



# The Flatteners

EECS 504, Fall 2020

Arnav Sharma, Austin Jeffries, Ali Badreddine

# Executive Overview

- Folded and bent paper when scanned can be very difficult to read
- Our idea is to:
  - Flatten folded sheets of paper using deep learning and transfer learning
  - Generate answer keys for math worksheets using template matching and K-means clustering
  - Incorporate a web-app as a visual aid of the dataset and worksheet grading
    - Demo at the end of presentation

# Background and Impact

- COVID-19 has affected the education industry deeply
- Teachers are grading homework assignments, such as math worksheets, via images sent by the student instead of in-person
- Students may not necessarily submit perfectly flat sheets and instead can be folded or creased

# Computer Vision Methods

- Deep Learning and Transfer Learning
- Dataset Generation
  - Creates 3D distorted paper sheets
  - Realistic lighting and shadows
- Answer Key Generation
  - Template matching
  - K-means clustering to tease out individual numerical problems

# Prototype

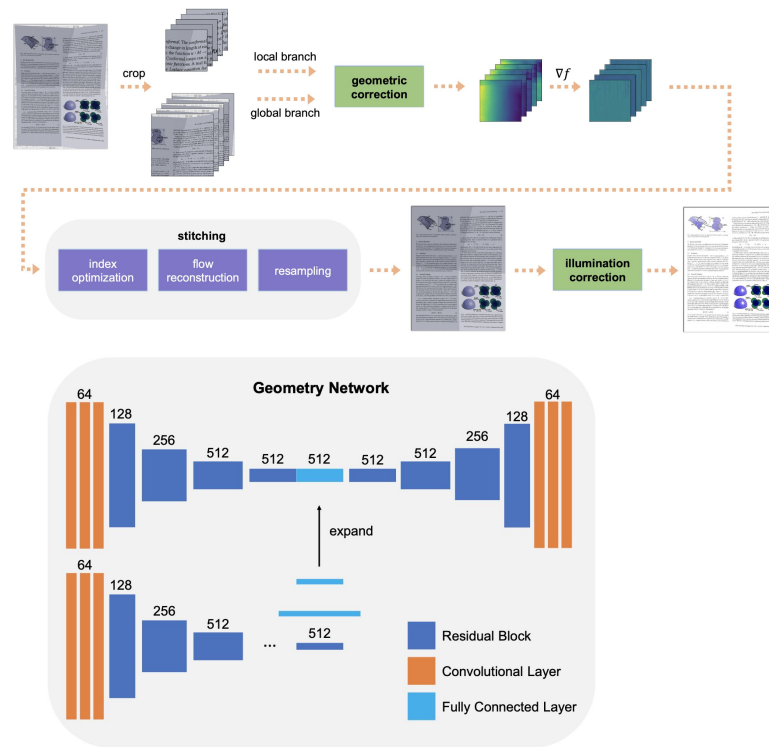
- Dataset Generation Examples
  - 3D distorted mesh developed by perturbing random vertices
  - Used these meshes to generate 2D flow maps
  - Introduced Plotly lighting for realistic lighting and shadows to highlight the folds and creases as compared to existing datasets
  - Applied worksheet image on to different backgrounds
  - Mask image generated to represent boundaries of perturbed worksheet image against the background

# Prototype - Model Architecture

## Model Training

- Input: Local and Global Image Patches
- Output: 2D Estimated Flow Maps
- Loss Function:

$$\mathcal{L}(F_s, F_t) = \frac{1}{HW} \sum_p \|F_s(p) - F_t(p)\|_2$$



Figures and architecture adapted from the following source:  
Xiaoyu Li, Bo Zhang, Jing Liao, and Pedro V. Sander. Document rectification and illumination correction using a patch-based cnn, 2019.

# Prototype

- Answer Key Generation

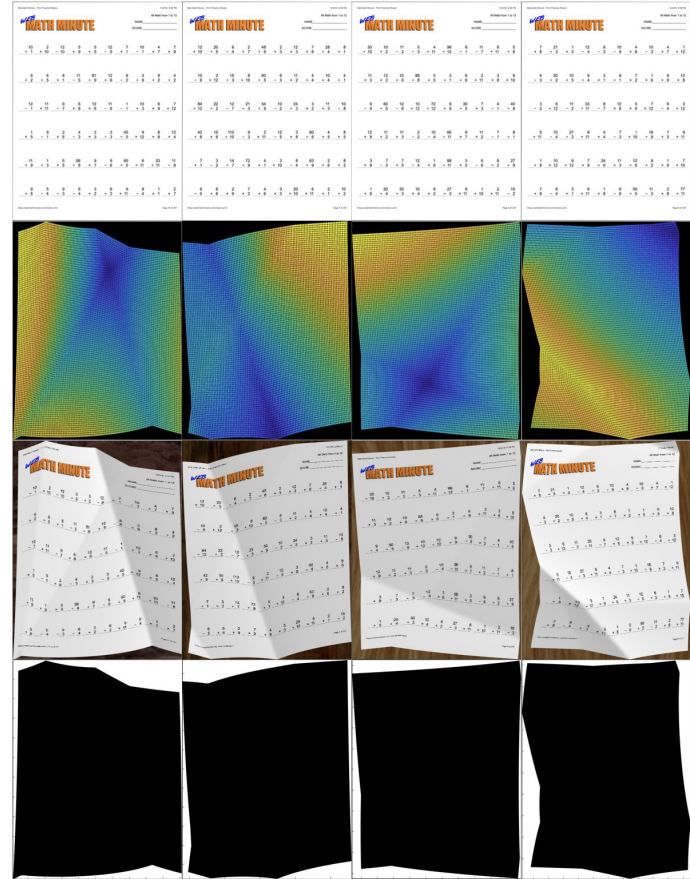
- Template matching for individual digits and math symbols while avoiding duplicate detections using a mask image

0 1 2 3 4 5 6 7 8 9 + - × ÷

- K-means clustering for each problem
  - Initiated algorithm to search for 60 distinct clusters as there are 60 different problems in each math worksheet

# Results - Data Generation

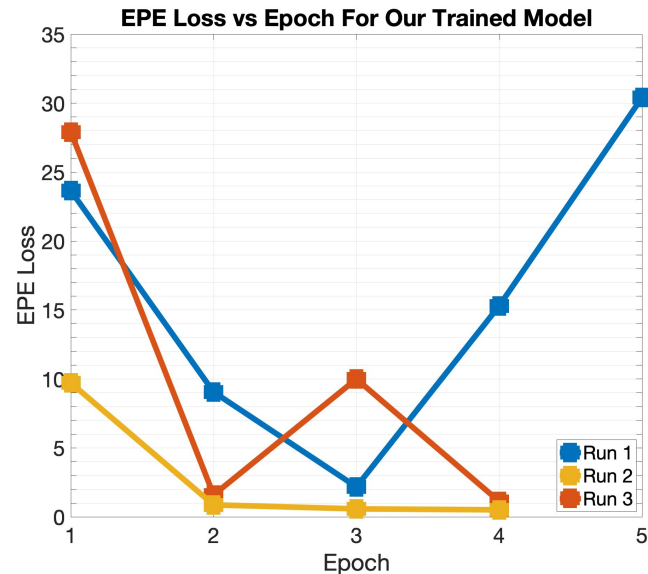
- Produces realistic 3D distorted sheets as seen in the third row
- Lighting and shadows are realistic
- Quick and efficient process due to using batch





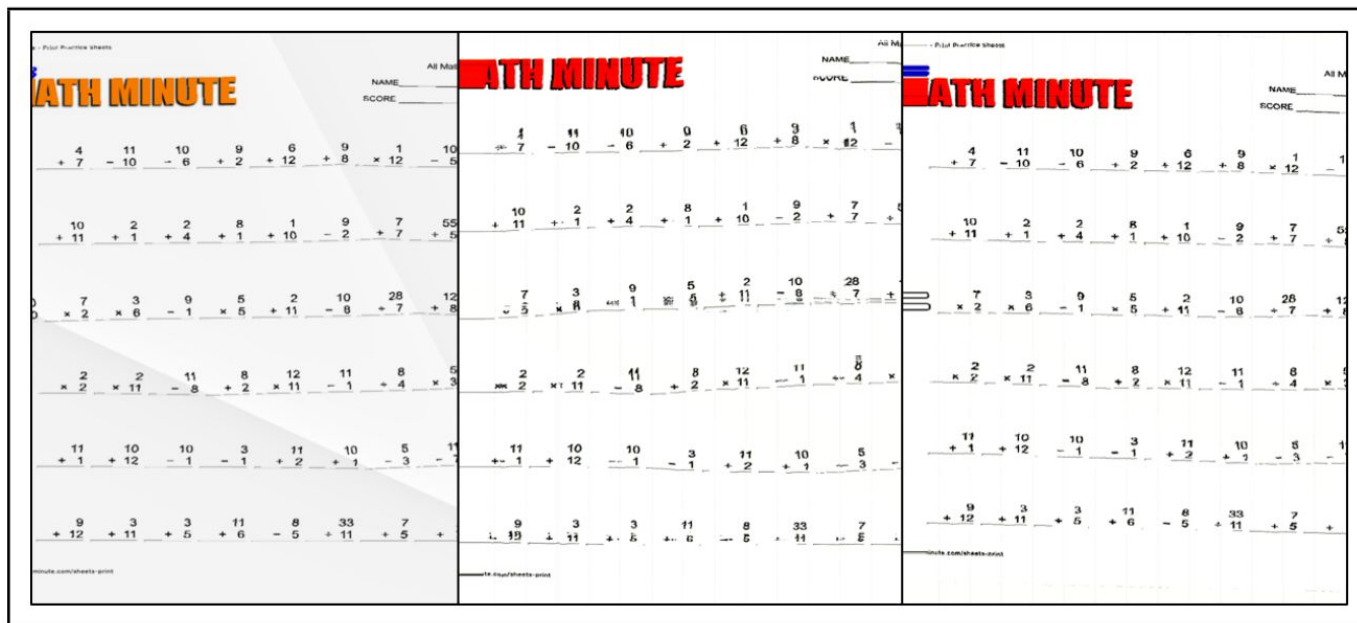
# Results

- Utilizing U-Net architecture
- Multiple data sets were generated at different specifications (*resolution factor, size, & test train data split*) and used to train the network
- To the right you can see three runs. Optimal loss achieved at Run 2 epoch 4 with a **Loss of 0.51**



Run	Epochs	Patch (px)	L-Rate	Res Factor	Training/Test Split (%)	Min. EPE Loss
1	5	128	0.001	4	70/30	2.17
2	4	64	0.01	6	80/20	0.51
3	5	64	0.001	4	80/20	1.12

# Results



Original Perturbed Image

Output of Our Trained  
Model **Loss = 0.51**

Output of Pre-Trained  
Model

# Answer Key Generation Performance

- Template matching requires significant tuning
- Our setup works most of the time
- K-means clustering also requires tuning for proper clustering but still works most of the time
- If unsatisfactory clustering is produced, a custom Error page is shown

Web Math Minute - Print Practice Sheets

12/4/20, 9:48 PM

**WEB MATH MINUTE**

All Math from 1 to 12  
NAME **ANSWER KEY**

SCORE \_\_\_\_\_

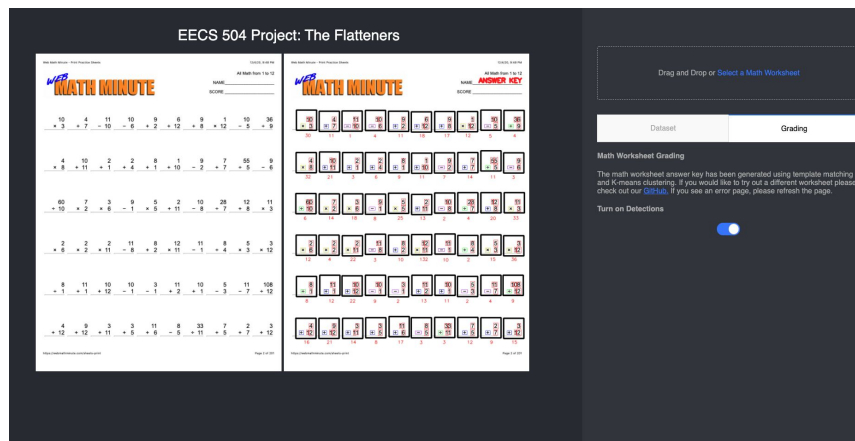
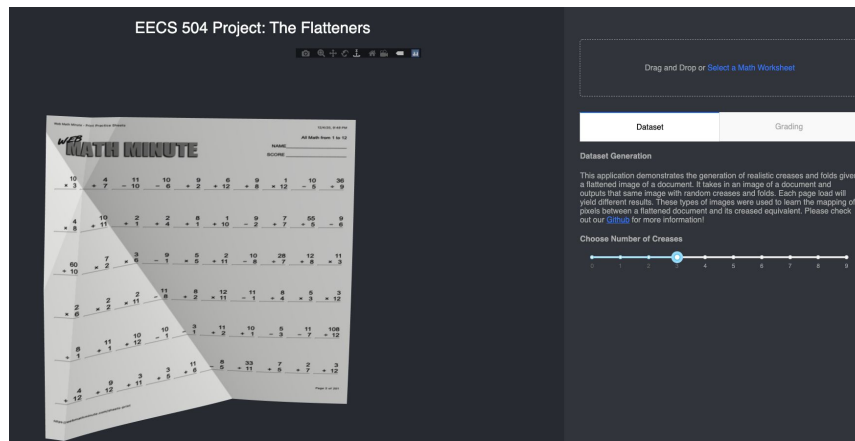
$\begin{array}{r} 10 \\ \times 3 \\ \hline \end{array}$	$\begin{array}{r} 4 \\ + 7 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ - 10 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ - 6 \\ \hline \end{array}$	$\begin{array}{r} 9 \\ + 2 \\ \hline \end{array}$	$\begin{array}{r} 6 \\ + 12 \\ \hline \end{array}$	$\begin{array}{r} 9 \\ + 8 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ \times 12 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ - 5 \\ \hline \end{array}$	$\begin{array}{r} 36 \\ + 9 \\ \hline \end{array}$
30	11	1	4	11	18	17	12	5	4
$\begin{array}{r} 4 \\ \times 8 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ + 11 \\ \hline \end{array}$	$\begin{array}{r} 2 \\ + 1 \\ \hline \end{array}$	$\begin{array}{r} 2 \\ + 4 \\ \hline \end{array}$	$\begin{array}{r} 8 \\ + 1 \\ \hline \end{array}$	$\begin{array}{r} 1 \\ + 10 \\ \hline \end{array}$	$\begin{array}{r} 9 \\ - 2 \\ \hline \end{array}$	$\begin{array}{r} 7 \\ + 7 \\ \hline \end{array}$	$\begin{array}{r} 55 \\ + 5 \\ \hline \end{array}$	$\begin{array}{r} 9 \\ - 6 \\ \hline \end{array}$
32	21	3	6	9	11	7	14	11	3
$\begin{array}{r} 60 \\ \times 10 \\ \hline \end{array}$	$\begin{array}{r} 7 \\ \times 2 \\ \hline \end{array}$	$\begin{array}{r} 3 \\ \times 6 \\ \hline \end{array}$	$\begin{array}{r} 9 \\ - 1 \\ \hline \end{array}$	$\begin{array}{r} 5 \\ \times 5 \\ \hline \end{array}$	$\begin{array}{r} 2 \\ + 11 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ - 8 \\ \hline \end{array}$	$\begin{array}{r} 28 \\ + 7 \\ \hline \end{array}$	$\begin{array}{r} 12 \\ + 8 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ \times 11 \\ \hline \end{array}$
6	14	18	8	25	13	2	4	20	33
$\begin{array}{r} 2 \\ \times 6 \\ \hline \end{array}$	$\begin{array}{r} 2 \\ \times 2 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ \times 11 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ - 8 \\ \hline \end{array}$	$\begin{array}{r} 8 \\ + 2 \\ \hline \end{array}$	$\begin{array}{r} 12 \\ \times 11 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ - 1 \\ \hline \end{array}$	$\begin{array}{r} 8 \\ + 4 \\ \hline \end{array}$	$\begin{array}{r} 5 \\ \times 3 \\ \hline \end{array}$	$\begin{array}{r} 3 \\ \times 12 \\ \hline \end{array}$
12	4	22	3	10	132	10	2	15	36
$\begin{array}{r} 8 \\ + 1 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ + 1 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ + 12 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ - 1 \\ \hline \end{array}$	$\begin{array}{r} 3 \\ - 1 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ + 2 \\ \hline \end{array}$	$\begin{array}{r} 10 \\ + 1 \\ \hline \end{array}$	$\begin{array}{r} 5 \\ - 3 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ - 7 \\ \hline \end{array}$	$\begin{array}{r} 108 \\ + 12 \\ \hline \end{array}$
8	12	22	9	2	13	11	2	4	9
$\begin{array}{r} 4 \\ + 12 \\ \hline \end{array}$	$\begin{array}{r} 9 \\ + 12 \\ \hline \end{array}$	$\begin{array}{r} 3 \\ + 11 \\ \hline \end{array}$	$\begin{array}{r} 3 \\ + 5 \\ \hline \end{array}$	$\begin{array}{r} 11 \\ + 8 \\ \hline \end{array}$	$\begin{array}{r} 8 \\ + 5 \\ \hline \end{array}$	$\begin{array}{r} 33 \\ + 11 \\ \hline \end{array}$	$\begin{array}{r} 7 \\ + 5 \\ \hline \end{array}$	$\begin{array}{r} 2 \\ + 7 \\ \hline \end{array}$	$\begin{array}{r} 3 \\ + 12 \\ \hline \end{array}$
16	21	14	8	17	3	3	12	9	15

<https://webmathminute.com/sheets-print>

Page 2 of 21

# Web-App

This demonstration application was created using the Dash/Plotly framework and was deployed using Heroku. All code was generated in python and adapted for use in the web application.



Thank you,  
Professor Corso  
Oscar de Lima  
& Parker Koch  
!!!