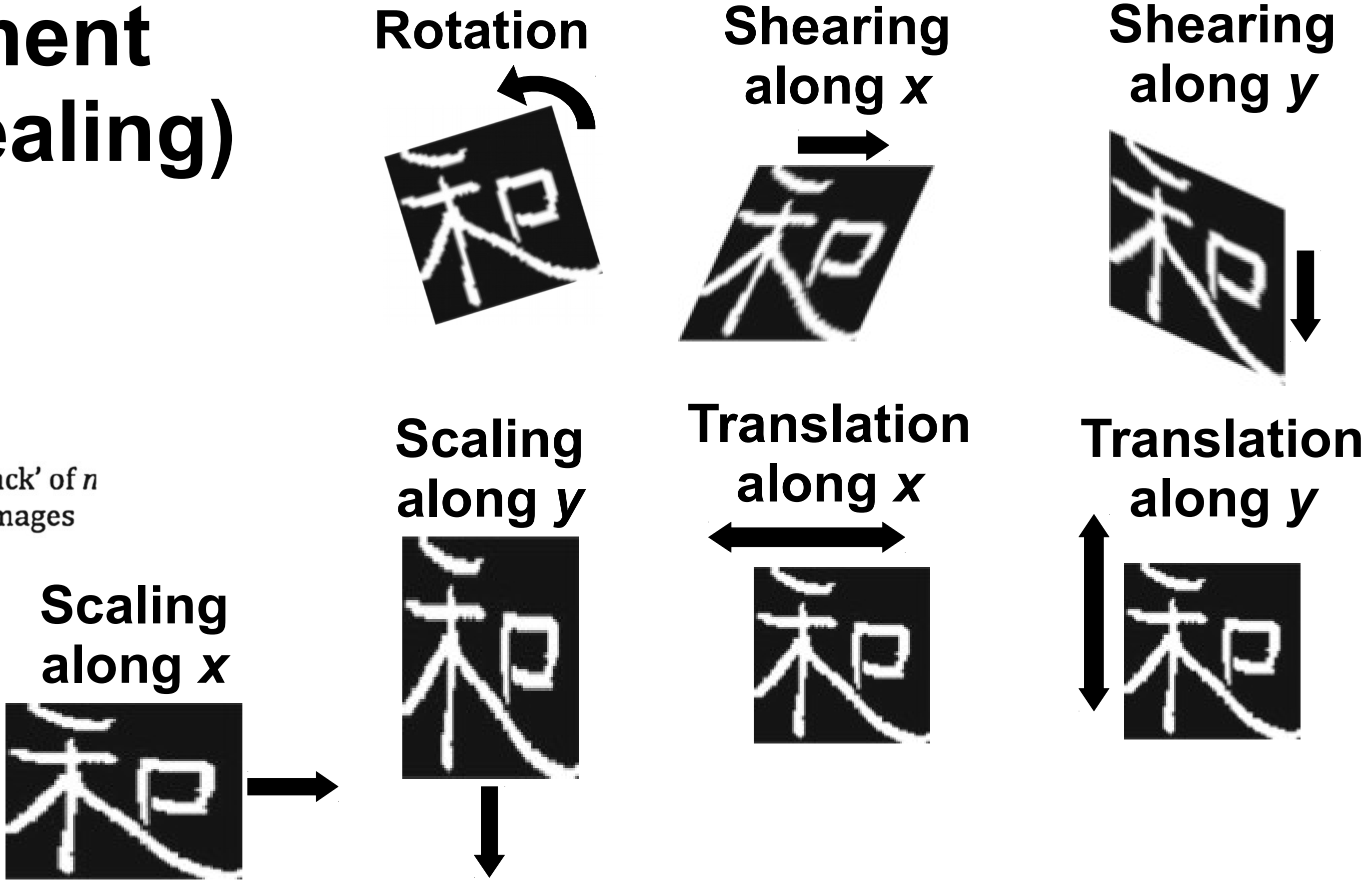
During the training, we need to make models that takes into account the different ways of writing a symbol



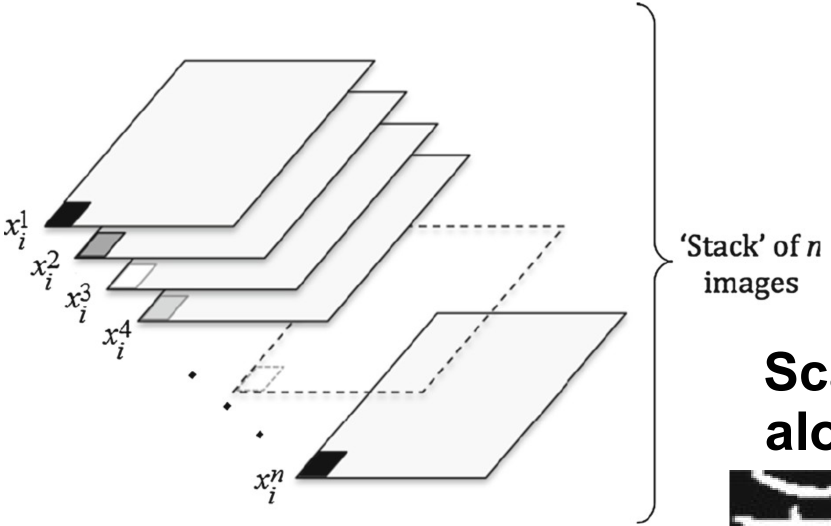
By measuring the entropy on each pixel stack, we can determine how aligned are the symbols

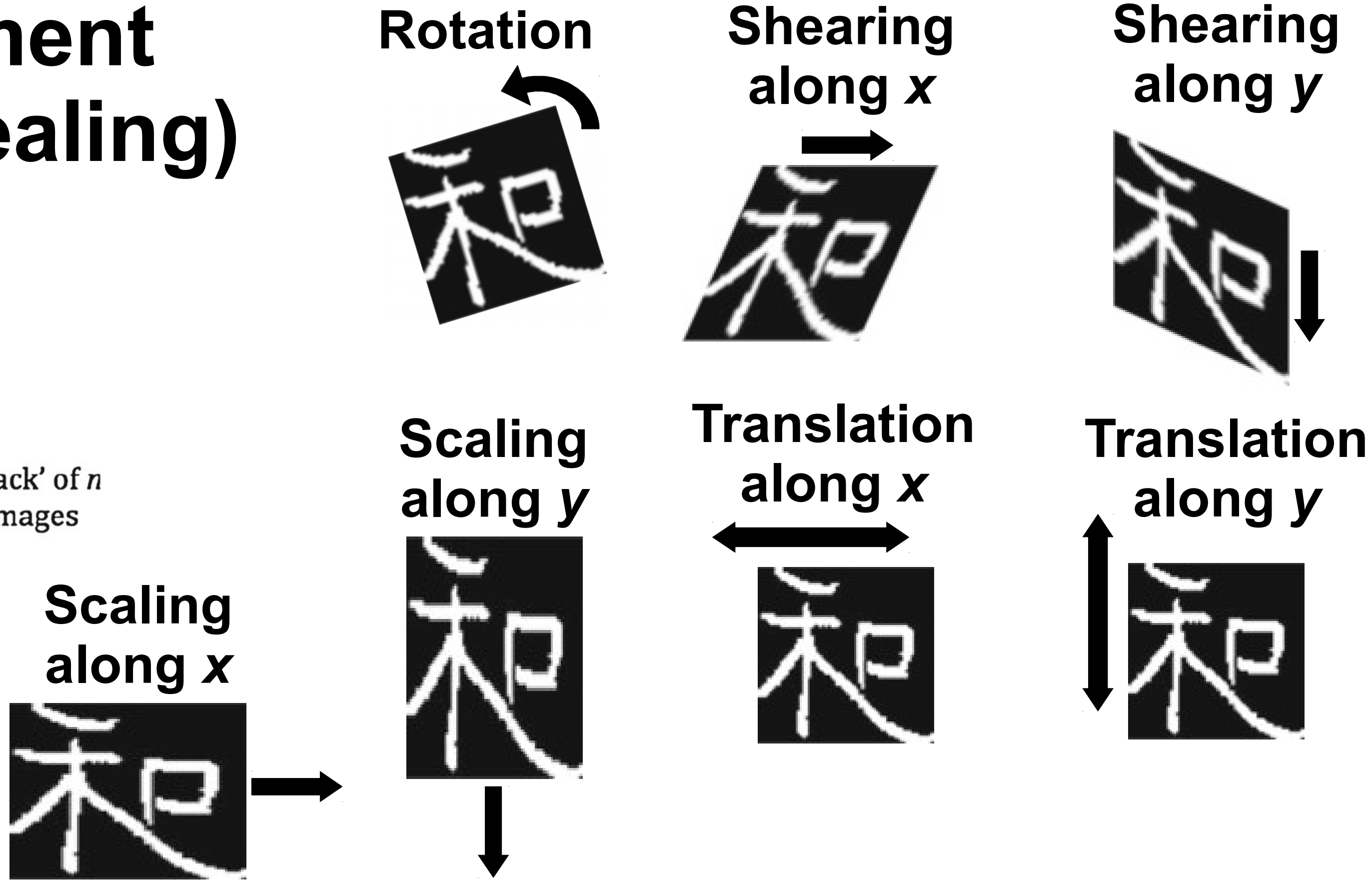
Picture alignment (or image congealing)





Picture alignment (or image congealing)





$$\begin{bmatrix} cos(\theta) \times e^{s_x} + (cos(\theta) \times e^{s_x} \times h_x - sin(\theta) \times e^{s_y}) \times h_y & cos(\theta) \times e^{s_x} \times h_x - sin(\theta) \times e^{s_y} & t_x \\ sin(\theta) \times e^{s_x} + (cos(\theta) \times e^{s_y} + sin(\theta) \times e^{s_x} \times h_x) \times h_y & cos(\theta) \times e^{s_y} + sin(\theta) \times e^{s_x} \times h_x & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Classic entropy computation

$$\sum_{i=1}^{n}\left(-\sum_{m\times m}\left(\frac{1}{n}\sum_{j}x_i^{j'}(k)log_2\frac{1}{n}\sum_{j}x_i^{j'}(k)\right)\right)$$

Uses multinomial distributions to judge the coherence of pixel among its pixel stack

➔ **Low probabilities result in high entropy, and vice versa**

Does not deal very well with diverse writings

Introducing the fuzzy set theory

(with an inheritance scenario)

	House	Money
Alice	1	0
Bob	0	1
Charlie	0	0

Introducing the fuzzy set theory

(with an inheritance scenario)

	House	Money
Alice	**1**	**0**
Bob	**0**	**1**
Charlie	**0**	**0**

	House	Money
Alice	**0.6**	**0**
Bob	**0**	**0.5**
Charlie	**0.4**	**0.5**

The house is a member of Alice’s set with a degree of 0.6

Introducing the fuzzy set theory

(with an inheritance scenario)

	House	Money
Alice	**1**	**0**
Bob	**0**	**1**
Charlie	**0**	**0**

	House	Money
Alice	(#visitsFromAlice/month)/31	**0**
Bob	**0**	BobKindnessPercentage/100
Charlie	1 - (#visitsFromAlice/month)/31	1 - BobKindnessPercentage/100

Similar to the indicator functions in classical set theory

Fuzzy entropy computation

In a pixel stack, each pixel is now compared to all the other ones

A pixel stack's fuzzy entropy is computed by summing up all the membership degrees

The image stack's fuzzy entropy is obtained by summing up all the sub-entropies (pixel stacks' entropies)

Fuzzy entropy computation

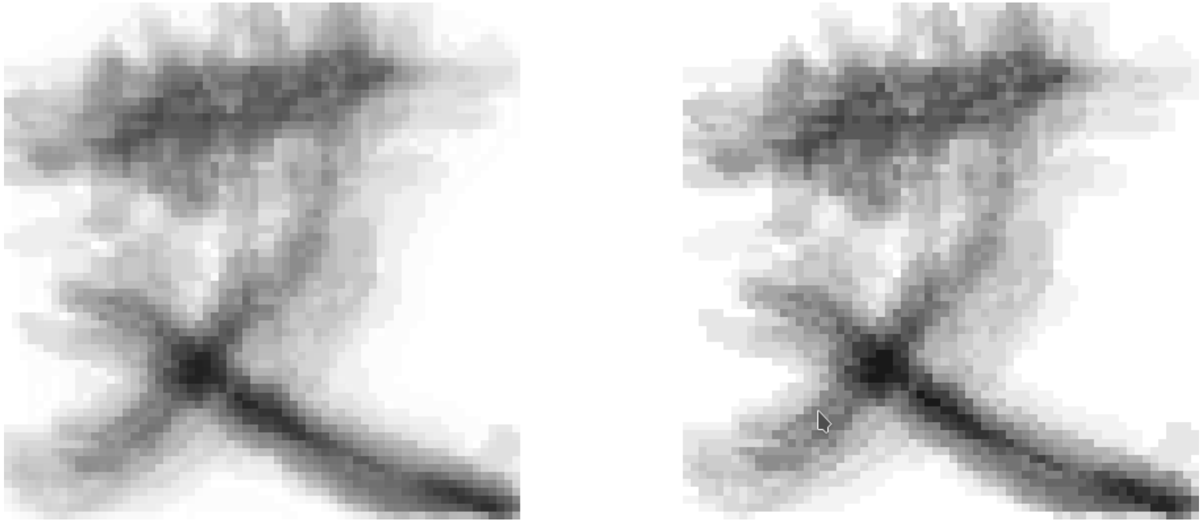
In a pixel stack, each pixel is now compared to all the other ones

A pixel stack's fuzzy entropy is computed by summing up all the membership degrees

The image stack's fuzzy entropy is obtained by summing up all the sub-entropies (pixel stacks' entropies)

$$\binom{n}{2} \times m^2 \times \#iterations$$

To align one stack for only one symbol, the membership function will be executed 11,673,600 times.



Fuzzy entropy computation

$$\mu_i(x,y) = 1 - \frac{|i(x) - i(y)|}{|i_{max} - i_{min}|}$$

$$\mu_i(x,y) = exp\left(-\frac{(i(x) - i(y))^2}{2\sigma_i^2}\right)$$

$$\mu_i(x,y) = max\left(min\left(\frac{(i(y) - (i(x) - \sigma_i))}{(i(x) - (i(x) - \sigma_i))}, \frac{i(x) + \sigma_i - i(y)}{i(x) + \sigma_i - i(x)}\right), 0\right)$$

Fuzzy entropy computation

$$\mu_i(x,y) = 1 - \frac{|i(x) - i(y)|}{|i_{max} - i_{min}|}$$

$$\mu_i(x,y) = exp\left(-\frac{(i(x) - i(y))^2}{2\sigma_i^2}\right)$$

$$\mu_i(x,y) = max\left(min\left(\frac{(i(y) - (i(x) - \sigma_i))}{(i(x) - (i(x) - \sigma_i))}, \frac{i(x) + \sigma_i - i(y)}{i(x) + \sigma_i - i(x)}\right), 0\right)$$

$$\mu_i(x,y) = 1 - |i(x) - i(y)|$$

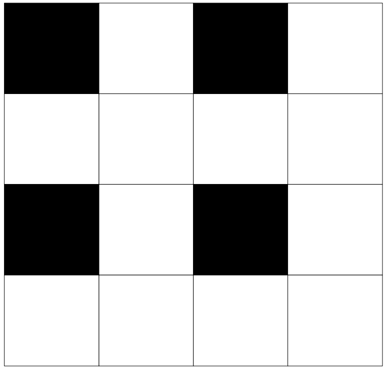
$$\mu_i(x,y) = exp\left(-(i(x) - i(y))^2\right)$$

Entropy subsampling

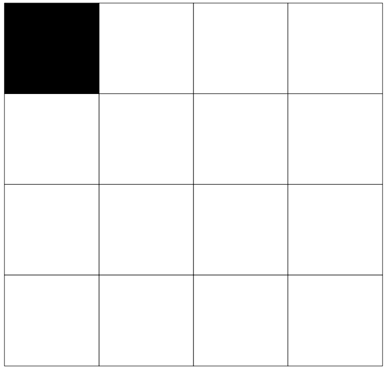
When subsampling pixel stacks, computation is faster.

1

1/2



1/4



Consequences of the new membership functions

		×3.13	×2.48
Character subset	Original linear	Simplified linear	Simplified exponential
Single	100	97.18	96.94
Left and right	92.31	84.59	82.64
Upper and lower	85.00	78.12	80.06
Surrounded by	83.33	74.33	74.10
Framework	100	98.84	97.83

Consequences of the entropy subsampling

Fraction of stacks processed	1	1/2	1/3	1/4
Speed of execution	1	1.91	2.93	3.93
Precision	1	0.97	0.94	0.83