DIP Project Proposal
LaTeX Code Generation from Printed Equations
Team Escher

Kritika Prakash[1] and Karthik Chintapalli[2]

[1]20161039
[2]201501207

29th September 2018

1 GitHub Repository

`https://github.com/Kritikalcoder/Latex-Generation-from-Printed-Equations`

2 Main Goals

To create a system that takes a photograph, scan or screenshot of a printed mathematical equation and produces a valid LaTeX markup code representation to be able to generate the equation.

3 Problem Definition

Working with lengthy involved mathematical equations can be cumbersome, tedious and error-prone. Mathematical equations in the printed form are not easily reproducible in new LaTeX documents. This is because, once a LaTeX document has been rendered, the underlying producer's code to recreate it is inaccessible.

3.1 Approach

The Latex Code Generation from Printed Equations Project aims to automatically generate valid LaTeX expressions for a photograph, scan or a screenshot of a printed mathematical equation.

3.2 Scope

The scope is defined by the kind of mathematical operations and expressions the system will be able to recognize and recreate. The equation is assumed to be the primary data in the input image, as opposed to extraction of the mathematical equation from an entire page full of content other than the equation.

4 Results

4.1 What Will Be Done

- Input Skewed photograph, screenshot or scan of a printed mathematical equation, for a given scope of equations and symbols
- Output Valid LaTeX code for the mathematical equation
- Testing Data Images of mathematical expressions with original LaTeX that generated them.
- Pipeline
  - Page Optimization
    We will use Adaptive Thresholding with Mean Filtering, Morphological Edge Smoothing, and Hough Transform for Image Binarization and Skew Correction of the input image.
  - Character Recognition
    Characters and tokens are first segmented from the cleaned input image and then matched with an existing database of characters. We will use Hu Invariant Moments and Circular Topology to identify characters against a database of characters. We will classify each character using Nearest Neighbour Classification.
  - LaTeX Compilation
    We will assemble the corresponding LaTeX code from the identified sequence of input characters.

4.2 Expected Final Result

Once we have build this baseline model, we will test it extensively. We will try to improve it by analyzing each component in the pipeline by different approaches to solving each sub problem. We will showcase a working model and a comparative study of the various approaches to solving this problem.

5 Tasks

S.No	Stage		Task	Member
1	Baseline Model	Page Optimization	Image Thresholding	Karthik
2			Binarization	Karthik
3			Skew Correction	Kritika
4		Character Recognition	Character Segmentation	Karthik
5			Character Identification	Kritika
6			Character Matching	Kritika
7		LaTeX Compilation	Equation Assembly	Both
8	Improvement		Testing	Both
9			Improving Baseline Model	Both
10			Comparative Study	Both
11	Presentation		Preparing Presentation	Both

6 Timeline

No.	Milestone	Expected Date
1.	Project Proposal Submission	Sept 29th, 2018
2.	Understanding Problem Statement, Reading Background	Oct 7th, 2018
3.	Finalizing Solution Approach	Oct 8th, 2018
4.	Implementation of Phase 1 (Baseline Model)	Nov 8th, 2018
5.	TA Review (prototype)	Nov 10th, 2018
6.	Implementation of Phase 2 (Improvement Phase)	Nov 23rd, 2018
7.	Implementation of Phase 3 (Presentation & Report)	Nov 27th, 2018
8.	Project Report Submission	Nov 28th, 2018
9.	Final Presentation	Nov 29th, 2018