Improving the Accessibility of Mathematical and other STEM Content in Engineering courses through Machine Learning Models

Louis Asanaka, University of Illinois at Urbana - Champaign
Delu Zhao, University of Illinois at Urbana - Champaign

Delu Zhao is a Senior at the University of Illinois at Urbana - Champaign majoring in Computer Science and Economics. He is an undergraduate research assistant passionate about improving education for all students.

Meghana Gopannagari, University of Illinois at Urbana - Champaign
Sonika Tamilarasan, The University of Illinois at Chicago
Alan Tao, University of Illinois Urbana-Champaign
Nancy Zhang, University of Illinois at Urbana - Champaign
Grace Elizabeth Sletten
Adelia Solarman, University of Illinois at Urbana - Champaign
Xiuhao Ding, University of Illinois at Urbana - Champaign

Xiuhao Ding is a master's student in Computer Science at the University of Illinois Urbana-Champaign with interests in CS education and AIGC

Dr. Pablo Robles-Granda, University of Illinois at Urbana - Champaign

Pablo Robles-Granda is a Teaching Assistant Professor at the University of Illinois at Urbana-Champaign.

Yang Victoria Shao, University of Illinois Urbana Champaign

Yang V. Shao is a Teaching Associate Professor in electrical and computer engineering department at University of Illinois Urbana-Champaign (UIUC). She earned her Ph.D. degrees in electrical engineering from Chinese Academy of Sciences, China. She has worked with University of New Mexico before joining UIUC where she developed some graduate courses on Electromagnetics. Dr. Shao has research interests in curriculum development, assessment, student retention and student success in engineering, developing innovative ways of merging engineering fundamentals and research applications.

Dr. Chrysafis Vogiatzis, University of Illinois at Urbana - Champaign

Dr. Chrysafis Vogiatzis is a teaching associate professor for the Department of Industrial and Enterprise Systems Engineering at the University of Illinois Urbana-Champaign. Prior to that, Dr. Vogiatzis was an assistant professor at North Carolina Agricultural and Technical State University. His current research interests lie in network optimization and combinatorial optimization, along with their vast applications in modern socio-technical and biological systems. He is teaching an introductory probabilities and statistics class for the college of engineering, a simulation class for industrial engineers, and an analysis of network data class for the graduate program. At Illinois, he is serving as the faculty advisor of the Institute of Industrial and Systems Engineers, and was awarded the 2020, 2023, and 2024 Faculty Advisor award for the North-Central region of IISE. Dr. Vogiatzis was awarded ASEE IL/IN Teacher of the Year in 2023, and received a Runner-up recognition for best case study by INFORMS in 2023.

Prof. Lawrence Angrave, University of Illinois Urbana-Champaign

Dr. Lawrence Angrave is an award-winning computer science Teaching Professor at the University of Illinois Urbana-Champaign. He creates and researches new opportunities for accessible and inclusive equitable education.

Dr. Hongye Liu, University of Illinois at Urbana - Champaign

Hongye Liu is a Teaching Assistant Professor in the Dept. of Computer Science in UIUC. She is interested in education research to help students with disability and broaden participation in computer science.

Improving the Accessibility of Mathematical and other STEM Content in Engineering courses through Machine Learning Models

Abstract

Transcribing equations and diagrams from STEM lecture slides pose significant challenges due to their varied structures. Existing methods focus on well-formatted research papers or isolated Mathematical content, lacking the flexibility to handle diverse Engineering educational materials. Furthermore, there is no widely available open-source software capable of both detecting and transcribing equations or diagrams from real-world STEM slides to Mathematical markup such as LaTeX, much less to natural language. This gap limits the accessibility of STEM content for students with disabilities or students with generally unmet needs, particularly in higher education settings as equations and diagrams become increasingly complex.

To address this, we evaluate and enhance existing machine learning models in computer vision for detection and transcription of equations and diagrams from STEM slides. To understand the strengths and limitations of existing methods we score them on their ability to handle different course materials. Then, we plan to improve both accuracy and efficiency in handling diverse content types, including handwritten equations and varied font styles.

We test these models on a custom dataset of lecture materials for six STEM courses at the University of Illinois Urbana-Champaign, impacting more than 1,000 engineering students per semester (mostly undergraduate). We apply character-error metrics for transcription to assess the performance of these models, with the consideration of computation availability to support our goal of integrating these models into our previous digital learning platform.

Through our evaluation of the models, we then extract the desirable components of the models to build a robust end-to-end Mathematical transcription pipeline for STEM content.We develop an open-source tool capable of accurately transcribing mathematical content from STEM slides, significantly enhancing accessibility for diverse learners. The state-of-the-art models are often either closed-source or lack the necessary flexibility to handle educational content. By refining and integrating open-source models into our previous digital learning platform, we aim to improve the accessibility of MATH and other STEM education.

Introduction

To improve the accessibility of STEM course content, this paper addresses the challenge of transcribing unstructured slides with a mix of equations, tables, figures, and handwriting into a structured output. While rapid advances in multimodal machine learning models have brought vast improvements in structured document and textual understanding, the varied structure of

STEM slides containing a mix of equations, tables, figures, and handwriting often leads to missed or incorrect transcriptions. Common models specialize in extracting information from well-formatted and isolated content commonly found in papers or single Mathematical formulas. Furthermore many models either cost money or require significant computational power in the form of GPUs that are often inaccessible to educational institutions with a limited budget or limited IT resources. This leads to inequality among faculty members that hope to improve accessibility of their course content, as creating such transcriptions would be infeasible and inconvenient. While this paper explores many capable models, it puts an emphasis on open-source models that can run in accelerator-less runtimes.

The most performant and reliable models that are capable of mathematical transcription, are scattered across the machine learning research publication literature, making it challenging for engineering educators to find and incorporate modern models in their workflow. This paper explores the specialized models that perform the different tasks of detection and transcription, then presents an ensemble end-to-end tool that combines the strengths of each model to provide the most accurate transcriptions of the slides. Our tool is model-agnostic to facilitate the integration of more capable future models.The tool provides engineering educators a unified method to generate transcriptions of course content, allowing students of diverse accessibility needs to access the content without solely relying on unstructured images of slides.

Background

Accessibility of STEM Course Materials When engineering instructors reuse or create slides and corresponding lecture videos, they rarely consider post-lecture accessibility, but rather are focused on the immediate needs of an upcoming live or recorded lectures. For instance, in our companion ASEE paper, a survey of students in Engineering courses found that students preferred LaTeX over plain text as a primary representation of Mathematical equations. However, rendered equations cannot be trivially recovered as LaTeX, versus text which can be semi-legibly parsed by optical character recognition software. Similarly, pedagogically important diagram details are often lost in transcription and prose, even though there are accessible ways to present them such as through textual summaries. Instructors are also often unaware of actions they can do to improve accessibility of their course material, such as providing alternative delivery formats that can be better parsed and presented by assistive technology. Even instructors who are familiar with such practices may not have time to use and master tools that make their material accessible. Our tool aims to streamline the process of building accessible course materials to save instructors time and unnecessary friction. For existing material only available in visual form, this tool also provides similar capabilities without hassle.

Universal Design for Learning Universal Design for Learning (UDL) is an inclusive teaching framework that improves learning for all students by providing multiple means of engagement, representation, and expression. It takes the approach of building flexibility into learning, most relevantly by offering content in various formats (e.g., text, audio, video) to accommodate different needs, particularly benefiting students with disabilities (SWD). In STEM education, where non-prose content including equations, graphs, and figures is commonly used, the development of transcription tools – which improve the translation of such content into multiple

formats – is critical to bringing UDL principles into the classroom.

Methods

We compare the capabilities of models for use in Detection (Segmentation) and Transcription tasks. Detection models take an image of a lecture slide as input, and output bounding boxes with labels that mark the segments of interest. Models that perform general layout perception beyond Mathematical Formula Detection (MFD) are also considered, but the primary metric of interest is the accuracy of MFD. Transcription models, or recognition models, take a localized image of an equation and transcribe it into LaTeX. To narrow the scope of this paper, only models that were developed or updated in the past 3 years are considered. The general performance of the models are evaluated using a representative sample of 9 slides from classes that participated in the project. The classes include 3 computer science courses, 2 electrical engineering and computer engineering courses and 1 industrial engineering course.

Name	Type (Detection/Transcription)	Year	Capabilities
Pix2Text	Both	2024	Equations, tables
PDF-Extract-Kit	Both	2024	Equations, tables
Marker	Transcription	2023	Equations, tables
Texify	Transcription	2023	Equations

Table 1: Comparison of OCR tools

For the end-to-end tool, we introduce a model ensemble that chains together detection and transcription models to output bounding boxes and transcription results from an image of a slide. During this process, we iteratively mask out content to ensure that the next model used can focus on new potential detections rather than re-discovering content areas that were already processed.

LaTeX Transcription Metrics

To evaluate machine-transcribed text, text-based metrics (e.g. edit distance and BLEU) are commonly used to measure precision of the output. However, treating LaTeX transcription as a traditional problem of machine translation ignores the problem that very different LaTeX formulas can render similar equation outputs (e.g. Fig 1 below). There can be disparities between text-based metric scores and visual judgement, which result in model outputs that are scored badly but look correct.

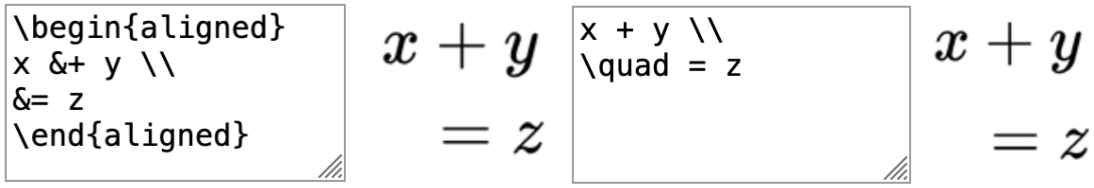


Figure 1: Different LaTeX formulas producing similar outputs

To mitigate this, we also use a metric called Character Detection Matching (CDM) [1] that includes a visual comparison in addition to a textual comparison.

We next introduce the AI models involved or evaluated in this work.

Mathpix

Mathpix is a piece of proprietary software with support for transcribing images that contain equations. While Mathpix is not a model that we surveyed due to its proprietary and paid nature, it is the bar for a good transcription model—in document-level transcriptions, Mathpix has consistently been the top performer [2].

Pix2Text

Pix2Text is a free alternative to Mathpix and is designed to turn images of math equations into LaTeX [3]. For detection, it uses a fine-tuned "You Only Look Once" (YOLO) model, which is a standard real-time object detection model. For transcription, it builds upon TrOCR [4], a strong base vision transformer model that allows effective fine-tuning using further human data. Pix2Text was able to work the best when the original image contained equations that were formatted similarly to LaTeX. When given a slide with large areas of large text and images, the resulting LaTeX would include random characters and symbols. The model performed best when the equations were isolated and single lined compared to using a multiline equation.

PDF-Extract-Kit

PDF-Extract-Kit is an open-source framework designed for content extraction from complex PDF documents [5]. It incorporates state-of-the-art models for diverse document analysis tasks, including formula detection via YOLO, formula recognition using UniMERNet [2], and table recognition through StructEqTable. These models are rigorously fine-tuned on diverse datasets, ensuring high accuracy and adaptability across a wide array of document formats; more importantly, they are able to accurately transcribe more complex mathematical content including multiline equations and matrices (Fig. 2, 3, 4, 5). UniMERNet is backed by a custom sequence-to-sequence model utilizing a vision transformer on top of Texify, another model we tested below [6].

$$\begin{aligned}
&\frac{\partial \tilde{f}}{\partial s_m}(s_1, s_2 \dots s_M) \\
&\quad = \sum_{n=1}^{N} \frac{\partial f}{\partial x_n}\left(g_1(s_1, s_2 \dots s_M), g_2(s_1, s_2 \dots s_M) \cdots g_N(s_1, s_2 \dots s_M)\right) \\
&\qquad \times \frac{\partial g_n}{\partial s_m}(s_1, s_2 \dots s_N)
\end{aligned}$$

Figure 2: Multiline equation as input

```
\begin{array} { r l } & { \displaystyle \frac { \partial \tilde { f } } { \partial S _ { m } } ( s _ { 1 } , s _ { 2 } \ldots s _ { M } ) } \\ & { \quad \quad = \displaystyle \sum _ { n = 1 } ^ { N } \frac { \partial f } { \partial x _ { n } } \left( g _ { 1 } ( s _ { 1 } , s _ { 2 } \ldots s _ { M } ) , g _ { 2 } ( s _ { 1 } , s _ { 2 } \ldots s _ { M } ) \cdot \cdot \cdot g _ { N } ( s _ { 1 } , s _ { 2 } \ldots s _ { M } ) \right) } \\ & { \quad \quad \quad \times \displaystyle \frac { \partial g _ { n } } { \partial S _ { m } } ( s _ { 1 } , s _ { 2 } \ldots s _ { N } ) } \end{array}
```

$$\begin{array} { r l } & { \displaystyle \frac { \partial \tilde { f } } { \partial S _ { m } } (s _ { 1 } , s _ { 2 } \ldots s _ { M }) } \\ & { \quad \quad = \displaystyle \sum _ { n = 1 } ^ { N } \frac { \partial f } { \partial x _ { n } } \left(g _ { 1 } (s _ { 1 } , s _ { 2 } \ldots s _ { M }) , g _ { 2 } (s _ { 1 } , s _ { 2 } \ldots s _ { M }) \cdot \cdot \cdot g _ { N } (s _ { 1 } , s _ { 2 } \ldots s _ { M }) \right) } \\ & { \quad \quad \quad \times \displaystyle \frac { \partial g _ { n } } { \partial S _ { m } } (s _ { 1 } , s _ { 2 } \ldots s _ { N }) } \end{array}$$

Figure 3: Model output as LaTeX (top) and rendered (bottom)

```
A - I \lambda = \left[ \begin{array} { c c c c c } { 3 - \lambda } & { 4 } & { 1 } & { 8 } & { 1 0 } \\ { 0 } & { 1 - \lambda } & { - 3 } & { 2 } & { 6 } \\ { 0 } & { 0 } & { - 5 - \lambda } & { 5 } & { - 3 } \\ { 0 } & { 0 } & { 0 } & { - 2 - \lambda } & { 1 } \\ { 0 } & { 0 } & { 0 } & { 0 } & { 3 - \lambda } \end{array} \right]
```

$$A - I \lambda = \left[\begin{array} { c c c c c } { 3 - \lambda } & { 4 } & { 1 } & { 8 } & { 1 0 } \\ { 0 } & { 1 - \lambda } & { - 3 } & { 2 } & { 6 } \\ { 0 } & { 0 } & { - 5 - \lambda } & { 5 } & { - 3 } \\ { 0 } & { 0 } & { 0 } & { - 2 - \lambda } & { 1 } \\ { 0 } & { 0 } & { 0 } & { 0 } & { 3 - \lambda } \end{array} \right]$$

Figure 4: Matrix equation as input

```
\begin{array} { r } { A - I \lambda = \left[ { \begin{array} { c c c c c } { 3 - \lambda } & { 4 } & { 1 } & { 8 } & { 1 0 } \\ { 0 } & { 1 - \lambda } & { - 3 } & { 2 } & { 6 } \\ { 0 } & { 0 } & { - 5 - \lambda } & { 5 } & { - 3 } \\ { 0 } & { 0 } & { 0 } & { - 2 - \lambda } & { 1 } \\ { 0 } & { 0 } & { 0 } & { 0 } & { 3 - \lambda } \end{array} } \right] } \end{array}
```

$$\begin{array} { r } { A - I \lambda = \left[{ \begin{array} { c c c c c } { 3 - \lambda } & { 4 } & { 1 } & { 8 } & { 1 0 } \\ { 0 } & { 1 - \lambda } & { - 3 } & { 2 } & { 6 } \\ { 0 } & { 0 } & { - 5 - \lambda } & { 5 } & { - 3 } \\ { 0 } & { 0 } & { 0 } & { - 2 - \lambda } & { 1 } \\ { 0 } & { 0 } & { 0 } & { 0 } & { 3 - \lambda } \end{array} } \right] } \end{array}$$

Figure 5: Model output as LaTeX (top) and rendered (bottom)

Another key feature of the PDF-Extract-Kit is its ability to operate efficiently in environments without GPU resources. Its models are able to run on CPU-only computers while maintaining acceptable performance by using their downscaled siblings that contain smaller networks.

Marker: PDF to Markdown breakdown

Marker is an open-source tool designed to convert PDFs into markdown, JSON, and HTML formats with high accuracy [7]. It handles a variety of documents and languages, and it's

particularly good at formatting tables, forms, code blocks, and equations. It can also extract and save images embedded in PDFs. Marker typically requires around 6 GB of VRAM per task, allowing up to eight documents to be processed in parallel on a high-end GPU. While it occasionally failed to detect handwritten text on slides, it accurately transcribed multi-line equations, resulting in consistent text recognition (Fig. 6 and 7).

General result: If $f : \mathbb{R}^N \to \mathbb{R}$, $g_n : \mathbb{R}^M \to \mathbb{R}$ for $n \in \{1, 2 \dots N\}$ and

$$\tilde{f}(s_1, s_2 \dots s_M) \stackrel{\text{def}}{=} f\left(g_1(s_1, s_2 \dots s_M), g_2(s_1, s_2 \dots s_M) \cdots g_N(s_1, s_2 \dots s_M)\right)$$

then

$$\frac{\partial \tilde{f}}{\partial s_m}(s_1, s_2 \dots s_M) = \sum_{n=1}^{N} \frac{\partial f}{\partial x_n}\left(g_1(s_1, s_2 \dots s_M), g_2(s_1, s_2 \dots s_M) \cdots g_N(s_1, s_2 \dots s_M)\right) \times \frac{\partial g_n}{\partial s_m}(s_1, s_2 \dots s_N)$$

Alternate example (thinking of polar coordinates):

$$f(a, b) = a^2 b^5 \qquad g_1(r, \theta) = r\cos(\theta) \qquad g_2(r, \theta) = r\sin(\theta)$$

Then

$$\tilde{f}(r, \theta) = (r\cos(\theta))^2 (r\sin(\theta))^5$$

Figure 6: A multiline equation serving as the input to the Marker model

General result: If $f : R^N \to R$, $g_n : R^M \to R$ for $n \in \{1, 2 \dots N\}$ and $\widetilde{f}(s_1, s_2 \dots s_M) \stackrel{\text{def}}{=} f\left(g_1(s_1, s_2 \dots s_M), g_2(s_1, s_2 \dots s_M) \cdots g_M(s_1, s_2 \dots s_M)\right)$. then

$$\frac{\partial \vec{f}}{\partial s_m}(s_1, s_2 \dots s_M) = \sum_{n=1}^{N} \frac{\partial f}{\partial x_n}\left(g_1(s_1, s_2 \dots s_M), g_2(s_1, s_2 \dots s_M) \cdots g_N(s_1, s_2 \dots s_M)\right) \times \frac{\partial g_n}{\partial s_m}(s_1, s_2 \dots s_N)$$

Alternate example (thinking of polar coordinates):
$f(a, b) = a^2 b^5 g_1(r, \theta) = r\cos(\theta) g_2(r, \theta) = r\sin(\theta)$ Then
$f(r, \theta) = (r\cos(\theta))^2 (r\sin(\theta))^5$

Figure 7: The output from the Marker model for the input shown in Fig. 6

Texify: Image and PDF to markdown and LaTeX

Texify is an optical character recognition (OCR) model that uses images and PDFs of equations and converts them into markdown and LaTeX [6]. It builds upon Donut, powered by a vision transformer [8]. The output can be directly used in MathJax, a common method of displaying Math and transcribing Math for screen readers [9]. When we tested Texify on our dataset of lecture slides and individual equations, we found that it performed well on small, simple equations, and it was somewhat efficient with an average time of 1 minute. However, it performed poorly on the lecture slides from our dataset. This is likely due to challenging complex, multi-line equations and varying formats.

Results

We compared the models by running the models on a dataset of 9 lecture slides (Images included in the Table 1 of Appendix), each with multiple equations. The results are presented in Table 2. We determined the accuracy by dividing the number of correctly transcribed equations (through visual inspection) by the total number of equations on each test slide. Overall, we found that models failed at transcribing tables accurately, and they generally performed better when transcribing short, one-line quotations compared to transcribing multi-line equations.

Model Name	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Pix2Text	1.0	0.5	1.0	n/a	n/a	n/a	n/a	n/a
PDF-Extract-Kit	1.0	1.0	1.0	1.0	1.0	0.83	1.0	1.0
Marker	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Texify	0.0	0.0	0.5	0.67	0.0	0.17	0.22	0.67

Table 2: Accuracy scores of Models tested on eight different lecture notes (Appendix Table 1)

Transcription Model Ensembling Algorithm

Individually, many of these open-source models serve a clear purpose in which they perform well. PDF Extract Kit provided pre-trained models using network architectures like YOLO and UniMERNet [2], which each specialized in tasks spanning Mathematical formula detection to table transcription. However, no unified framework properly extracted all desirable components, including text, equations, and tables, from a single image. Furthermore, models specialized for text extraction would be less accurate in transcribing mathematical formulas.

To address this limitation, we introduce a cohesive method of model ensembling to tackle this task of information extraction. Namely, we used a formula detection neural network to predict bounding boxes for all potential desirable components. We then fix a sequence of model passes for all transcription models which may vary in specialization. Noting that equations can be considered as a subset of text, we generalize this notion of some types of desirable components being subsets of other types to list the specialized models from smallest to largest in scope. Our algorithm is an ordered iteration of processing for this list of specialized models.

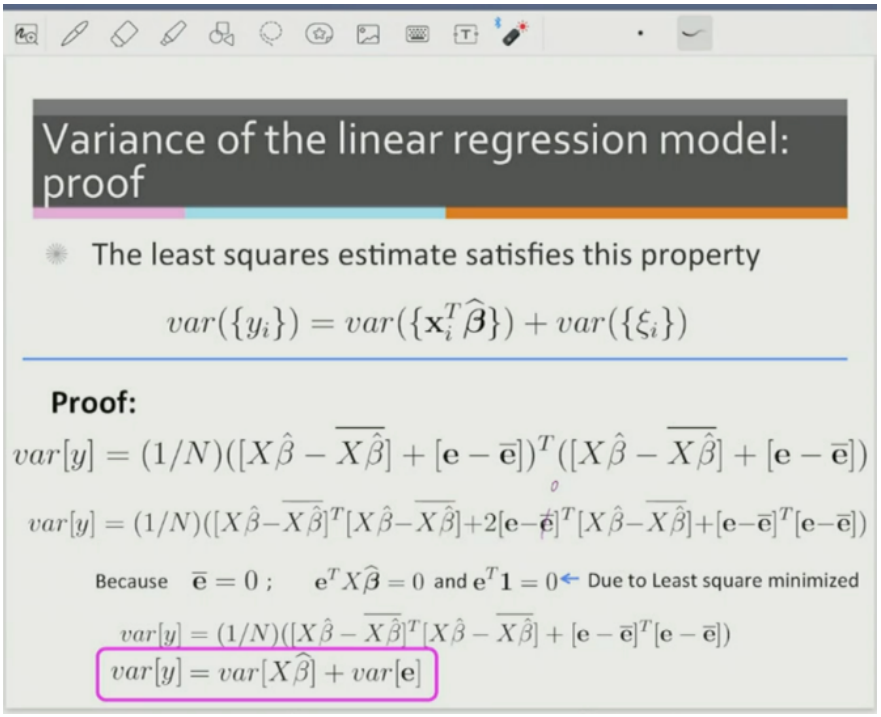


Figure 8: Example slide with multiple modalities of information across text, equations, and icons

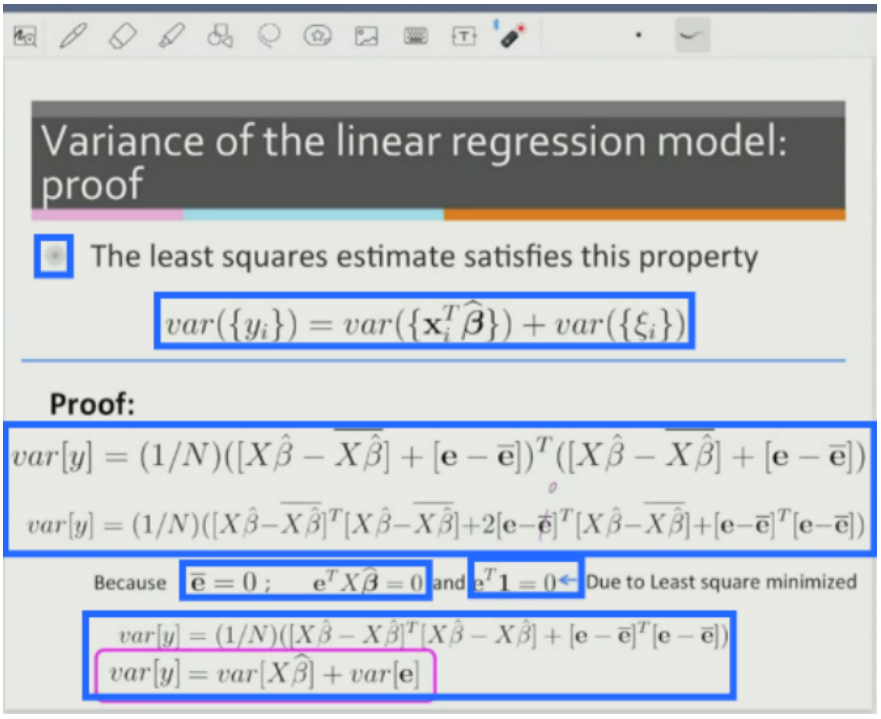


Figure 9: Bounding boxes of equations detected by YOLO and transcribed by UniMERNet

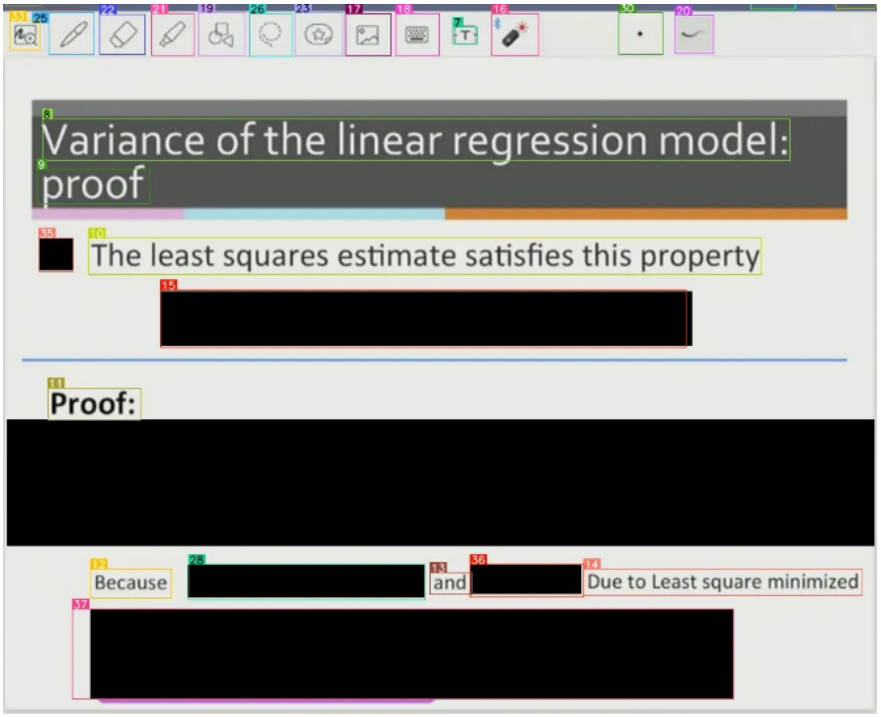


Figure 10: Bounding boxes of text and icons detected and transcribed by Omniparser [10] after equation bounding boxes are removed from the image

Intuitively and from experimentation, text models are more likely to incorrectly transcribe a Mathematical equation compared to mathematical models, as non-alphanumeric symbols and various Mathematical structures are out-of-domain for many text models. Inspired by this result, for each image, our unified algorithm first processes potential mathematical formula candidates as extracted by a fine-tuned YOLO detection model and generates a LaTeX transcription for each candidate through UniMERNet, as demonstrated in Fig. 9. Then, we blacken areas in the image as defined by the bounding boxes for those mathematical formula candidates before re-processing the image with the next specialized model in the list, as demonstrated in Fig. 10. This forces future models in the list to ignore previously processed transcription candidates and minimize interference from out-of-domain components. An example of this algorithm is shown in the figures above, displaying two specialized models in equation transcription and text transcription. Purely for visualization purposes, the transcribed equations are denoted by the blue bounding boxes in Fig. 9 and the text and icons are displayed with miscellaneously colored bounding boxes in Fig. 10. After each model, transcriptions are collected into a JSON dictionary, which can be directly used for any purpose including textual reconstructions of the original image.

This algorithm uniquely enables processing of complex images consisting of multiple modalities of information from text and equations to tables and visual icons. Independently, each model within the algorithm often fails to transcribe all elements of the images due to the previously mentioned training domain issues. Therefore, the core contribution of this algorithm is a flexible end-to-end approach for information transcription.

Transcribing slides to English

For further accessibility, translating LaTeX into English through spoken Math systems was explored with the Speech Rule Engine [11] and MathJax [9]. These models can translate LaTex to two different spoken Math systems: MathSpeak and ClearSpeak. MathSpeak focuses on concise language while ClearSpeak focuses on more natural language. For example the LaTex of a^b would give the output "a Superscript b" with the MathSpeak domain and "a to the b-th power" with the ClearSpeak domain. This enables further accessibility to Mathematical material through verbal methods.

We also looked into using Large Language Models (LLMs) to transcribe diagrams and explain equations from class slides. Specifically, we compared the models GPT-4o [12], Gemini, and llama 3.2-vision [13]. GPT-4o was able to explain equations step-by-step and provide a general summary of the goals of the equation. Gemini also explained the equations, however was also able to provide enrichment examples of applications of equations. Llama 3.2-vision was the least accurate of the three LLMs, providing general summaries that sometimes did not match the premise of the equation. A similar pattern followed with explaining diagrams: all three LLMs were able to extract the main subject of the diagram, but GPT-4o and Gemini were able to add enrichment information as well as go more in depth with the explanations of parts of the diagram.

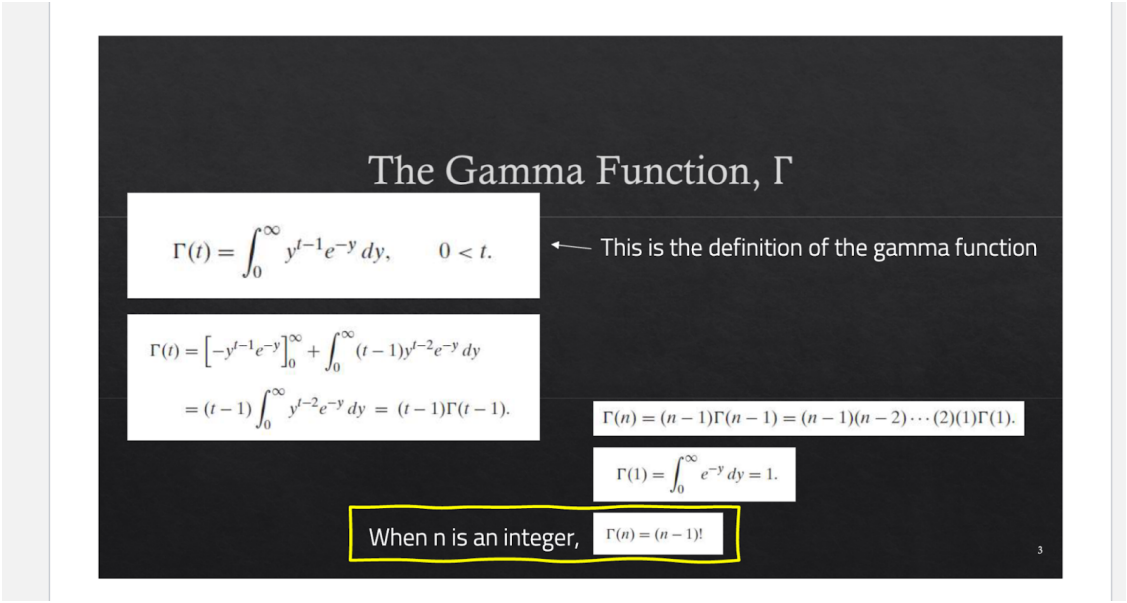


Figure 11: Example input lecture slide for LLMs

Development of methods for complex multi-line equations and tables

A category of equations to note is multiline equations. These are important in areas such as industrial engineering, where optimization problems are often written with a single constraint on each line. An understanding of each constraint does not equate to an understanding of the optimization program as a whole, which means that it is important for models and ultimately our tool to be able to reconstruct multiline equations without losing context.

Model Name	Summary of Response
GPT-4o	The model provides an overview of the gamma function, states the definition of the gamma function, explains the recursive property, and discusses special cases. The output does not convert the equations into LaTeX.
Gemini	The model defines the gamma function, explains the conditions for the gamma function, and provides an example of a gamma function. The output does not convert the equations into LaTeX.
Llama 3.2-vision	The model defines the gamma function and explains its applications. It does not include information about the lecture slide or convert the equations into LaTeX.

Table 3: Summary of LLM responses for the lecture slide in Fig. 11

Multiline equations pose a special challenge due to the combination of bounding box splitting and transcription formatting. When a detection model sees a multiline equation, it may regard each line as an isolated equation, severing any connection between the lines. Regardless of how the downstream transcription model performs, it would be impossible to reconstruct the full multiline equation. However, if the detection model does return the correct bounding box, the transcription model could lack the ability to reason about the formatting of the equation in LaTeX. A naive model may output spaces rather than introduce an align environment, for instance, which would produce an incorrect layout.

Many of the models mentioned above failed when transcribing tabular data. As a result, we decided to find models specifically for transcribing tables. We compared two models for our analysis: Table Transformer and StructEqTable-Deploy. Table Transformer is an open-source deep-learning model that analyzes table structure to identify tables and their structures [14]. The model uses an image or PDF as an input, and provides bounding boxes for rows, columns, and headers for the table. Additional OCR would be necessary to fully transcribe the table. Overall, the model had a 90% accuracy at identifying whether a table was present but only had a 30% accuracy of correctly identifying the cells and headers of the table. Two major errors included the cropping of tables and incorrectly identifying the headers of the table.

Name	Num.	df	Coefficient	Std. Error	Z Score	Nonlinear P-value
Positive effects						
our	5	3.9	0.566	0.114	4.970	0.052
over	6	3.9	0.244	0.195	1.249	0.004
remove	7	4.0	0.949	0.183	5.201	0.093
internet	8	4.0	0.524	0.176	2.974	0.028
free	16	3.9	0.507	0.127	4.010	0.065
business	17	3.8	0.779	0.186	4.179	0.194
hpl	26	3.8	0.045	0.250	0.181	0.002
ch!	52	4.0	0.674	0.128	5.283	0.164
ch$	53	3.9	1.419	0.280	5.062	0.354
CAPMAX	56	3.8	0.247	0.228	1.080	0.000
CAPTOT	57	4.0	0.755	0.165	4.566	0.063
Negative effects						
hp	25	3.9	−1.404	0.224	−6.262	0.140
george	27	3.7	−5.003	0.744	−6.722	0.045
1999	37	3.8	−0.672	0.191	−3.512	0.011
re	45	3.9	−0.620	0.133	−4.649	0.597
edu	46	4.0	−1.183	0.209	−5.647	0.000

Name	Num.	df	Coefficient	Std. Error	Z Score	Nonlinear P-value
			Positive effects			
our	5	3.9	0.566	0.114	4.970	0.052
over	6	3.9	0.244	0.195	1.249	0.004
remove	7	4.0	0.949	0.183	5.201	0.093
internet	8	4.0	0.524	0.176	2.974	0.028
free	16	3.9	0.507	0.127	4.010	0.065
business	17	3.8	0.779	0.186	4.179	0.194
hpl	26	3.8	0.045	0.250	0.181	0.002
ch!	52	4.0	0.674	0.128	5.283	0.164
ch$	53	3.9	1.419	0.280	5.062	0.354
CAPMAX	56	3.8	0.247	0.228	1.080	0.000
CAPTOT	57	4.0	0.755	0.165	4.566	0.063
			Negative effects			
hp	25	3.9	−1.404	0.224	−6.262	0.140
george	27	3.7	−5.003	0.744	−6.722	0.045
1999	37	3.8	−0.672	0.191	−3.512	0.011
re	45	3.9	−0.620	0.133	−4.649	0.597
edu	46	4.0	−1.183	0.209	−5.647	0.000

Figure 12: Example of TableTransformer correctly identifying a table and its structure

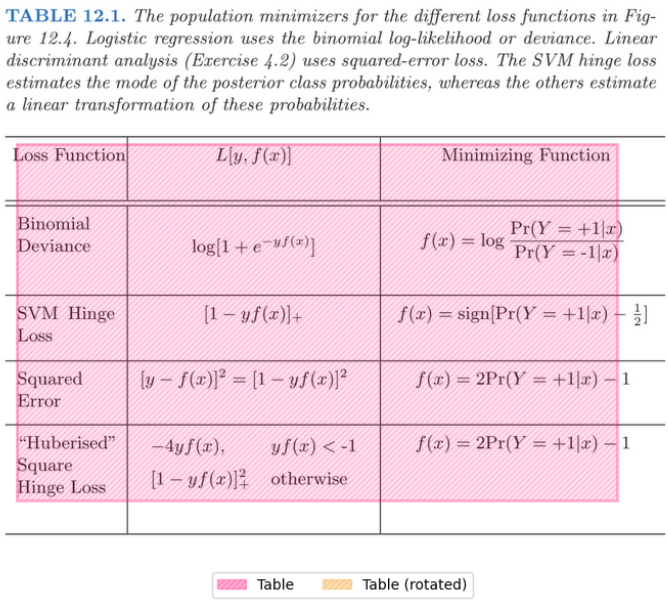


TABLE 12.1. *The population minimizers for the different loss functions in Figure 12.4. Logistic regression uses the binomial log-likelihood or deviance. Linear discriminant analysis (Exercise 4.2) uses squared-error loss. The SVM hinge loss estimates the mode of the posterior class probabilities, whereas the others estimate a linear transformation of these probabilities.*

Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\Pr(Y = +1\|x)}{\Pr(Y = -1\|x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x) = \text{sign}[\Pr(Y = +1\|x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2 = [1 - yf(x)]^2$	$f(x) = 2\Pr(Y = +1\|x) - 1$
"Huberised" Square Hinge Loss	$-4yf(x)$, $yf(x) < -1$ $[1 - yf(x)]_+^2$ otherwise	$f(x) = 2\Pr(Y = +1\|x) - 1$

Table Table (rotated)

Figure 13: Example of TableTransformer poorly cropping a table

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Figure 14: Example of TableTransformer incorrectly cropping the second column of the table resulting in incorrectly marking the "Coefficient" header

On the other hand, StructEqTable-Deploy [15] effectively detected tables within text and accurately transcribed their content, including associated captions and descriptions. The model excelled in handling complex scenarios, such as tables with multiple headers, asymmetric layouts, or empty cells. It also performed well with tables containing mathematical symbols and equations and successfully transcribed text with varied color formatting. However, the model occasionally rearranged columns, particularly shifting the last column to the first when space was constrained. Debugging efforts revealed that these errors were likely artifacts of the underlying transformer architecture used for processing. Addressing these challenges required careful adjustments to the input data and model parameters. Additionally, a failure case involved difficulty in transcribing images or symbols, e.g., triangles, resulting in errors. Despite these flaws, on average, the model's processing time was approximately 24 seconds per table, demonstrating a balance between accuracy and speed.

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲*= good,* ◆*=fair, and* ▼*=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

Table 1: **Some characteristics of different learning methods.** *Key:* ▲ *= good,* ►*=fair, and =poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▲	►	▲	▲	▲
Handling of missing values	▲	►	▲	△	▲
Robustness to outliers in input space	▲	▲	△	△	△
Insensitive to monotone transformations of inputs	▲	▲	▲	▲	▲
Computational scalability (large N)	▲	▲	△	▽	▼
Ability to deal with irrelevant inputs	▲	▲	△	◯	►
Ability to extract linear combinations of features	▲	▲	▲	▲▲	▲
Interpretability	▲	▲	▲	▲	▲
Predictive power	▲	▲	▲	▲	▲

Consider, for example, the microarray data in Table 18.4, taken from a study on the sensitivity of cancer patients to ionizing radiation treatment (Rieger et al., 2004). Each row consists of the expression of genes in 58 patient samples: 44 samples were from patients with a normal reaction, and 14 from patients who had a severe reaction to radiation. The measurements were made on oligonucleotide microarrays. The object of the experiment was to find genes whose expression was different in the radiation sensitive group of patients. There are $M = 12,625$ genes altogether; the table shows the data for some of the genes and samples for illustration. To identify informative genes, we construct a two-sample t-statistic for each gene.

Figure 15: An example of StructEq-Table incorrectly transcribing a table, where the table on the left is the input and the table on the right is StructEq-Table's output, produced in 25 seconds.

TABLE 12.3. *Vowel recognition data performance results. The results for neural networks are the best among a much larger set, taken from a neural network archive. The notation FDA/BRUTO refers to the regression method used with FDA.*

	Technique	Error Rates	
		Training	Test
(1)	LDA	0.32	0.56
	Softmax	0.48	0.67
(2)	QDA	0.01	0.53
(3)	CART	0.05	0.56
(4)	CART (linear combination splits)	0.05	0.54
(5)	Single-layer perceptron		0.67
(6)	Multi-layer perceptron (88 hidden units)		0.49
(7)	Gaussian node network (528 hidden units)		0.45
(8)	Nearest neighbor		0.44
(9)	FDA/BRUTO	0.06	0.44
	Softmax	0.11	0.50
(10)	FDA/MARS (degree = 1)	0.09	0.45
	Best reduced dimension (=2)	0.18	0.42
	Softmax	0.14	0.48
(11)	FDA/MARS (degree = 2)	0.02	0.42
	Best reduced dimension (=6)	0.13	0.39
	Softmax	0.10	0.50

Table 4: 12.3. Vogue recognition data performance results. The results for neural networks are the best among a much larger set, taken from a neural network architecture. The notation *FDA/BRUTO* refers to the regression method used with *FDA*.

	Technique	Error Rates	
		Training	Test
(1)	LDA	0.32	0.56
	Softmax	0.48	0.67
(2)	QDA	0.01	0.53
(3)	CART	0.05	0.56
(4)	CART (linear combination splits)	0.05	0.54
(5)	Single-layer perceptron		0.67
(6)	Multi-layer perceptron (88 hidden units)		0.49
(7)	Gaussian node network (528 hidden units)		0.45
(8)	Nearest neighbor		0.44
(9)	FDA/BRUTO	0.06	0.44
	Softmax	0.11	0.50
(10)	FDA/MARS (degree = 1)	0.09	0.45
	Best reduced dimension (=2)	0.18	0.42
	Softmax	0.14	0.48
(11)	FDA/MARS (degree = 2)	0.02	0.42
	Best reduced dimension (=6)	0.13	0.39
	Softmax	0.10	0.50

Figure 16: The left side shows the model's input, and the right side is the output, which closely matches the original LaTeX version converted to PNG via Overleaf in 24 seconds.

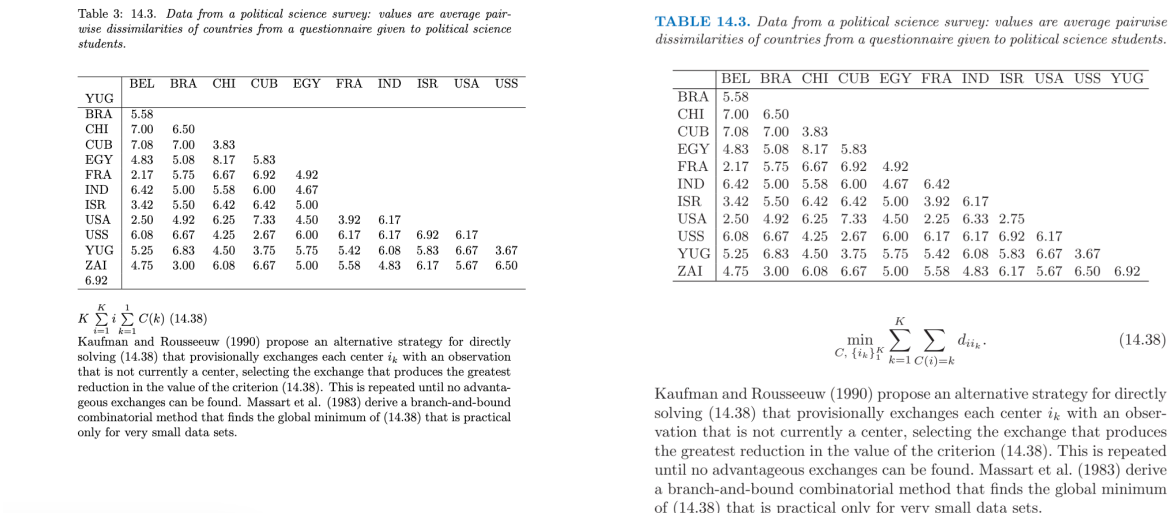


Table 3: 14.3. *Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.*

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS
YUG										
BRA	5.58									
CHI	7.00	6.50								
CUB	7.08	7.00	3.83							
EGY	4.83	5.08	8.17	5.83						
FRA	2.17	5.75	6.67	6.92	4.92					
IND	6.42	5.00	5.58	6.00	4.67					
ISR	3.42	5.50	6.42	6.42	5.00					
USA	2.50	4.92	6.25	7.33	4.50	3.92	6.17			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17	
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50
6.92										

$K \sum_{i=1}^{K} i \sum_{k=1}^{1} C(k)$ (14.38)

Kaufman and Rousseeuw (1990) propose an alternative strategy for directly solving (14.38) that provisionally exchanges each center i_k with an observation that is not currently a center, selecting the exchange that produces the greatest reduction in the value of the criterion (14.38). This is repeated until no advantageous exchanges can be found. Massart et al. (1983) derive a branch-and-bound combinatorial method that finds the global minimum of (14.38) that is practical only for very small data sets.

TABLE 14.3. *Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.*

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

$$\min_{C,\,\{i_k\}_1^K} \sum_{k=1}^{K} \sum_{C(i)=k} d_{ii_k}. \tag{14.38}$$

Kaufman and Rousseeuw (1990) propose an alternative strategy for directly solving (14.38) that provisionally exchanges each center i_k with an observation that is not currently a center, selecting the exchange that produces the greatest reduction in the value of the criterion (14.38). This is repeated until no advantageous exchanges can be found. Massart et al. (1983) derive a branch-and-bound combinatorial method that finds the global minimum of (14.38) that is practical only for very small data sets.

Figure 17: An example of StructEq-Table struggling with excess columns, placing the last column in the first position, which required 33 seconds to complete the conversion.

StructEq-Table was tested with 10 unique tables (see appendix 3). The outputs were evaluated using structural differences and textual differences as key comparison metrics. Structural differences assess the number of cells and their alignment, measuring how accurately the model preserves the original table's structure. The ratio used can be represented as:

$$\text{Structural Difference} = \frac{\text{Number of misaligned cells}}{\text{Total number of cells in the original table}}$$

Then, for textual comparison, we use the Levenshtein Distance, which uses an equation to calculate the number of insertions, deletions, and substitutions required to transform the transcribed text into the original text. The difference score is given by:

$$\text{Levenshtein Difference} = \frac{\text{Levenshtein Distance}(S_1, S_2)}{\max(|S_1|, |S_2|)}$$

By combining structural difference analysis and Levenshtein similarity, we were able to evaluate both layout preservation and content accuracy, ensuring a comprehensive assessment of the model's transcription performance.

Overall, both Table Transformer and StructEqTable-Deploy provided valuable insights that helped shape our own approach, enabling us to better detect and transcribe complex tables—especially those involving intricate mathematical structures.

Towards the development of a prototype equation/diagram converter

To visualize and demonstrate the capabilities of the extraction pipeline, we are currently in progress of creating a standalone web based application that converts image file inputs of instructional materials to an output that highlights important features such as table detection and

extraction, as well as transcribed mathematical content. To use the application, the user will first upload an image or slide-set of the material that they want to be analyzed. Then, we will run the pipeline to extract the desirable components from the input image, including the coordinates of the bounding boxes, as well as the transcribed content within these bounding boxes. The output of this pipeline can be found in Fig. 4 of the Appendix. As the output to the user, we will display the image with the highlighted bounding boxes and the corresponding transcription. To prototype this application, we use a JavaScript frontend framework, Vue.js. For the backend, which runs the detection, extraction, and transcription operations, we use a Python web framework FastAPI alongside a task queue Huey for future work scaling. We currently have a functional prototype that takes an input image and outputs mathematical equations in LaTeX alongside other text found in the image. With this tool, we demonstrate a potential resource both educators and students could utilize to engage with material as effectively as possible.

Conclusion

In this paper, we surveyed and evaluated many of the existing open-source machine learning models relating to STEM content transcription. Through extensive evaluation methods, our paper outlines the strengths and limitations of these machine learning networks. PDF-Extract Kit performed the best when converting lecture slides into a LaTeX format. Additionally, StructEqTable-Deploy performed the best at identifying table structures in educational materials.

However, none of these methods, including large language models, were able to effectively transcribe complex images containing multiple modes of information across text and visual modalities. The specialized models demonstrated trade-offs between performance in information candidate detection, text transcription, and equation transcription. Through our model ensembling algorithm, users can leverage the orthogonal strengths of various models to more comprehensively transcribe the information transmitted by an image.

Our work emphasizes the value of accessible models, whether it be through open-source contributions e.g., PDF-Extract Kit or company-subsidized models e.g., GPT-4o. Through our published code of modeling pipelines and user interface software, instructors gain access to a transcription algorithm optimized for cost and performance. With minimal additional scripting, users can ensemble the latest transcription models to process complex images containing text, equations, and visual icons.

Discussion

This paper addresses the problem of automating the processing and parsing of mathematical content in various forms, including equations, tables, slide layouts, and diagrams. Equations are a significant focus in our work, encompassing single-line, multi-line, handwritten, and equations with unconventional formatting or fonts. Multi-line equations present significant challenges, as models often split them into separate components, losing their contextual meaning. These equations are transcribed into LaTeX for accessibility and can be further converted into spoken formats e.g., MathSpeak and ClearSpeak for verbal communication. Tables, another significant element of our work, range from simple structures to complex layouts involving multiple headers, asymmetry, and embedded mathematical symbols. These tables are often misinterpreted during transcription, with common issues including cropping errors, header misidentification, and

rearranged columns. In addition to equations and tables, unstructured slide layouts containing mixed content (e.g., text, equations, diagrams, handwriting) pose unique challenges for transcription models. Diagrams and visual icons also require detailed textual explanations to ensure accessibility, particularly for students with disabilities.

The libraries and machine learning algorithms we studied in this paper included Pix2Text, PDF-Extract-Kit, Marker, and Texify. We also analyzed the problem of transcription in a multimodal setting. We analyzed additional tools, Speech Rule Engine [11] and MathJax [9], which translate LaTeX into spoken formats, enhancing accessibility and AI models for contextual enrichment and explanation. We have also studied the problem of transcription of complex layouts in tables and multiline equations. All these elements indicate the need for further experimentation, development, and research to create versatile, robust, and scalable tools for Mathematical annotation and processing, which is critical for higher education in STEM fields.

Future Work

Due to the fast-paced developments in the field, we cannot evaluate all new models in this paper. Newer models like DeepSeek VL2 [16] (a multi-modal vision language model) and LaTeXNet [17] (another specialized ensemble model) are being evaluated for implementation in our tool under `https://github.com/classtranscribe/latextranscribe`. Texify / Marker have also undergone significant development since our initial evaluation.

Acknowledgments

This work was supported in part by a GIANT 2024-25 grant sponsored by the Institute for Inclusion, Diversity, Equity and Access (IDEA) in the Grainger College of Engineering at the University of Illinois Urbana-Champaign, and by a Strategic Instructional Innovations Program (SIIP) award from the Grainger College of Engineering. Additional support was provided by a Microsoft grant for research on Universal Design for Learning (UDL) practices in higher education.

References

[1] B. Wang, F. Wu, L. Ouyang, Z. Gu, R. Zhang, R. Xia, B. Zhang, and C. He, "Image over text: Transforming formula recognition evaluation with character detection matching," 2025. [Online]. Available: https://arxiv.org/abs/2409.03643

[2] B. Wang, Z. Gu, G. Liang, C. Xu, B. Zhang, B. Shi, and C. He, "Unimernet: A universal network for real-world mathematical expression recognition," 2024. [Online]. Available: https://arxiv.org/abs/2404.15254

[3] breezedeus. (2024) Pix2text. [Online]. Available: https://github.com/breezedeus/pix2text

[4] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: Transformer-based optical character recognition with pre-trained models," 2022. [Online]. Available: https://arxiv.org/abs/2109.10282

[5] B. Wang. (2024) Pdf extract kit. [Online]. Available: https://github.com/opendatalab/PDF-Extract-Kit

[6] V. Paruchuri. (2024) Texify. [Online]. Available: https://github.com/VikParuchuri/texify

[7] ——. (2024) Marker. [Online]. Available: https://github.com/VikParuchuri/marker

[8] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park, "Ocr-free document understanding transformer," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 498–517. [Online]. Available: https://doi.org/10.1007/978-3-031-19815-1_29

[9] D. Cervone, "Mathjax: A platform for mathematics on the web," *Notices of the American Mathematical Society*, vol. 59, no. 02, p. 1, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1090/noti794

[10] Y. Lu, J. Yang, Y. Shen, and A. Awadallah, "Omniparser for pure vision based gui agent," 2024. [Online]. Available: https://arxiv.org/abs/2408.00203

[11] V. Sorge. (2024) Speech rule engine. [Online]. Available: https://github.com/Speech-Rule-Engine/speech-rule-engine

[12] O. et al., "Gpt-4o system card," 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[13] A. G. et al., "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

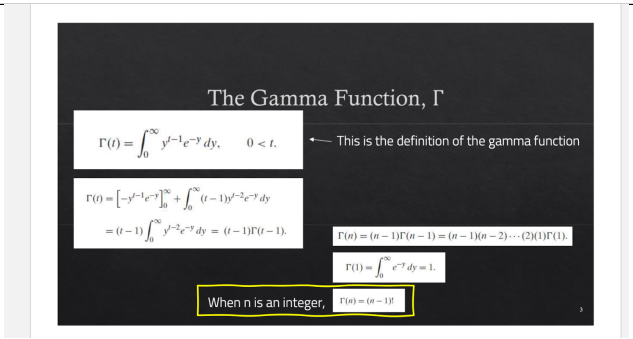
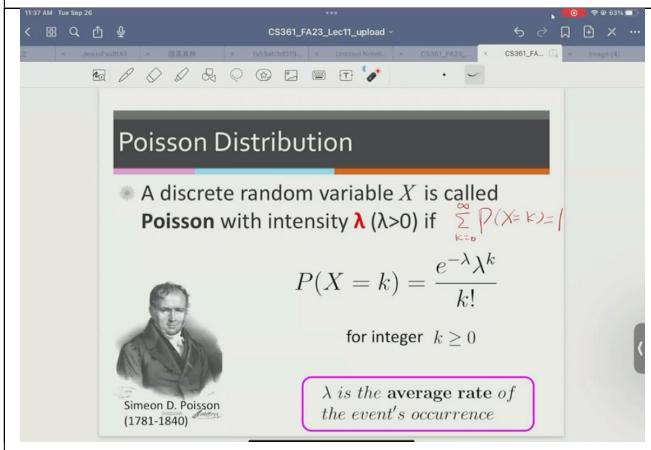
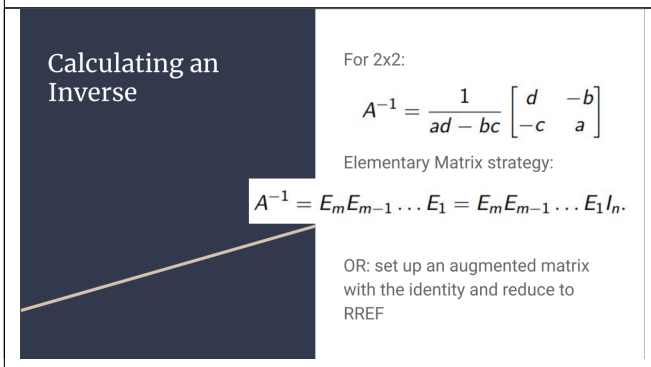
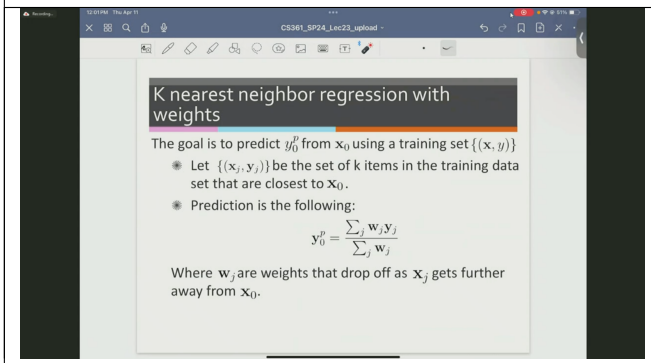
[14] B. Smock, R. Pesala, and R. Abraham, "Pubtables-1m: Towards comprehensive table extraction from unstructured documents," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4624–4632.

[15] R. Xia, S. Mao, X. Yan, H. Zhou, B. Zhang, H. Peng, J. Pi, D. Fu, W. Wu, H. Ye *et al.*, "Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models," *arXiv preprint arXiv:2406.11633*, 2024.

[16] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Z. Xie, Y. Wu, K. Hu, J. Wang, Y. Sun, Y. Li, Y. Piao, K. Guan, A. Liu, X. Xie, Y. You, K. Dong, X. Yu, H. Zhang, L. Zhao, Y. Wang, and C. Ruan, "Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding," 2024. [Online]. Available: https://arxiv.org/abs/2412.10302

[17] R. Xia, H. Zhou, Z. Feng, H. Liu, B. Chen, B. Zhang, and J. Yan, "Latexnet: A specialized model for converting visual tables and equations to latex code," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

Appendix

Test Image	Pix2Text	PEK	Marker	Texify
	1.0	1.0	N/A	0.0
	0.5	1.0	N/A	0.0
	1.0	1.0	N/A	0.5
	N/A	1.0	N/A	0.67

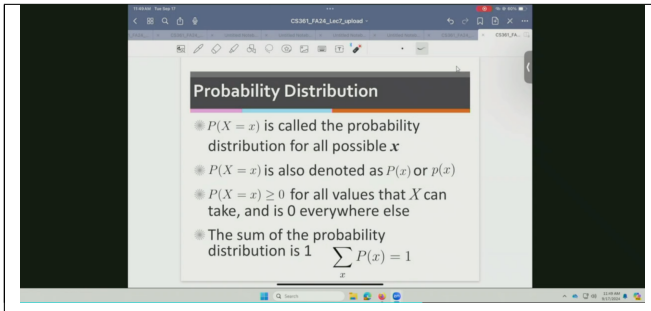
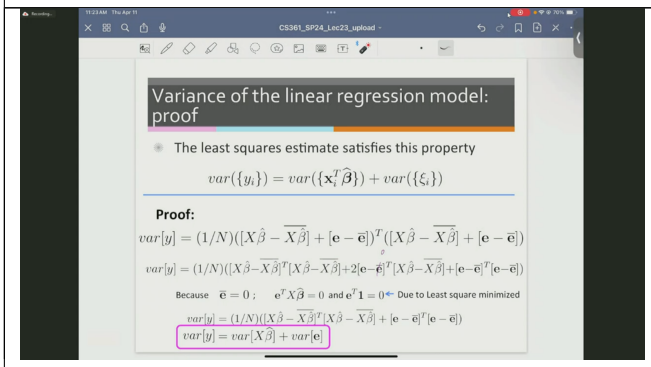
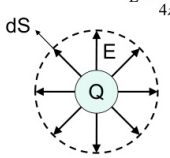
	N/A	1.0	N/A	0.0
	N/A	0.83	N/A	0.17
General result: If $f : \mathbb{R}^N \to \mathbb{R}$, $g_n : \mathbb{R}^M \to \mathbb{R}$ for $n \in \{1, 2 \ldots N\}$ and $\tilde{f}(s_1, s_2 \ldots s_M) \stackrel{\text{def}}{=} f\left(g_1(s_1, s_2 \ldots s_M), g_2(s_1, s_2 \ldots s_M) \cdots g_N(s_1, s_2 \ldots s_M)\right)$ then $\frac{\partial \tilde{f}}{\partial s_m}(s_1, s_2 \ldots s_M) = \sum_{n=1}^{N} \frac{\partial f}{\partial x_n}\left(g_1(s_1, s_2 \ldots s_M), g_2(s_1, s_2 \ldots s_M) \cdots g_N(s_1, s_2 \ldots s_M)\right) \times \frac{\partial g_n}{\partial s_m}(s_1, s_2 \ldots s_N)$ Alternate example (thinking of polar coordinates): $f(a, b) = a^2 b^5 \quad g_1(r, \theta) = r\cos(\theta) \quad g_2(r, \theta) = r\sin(\theta)$ Then $\tilde{f}(r, \theta) = (r\cos(\theta))^2 (r\sin(\theta))^5$	N/A	1.0	N/A	0.22
Electric Flux = Charge Enclosed • Coulomb's Law for the Electric field of a point charge: $\vec{E} = \frac{Q}{4\pi\varepsilon_0 R^2}\hat{a}_R$  Define D=ε_0E to be the "displacement flux density" $\psi_E = \oiint_S \vec{D} \bullet d\vec{S} = \oiint_S \varepsilon_0 \vec{E} \bullet d\vec{S}$ $= \varepsilon_0 E(Surf\ Area)$ $= \varepsilon_0 \frac{Q}{4\pi\varepsilon_0 R^2}(4\pi R^2) = Q$ 8 ECE329 Lectures 3	N/A	1.0	N/A	0.67

Table A.1: Accuracy scores of transcribing lecture slides from various models

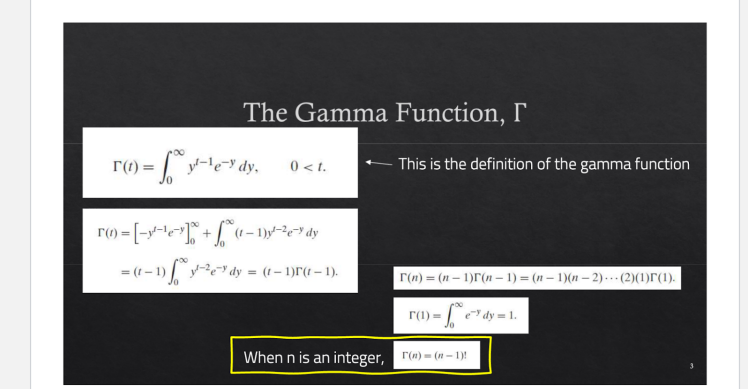


Figure A.1: Comparison between the summaries of LLM models for the first dataset image

GPT-4o

This slide explains the Gamma function, denoted as $\Gamma(t)$, which is a generalization of the factorial function to real and complex numbers. Here's a step-by-step explanation of what's shown:

Definition of the Gamma Function

The Gamma function is defined as:

$$\Gamma(t) = \int_0^\infty y^{t-1}e^{-y}\,dy, \quad t > 0$$

y^{t-1}: This represents a power function of y, where the exponent depends on t.

Recursive Property

By integration by parts, the Gamma function satisfies a recursive property:

$$\Gamma(t) = (t-1)\Gamma(t-1)$$

This is derived as follows: Start with the definition:

$$\Gamma(t) = \int_0^\infty y^{t-1}e^{-y}\,dy$$

Apply integration by parts:

$$\int u\,dv = uv - \int v\,du$$

Let $u = y^{t-1}$ and $dv = e^{-y}dy$.

Special Cases

- **Gamma Function at 1:**
$$\Gamma(1) = \int_0^\infty e^{-y}\, dy = 1$$
This establishes a base case.

- **When $t = n$ (a positive integer):** Using the recursive property:
$$\Gamma(n) = (n-1)\Gamma(n-1) = (n-1)(n-2)\Gamma(n-2) = \cdots = (n-1)!\Gamma(1) = (n-1)!$$
Thus, the Gamma function generalizes the factorial function, satisfying:
$$\Gamma(n) = (n-1)!$$

Summary

- $\Gamma(t)$ generalizes factorials to non-integer values.
- Recursive relation: $\Gamma(t) = (t-1)\Gamma(t-1)$.
- For integers n: $\Gamma(n) = (n-1)!$.

Gemini

The Gamma Function, Γ

The equation you've provided is the definition of the gamma function, denoted by $\Gamma(t)$. It's defined as an integral:
$$\Gamma(t) = \int_0^\infty y^{t-1}e^{-y}\, dy$$

Conditions: The gamma function is defined for all complex numbers t except for non-positive integers ($t \neq 0, -1, -2, \ldots$).

Key Points:

- The gamma function is a generalization of the factorial function. For positive integers n, $\Gamma(n) = (n-1)!$.
- It has applications in various fields of mathematics and physics, including probability theory, statistics, and quantum mechanics.

Example: Let's calculate $\Gamma(3)$:
$$\begin{aligned}\Gamma(3) &= \int_0^\infty y^{3-1}e^{-y}\, dy \\ &= \int_0^\infty y^2 e^{-y}\, dy\end{aligned}$$

This integral can be evaluated using integration by parts twice, and the result is:
$$\Gamma(3) = 2! = 2$$

This confirms the relationship between the gamma function and the factorial function for positive integers.

Additional Notes:

- The gamma function has many interesting properties and relationships with other special functions.
- It's a fundamental tool in many areas of mathematics and science.

Llama 3.2-vision

The Gamma function, denoted by the Greek letter Γ (Gamma), is a fundamental concept in mathematics that has numerous applications in various fields. It can be defined as:

$\Gamma(n) = (n-1)!$ for any positive integer n. This equation represents an integral transform of the factorial function, which is crucial for understanding and manipulating sequences and series that involve factorials. The Gamma function plays a significant role in number theory, probability theory, statistical mechanics, combinatorial analysis, special functions in mathematics, and physics, making it an essential tool in many mathematical disciplines.

In essence, the Gamma function can be thought of as an extension of the factorial function to real and complex numbers, providing a way to generalize the concept of factorials beyond integers. This makes it a powerful tool for solving problems that involve sequences and series with factorial terms.

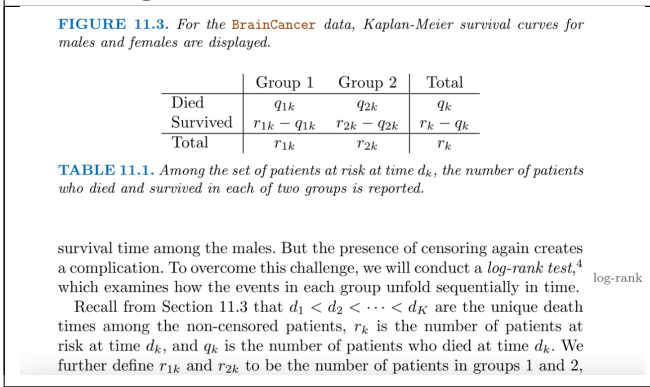
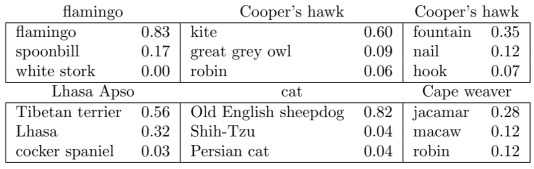
Test Image	(Structural, Levenshtein) Difference
 **FIGURE 11.3.** *For the* BrainCancer *data, Kaplan-Meier survival curves for males and females are displayed.* | | Group 1 | Group 2 | Total | |---|---|---|---| | Died | q_{1k} | q_{2k} | q_k | | Survived | $r_{1k} - q_{1k}$ | $r_{2k} - q_{2k}$ | $r_k - q_k$ | | Total | r_{1k} | r_{2k} | r_k | **TABLE 11.1.** *Among the set of patients at risk at time d_k, the number of patients who died and survived in each of two groups is reported.* survival time among the males. But the presence of censoring again creates a complication. To overcome this challenge, we will conduct a *log-rank test*,[4] which examines how the events in each group unfold sequentially in time. log-rank Recall from Section 11.3 that $d_1 < d_2 < \cdots < d_K$ are the unique death times among the non-censored patients, r_k is the number of patients at risk at time d_k, and q_k is the number of patients who died at time d_k. We further define r_{1k} and r_{2k} to be the number of patients in groups 1 and 2,	(6.25%, 3.09%)
 flamingo: flamingo 0.83, spoonbill 0.17, white stork 0.00 Cooper's hawk: kite 0.60, great grey owl 0.09, robin 0.06 Cooper's hawk: fountain 0.35, nail 0.12, hook 0.07 Lhasa Apso: Tibetan terrier 0.56, Lhasa 0.32, cocker spaniel 0.03 cat: Old English sheepdog 0.82, Shih-Tzu 0.04, Persian cat 0.04 Cape weaver: jacamar 0.28, macaw 0.12, robin 0.12	(0.0%, 0.0%)

TABLE 10.1. *Some characteristics of different learning methods. Key: ▲= good, ◆=fair, and ▼=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

(0.0%, 51.1%)

TABLE 9.2. *Significant predictors from the additive model fit to the spam training data. The coefficients represent the linear part of $\hat{f}_j$, along with their standard errors and Z-score. The nonlinear P-value is for a test of nonlinearity of $\hat{f}_j$.*

Name	Num.	df	Coefficient	Std. Error	Z Score	Nonlinear P-value
Positive effects						
`our`	5	3.9	0.566	0.114	4.970	0.052
`over`	6	3.9	0.244	0.195	1.249	0.004
`remove`	7	4.0	0.949	0.183	5.201	0.093
`internet`	8	4.0	0.524	0.176	2.974	0.028
`free`	16	3.9	0.507	0.127	4.010	0.065
`business`	17	3.8	0.779	0.186	4.179	0.194
`hpl`	26	3.8	0.045	0.250	0.181	0.002
`ch!`	52	4.0	0.674	0.128	5.283	0.164
`ch$`	53	3.9	1.419	0.280	5.062	0.354
`CAPMAX`	56	3.8	0.247	0.228	1.080	0.000
`CAPTOT`	57	4.0	0.755	0.165	4.566	0.063
Negative effects						
`hp`	25	3.9	−1.404	0.224	−6.262	0.140
`george`	27	3.7	−5.003	0.744	−6.722	0.045
`1999`	37	3.8	−0.672	0.191	−3.512	0.011
`re`	45	3.9	−0.620	0.133	−4.649	0.597
`edu`	46	4.0	−1.183	0.209	−5.647	0.000

(0.0%, 0.0%)

TABLE 14.3. *Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.*

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

$$\min_{C,\,\{i_k\}_1^K} \sum_{k=1}^{K} \sum_{C(i)=k} d_{ii_k}. \tag{14.38}$$

Kaufman and Rousseeuw (1990) propose an alternative strategy for directly solving (14.38) that provisionally exchanges each center i_k with an observation that is not currently a center, selecting the exchange that produces the greatest reduction in the value of the criterion (14.38). This is repeated until no advantageous exchanges can be found. Massart et al. (1983) derive a branch-and-bound combinatorial method that finds the global minimum of (14.38) that is practical only for very small data sets.

(9.09%, 18.9%)

TABLE 18.4. *Subset of the 12,625 genes from microarray study of radiation sensitivity. There are a total of 44 samples in the normal group and 14 in the radiation sensitive group; we only show three samples from each group.*

	Normal				Radiation Sensitive			
Gene 1	7.85	29.74	29.50	...	17.20	-50.75	-18.89	...
Gene 2	15.44	2.70	19.37	...	6.57	-7.41	79.18	...
Gene 3	-1.79	15.52	-3.13	...	-8.32	12.64	4.75	...
Gene 4	-11.74	22.35	-36.11	...	-52.17	7.24	-2.32	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Gene 12,625	-14.09	32.77	57.78	...	-32.84	24.09	-101.44	...

Consider, for example, the microarray data in Table 18.4, taken from a study on the sensitivity of cancer patients to ionizing radiation treatment (Rieger et al., 2004). Each row consists of the expression of genes in 58 patient samples: 44 samples were from patients with a normal reaction, and 14 from patients who had a severe reaction to radiation. The measurements were made on oligo-nucleotide microarrays. The object of the experiment was to find genes whose expression was different in the radiation sensitive group of patients. There are $M = 12,625$ genes altogether; the table shows the data for some of the genes and samples for illustration.

To identify informative genes, we construct a two-sample t-statistic for each gene.

$$\bar{x}_{2i} - \bar{x}_{1i}$$

(43.75%, 61.9%)

TABLE 12.3. *Vowel recognition data performance results. The results for neural networks are the best among a much larger set, taken from a neural network archive. The notation FDA/BRUTO refers to the regression method used with FDA.*

	Technique	Error Rates	
		Training	Test
(1)	LDA	0.32	0.56
	Softmax	0.48	0.67
(2)	QDA	0.01	0.53
(3)	CART	0.05	0.56
(4)	CART (linear combination splits)	0.05	0.54
(5)	Single-layer perceptron		0.67
(6)	Multi-layer perceptron (88 hidden units)		0.49
(7)	Gaussian node network (528 hidden units)		0.45
(8)	Nearest neighbor		0.44
(9)	FDA/BRUTO	0.06	0.44
	Softmax	0.11	0.50
(10)	FDA/MARS (degree = 1)	0.09	0.45
	Best reduced dimension (=2)	0.18	0.42
	Softmax	0.14	0.48
(11)	FDA/MARS (degree = 2)	0.02	0.42
	Best reduced dimension (=6)	0.13	0.39
	Softmax	0.10	0.50

(0.0%, 0.0%)

TABLE 12.1. *The population minimizers for the different loss functions in Figure 12.4. Logistic regression uses the binomial log-likelihood or deviance. Linear discriminant analysis (Exercise 4.2) uses squared-error loss. The SVM hinge loss estimates the mode of the posterior class probabilities, whereas the others estimate a linear transformation of these probabilities.*

Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\Pr(Y=+1\|x)}{\Pr(Y=-1\|x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x) = \text{sign}[\Pr(Y=+1\|x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2 = [1 - yf(x)]^2$	$f(x) = 2\Pr(Y=+1\|x) - 1$
"Huberised" Square Hinge Loss	$-4yf(x)$, $yf(x) < -1$ $[1 - yf(x)]_+^2$ otherwise	$f(x) = 2\Pr(Y=+1\|x) - 1$

(0.0%, 0.28%)

TABLE 12.2. *Skin of the orange: Shown are mean (standard error of the mean) of the test error over 50 simulations. BRUTO fits an additive spline model adaptively, while MARS fits a low-order interaction model adaptively.*

	Method	Test Error (SE)	
		No Noise Features	Six Noise Features
1	SV Classifier	0.450 (0.003)	0.472 (0.003)
2	SVM/poly 2	0.078 (0.003)	0.152 (0.004)
3	SVM/poly 5	0.180 (0.004)	0.370 (0.004)
4	SVM/poly 10	0.230 (0.003)	0.434 (0.002)
5	BRUTO	0.084 (0.003)	0.090 (0.003)
6	MARS	0.156 (0.004)	0.173 (0.005)
	Bayes	0.029	0.029

(0.0%, 0.0%)

TABLE 1.1. *Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between* `spam` *and* `email`.

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

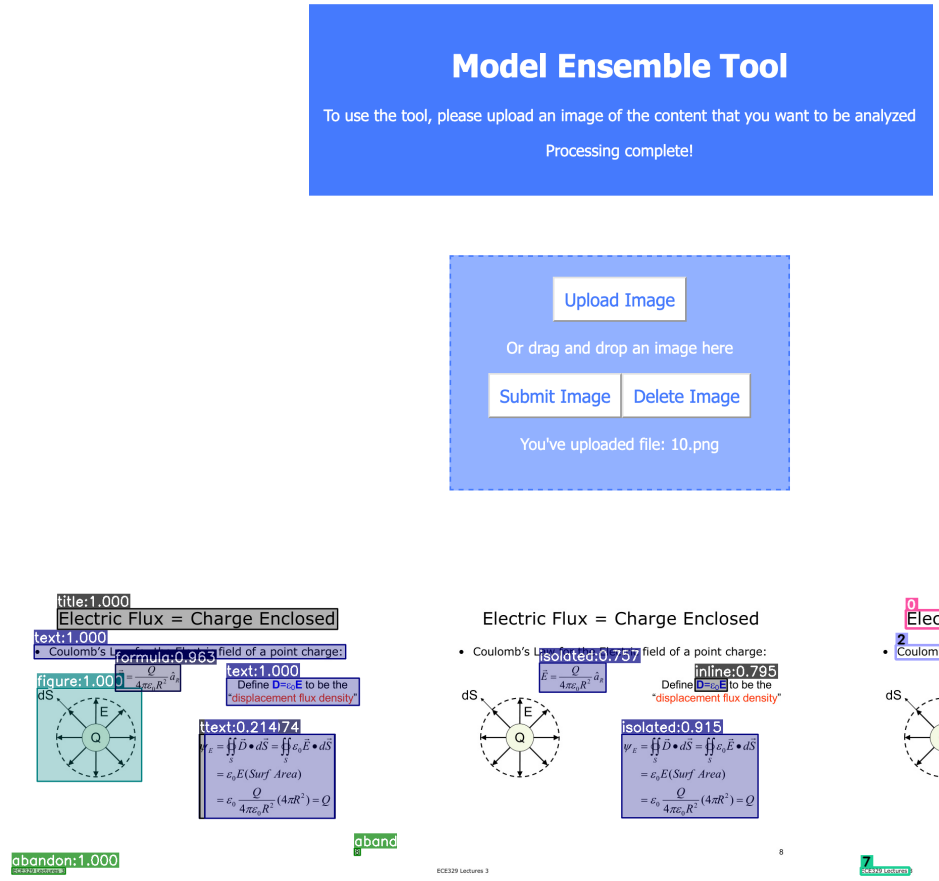
measurements for a set of objects (such as people). Using this data we build a prediction model, or *learner*, which will enable us to predict the outcome for new unseen objects. A good learner is one that accurately predicts such an outcome.

The examples above describe what is called the *supervised learning* problem. It is called "supervised" because of the presence of the outcome variable to guide the learning process. In the *unsupervised learning problem*, we observe only the features and have no measurements of the outcome. Our task is rather to describe how the data are organized or clustered. We devote most of this book to supervised learning; the unsupervised problem is less developed in the literature, and is the focus of Chapter 14.

Here are some examples of real learning problems that are discussed in this book.

(8.33%, 4.76%)

Table A.2: Accuracy scores of transcribing tables from various models



1. Type: formula

Position: 397.44, 312.53, 667.19, 477.53

$$\psi_E = \iiint_S \vec{D} \bullet d\vec{S} = \iint_S \varepsilon_0 \vec{E} \bullet d\vec{S}$$
$$= \varepsilon_0 E(Surf\ Area)$$
$$= \varepsilon_0 \frac{Q}{4\pi\varepsilon_0 R^2}(4\pi R^2) = Q$$

2. Type: formula

Position: 542.67, 202.8, 606.36, 227.36

D=ε_0E

2. Type: formula

Position: 542.67, 202.8, 606.36, 227.36

D=ε_0E

Figure A.2: Screenshots of work-in-progress tool and its outputs