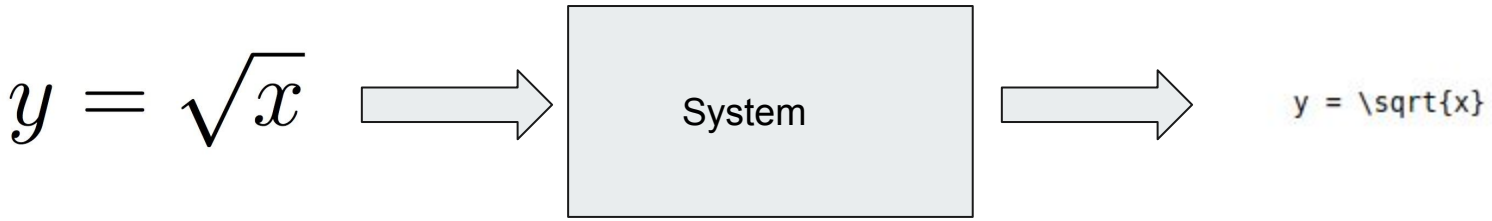
LaTeX Generation from Printed Equations

TEAM NAME - TAR--1
TEAM MEMBERS - SRIKAR M (20171138,CSE)
VIVEK N (20171182,CSD)
TA MENTOR - ADITYA AGGARWAL
REPO URL- https://github.com/SrikarMannepalli/Latex-Generation-from-Printed-Equations

PROBLEM STATEMENT

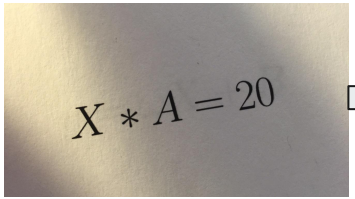
To build a system that takes a scan, PDF screenshot or a photograph of a printed mathematical equation and produces a LaTeX code representation that can generate the mathematical equation.



MOTIVATION

LaTeX is a powerful typesetting system that is extremely useful for technical documents, particularly mathematical equations. Working with lengthy mathematical equations can be a tedious and error-prone process. But, once rendered, the output cannot be modified, as we don’t have access to the underlying code. The ability to take a screenshot or a photograph of an existing equation and generate the LaTeX code for it can be extremely useful.

SOLUTION PIPELINE



Binarization

Skew Correction

Character Segmentation

Character Identification

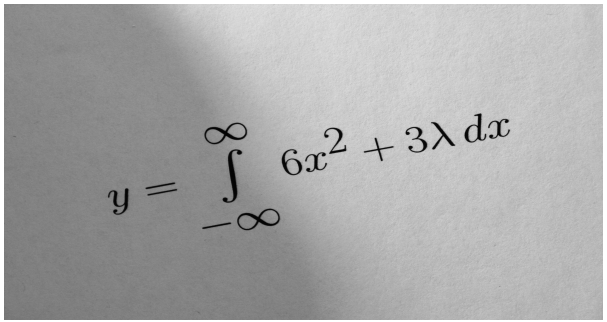
Character Matching

Equation Assembly

$X * A = 20$

Binarization

- Convert image to grayscale and classify as a screenshot or photograph.
- Remove high frequency noise.
- Adaptive thresholding to obtain binary image.
- Fill small holes and remove noise using morphological operations.



$$y = \int_{-\infty}^{\infty} 6x^2 + 3\lambda \, dx$$

Skew Correction

- Obtain dominant orientation using hough transform and obtain deskewing angle.
- Take top 4 peaks from hough transform.
- Rotate the image by deskewing angle.
- Soften the edges.

$$y = \int_{-\infty}^{\infty} 6x^2 + 3\lambda \, dx$$

$$y = \int_{-\infty}^{\infty} 6x^2 + 3\lambda \, dx$$

Character Segmentation

- Create the edge map of the input image.
- Obtain centroids and bounding boxes of each edge.
- Select the single largest character in each region.

$$y = \int_{-\infty}^{\infty} 6x^2 + 3\lambda \, dx$$

$$y = \int_{-\infty}^{\infty} 6x^2 \boxplus 3\lambda \, dx$$

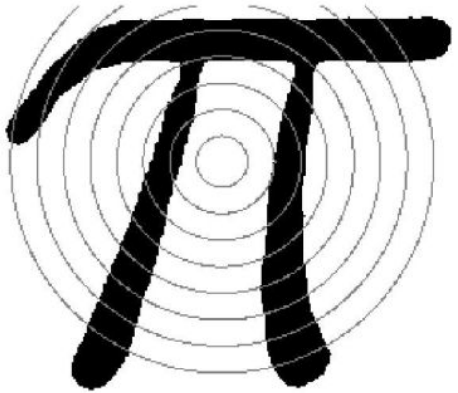
Character Identification

Created a 38-element Feature vector for each character - Identification Profile.

Features are invariant to Translation, Rotation and Scaling.

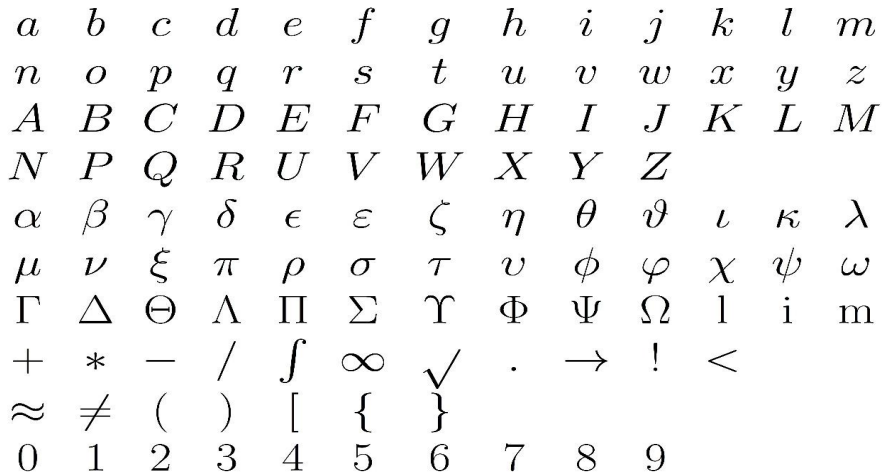
Features 1-38:

- 1 : Normalized central moment of inertia
- 2 - 32 : Circular Topology
  - 2-17: Number of times each circle crosses the character
  - 18-32: Spacing between character crossings for each circle
- 33 - 38: Hu Invariant Moments



Character Matching

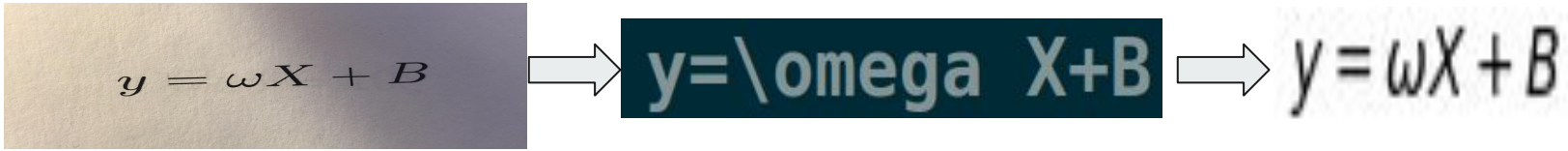
- Obtain and store feature vector for each character in the character palette.
- Find nearest neighbour for each segmented character from the above database.
- Use Manhattan distance metric.

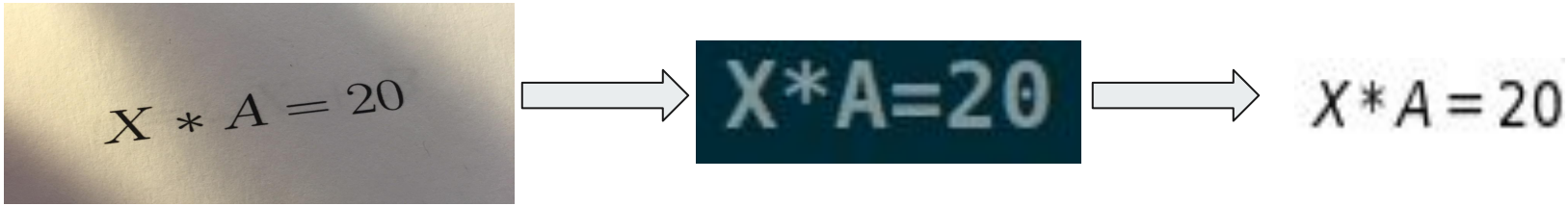


Equation Assembly

- Assemble from left to right.
- Recursive assembly for subequations.
- Handle Superscripts and subscripts.
- Bounding box overlaps.

Some Results..





Some Results..

$$6\frac{dx}{dt} + z\frac{d^2x}{dt^2} = 3x$$

```
6\frac{dx}{dt}+z\frac{d^{2}x}{dt^{2}}=3x
```

$$y = \sqrt{\frac{x^2+y^{n-1}}{2}}$$

```
y=\sqrt{\frac{x^{2}+y^{n-1}}{2}}
```

Analysis

Image type	Number of characters	Number of characters correctly identified
Photographs	98	84
Screenshots	98	93

Improvements

- Compatibility with handwritten equations.
- Robustness to resolution of images.
- Faster pipeline.

Work Division

Work item	Major Contributor
Binarization	Srikar M
Skew Correction	Vivek N
Character Segmentation	Srikar M
Character Identification	Vivek N and Srikar M
Character Matching	Vivek N
Equation Assembly	Srikar M and Vivek N
Debugging and Misc.	Vivek N

THANK YOU