Digital Image Processing: LaTeX generation from Printed Equations

Shubham Rathore, 201430101

Monsoon '16

Abstract

LaTeX is a strong document preparation system that is used for high quality typesetting. LaTeX is a popular tool when it comes to the publication of scientific documents in various fields that include mathematics, physics, computer science, economics and engineering. Multilingual characters can be represented using LaTeX. It is also used to convert XML based formats into PDF. The main issue that occurs when dealing with LaTeX is the irreversibility of the document once created. A document once rendered is unable to move back into its source code. It may be great importance to people dealing with LaTeX, to make some amendments in the existing document. The aim of this project is to make the source code accessible to the user by converted images of the document taken by the user into the LaTeX source code.

1 Introduction

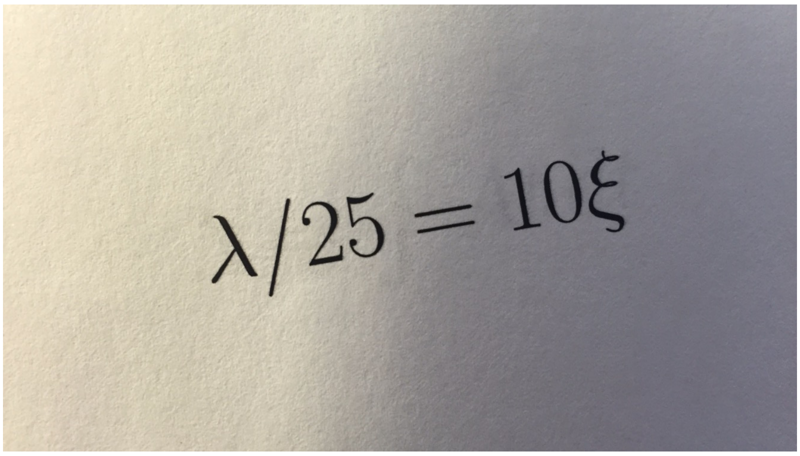
The project is broadly divided into three categories namely, page optimization, character recognition and LaTeX compilation. The character recognition part comes under the banner of ML whereas the image processing takes place in the page optimization part of the project. The LaTeX compilation part is not that complex and uses some naive assembly methods for execution. The aim of the project was basically to provide an efficient algorithm to accomplish the task. The scope of character recognition is limited to certain common operators. The code can be made to recognize more characters by just increasing the size of the dataset used for classification.

2 Details

Input: Photographs of printed scientific and mathematical expressions
Output: LaTeX code for the corresponding expressions
Data: Images of scientific and mathematical expressions generated using LaTeX for testing
Implementation: OpenCv, Can be extended to Android

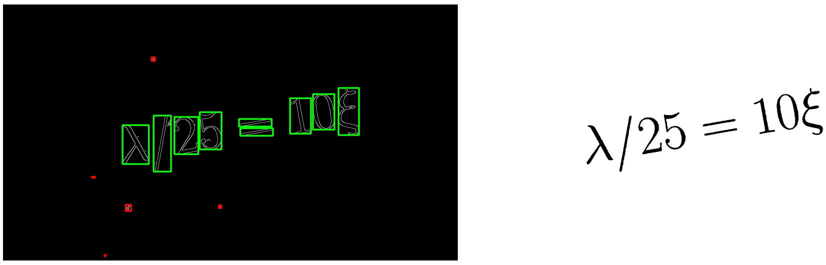
3 Process Flow

The challenges that are encountered in the process are the conversion of the image into a binary image, correction of skew from the horizontal, segmentation of each individual character, recognition of the characters using a database of characters, and finally the generation of LATEX representation of the equation[2].

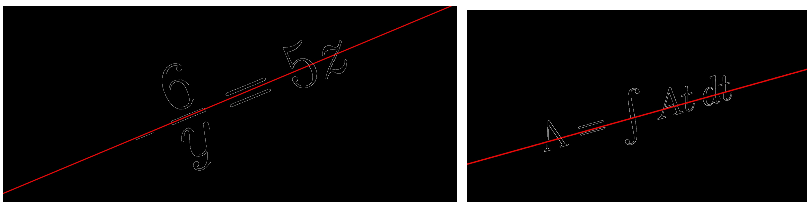


3.1 Binarization

Typical binarization algorithms like global and adaptive thresholding are not directly used, instead we use a method that is more efficient than these typical algorithms. An edge map of the original image is worked upon. Next step is to find the connected components in the edge map. Then bounding boxes are found for each connected component. Now thresholding is applied on these bounded boxes in the original image based upon standard deviation. For thresholding Otsu's algorithm is used.



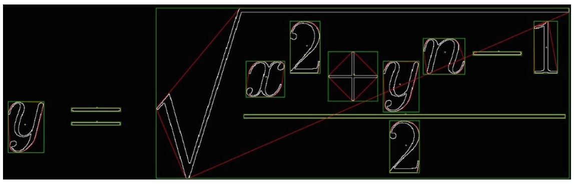
3.2 Skew Correction



The concept of Hough transform comes into play. Various Hough lines are obtained when transform is applied to an equation[5]. The main issue that arises is the selection of the angle to be used for the correction of skew. To encounter this problem binning is done on the angles and the range that has the maximum number of hough lines is selected. Now the box bounding the equation is rotated by an angle that is equal to the mean of the range that was selected in the previous step.

3.3 Segmentation

The characters need to be segmented out of the equation for use in the later stages in the pipeline namely character recognition and assembly.



In this part also, connected components are extracted and the image bounding the connected component is cropped out. It is to be noticed that the bounding box and the centroid for each connected component is saved to be later used in LaTeX compilation stage.

3.4 Recognition

This part includes portions of image processing and machine learning. Nearest neighbor algorithm using euclidean distance is used to classify the character. The main point of concern is the formulation of the feature vector for each image. The feature vector comprises of four parts[4].

3.4.1 Centralized moment of inertia

$$I_N = \frac{1}{N^2} \sum_{i=0}^{N} (x_i - c_x)^2 + (y_i - c_y)^2$$

The central moment of inertia is the first element of the feature vector.The centralized moment of inertia is normalized using the size of the input image.[3] This renders the moment as scale invariant. Moreover the centralized moment of inertia is translation as well as rotation invariant.Moment invariants are better thought of as mathematical tools that are of great benefit to computer vision and image processing

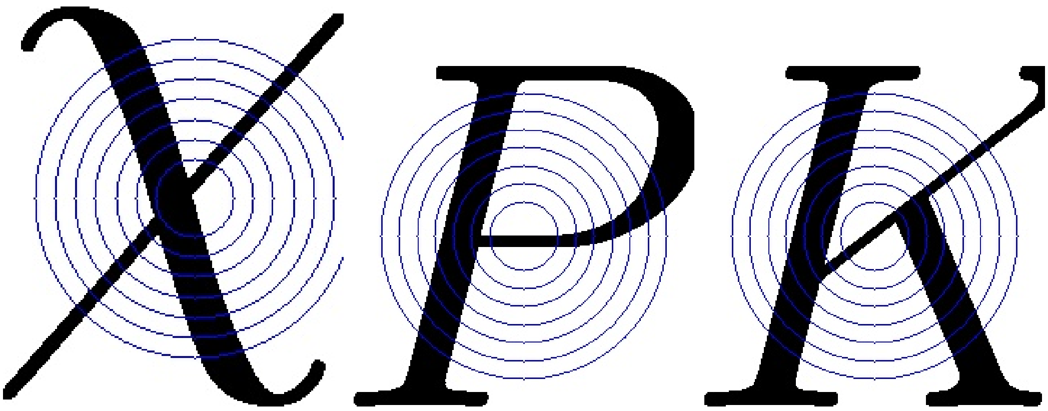
3.4.2 Hu Invariant Moments

The next seven features in the feature vector are derived from the Hu moments. These moments are again invariant to translation, scaling as well as rotation. The Hu moments are calculated using the central moments for different orders. Hu moments are very sensitive to noise and may create an issue in some cases. It is to be noted that Zernike moments or Affine moments could also be used, but Hu moments give the maximum accuracy in this case.

$$
\begin{aligned}
\eta_{pq} &= \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \qquad \gamma = 1 + \frac{p+q}{2} \\
H_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
H_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
H_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \\
H_5 &= (\eta_{30} - 3\eta_{12})^2(\eta_{30} + \eta_{12})^2[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
H_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{12} + \eta_{03}) \\
H_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}
$$

3.4.3 Circular Topologies

The circular topology is used to generate the next eight features for each image. Eight circles are drawn at equal spacing on the image and the crossings are calculated for each circle. The crossing is checked in clockwise order, with contour points sorted by the angle subtended by the segment to the centroid to achieve rotation invariance.[1] By crossings it is meant that the count of the number of times that the circle cuts the digits is maintained. Padding is done on the images so as to fit all the eight circles in the image, otherwise parts of the larger circles move out of the image and trash values are generated. The circles are centered at the character's centroid. As the circle is rotationally invariant, the topology of the character along the circular path remains invariant.



3.4.4 Fourier Descriptors

$$
Z[k] = DFT[z[n]] = \frac{1}{N}\sum_{n=0}^{N-1} z[n]e^{\frac{-2\pi mk}{N}} \tag{1}
$$

Fourier Descriptors are scale, rotation and shift invariant. Hence we use 15 fourier descriptors for each image.Let $x[n]$ and $y[n]$ be the coordinates of the nth pixel on the boundary of a given contour containing N pixels, a complex number can be formed as $z[n] = x[n] + jy[n]$, and the Fourier Descriptor (FD) of this image is defined as the DFT of $z[n]$:

3.5 Equation Assembly

This is the last part of the pipeline that merges all the components of the equation. An abstract syntax tree(AST) is created using the centroids of the bounding boxes of the characters that were previously saved. Each node of the tree denotes a construct occurring in the equation. Then the tree is traversed through a depth first search(DFS) and the equation is printed accordingly.

4 Further Work

The algorithm has the ability to detect and scrape out characters from any photograph with empty background. The algorithm can be modified such that characters with background clutter are also detected. Plan is to convert the project into an android application that increases the functionality of the algorithm. There happens to be a conflict between lowercase and uppercase characters sometimes, so there is a need to reduce error. A neural net can be used to classify instead of NN classifier which in turn will give better results.

References

[1] M Usman Akram, Zabeel Bashir, Anam Tariq, and Shoab A Khan. Geometric feature points based optical character recognition. In *Industrial Electronics and Applications (ISIEA), 2013 IEEE Symposium on*, pages 86–89. IEEE, 2013.

[2] Jim Brewer and James Sun. Latex generation from printed equations. 2015.

[3] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[4] Ashis Pradhan, Mohan P Pradhan, and Amit Prasad. An approach for reducing morphological operator dataset and recognize optical character based on significant features. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pages 1631–1638. IEEE, 2015.

[5] Heng-You Wang, Rui-Zhen Zhao, and Jing-An Cui. Fast and robust skew correction in scanned document images based on low-rank matrix decompositon. In *2014 International Conference on Machine Learning and Cybernetics*, volume 2, pages 883–887. IEEE, 2014.