**Testing Tesseract-OCR**

Tesseract-OCR was able to completely read in and recognise a sample of simplified Chinese writing, as shown in Fig. 1:

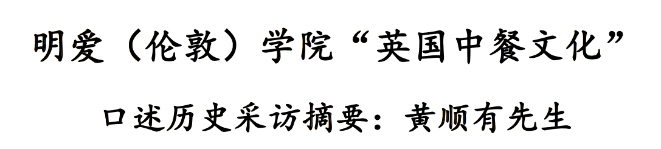


Figure 1: Sample of Chinese writing, fully recognised.

*Transliteration:* 明爱 (伦敦〉 学院 “英国中餐文化”

口迷历史采访摘要: 黃順有 先生

*Translation:* Caritas (London> College "British Chinese Culture" Interview with Mr. Huang Shunyou

When tested on a stylized form of Chinese writing, as in Fig. 2, Tesseract-OCR recognised zero characters.

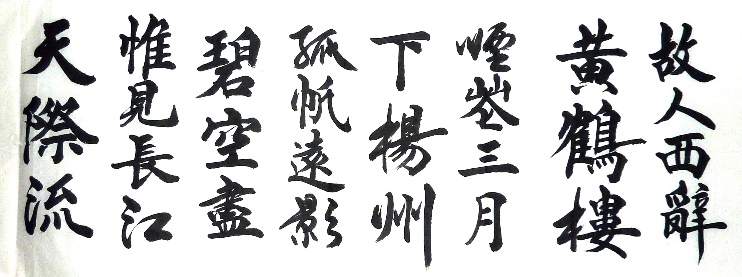


Figure 2: Example of stylized Chinese writing, no characters recognised.

Images of Unicode Chinese characters were generated using Python by generating the Unicode Chinese character for a specific Unicode number, then drawing that onto a blank canvas and saving it as a .png file. Examples of these can be seen in Fig. 3 where images have been generated of the Unicode characters 哀, 艾, and 阿.

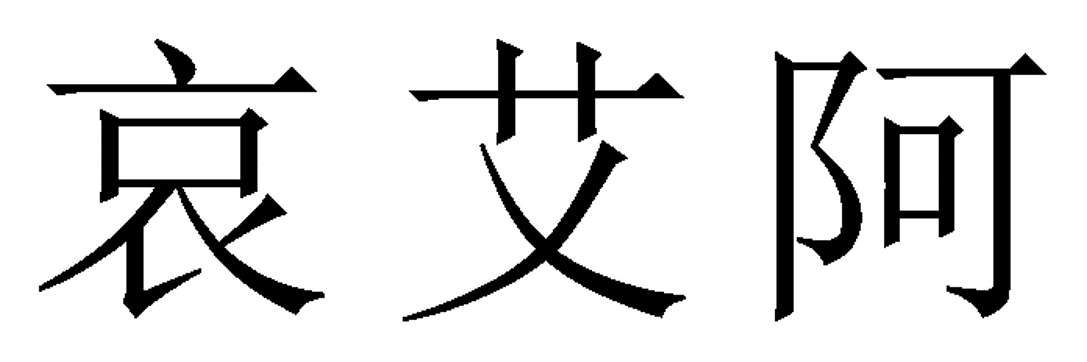


Figure 3: Images of Chinese characters generated from the Unicode characters: 哀, 艾, 阿.Pixel size of 200x200, border (white space) size of 32x32 pixels for the characters.

On a set of 30 such generated images, Tesseract-OCR recognises 4 (~10%). The accuracy changes to ~20% (23/100) if 100 images are tested. If the pixel size is increased to 400x400 pixels for the character, Tesseract-OCR accuracy diminishes to only 18 of the same 100 characters recognised correctly.

This seems strange, considering Tesseract-OCR’s accuracy before on a whole set of Chinese characters in Fig. 1. Tesseract-OCR is tested again on a set of Chinese characters in Fig. 4 to confirm its use. It recognised all the characters correctly in the simplified Chinese sample in Fig. 4.



Figure 4: Sample of simplified Chinese characters, which Tesseract-OCR recognised with complete accuracy.

*Transliteration:* 人人生而自由， 在尊严和权利上一律

平等。 他們賦有理性和 良心， 并应以

兄弟关系的精神互相对待。

*Translation:* Everyone is born and free in dignity and right equality. They are given reason and conscience and should be The spirit of brotherhood is treated with each other *(article 1 of the Universal Declaration of Human Rights).*

What if we string a series of Chinese Unicode characters together in a sentence-like form such as above? This is done in Fig. 5 for 10 characters. These 10 characters formed the first 10 characters of the 30 character-generated-dataset described above, where Tesseract-OCR only achieved 10% accuracy. Surprisingly, Tesseract-OCR has perfect accuracy on reading these characters, with the only difference being is that they are now in a sentence-like format. The pixel size is maintained at 300x300.

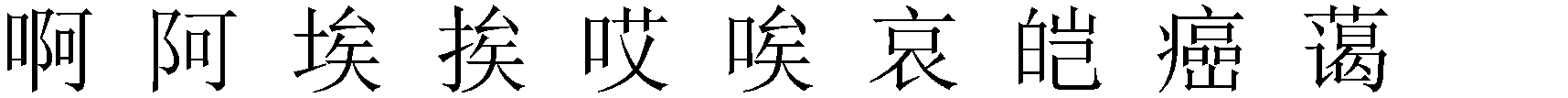


Figure 5: 10 Unicode-generated characters from the same generated data set as before, now put in a string. Tesseract recognises these characters with 100% accuracy.

*Transliteration: 啊 阿 埃 挨哎唉哀皑癌蔼*

Clearly, Tesseract-OCR has a problem with isolated characters. Is there a limit to how few characters we can put in a sentence-form before Tesseract stops reading them in correctly? Fig. 6 shows three pairs of Chinese characters individually fed into Tesseract-OCR. Set 1 was correctly identified, set 2 was unidentified, and set 3 was misidentified.

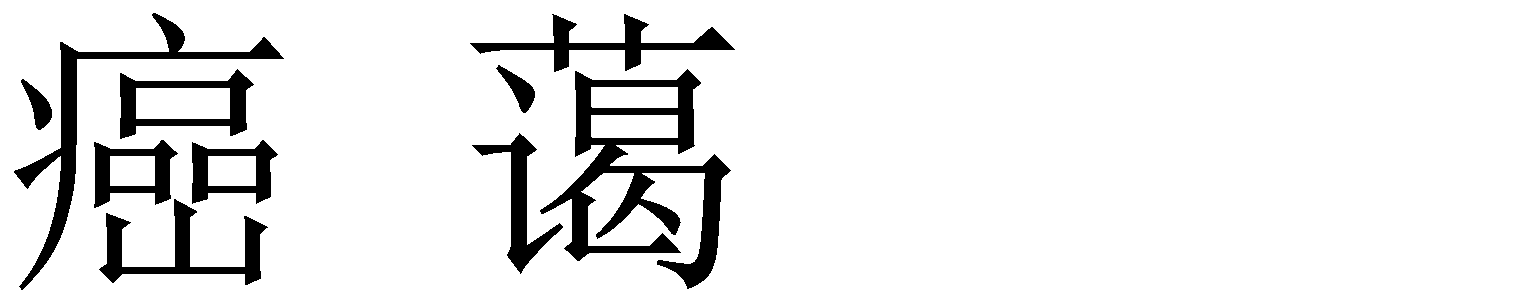
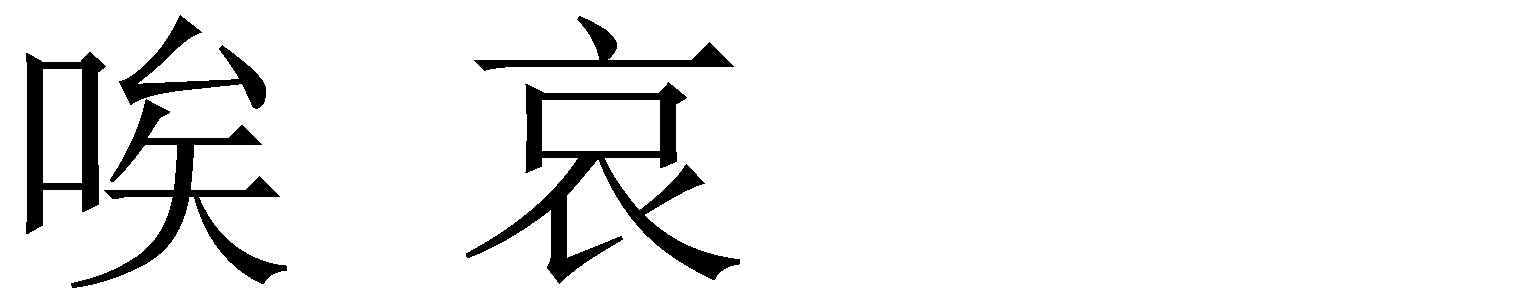
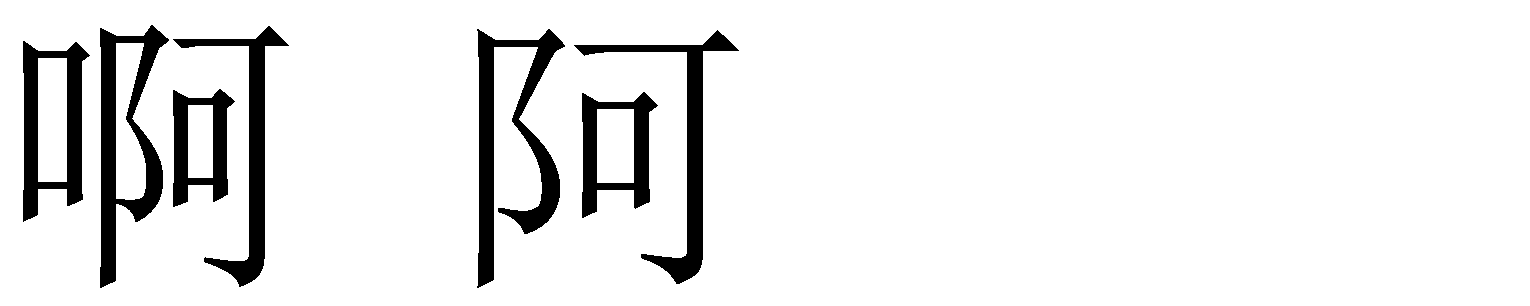


Figure 6: 3 sets of character pairs tested with Tesseract-OCR. Set 1 was identified correctly, set 2 was not identified, set 3 was mis-dentified.

Perhaps Tesseract-OCR working is instead a function of the size of the character in relation to the size of the image. 30 isolated characters corresponding to the initial 30 generated character dataset of size 50x50 pixels on a white background of total image size 2000x2000 were fed through Tesseract-OCR. Tesseract achieved 10% accuracy, similar to previous tests.

It seems that Tesseract-OCR does genuinely require sets of characters at a time, and cannot function on individual characters. There even seems to be a limit to the character size. Setting characters to be 2000x2000 pixels large reduces Tesseract accuracy.

This is a huge limitation in using Tesseract, and looking forward we do not think we can use Tesseract in our implemented solution.