

Semester Project: Chinese Type Design

Maxime Schoemans

École polytechnique fédérale de Lausanne

June 3, 2019

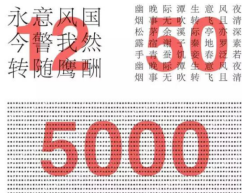
Overview

- 1 Introduction
- 2 Project Structure
- 3 Rough character set
- 4 Refined character set
- 5 Results and future work

Introduction

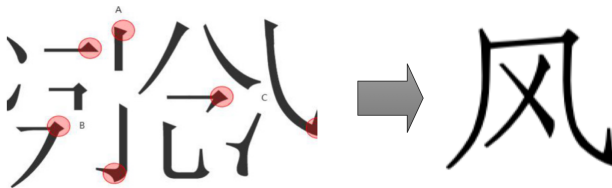


(a) 12 iconic characters



5000

(b) From 12 to 50 and to 5000



(c) From components to final character

Project structure

- Project split in 2 main parts
- Input set: Small set of base characters
- Intermediate set: Complete set of rough characters
- Output set: Complete set of refined characters

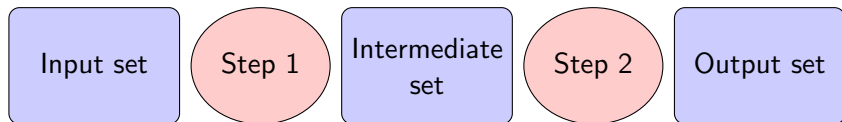


Figure: Project pipeline

- Most characters are combinations of simpler characters
- A base character is a character that cannot be decomposed further into simpler components
- Each character has a character definition, specifying its components and layout
- 2 possible layout types: general layout or precise definition

Example of character decomposition

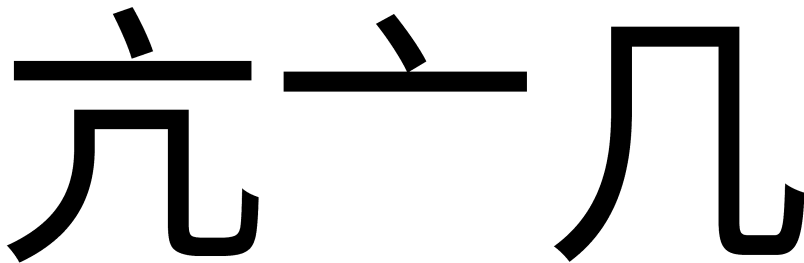


Figure: Character with its components

Character definition

Example of a character definition for the previous character:

- Character: 亢
- Unicode: 4ea2
- Layout id: 2
- Components: 亠, 儿
- Component unicodes: 4ea0, 51e0
- Precise definition: None

Layouts 1/2

AIZI - Layout System

bounding box : {x, y, dx, dy}
starting point: upper left corner
box size = {1,1}



水, 又, 万.....
layout id: lid00
name: duji // 独体
bid0: 0, 0, 1, 1



明, 灯, 林.....
layout id: lid01
name: zuoyou // 左右
bid0: 0, 0, 0.5, 1
bid1: 0.5, 0, 0.5, 1



哭, 画, 会.....
layout id: lid02
name: shangxia // 上下
bid0: 0, 0, 1, 0.5
bid1: 0, 0.5, 1, 0.5



床, 尚, 静.....
layout id: lid03
name: zuoshangbaowei // 左上包围
bid0: 0, 0, 1, 1
bid1: 0.5, 0.5, 0.5, 0.5



漏, 边, 建.....
layout id: lid04
name: zuoxiabaowei // 左下包围
bid0: 0, 0, 1, 1
bid1: 0.5, 0, 0.5, 0.5



得, 勺, 旬.....
layout id: lid05
name: youshangbaowei // 右上包围
bid0: 0, 0, 1, 1
bid1: 0, 0.5, 0.5, 0.5



风, 网, 冈.....
layout id: lid06
name: shangbaowei // 上包围
bid0: 0, 0, 1, 1
bid1: 1, 0.25, 0.5, 0.5



凶, 雨, 幽.....
layout id: lid07
name: xiabaowei // 下包围
bid0: 0, 0, 1, 1
bid1: 0.25, 0, 0.5, 0.5



医, 医, 医.....
layout id: lid08
name: zuobaowei // 左包围
bid0: 0, 0, 1, 1
bid1: 0.5, 0.25, 0.5, 0.5



因, 因, 因.....
layout id: lid09
name: quanbaowei // 全包围
bid0: 0, 0, 1, 1
bid1: 0.25, 0.25, 0.5, 0.5



羽, 羽, 羽.....
layout id: lid10
name: shanghongyou // 上中右
bid0: 0, 0, 0.3, 1
bid1: 0.3, 0, 0.3, 1
bid2: 0.6, 0, 0.3, 1



荣, 景, 景.....
layout id: lid11
name: shangzhongxia // 上中下
bid0: 0, 0, 1, 0.3
bid1: 0, 0.3, 1, 0.3
bid2: 0, 0.6, 1, 0.3



哭, 森, 森.....
layout id: lid12
name: sandie // 三叠
bid0: 0, 0, 1, 0.5
bid1: 0, 0.5, 0.5, 0.5
bid2: 0.5, 0.5, 0.5, 0.5



然, 雷, 雷.....
layout id: lid13
name: zuoyouxia // 左右下
bid0: 0, 0, 0.5, 0.5
bid1: 0.5, 0, 0.5, 0.5
bid2: 0, 0.5, 1, 0.5



差, 差, 差.....
layout id: lid14
name: duixiang // 对称
bid0: 0, 0, 1, 1
bid1: 0, 0.25, 0.5, 0.5
bid2: 0.5, 0.25, 0.5, 0.5



疑, 疑, 疑.....
layout id: lid15
name: siqie // 四叠
bid0: 0, 0, 0.5, 0.5
bid1: 0.5, 0, 0.5, 0.5
bid2: 0, 0.5, 0.5, 0.5
bid3: 0.5, 0.5, 0.5, 0.5



静, 静, 静.....
layout id: lid16
name: shangzuoyouxia // 上左中右
bid0: 0, 0, 1, 0.3
bid1: 0, 0.3, 0.5, 0.3
bid2: 0.5, 0.3, 0.5, 0.3
bid3: 0, 0.6, 1, 0.3



漏, 漏.....
layout id: lid17
name: shangzhongzuozhongyou // 上中左中右
bid0: 0, 0, 1, 0.25
bid1: 0, 0.25, 1, 0.25
bid2: 0, 0.5, 0.2, 0.5
bid3: 0.3, 0.5, 0.3, 0.5
bid4: 0.6, 0.5, 0.3, 0.5



雷, 雷.....
layout id: lid18
name: zuoyouzhongzuoyou // 左左中左右
bid0: 0, 0, 0.5, 0.3
bid1: 0.5, 0, 0.5, 0.3
bid2: 0, 0.3, 1, 0.3
bid3: 0, 0.6, 0.5, 0.3
bid4: 0.5, 0.6, 0.5, 0.3



静.....
layout id: lid19
name: shangzuozhongyouxia // 上左中右下
bid0: 0, 0, 1, 0.3
bid1: 0, 0.3, 0.3, 0.3
bid2: 0.3, 0.3, 0.3, 0.3
bid3: 0.6, 0.3, 0.3, 0.3
bid4: 0, 0.6, 1, 0.3

Layouts 2/2

AI21 -Layout System -additional

bounding box : (x, y, dx, dy)
starting point: upper left corner
box size = {L,1}



中, 中, 中.....
layout id: lid20
name: shuchuancha
// 竖穿编
bid0: 0, 0, 1, 1
bid1: 0, 0, 0, 3, 1



坐
layout id: lid25
name: zuosouchuancha
// 左右穿编
bid0: 0, 0, 1, 1
bid1: 0, 0, 0, 5, 0, 5
bid2: 0, 5, 0, 0, 5, 0, 5



面
layout id: lid30
name: chli // 齿
bid0: 0, 0, 1, 0, 5
bid1: 0, 0, 5, 1, 0, 5
bid2: 0, 3, 0, 5, 0, 3, 0, 3



侧, 侧, 侧
layout id: lid35
name: zuoshangzhongxin
// 左上中下
bid0: 0, 0, 0, 5, 1
bid1: 0, 5, 0, 0, 5, 0, 3
bid2: 0, 5, 0, 3, 0, 5, 0, 3
bid3: 0, 5, 0, 6, 0, 5, 0, 3



副, 副.....
layout id: lid21
name: shangxiayou
// 上下右
bid0: 0, 0, 0, 5, 0, 5
bid1: 0, 0, 5, 0, 5, 0, 5
bid2: 0, 5, 0, 0, 5, 1



侧, 侧, 侧.....
layout id: lid26
name: sheng // 侧
bid0: 0, 0, 1, 0, 2, 5
bid1: 0, 0, 2, 5, 1, 0, 2, 5
bid2: 0, 0, 5, 1, 0, 2, 5
bid3: 0, 0, 7, 5, 1, 0, 2, 5



里
layout id: lid31
name: sou // 里
bid0: 0, 0, 1, 0, 5
bid1: 0, 3, 0, 0, 3, 0, 5
bid2: 0, 0, 5, 1, 0, 5



里
layout id: lid36
name: wuheng // 五横
bid0: 0, 0, 1, 0, 2
bid1: 0, 0, 2, 1, 0, 2
bid2: 0, 0, 4, 1, 0, 2
bid3: 0, 0, 6, 1, 0, 2
bid4: 0, 0, 8, 1, 0, 2



侧, 侧, 侧.....
layout id: lid22
name: zuochuangxin
// 左上下
bid0: 0, 0, 0, 5, 1
bid1: 0, 5, 0, 0, 5, 0, 5
bid2: 0, 5, 0, 5, 0, 5, 0, 5



里
layout id: lid27
name: sidedechuancha
// 侧穿编
bid0: 0, 0, 1, 1
bid1: 0, 0, 0, 5, 0, 5
bid2: 0, 5, 0, 0, 5, 0, 5
bid3: 0, 5, 0, 5, 0, 5, 0, 5
bid4: 0, 5, 0, 5, 0, 5, 0, 5



侧
layout id: lid22
name: zuorhongyouxin
// 左中右下
bid0: 0, 0, 0, 3, 0, 5
bid1: 0, 3, 0, 0, 3, 0, 5
bid2: 0, 6, 0, 0, 3, 0, 5
bid3: 0, 0, 5, 1, 0, 5



原, 原, 原.....
layout id: lid23
name: zuobaoweishangxin
// 左包原上下
bid0: 0, 0, 1, 1
bid1: 0, 3, 0, 3, 0, 6, 0, 3
bid2: 0, 3, 0, 6, 0, 6, 0, 3



串, 串, 串.....
layout id: lid28
name: hengchuancha
// 横穿编
bid0: 0, 0, 1, 1
bid1: 0, 0, 3, 1, 0, 3



副
layout id: lid33
name: juan // 副
bid0: 0, 0, 1, 0, 2, 5
bid1: 0, 0, 2, 5, 1, 0, 2, 5
bid2: 0, 0, 5, 0, 5, 0, 5
bid3: 0, 5, 0, 5, 0, 5, 0, 5



副
layout id: lid24
name: zuobaoweizuoqiu
// 左包原左右
bid0: 0, 0, 1, 1
bid1: 0, 3, 0, 3, 0, 3, 0, 6
bid2: 0, 6, 0, 3, 0, 3, 0, 6



串
layout id: lid29
name: ce // 串
bid0: 0, 0, 0, 5, 1
bid1: 0, 5, 0, 0, 5, 1
bid2: 0, 0, 3, 1, 0, 3



里
layout id: lid34
name: ji // 里
bid0: 0, 0, 1, 0, 6
bid1: 0, 0, 3, 0, 3, 0, 3
bid2: 0, 6, 0, 3, 0, 3, 0, 3
bid3: 0, 0, 6, 1, 0, 3

Layouts: 2 methods

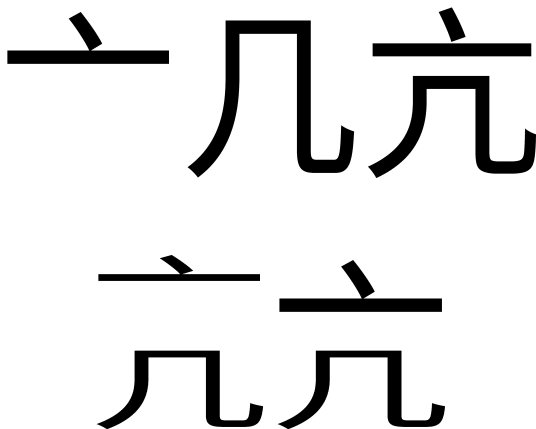
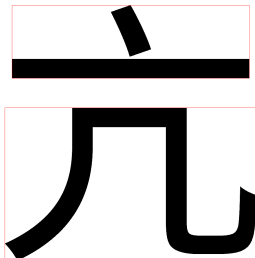


Figure: Comparison between the two basic layout methods

Precise layout definition

- 1 layout definition per character
- Achieves better results than the general layout methods
- Created manually or automatically
- Generalizable to multiple fonts
- Example of a character with a precise definition:



- Character: 亢
- Layout id: 2
- Components: 亠, 儿
- Precise definition:
 - ① [69, 30, 866, 266]
 - ② [43, 404, 918, 557]

Automatic definition

Version 1:

- Brute force
- Bad results and computationally intensive

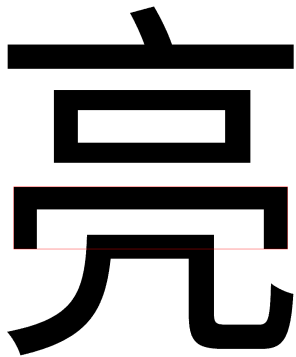
Version 2:

- Use of black patches
- Faster method, but not applicable to all characters
- Need for a preprocessing step

Version 3:

- Slight variation of version 2

Comparison between version 2 and 3

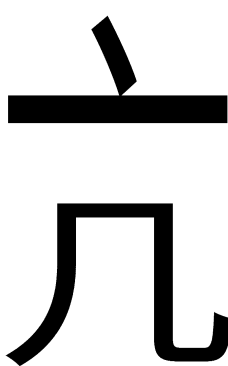


(a) Version 2

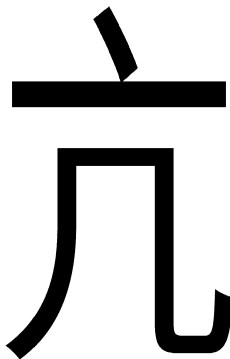


(b) Version 3

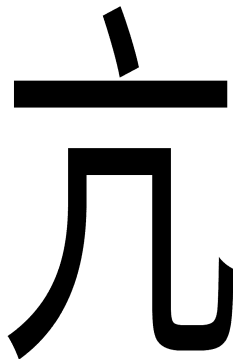
Rough results with and without precise definition



(a) Without



(b) With



(c) Real character

Refined character set

- Second step in the pipeline
- Transforms a rough character into a refined character
- Use of a neural network
- Architectures tested: unet, gan
- Different loss functions tested: l_1 , l_2 on the edges, constant loss

Architecture

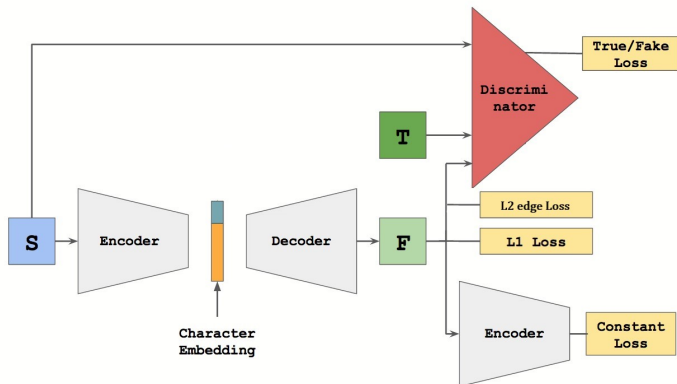
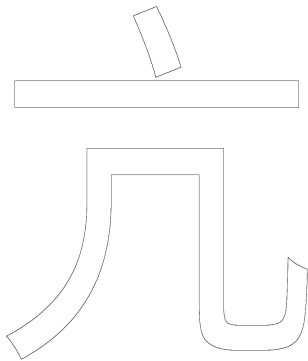


Figure: Network architecture

Loss functions

- L1 loss between output image and the real image
- L2 loss between the edges of the output and the real image
- Constant loss: L2 loss between the encoded input and output images
- Discriminator loss

Edge loss (version 1)



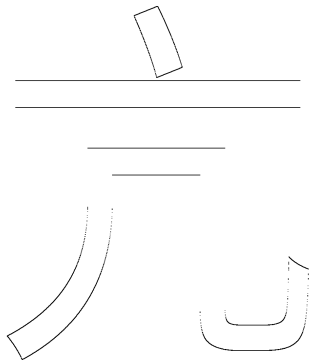
(a) Edges



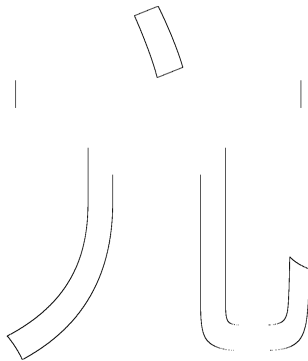
(b) Distance map

Edge loss (version 2)

Based on the Sobel filters



(a) S_x



(b) S_y

Different test done

- Comparison between unet and gan
- Effect of adding image augmentation to the training set
- Effect of adding the l2 edge loss and the constant loss to the network.
- Intersection over union is used as a metric for the tests

Current results 1/2

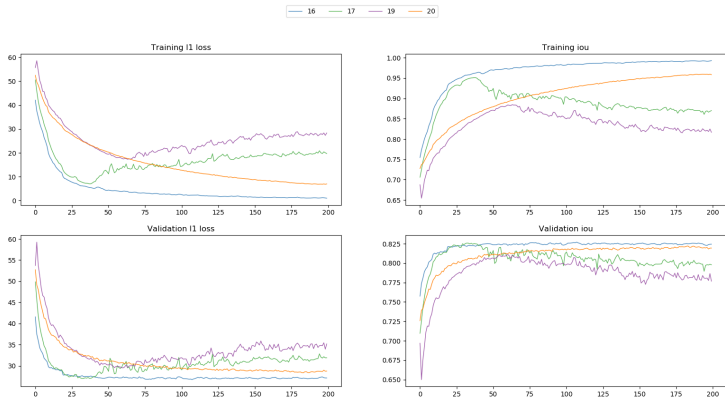


Figure: Comparison between unet and gan.

blue: unet, orange: unet with data augmentation
green: gan, purple: gan with image augmentation

Current results 1/2

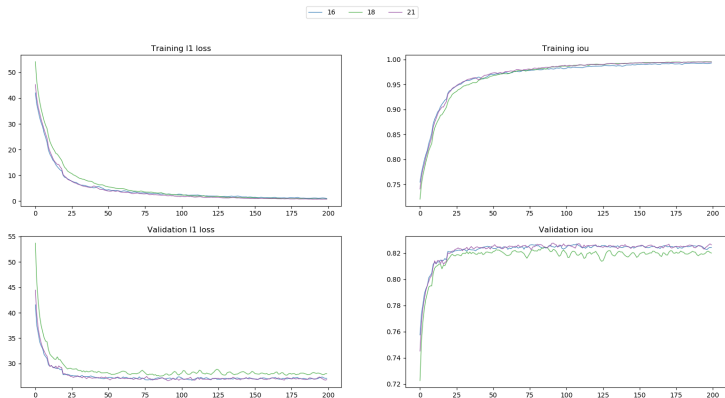


Figure: Effect of different losses

blue: no edge loss, no const loss, green: edge loss, no const loss
purple: const loss, no edge loss

唔 唔 唔 五
局 局 尸 可
诌 诌 诌 十

Figure: Validation sample

- Generalization to multiple fonts
- Train a network on a single character in multiple fonts
- Define precise definitions for all characters
- Extend database to contain minimal character set

- R. S. Zimmermann, J. N. Siems, "Faster Training of Mask R-CNN by Focusing on Instance Boundaries", arXiv:1809.07069v3, (2019).
- Yuchen Tian, zi2zi, (2017), GitHub repository, <https://github.com/kaonashi-tyc/zi2zi>.