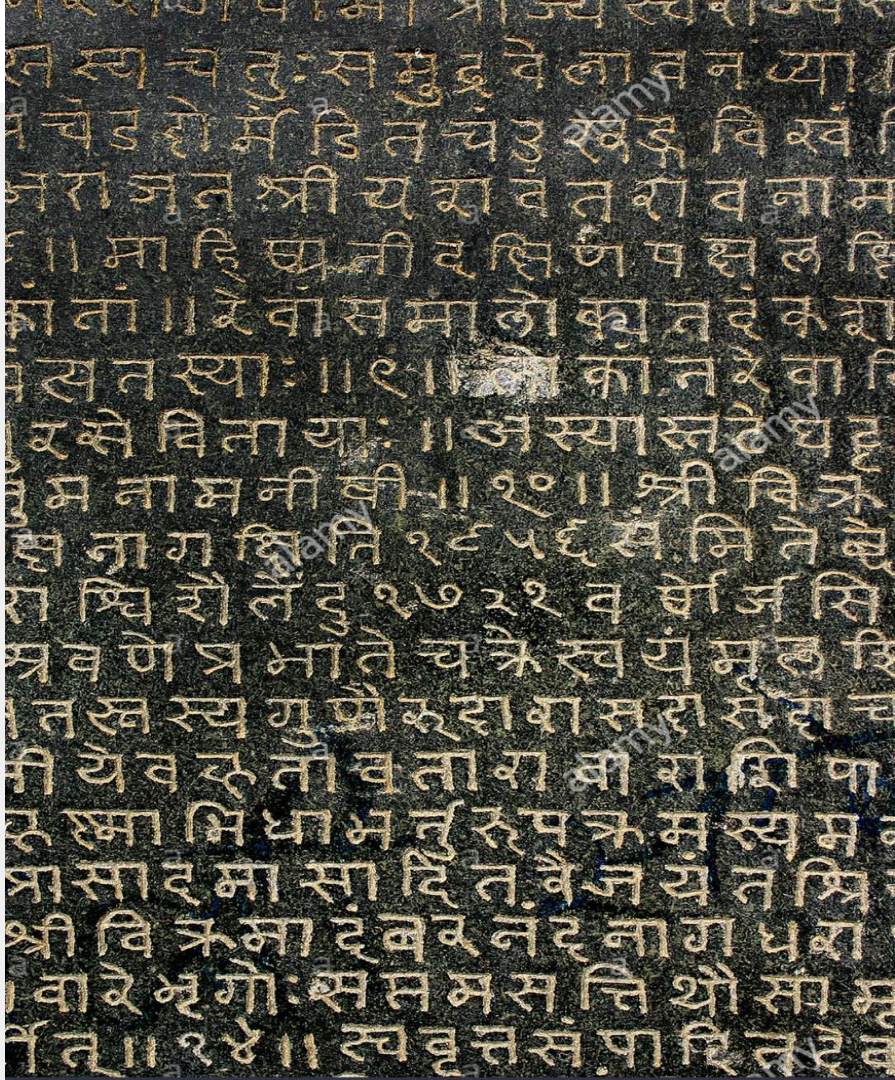
TEXT EXTRACTION FROM STONE INSCRIPTIONS USING CHARACTER SPOTTING

TA: Prathyakshun Rajashekhar

Sivani Pentapati (20171174)
Mayukha Lingampally(20171072)
Pranathi Bhuvanagiri(20171137)



CONTENTS

- ABSTRACT
- REQUIREMENTS
- WORKFLOW
- PROCEDURE AND NOVELTY
- CHALLENGES
- RESULTS

ABSTRACT

We implemented a semi-automated method for extracting text from images of stone inscriptions using the novel method of character spotting.
The project consists of robust character-analytic elements which include:

- Denoising with cascaded filters
- Pruning for ROI using NCC
- HOG based feature matching
- Unicode labelling to generate the text file

The deliverables include

- Editable text file for further analysis of the content of inscriptions.

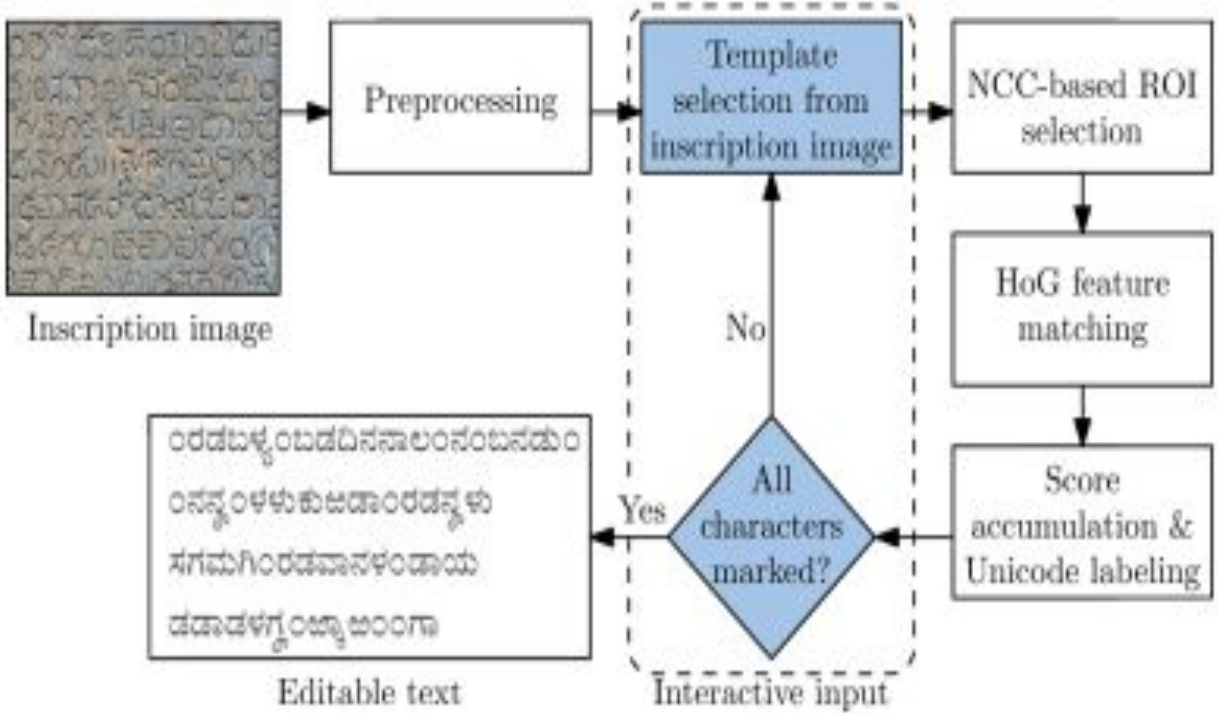
REQUIREMENTS

This requires a knowledgeable user, who is acquainted with the script, to convert the inscription to an editable text document.

The spotting process is robust to writing style and font, with a reasonable presumption that the entire inscription has a uniformity of script.

This framework supports only inscriptions in sanskrit or hindi script.
A labeling is made by the user through an interactive process, which is the input to subsequent processes.

WORKFLOW


Inscription image
Preprocessing
Template selection from inscription image
NCC-based ROI selection
HoG feature matching
Score accumulation & Unicode labeling
All characters marked?
No
Yes
Editable text
Interactive input

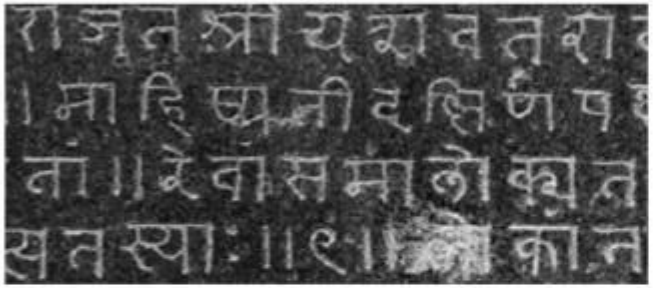
PROCEDURE

PREPROCESSING

- The following denoising algorithms in order are used for denoising of the input image
- The degraded grey image is first thresholded using Otsu’s method
- The thresholded image is later passed through a median filter(3x3) in order to remove the remaining salt and pepper.
- The image is then closed to counter any unnecessary deformation of characters due to median filter.

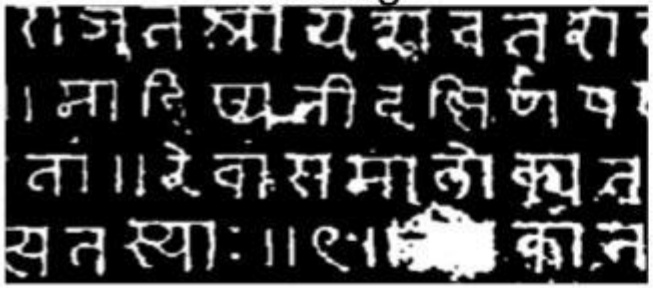
Outputs after different passes in the denoising block

original

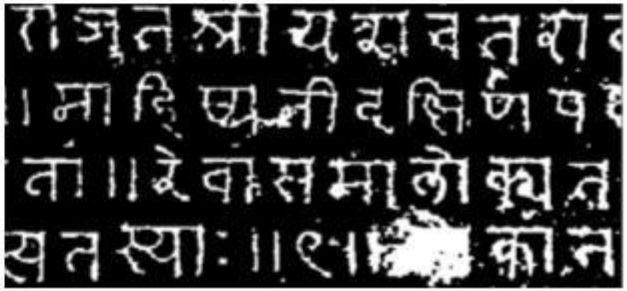


Otsu's Thresholding

closing



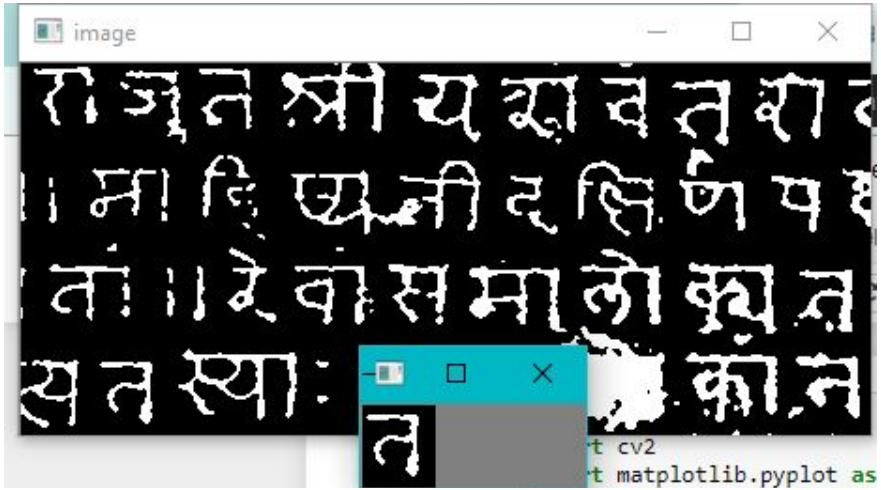
Median filter



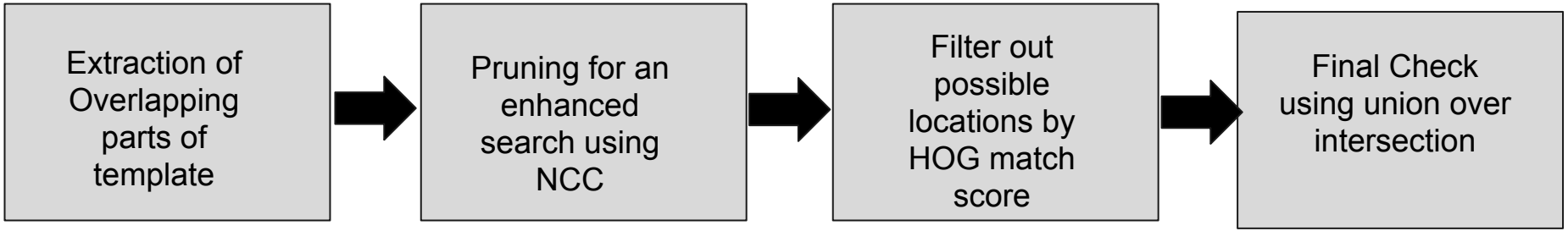
Selection of character by user

The user is asked to select a character (template) from the displayed image and also provide the unicode corresponding to the particular character in the modern script.

This process is repeated until all the distinct characters are marked.



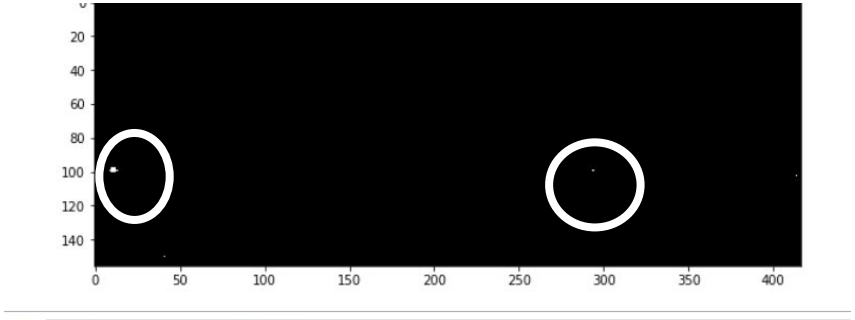
Template matching with its sub-blocks



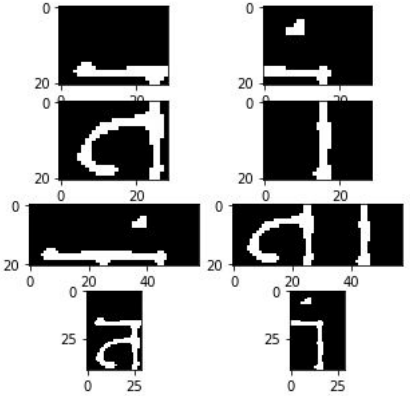
NCC based ROI selection

- The selected template is first split into 8 overlapping regions and for every region NCC (Normal cross correlation) is computed between the all possible locations in the original image and the overlapping part of the template
- The NCC matrix is then thresholded and the locations of the matched character centres are given by the white pixels.
- We observed clustering of high correlations for a match. The mean of all regions belonging to a cluster is used to represent the location of character
- The possible centers obtained are passed for HOG matching for further refinement

Outputs after thresholding using NCC



Circled regions represent the regions of maximas in the correlation image



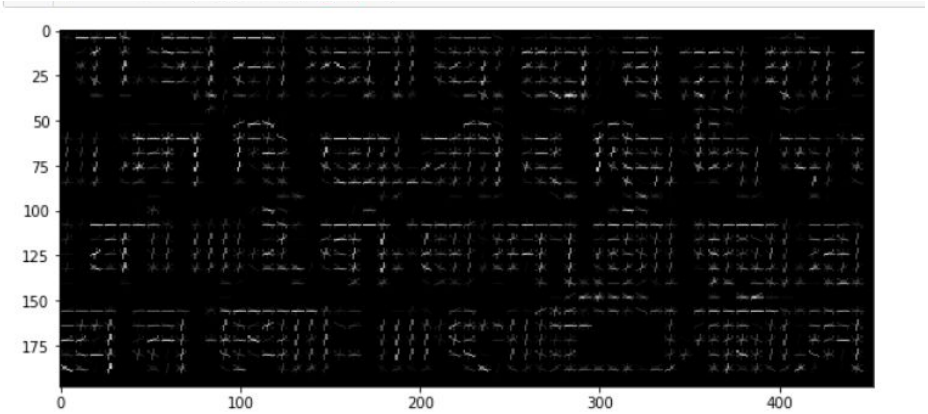
Obtained overlapping templates for a character

Enhanced matching using HOG character descriptors

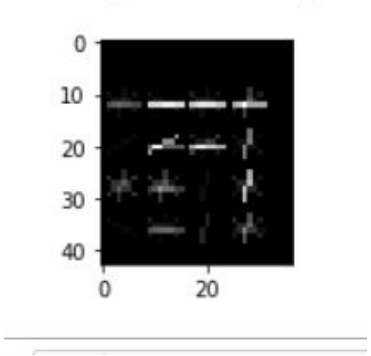
Computation of HOG

- A HOG mapping with a cell size of 2x2 and a block size of 8x8 with a bin size of 9 is used to obtain the feature vector of length 36 (4 blocks x 9 elements for each)
- For all locations flagged by NCC:
  - Matching score using Dot product is computed between the hog vector of the patch in the image with all the 8 overlapping parts of the template and is thresholded for extraction of character centers

Obtained images after computing HOG



HOG map for the entire image



HOG map obtained for the selected template

Improvisation of the algorithm:

- As a part of further enhancement we are performing union over intersection for the binary images of patches and the character template as final check
- It will be considered as a match if and only if it crosses the thresholds in all the respective blocks of NCC,HOG,and union over intersection
- A union over intersection image for character *('ma')*

Score accumulation and unicode labelling

The output locations along with their mapped unicode are continuously appended into a text-file until all the characters in the image are marked.

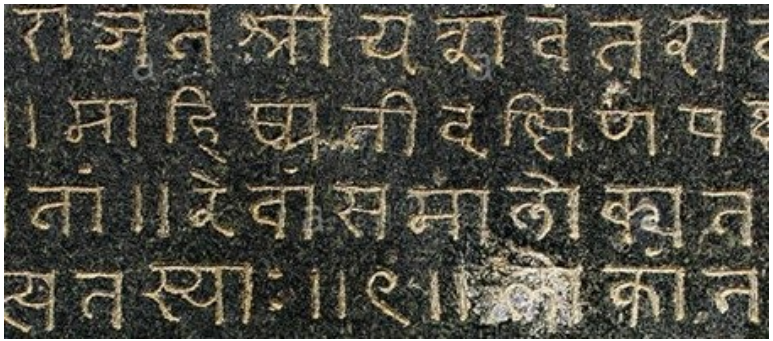
This process is done for every selected character and the text file with unicode and locations is converted into an editable file in the language of the inscription using a mapping script.

Challenges

- The results obtained after NCC couldn’t detect all the possible matches for some characters so we need multiple iterations for same character
- Area threshold cannot always resolve the ambiguity of dialects

Results

The editable text file that we got for the sample image:



```
1  ऱ ज न श य श व त थ
2  म ह ष न द घ ण प
3  त र व स म ल क त
4  स त स क न
```

GithubLink: https://github.com/mayukhalingam/DIP-Project/commits/master