

Reviewer #1 Concern #1: The paper is devoted to construction of a dataset of Uchen Tibetan handwritten characters, training and testing classifier network thereof. The work contains all necessary parts: introduction, related work, methodology, experiments, and conclusion. English language itself has some flaws but this does not affect understanding.

Author action: Thank you for your comments. We have carefully checked the grammar and expression of the manuscript, and revised and improved the English language problems.

Reviewer #1 Concern #2: The presented study can be split into two parts: the first is collecting the database and the second is training and testing the classifier. The first part in its turn consists of character annotation, character image extraction, and dataset construction/augmentation.

Author action: Thank you for your comments. In order to show our work more clearly, we split the main structure of the paper into two parts: character dataset construction and character recognition.

Reviewer #1 Concern #3: Annotation is done manually by experts. The question is: were there any contradictions in annotation? I can hardly believe that all eight experts have annotated completely equal. How many contradictions were there and how they were resolved?

Author action: Thank you for your questions. Because we did not express clearly the contents for annotation in the paper, which caused a little misunderstanding to the reviewers. We have carefully revised the expression of this part: “To ensure the accuracy of character annotation, we choose 8 Tibetans whose mother tongue is Tibetan and who have completed their graduate education as annotators to complete the annotation. Their annotation tasks do not overlap. An expert in the field of Tibetan language studies proofread their annotation results.”

Reviewer #1 Concern #4: Connected question. At string 301 (page 11) it is claimed that 56% of data were used. How these data were selected?

Author action: Thank you for your question. To ensure the annotation quality, the annotation results need to be proofread repeatedly, and the annotation process is time-consuming and laborious. In addition, the images of historical Tibetan documents have different degrees of quality fading, which increases the difficulty of annotation. Therefore, after the steps of annotation and proofreading, each set of data has approximately 5500 character blocks available, which is about 56% of the total. The data with valid annotation accounted for 56% of the total, rather than 56% of the selected data.

Reviewer #1 Concern #5: Concerning character image extraction. I cannot understand it. As I can see from line 101 (page 3) and figures, the characters are already outlined in images with different colors (unique color corresponds to a single character) by 'previous work'. To my mind, starting from this point obtaining images of distinct characters and the order (position in image) is a schoolboy's task, which is hardly worth to mention. Just enumerate all unique RGB triplets in the image, find bounding box for pixels with each triplet, and sort the X coordinated of centers of bounding boxes in increasing order, that all. Why do the authors use Algorithm 1 and Algorithm 2 here? Especially for the Algorithm 1: is there any sense in a special (and very poorly described) algorithm of producing 32-bit integer from RGB triplet, while RGB triplet is by its nature already 24-bit integer?

Author action: Thank you for your questions.

(1) If the coloring order is fixed, it is really easy to extract characters in order from character blocks that have completed character segmentation in the previous work. However, the color values of characters between each character block in this work are not in a fixed order, character strokes are a certain degree of arbitrariness, and the character strokes are offset from left to right. Therefore, it is difficult to extract the characters in sequence by using the centroid or coordinate position information of the connected component, which easily leads to disorder between the extracted character numbering sequence and the sequence. To solve the above problems, we first realize color value conversion by performing secondary encoding on the color values of the RGB three channels into a 32-bit grayscale and then extract the single characters, which can avoid the possibility that the color values cannot be uniquely corresponding when the RGB image is converted to the gray-scale image.

(2) Algorithm 1 is simple to implement, so we have removed it.

Reviewer #1 Concern #6: In general, all 'algorithms' are poorly and incompletely described. For instance, where from do we take 'NsingleChar' value in Algorithm 3? Or how do we rotate strokes in Algorithm 4? How do we extract upper and lower strokes? What does it mean 'separate images'? And so on.

Author action: Thank you for your questions.

(1) We have revised three algorithms into clearer pseudo code form.

(2) In the previous version, 'NsingleChar' is the number of characters after extraction.

(3) The specific operation of rotating strokes is to rotate the stroke 90 degrees counterclockwise.

(4) The extraction of upper and lower strokes is completed by using the Y coordinate of the centroid. First, the connected component analysis of strokes is carried out. Second, upper and lower strokes are determined according to the size relationship of the Y coordinate of the centroid. Final, the upper and lower strokes are extracted.

(5) In the previous version, 'separate images' refer to the upper and lower strokes after extraction, and we have revised this expression.

Reviewer #1 Concern #7: The description of algorithms (and the whole first stage) should be rewritten in clear manner by using formulas and formal definitions.

Author action: Thank you for your comments. We have revised three algorithms into pseudo code.

Algorithm 1: Character extraction algorithm based on real location information

Input: Encoded character block ECB

Output: Single character SC

```
1: H, W ← size function(ECB)
2: NC ← count the number of colors in ECB
3: for l = 1: 2: H do
4:   ECBL ← ECB(1: l,:) = 0, ECB(l+2, :) = 0
5:   NCL ← count the number of colors in ECBL
6:   if NC == NCL
7:     for c = 1: NC do
8:       for h, w = 1: H, 1: W do
9:         if ECB(h, w) ≠ NCL(c)
10:          ECB(h, w) = 0
11:         end if
12:       end for
13:       SC ← cropping function(ECB)
14:     end for
15:   end if
16: end for
```

Algorithm 2: Construction of the algorithm of the character dataset based on real position sequence matching

Input: Single character annotations SCA, Single characters SC

Output: Preliminary character dataset PCD

```
1: NSCA  $\leftarrow$  count the number of SCA
2: for  $d = 1 : NSCA$  do
3:   MN  $\leftarrow$  read the mapping name of SCA[d]
4:   ISEF  $\leftarrow$  exist the class folder of SCA[d] or not
5:   if not ISEF
6:     CF  $\leftarrow$  create the class folder of SCA[d]
7:     the MN in SC is written into CF //the MN corresponds to a single character name
8:   else
9:     the MN in SC is written into CF
10:  end if
11:  PCD  $\leftarrow$  save function(CF) //each CF represents a character class
12: end for
```

Algorithm 3: Special character synthesis algorithm based on the random distance between upper and lower strokes

Input: Special symbols SS

Output: Synthesized special symbols SSS

```
1: NSS  $\leftarrow$  count the number of SS
2: for  $s = 1 : NSS$  do
3:   SSU, SSL  $\leftarrow$  extract the upper and lower strokes of SS //use the Y coordinate of centroid
4:   SSUs, SSLs  $\leftarrow$  rotate function(SSU, SSL) //rotate 90 degrees counterclockwise
5:   RI  $\leftarrow$  generate a random integer in [-2,12] //[-2,12] is the priori distance range
6:   SSL, SSU  $\leftarrow$  SSUs, SSLs //the upper(lower) stroke is as the lower(upper) stroke of SSS
7:   SSS  $\leftarrow$  synthesis function(SSL, SSU, RI)
8: end for
```

Reviewer #1 Concern #8: The second part conforms to standard practices of this kind and does not arise obvious questions. Maybe the questions will appear after clarifying the previous part.

Author action: Thank you for your suggestions. We have revised the first part (dataset construction) and improved the second part (character recognition).

Reviewer #1 Concern #9: Minor drawbacks. 1). Figure 5. As I can see, (b), (c), and (d) are G, B, and R components of (a) rather than R, G, B. 2). Figure 5. (e) is a plain grey square, useless, should be removed.

Author action: Thank you for your suggestions. We carefully checked the content involved in Figure 5 and found that it was not a necessary part. We have removed Figure 5.