Manual grading in mathematics creates inefficiencies for students and teachers

Comprehensive Grading

Teacher spends countless hours providing complete feedback on each question, taking time away from other activities and leading to burnout

Student receives feedback days or weeks later, and is likely to have either moved on to a new topic or lost interest

Grading for Completeness

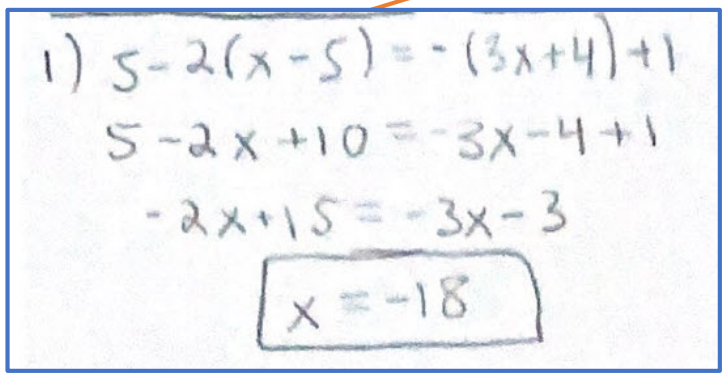
Teacher lacks insight into comprehension at individual or class level, and cannot review weak areas or target struggling students

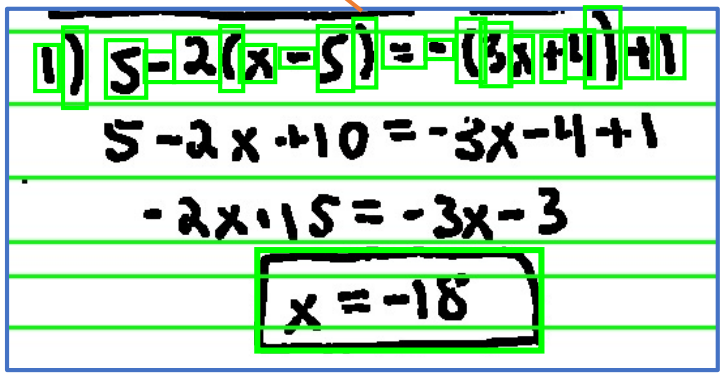
Student lacks understanding of comprehension gaps, and cannot prioritize areas for additional study

When students and teachers are provided with instant feedback on homework assignments, both parties can adjust their approaches in real time and achieve better academic outcomes with less human effort.

The InstaGrade pilot combines multiple data analysis methods to translate student response images into real-time feedback

Response Submitted → Image Cleansing → Character ID/Page Geography → Character Recognition and Evaluation → Feedback received





$$5 - 2(x-5) = -(3x+4) + 1$$

$$5 - 2x + 10 = -3x - 4 + 1$$

$$-2x + 15 = -3x - 3$$

$$x = -18$$

Goals: Remove noise and background from image, convert to black and white

Method: Create ensemble-CNN using multiple pre-processing techniques as inputs

- Utilize and augment open-source Kaggle "denoising dirty documents" dataset for training and testing
- Perform median filtering (background removal), gaussian blur, canny edge detection, adaptive thresholding on "dirty" and "clean" images
- Train CNN to combine outputs of basic cleansing techniques into "clean" image

Technical Environment: Python (OpenCV, keras)

Goals: Identify distinct solution steps, lines within these steps, and characters within these lines; identify final answer by box/circle or by page geography

Method: Identify shapes and characters with contour extraction, group these into distinct sections and lines with statistical analysis of pixel distribution

Technical Environment: Python (OpenCV, keras)

Goals: Convert characters to machine-readable text, evaluate final answer

Method: Train CNNs to categorize numbers, letters, and math symbols; vote on outputs of these three models to determine final classification, identify mistakes and evaluate final answer with SymPy

- Leverage and augment open-source MNIST and Kaggle datasets for numbers, letters, and math symbols

Technical Environment: Python (keras, SymPy)

Planned enhancements will add nuance to feedback and improve user experience

An online digital interface will provide students and teachers with a common access point

Historical tracking will identify learning gaps and trends to provide tailored recommendations for each student

Advanced analysis methods like Markov chains and Mathematical Language Processing will pinpoint specific mistakes and tendencies more accurately

Class-level analytics will help teachers address common issues and modify their teaching approach

The image-to-feedback pipeline created in the InstaGrade pilot will serve as a launching point for further capabilities in the grading and analysis of math homework.