22/9/2016

Not needed 74k char data set can be found here
http://francescopochetti.com/text-recognition-natural-scenes/

Image language detection can be found here http://www.bmva.org/bmvc/1997/papers/050/

Decided to follow our OCR and compare them with existing language detection in image techniques. Two methods using character and confidence model ours. Other that is in paper.

23/9/2016

All kind of features can be found here https://arxiv.org/pdf/1206.0238.pdf but can't use them for character

From text language can be found langid.py
http://www.anthology.aclweb.org/P/P12/P12-3.pdf#page=37

Language identification into 3 categories:
statistical analysis of text line, texture analysis, and template matching.
Statistical analysis:
http://www.umiacs.umd.edu/~zhugy/HandwritingLanguageID_PR2009.pdf
Texture analysis:
http://www98.griffith.edu.au/dspace/bitstream/handle/10072/4262/31447.pdf?sequence=1&origin=publication_detail
Template matching:
https://pdfs.semanticscholar.org/a495/7c1c2564e62d40fb91d5b7bbd57a8fcf203d.pdf

So we can solve drawbacks in them or combine them someway.

24/9/2016

Notes for Script and Language Identification from Document Images:
An uniform text area taken from image. Multiple channel (Gabor) filters and grey level co-occurrence matrices are computed then. K-NN classifier applied over these features. Language detection can be done with char segmentation or not. Here they gave methods for both of them. They followed without segmentation.

Notes for Language Identification by Using SIFT Features:
Size of char estimated then sliding window used to get SIFT features. Then visual vocabulary or fischer vector generated for each image and SVM used to classification. Vocabulary they have done in two ways. Bag of features and vector. Then using SVM classification done. Bag of features is not that good.

Notes for Script and Language Identification in Noisy Degraded Document Images:
Transform each image into electronic document vector. Then by constructing these vectors over train set. Test using distance from trained vectors. Some features from images are generated using language knowledge.

Can we use mixture of features from above papers and do classification correctly.

26/9/2016

Reading Script and Language Identification from Document Images:

Steps to implement:
Skew compensation DONE
Non textual info removal DONE
Horizontal projection profile then limited smoothing DONE
Non overlapping lines checked DONE
Outsize text lines removed DONE
White space between lines made same DONE
Left justification DONE
Vertical words space made < 5px SKIP
Right padding DONE
left padding, DONE
top padding also. DONE
Bottom padding DONE
Top left block taken DONE
Gabor filtering DONE With some doubt
Knn classifier

27/9/2016

Applications can be in soft copy of the document images in any language website. Also if you have a book in hard copy and you want to store it for soft copy then you can use our application directly.
Got datasets for english and hindi.
https://www.freepdfconvert.com/pdf-image

PDF to jpg
http://pdftojpg.me/

30/9/2016

Median filtering

```
source = cv2.imread("Medianfilterp.png", CV_LOAD_IMAGE_GRAYSCALE)
final = cv2.medianBlur(source,3)
```

Black and white

```
from PIL import Image image_file = Image.open("convert_image.png") # open colour image image_file = image_file.convert('1') # convert image to black and white image_file.save('result.png')
```

Dilation and erosion

4/10/2016

While aligning lines we are aligning wrt centre of each line

Gabor filtering : http://scikit-image.org/docs/dev/auto_examples/plot_gabor.html
http://www.it.lut.fi/project/simplegabor/downloads/src/simplegabortb/simplegabortb-v1.0.0/htmldoc/simplegabortb-v1.0.0/
http://freesourcecode.net/matlabprojects/57293/gabor-filtering-on-an-image-in-matlab#.V_U_bHV97Eo

sigma = lambda/pi*sqrt(log(2)/2)*(2^bw+1)/(2^bw-1);

http://freesourcecode.net/matlabprojects/57291/gabor-filter-in-matlab#.V_VCo3V97Eo

8/10/2016

Paper we are implementing
http://www.bmva.org/bmvc/1997/papers/050/

http://hanzratech.in/2015/05/30/local-binary-patterns.html

http://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

Now on we are implementing LBP for image classification.

10/10/2016

Implemented LBP

Getting 100 percent accuracy
Worst case ocr online
http://www.onlineocr.net/
http://www.ocrconvert.com/
https://www.newocr.com/
http://www.free-ocr.com/

Online api for text from image when langauge known:
http://www.ocrwebservice.com/api/restguide

One application can be finding language by LBP method and then OCR with online api.

24/10/2016

Tilting angle invariant as LBP is rotation invariant
All datasets:
http://www.iapr-tc11.org/mediawiki/index.php/Datasets_List

Most people using this ocr
https://github.com/tesseract-ocr/tesseract
https://pypi.python.org/pypi/pytesseract

Scanned images

https://www.google.co.in/search?q=scanned+images&espv=2&tbm=isch&imgil=kFuec44gnmMfaM%253A%253Bh0Quj3Safbeh5M%253Bhttp%25253A%25252F%25252Fwww.ampercent.com%25252Fconvert-scanned-pdf-documents-to-text-using-google-docs%25252F9745%25252F&source=iu&pf=m&fir=kFuec44gnmMfaM%253A%252Ch0Quj3Safbeh5M%252C_&usg=__RUpqKb5AWzuiKZ9qgWjvVTlU4pM%3D&biw=1366&bih=638&ved=0ahUKEwjW6OW6kfPPAhVBNo8KHbobCO4QyjcIPA&ei=hdUNWJaSM8HsvAS6t6DwDg#imgrc=kFuec44gnmMfaM%3A

https://www.google.co.in/search?espv=2&biw=1366&bih=638&tbm=isch&sa=1&q=telugu+scanned+images&oq=telugu+scanned+images&gs_l=img.3...56577.61595.0.61795.21.18.0.0.0.0.363.2077.0j1j4j3.8.0....0...1c.1.64.img..13.5.1072...0j0i67k1.JZaUfLugptk#imgrc=4xl5EH3eLhTIwM%3A

https://www.google.co.in/search?espv=2&biw=1366&bih=638&tbm=isch&sa=1&q=hindi+scanned+images&oq=hindi+scanned+images&gs_l=img.3...79089.87503.0.88338.13.10.0.0.0.0.589.1663.2-1j1j1j1.4.0....0...1c.1.64.img..9.2.1019...0j0i24k1.Bs1IUy51gu8#imgdii=tX8BXIVLaCBYSM%3A%3BtX8BXIVLaCBYSM%3A%3BckJCYWwD5x-6PM%3A&imgrc=tX8BXIVLaCBYSM%3A

In worst case you can use this tesseract OCR
https://pypi.python.org/pypi/doc2text

25/10/2016

Scanned images also working properly if images are clear
Removing images
Tilting not necessary as LBP rotation invariant
Histograms also different
Baseline need to decide

We can implement template matching for comparison.

Notes for Automatic Script Identification from Images Using Cluster-based Templates:
From training images we need to make templates of symbols which are occurring frequently. Then when new image comes, we take only a few symbols and compare them with the present symbols and decide the language.

Steps to implement:
How about taking symbols from training and comparing similar symbols in test and then classification. I think it is best idea.

Make symbols for english alphabets. And then match them with number of occurrences in the test image.

http://stackoverflow.com/questions/34690774/opencv-template-match-similar-object
Search template in a image python

26/10/2016

Skew angle compensation completed within -10 to 10 degrees possible
Skew compensation
http://engr.case.edu/merat_francis/EECS%20490%20F04/References/Document%20Deskew/00619830.pdf

27/10/2016

Implementing template matching
http://stackoverflow.com/questions/32664481/matlab-template-matching-using-fft

If templates are ready, then can find how many times it came in test image
How to make templates from train images

Take templates from online and resize it according to test image.
Or else crop from train images
Found templates
http://www.symbols.com/gi.php?type=1&id=237&i=1
https://en.wikipedia.org/wiki/Telugu_script
http://tdil-dc.in/tdildcMain/articles/534028Devanagari%20Script%20Behaviour%20for%20Hindi%20%20ver%201.4.7.pdf

Crop and try
Resize and try
Dinally break them to templates DONE
Algo for classifying test images
Accuracy very pathetic with templates from net

So make templates from training images.

2/11/2016

Interpreting

Justify by shape
Character wise histogram

4/11/2016

Minimum lines/words DONE
Baseline implement almost complete just need to run DONE
Histogram analyse DONE

Because making templates manually is very tedious, going with internet templates.

7/11/2016

Baseline is giving accuracy of about 73% only.

Because we have noise. We have got peak for >23 for all languages. So we have removed that part. May be this noise part is wrongly interpreted. May it because of simple curves in all languages.

Telugu has a peak at lower, middle, high integers.
As there are more curves in telugu than straight lines. And 45, 135 degrees curves at both sides result in these 3 peaks.
English has peak at lower, middle, somewhat low to high integers.
This is because of many vertical lines present in english.
Histogram we are interpreting as hindi has peak in the middle.
Hindi has many horizontal lines and many curves. So as we can see LBP figure we get more 1s at centre of the eight bit integer and they outnumbered remaining integers. So we have peak at center. Actually we think we need get peak at high integers also but results shown that this is not true.

4 lines is the limit.

NEED TO MAKE POSTER
https://drive.google.com/drive/folders/0BxfIOSlQJT3hcDlfbnAwc29OdlU?usp=sharing // IITR posters
https://en.wikipedia.org/wiki/Poster_session
http://people.eku.edu/ritchisong/posterpres.html
http://nau.edu/undergraduate-research/poster-presentation-tips/

8/11/2016

---

https://projects.ncsu.edu/project/posters/examples/Flounder/ GOOD one poster
https://projects.ncsu.edu/project/posters/examples/Manatees/ GOOD one poster

take template from here
http://www.craftofscientificposters.com/templates.html
http://www.posterpresentations.com/html/free_poster_templates.html
http://www.makesigns.com/SciPosters_Templates.aspx
https://www.genigraphics.com/templates

Last 3 awesome

After templates, make poster in powerpoint http://people.eku.edu/ritchisong/posterpres.html

Poster headings

Title DONE
Members names DONE
bstract DONE
Introduction DONE
Methods DONE
First Approach DONE
Second Approach DONE
Uniqueness and Contribution DONE
Experimental Analysis
Results DONE
Conclusion DONE
Future Work DONE
References DONE

Make all headings and then choose template to present them.

Quality work done:

Made gold standard dataset
Literature survey about 3 methods of language detection
Implemented and can compare texture, template matching
While implemented both methods, we have chosen good algorithm for each step and made robust
Compared with scanned images performance.
For implementing we tried different methods for texture, templates.
Can show sample results for each step, final 2 methods results.

Some number:
360 train images, 120 test images for texture approach

9/11/2016

Need to make poster

Will copy everything from first 3 papers at beginning.
Title -
Texture, Template matching approaches for Language Detection in document images.
Members names -
Deepak Jannarapu, Nallagatla Manikanta, Vinod Pankajakshan IIT Roorkee.
Abstract -
The problem of recognising the language of a document image has large number of applications in many fields like document analysis such as indexing the document image, as an initial step

for optical character recognition. A uniform text block on which texture analysis can be performed is produced from a document image via simple image processing. We compare two approaches for language detection - texture and template matching. Local binary patterns are used for texture features and for template matching character templates for three languages english, hindi, telugu were hand picked from respective scripts. Classification of test document images is made based on their comparison with the features of training documents. Comparing both approaches, texture features, achieved over 95% accuracy on the classification of 60 test document images from 3 languages are very promising over 73% accuracy with template matching features. Gold standard dataset for document images is also made.

Introduction-

The world we live in is becoming increasingly multilingual and, at the same time, increasingly automated. Hence the need of automatic language identification is increasing. This topic of research is continuing to be a fundamental research problem in fields of Optical character recognition(OCR) and document image analysis.OCR tools for different languages had been developed but these works fine only for specific languages. Almost all existing work on OCR makes an important assumption that the language of the document to be processed is known beforehand. Even in the applications dealing with many languages the language should be specified for further processing.

Various methods have been developed for language detections. But among these 3 methods namely texture analysis, template matching and statistical analysis are found to be accurate and feasible.

we have implemented two of those methods ,texture analysis and template matching.

In texture analysis, documents are classified on the basis of texture features known as local binary patterns. Document images must be normalized to ensure accuracy and algorithms for such preprocessing stages, includes gre scaling, skew detection and correction, and normalization of text.The template matching identifies textual symbols in a representative set of training documents, clusters them, and calculates each cluster's centroid, or pixel-by-pixel average. This serves as a representative symbol, or template, for the cluster. To identify the script used in a new document, we compare a subset of its symbols to the templates for each script, and choose the script whose templates provide the best match.The details about these methods are explained in the further sections.

Diagram https://www.draw.io/#G0B55x8fIQc5xTM3JROUdoeFJuWk0

Methods -

The image processing part for both methods is same as shown in figure. After image processing, respective features are extracted from the image.

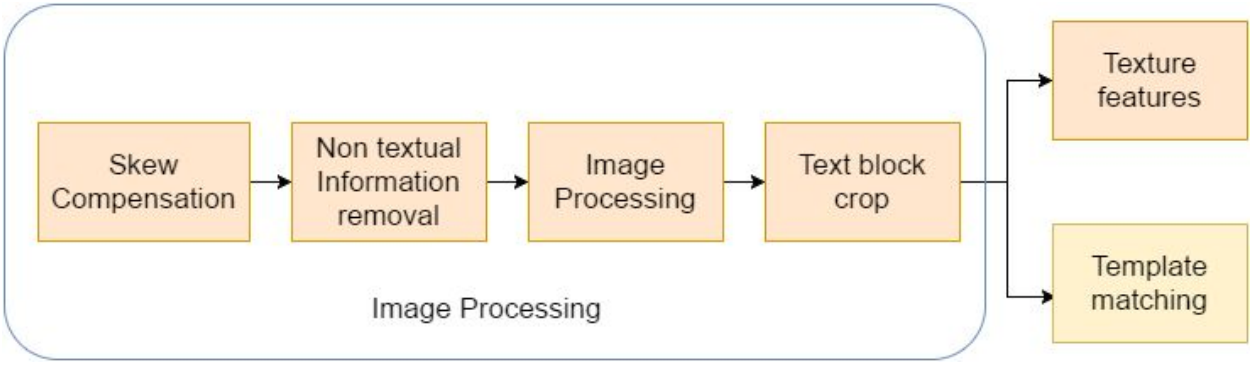
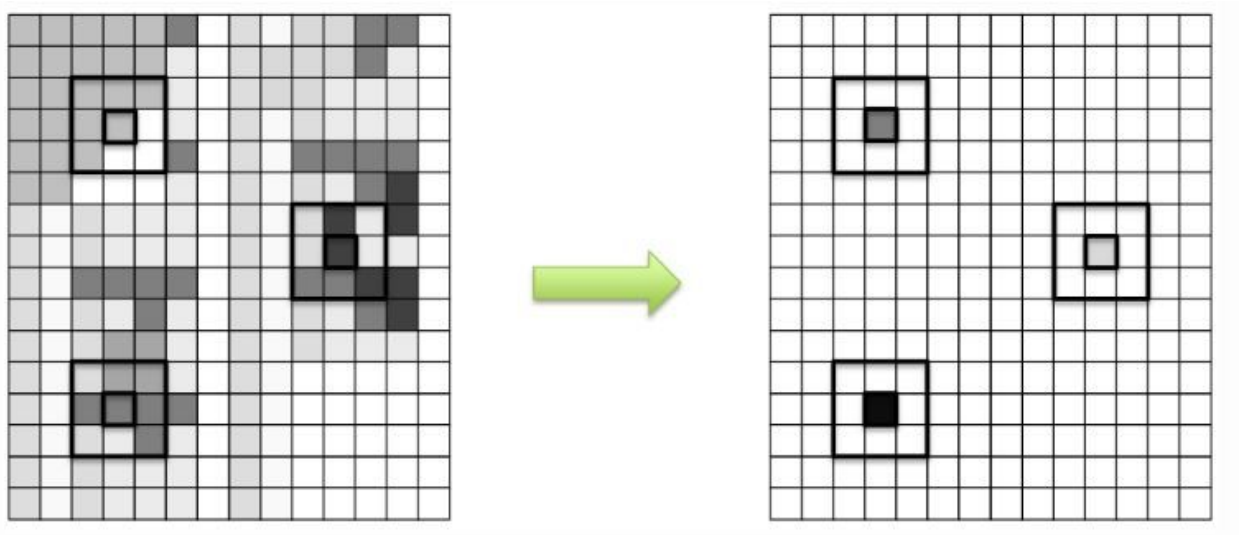


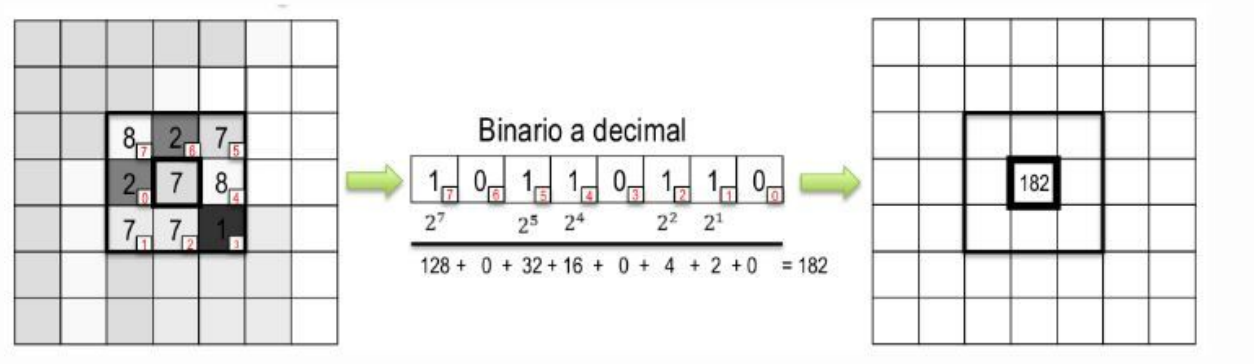
Image Processing:
Document image is converted into grayscale image using default threshold mechanism in python and then if the document is scanned with a skew angle it is compensated using Gradient direction. Then graphics, pictures are removed using connected components algorithm. The text may contain lines with different point sizes and variable spaces between lines, words and characters. Punctuation symbols and foreign characters may appear.

The horizontal projection profile (HPP) of the document is computed and smoothed. The peaks correspond to the centre of the text lines, and the valleys correspond to the blank areas between lines. Text lines are checked to ensure they do not overlap. All lines with a height much greater or much smaller than the mean line height are removed. Once the vertical bounds of each line are known, we can adjust the lines so that all line spacings are set to be the same predefined value. Then Left justification is done with 5pixels left padding. Similar approach is followed to achieve inter-word spacing is normalized to a maximum of 5 pixels: All runs of white pixels greater than 5 pixels wide are reduced to 5. This step forces the maximum gap between two characters to be 5 pixels. Gaps smaller than this are allowed to remain, because they may well, depending on the language. Text block is cropped from the image and padded on all sides with predefined value.

Texture approach:
Using local binary patterns texture features are extracted. For each pixel in the grayscale image, a neighbourhood is selected around the current pixel and then we calculate the LBP value for the pixel using the neighbourhood. After calculating the LBP value of the current pixel, we update the corresponding pixel location in the LBP mask as shown below.



To calculate the LBP value for a pixel in the grayscale image, we compare the central pixel value with the neighbouring pixel values. Since there are 8 neighbouring pixels – for each pixel, we will perform 8 comparisons. The results of the comparisons are stored in a 8-bit binary array. If the current pixel value is greater or equal to the neighbouring pixel value, the corresponding bit in the binary array is set to 0 else if the current pixel value is less than the neighbouring pixel value, the corresponding bit in the binary array is set to 1. The whole process is shown in the image below.



Then calculate the LBP Histogram and normalize it for the features.Same process is followed for test image and calculated chi square distance with the features of train images for classifying the test image.

Template matching approach:
Templates for characters of three languages english, hindi, telugu are hand picked from their scripts. Then after performing image processing as above, the instances of these templates are searched in test document images.
Sample of templates for each language are shown in below figure.

A	B	C	D	E	F	G	H
a	b	c	d	e	f	g	h

English Templates

క	కా	కి	కీ	కు	కూ	కృ	కౄ
ఖ	ఖా	ఖి	ఖీ	ఖు	ఖూ	ఖృ	ఖౄ

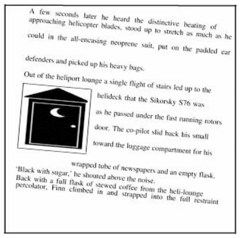
Telugu Templates

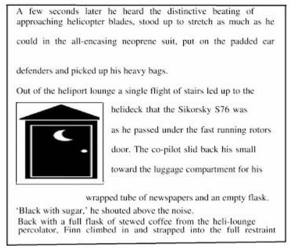
क	क़	ख	ख़	ग
ग़	घ	ङ	च	छ

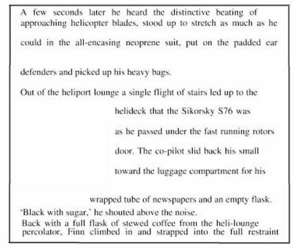
Hindi Templates

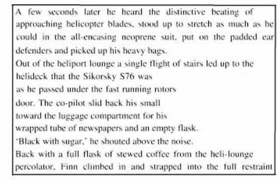
Uniqueness-
Texture features approach is rotation invariant as local binary patterns are rotation invariant. As we are using texture for language classification texture approach is independent on number of lines, size of image. This approach is also working for scanned documents also which has various applications like indexing, digitalizing a hard copy.
Experimental Analysis not in poster-

Results-
The color images used were taken as screenshots from english, hindi, telugu content pdfs. 80 document examples were chosen for each of: english, hindi and telugu. Images contain graphics, and would resemble reasonably closely the output from a document segmentation system. Also Foreign characters, numerals and italicized text were present in many of the images. Figure below shows step by step processing of a sample image used in the experiments. The images were divided into 60 training and 20 test images per language. For non text removal, connected component of more than 2.25 times average area of a character is removed. For template matching, 52 for english, 425 for hindi, 756 for telugu template characters are hand picked.


A few seconds later he heard the distinctive beating of approaching helicopter blades, stood up to stretch as much as he could in the all-encasing neoprene suit, put on the padded ear defenders and picked up his heavy bags.
Out of the heliport lounge a single flight of stairs led up to the helideck that the Sikorsky S76 was as he passed under the fast running rotors door. The co-pilot slid back his small toward the luggage compartment for his wrapped tube of newspapers and an empty flask.
'Black with sugar,' he shouted above the noise.
Back with a full flask of stewed coffee from the heli-lounge percolator, Finn climbed in and strapped into the full restraint


A few seconds later he heard the distinctive beating of approaching helicopter blades, stood up to stretch as much as he could in the all-encasing neoprene suit, put on the padded ear defenders and picked up his heavy bags.
Out of the heliport lounge a single flight of stairs led up to the helideck that the Sikorsky S76 was as he passed under the fast running rotors door. The co-pilot slid back his small toward the luggage compartment for his wrapped tube of newspapers and an empty flask.
'Black with sugar,' he shouted above the noise.
Back with a full flask of stewed coffee from the heli-lounge percolator, Finn climbed in and strapped into the full restraint


A few seconds later he heard the distinctive beating of approaching helicopter blades, stood up to stretch as much as he could in the all-encasing neoprene suit, put on the padded ear defenders and picked up his heavy bags.
Out of the heliport lounge a single flight of stairs led up to the helideck that the Sikorsky S76 was as he passed under the fast running rotors door. The co-pilot slid back his small toward the luggage compartment for his wrapped tube of newspapers and an empty flask.
'Black with sugar,' he shouted above the noise.
Back with a full flask of stewed coffee from the heli-lounge percolator, Finn climbed in and strapped into the full restraint


A few seconds later he heard the distinctive beating of approaching helicopter blades, stood up to stretch as much as he could in the all-encasing neoprene suit, put on the padded ear defenders and picked up his heavy bags.
Out of the heliport lounge a single flight of stairs led up to the helideck that the Sikorsky S76 was as he passed under the fast running rotors door. The co-pilot slid back his small toward the luggage compartment for his wrapped tube of newspapers and an empty flask.
'Black with sugar,' he shouted above the noise.
Back with a full flask of stewed coffee from the heli-lounge percolator, Finn climbed in and strapped into the full restraint

The texture features approach achieved over 95% accuracy on the classification of 60 test document images from 3 languages are very promising over 73% accuracy with template matching features.

Conclusion-
In this poster, we compared two methods of language detection in scanned documents for three different languages. We have also found the effectiveness of texture analysis over template matching. In texture analysis, We saw that that there is no need of character segmentation or connected component analysis. Our system can also overcome noise which is in the form of moderate skew, numerals, foreign characters, illustrations, and blurred or fragmented characters.

Future Work-
Language detection is only the first step in document image analysis. The next step in our research is to link this algorithm with appropriate language character identification. Number of languages to be identified can also be increased. The system should be made more robust towards external noise.

References-
Statistical analysis:
http://www.umiacs.umd.edu/~zhugy/HandwritingLanguageID_PR2009.pdf
Texture analysis:
http://www98.griffith.edu.au/dspace/bitstream/handle/10072/4262/31447.pdf?sequence=1&origin=publication_detail
Template matching:
https://pdfs.semanticscholar.org/a495/7c1c2564e62d40fb91d5b7bbd57a8fcf203d.pdf
LBP:
http://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/
Skew compensation:
http://engr.case.edu/merat_francis/EECS%20490%20F04/References/Document%20Deskew/00619830.pdf

MSB not doing. As just the position of height shifts wrt MSB position. So we have shown this position of MSB.

binary image Not much change. As we are comparing the pixel values. After thresholding also same comparison.
font size need to try. Or else let us say our method is working for same font images. This is our drawback and we are working on it in future work.

Not in poster contribution:
Have done a literature survey and tried different algorithms while implementing different steps of the method and chosen the best ones and made the system robust against skew angle, number of lines, size and also working for scanned images.

10/11/2016

_____JUST A COPY HERE AND SHORTENING THE CONTENT_____

Title -
Texture, Template matching approaches for Language Detection in document images.

Members names -
Deepak Jannarapu, Nallagatla Manikanta, Vinod Pankajakshan IIT Roorkee.

Abstract -
The problem of recognising the language of a document image has large number of applications in many fields like document analysis, as an initial step for optical character recognition. We analysed and improved two approaches for language detection- texture and template matching. Local binary patterns are used or texture features and for template matching character templates for three languages English, Hindi, Telugu are handpicked from respective scripts. Comparing both approaches, texture features, achieved over 95% accuracy on the classification of document images from 3 languages.

Introduction-
The world we live in is becoming increasingly multilingual and, at the same time, increasingly automated. Hence the need of automatic language identification is increasing. Various methods have been developed for language detections. But among these 3 methods namely texture analysis, template matching and statistical analysis are found to be accurate and feasible. In texture analysis, documents are classified on the basis of texture features known as local binary patterns. The template matching identifies textual symbols in the test images. The details about these methods are explained in the further sections.

A	B	C	D	E	F	G	H
a	b	c	d	e	f	g	h

English Templates

క	కా	కి	కీ	కు	కూ	కృ	కౄ
ఖ	ఖా	ఖి	ఖీ	ఖు	ఖూ	ఖృ	ఖౄ

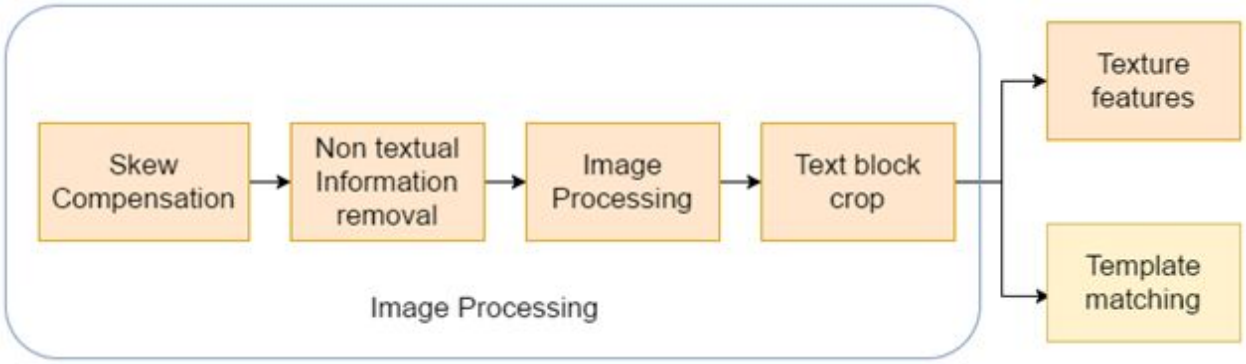
Telugu Templates

क	क़	ख	ख़	ग
ग़	घ	ङ	च	छ

Hindi Templates

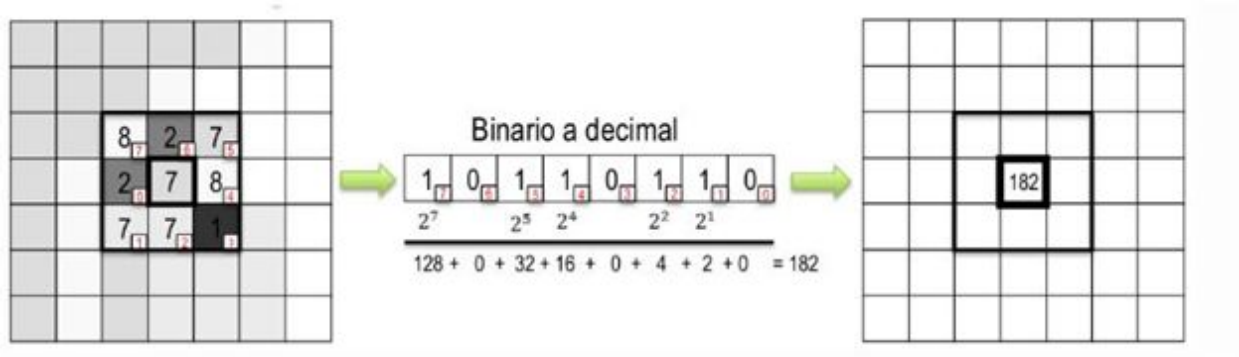
Methods -

After image processing step shown in figure 2, respective features are extracted from the image.



Texture approach:

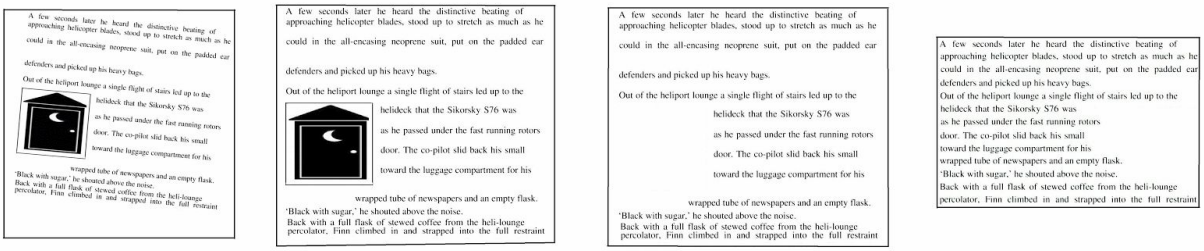
Using local binary patterns texture features are extracted. To calculate the LBP value for a pixel in the gray scale image, we compare the central pixel value with the neighbouring pixel values. Since there are 8 neighbouring pixels – for each pixel, we will perform 8 comparisons. The results of the comparisons are stored in a 8-bit binary array. I The whole process is shown in the figure 3.



Then LBP Histograms are calculated and normalized for the features.We used chi square distance with the features of train images for classifying the test image.

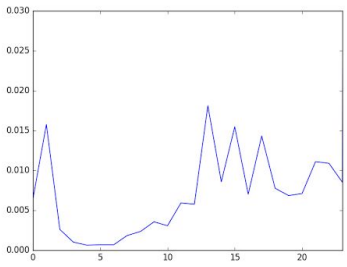
Template matching approach:
Templates for characters of three languages English, Hindi, Telugu are handpicked from their scripts. Then after performing image processing as above, the instances of these templates are searched in test document images. Sample of templates for each language are shown in below figure 1.

Results-
The color images used were taken as screenshots from english, hindi, telugu content pdfs. 80 document examples were chosen for each of: english, hindi and telugu. Images contain graphics, and would resemble reasonably closely the output from a document segmentation system. Also Foreign characters, numerals and italicized text were present in many of the images. Figure below shows step by step processing of a sample image used in the experiments. The images were divided into 60 training and 20 test images per language. For non text removal, connected component of more than 2.25 times average area of a character is removed. For template matching, 52 for english, 425 for hindi, 756 for telugu template characters are hand picked.

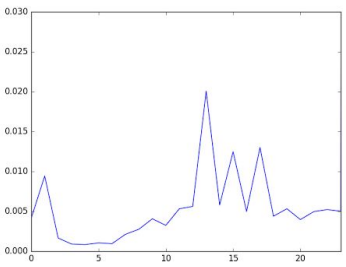


Original Image | Skew Compensated | Graphics Removed | Text Block

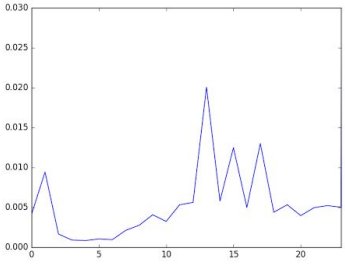
The texture features approach achieved over 95% accuracy on the classification of 60 test document images from 3 languages are very promising over 73% accuracy with template matching features.



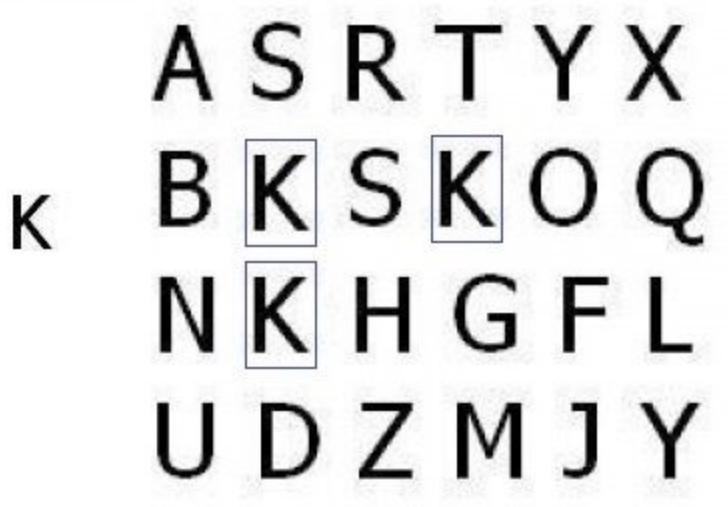
English Histogram



Hindi Histogram



Telugu Histogram



Template matching

Uniqueness-

Texture features approach is rotation invariant as local binary patterns are rotation invariant. As we are using texture for language classification texture approach is independent on number of lines work, size of image. This approach is also working for scanned documents also which has various applications like indexing, digitalizing a hard copy.

Conclusion and Future Study

We analysed and improved two methods of language detection in scanned documents for three different languages. We have also found the effectiveness of texture analysis over template matching.Our system can also overcome noise which is in the form of moderate skew, numerals, foreign characters, illustrations, and blurred or fragmented characters. But main drawback of texture approach is that it is sensitive for font size in training images. The next step in our research is to link this algorithm with appropriate language character identification. Number of languages to be identified can also be increased. system need to overcome the sensitivity of font size.

References-

[1] Peake, G. S., and T. N. Tan. "Script and language identification from document images." *Document Image Analysis, 1997.(DIA'97) Proceedings., Workshop on*. IEEE, 1997.

[2] Hochberg, Judith, et al. "Automatic script identification from images using cluster-based templates." *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. Vol. 1. IEEE, 1995.

[3] Pietikäinen, Matti. "Local binary patterns." *Scholarpedia* 5.3 (2010): 9775.

DEEPAK SEE MAIL OF RAJIV SIR FOR GUIDELINES FOR BTP POSTER

See the dpi of images. Important in image processing.

Changing dpi of images

Converting image to black and white

Graph width in results

Remaining works:

22/11/2016

Talk track:

To read:

How we did Skew compensation DONE

How we did template matching DONE

References DONE

We present a new fast and accurate approach based on Local Binary Patterns (LBP) for the extraction of the features that is combined with the new classifier Neighboring Support Vector classifier (NSVC) for classification.

The skew angle is obtained by searching for a peak in the histogram of the gradient orientation of the input greylevel image.

Perform gradient operation on the image;

Taninverse as orientation

Search for maximum in this histogram to obtain an initial skew angle;

```
c = real(ifft2(fft2(background) .* fft2(template, by, bx)));
```

24/11/2016

COMMENTS:
PP ROY 100% accuracy needed
Ghosh we have done script identification not language identification
RKP Poster bullets and graphs labelling
What is texture?