https://youtu.be/MDS35h-oiCY
https://github.com/andialexandrescu/LatexOCR_Azure

LaTex OCR Engine via Azure Document Intelligence compared to pix2tex pretrained model, deployed in Azure Container Apps service

Empirical approach to auto detecting the bounded formula boxes

The first script, formula_detector.py, detects mathematical formulas in pdf pages, crops them, and saves each formula as an image. It uses PyMuPDF (fitz) to render pages and OpenCV (cv2) plus NumPy for image preprocessing, morphology, contour detection, and cropping.

- Besides rendering and image preprocessing, the algorithm detect candidate regions via morphological dilation to connect strokes and then apply contour detection, these are then merged to combine fragmented parts of the same formula.
- It filters by size and optionally screens out prose via some heuristics which decide what is the text most similar to (based on a computed factor), out of two options, which are formula and prose

The next script, debug_detections.py, is a practical tool for inspecting and validating the performance of the formula detection pipeline, meaning it is designed to visualize and debug the detections based on the FormulaDetector class defined previously.

- The script creates a copy of the page image for annotation.
- For each detected formula region, it draws a green rectangle around the region and labels it with a number.

OCR recognition for mathematical formulas

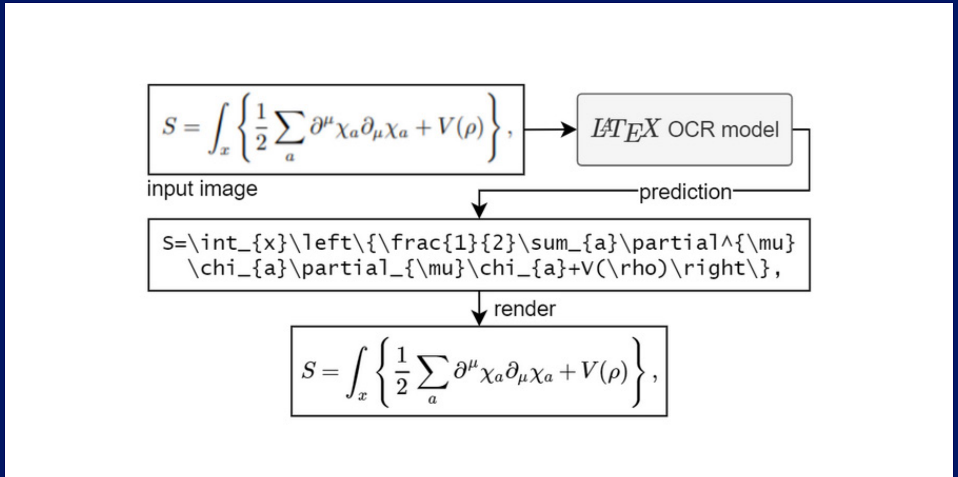
pix2tex VS Azure Document Intelligence

Before developing the Python and FastAPI app, a key decision for the LaTeX OCR involved identifying differences between the pix2tex LaTeX model and Azure Document Intelligence.

pix2tex

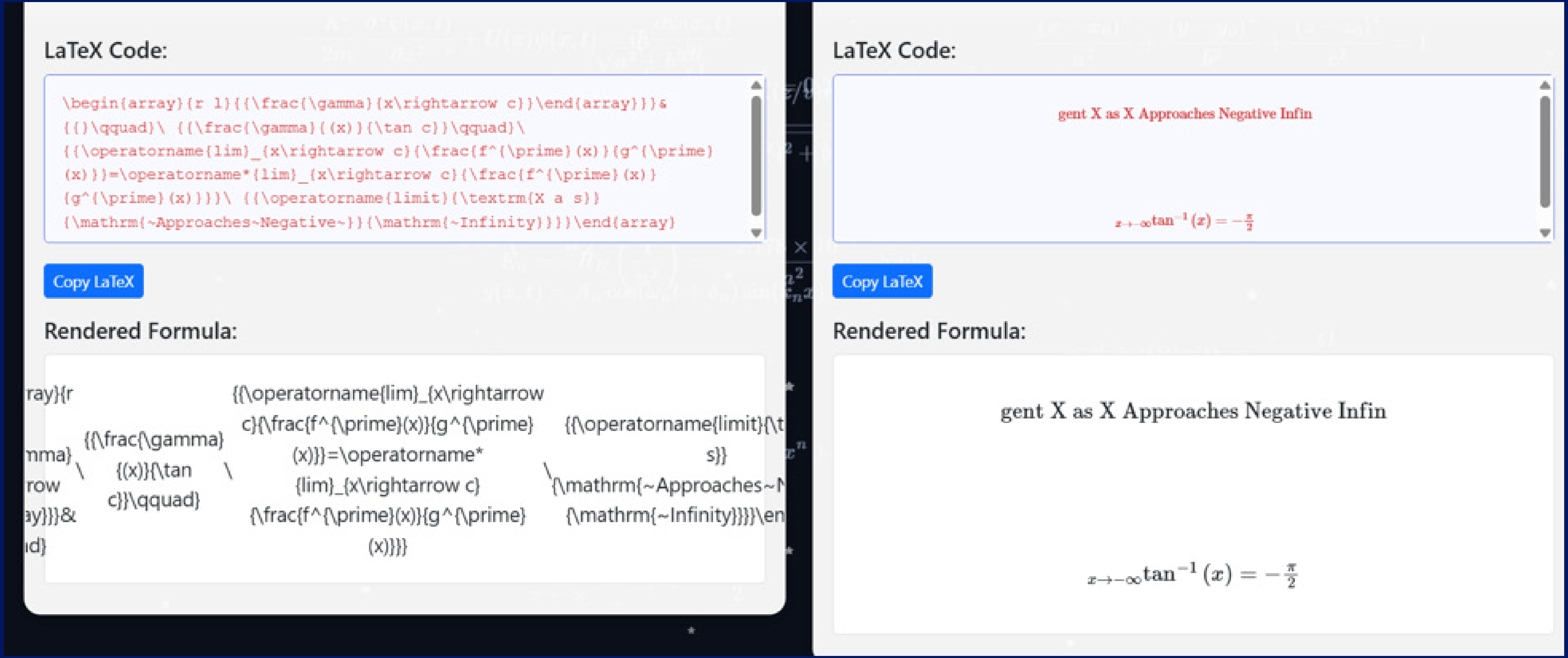
Part of the app.py integrations is pix2tex, which loads the pre-trained pix2tex model, ensuring that the neural network is ready for inference.

By leveraging the Pix2Tex model, it achieves high accuracy in recognizing complex mathematical notation, therefore an interesting approach when compared to Azure Document Intelligence sparked some interest, which is the fact that by cropping the formula region tightly without adding any extra padding to the bounding box gives optimal recognition results, unlike Azure's service.



To test the pix2tex LaTeX specialized method, another approach involved adding padding to the cropped formula by filling it with the surrounding area.

The output was not optimal since the pix2tex model attempts to parse prose as mathematical formulas, leading to failure in recognizing the actual formula boundaries.


LaTeX Code:
\begin{array}{r l}{{\frac{\gamma}{x\rightarrow c}}\end{array}}}&
{{}\qquad}\ {{\frac{\gamma}{(x)}{\tan c}}\qquad}\
{{\operatorname{lim}_{x\rightarrow c}{\frac{f^{\prime}(x)}{g^{\prime}
(x)}}=\operatorname*{lim}_{x\rightarrow c}{\frac{f^{\prime}(x)}
{g^{\prime}(x)}}}\ {{\operatorname{limit}{\textrm{X a s}}
{\mathrm{~Approaches~Negative~}}{\mathrm{~Infinity}}}}\end{array}
Copy LaTeX
Rendered Formula:
LaTeX Code:
gent X as X Approaches Negative Infin
Copy LaTeX
Rendered Formula:
gent X as X Approaches Negative Infin

Box #15 Page 2

LaTeX Code:

Extra close brace or missing open brace

Copy LaTeX

Rendered Formula:

Extra close brace or missing open brace

Box #16 Page 2

LaTeX Code:

rm time domain multiplication (z domanvolution)·

$$h(n)x(n) \Leftrightarrow \frac{1}{2\pi j} \oint_C H(v) X(z/v) v^{-1}\, dv$$

Copy LaTeX

Rendered Formula:

rm time domain multiplication (z domanvolution)·

$$h(n)x(n) \Leftrightarrow \frac{1}{2\pi j} \oint_C H(v) X(z/v) v^{-1}\, dv$$

Azure Document Intelligence

Part of the app.py integrations is Azure Document Intelligence, which initializes the cloud-based OCR service, ensuring robust document processing for mixed content analysis.

We are using Read which is a more general purpose model part of Document Intelligence Optical Character Recognition (OCR). The model runs at a higher resolution than Azure Vision Read and extracts print and handwritten text from pdf documents and scanned images. It can also detect paragraphs, text lines, words, locations, and languages.

Out use case of a pdf document containing both prose and formulas was tested in document_intelligence_extractor.ipynb, where we gave the Document Intelligence specific method begin_analyze_document a parameter specifing features=[DocumentAnalysisFeature.FORMULAS, DocumentAnalysisFeature.STYLE_FONT].

The Azure model detects formulas and replaces them with temporary :formula: placeholder in content, while the actual formulas are stored separately in page.formulas[] with full LaTeX code. Therefore, we had to implement a reconstruction method to display the LaTex code inline with prose text.

```
reconstructed_content = result.content  # "Text :formula: text"
for formula_data in all_formulas:
    reconstructed_content =
reconstructed_content.replace(":formula:",
formula_data['value'], 1)
# RESULT: "Text x=\\frac{-b...}{2a} text"
```

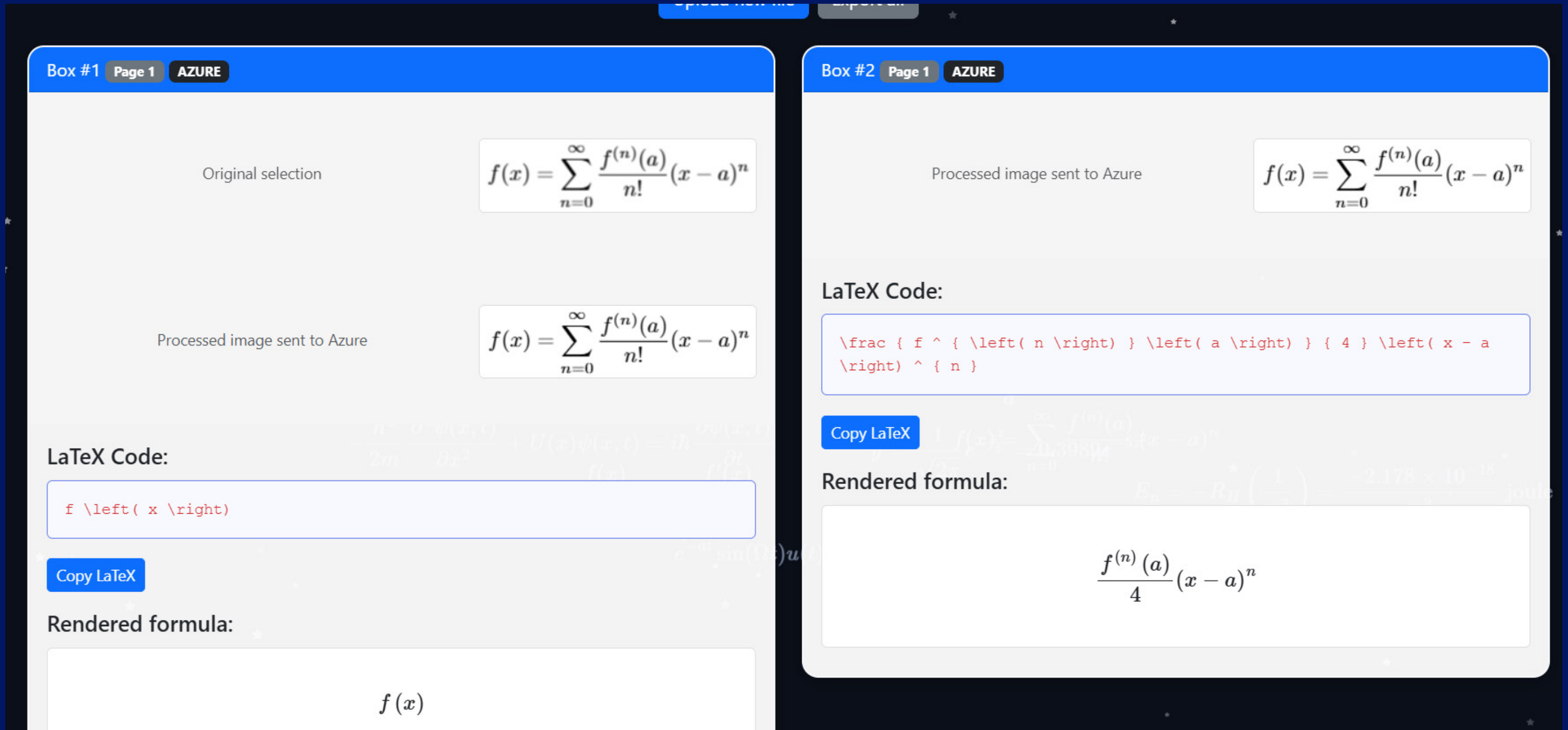
The document contains 'DocumentFontStyle.NORMAL' font style, applied to the following text:
Taylor Series,Limit of Arctangent,as,Approaches Negative Infinity,Standing-wave function,Relationship between Energy and
Page #1: 8 formula(s) detected
Page #2: 10 formula(s) detected
Taylor Series
:formula: :formula: :formula: :formula:
:formula:
:formula: :formula: :formula:
:formula: :formula: :formula:
Limit of Arctangent :formula: as :formula: Approaches Negative Infinity
:formula:
Standing-wave function
:formula: :formula:
Relationship between Energy and Principal Quantum Number
:formula:
Z-transform time domain multiplication (z domain convolution) property
:formula:
2
Taylor Series

```
f \left( x \right) = \sum _ { n = 0 } ^ { \infty } \frac { f ^ { \left( n \right) } \left( a \right) } { n ! } \left( x -
\int \sec \left( a x \right) d x = \frac { 1 } { a } \ln | \tan \left( \frac { a x } { 2 } + \frac { \pi } { 4 } \right)
e ^ { - a t } \sin \left( \Omega t \right) u \left( t \right) \Leftrightarrow \frac { \Omega } { \left( s + a \right) ^ {
e = \lim _ { n \rightarrow \infty } \left( 1 + \frac { 1 } { n } \right) ^ { n } \Gamma \left( a \right) = \int _ { 0 } ^
```

Limit of Arctangent \mathrm { X } as \mathrm { X } Approaches Negative Infinity

```
\lim _ { x \rightarrow - \infty } \tan ^ { - 1 } \left( x \right) = - \frac { \pi } { 2 }
```

Padding around the formula bounding box was necessary because Azure Document Intelligence is designed for full document analysis, not isolated formula images. When given only a cropped formula, it fails to identify the complete expression, recognizing only fragments as shown below.



In the approach where we add padding to the bounding box and fill it with the surrounding area, the formula retrived by Document Intelligence remains only be the first instance found in that given area. Therefore, a descriptive text above a formula, which contains details about the formuala's variables is retrieved as the first variable displayed and not the formula below it.

Box #1 Page 2 AZURE

Original selection

Limit of Arctangent X as X Approaches Negative Infinity

$$\lim_{x \to -\infty} \tan^{-1}(x) = -\frac{\pi}{2}$$

rocessed image sent to Azure (40px padding)

Limit of Arctangent X as X Approaches Negative Infinity

$$\lim_{x \to -\infty} \tan^{-1}(x) = -\frac{\pi}{2}$$

LaTeX Code:

```
X
```

Copy LaTeX

Rendered formula:

$$X$$

Box #2 Page 2 AZURE

rocessed image sent to Azure (40px padding)

Limit of Arctangent X as X Approaches Negative Infinity

$$\lim_{x \to -\infty} \tan^{-1}(x) = -\frac{\pi}{2}$$

LaTeX Code:

```
X
```

Copy LaTeX

Rendered formula:

$$X$$

Containerizing the app using Azure Container Apps service

latex-ocr-container-app

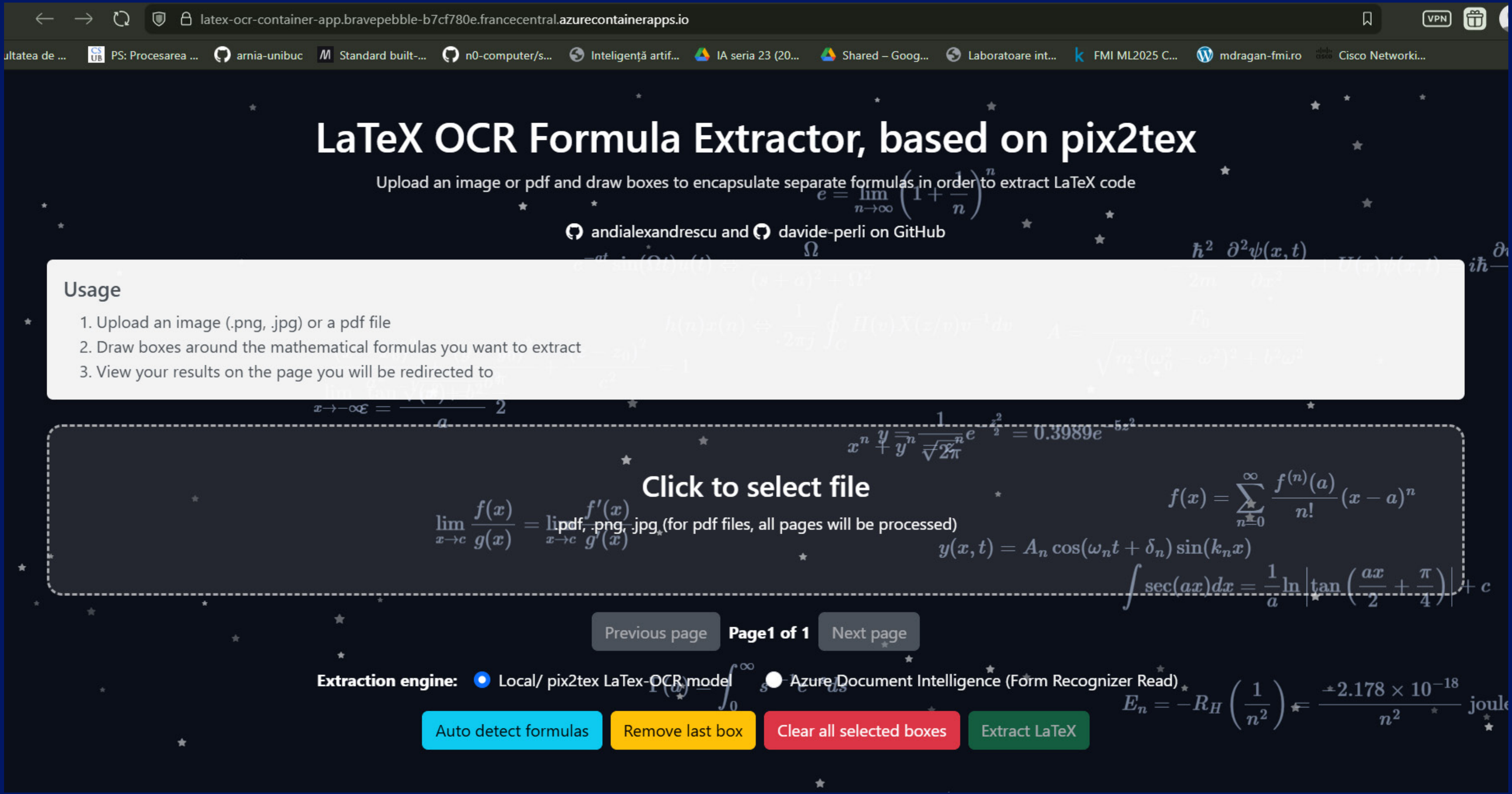
Container App

This part of the project involved both Docker and Azure Container Apps which is a serverless platform that allows you to maintain less infrastructure and save costs while running containerized applications.

```dockerfile
FROM python:3.11-slim

WORKDIR /app
# creates app inside the container

RUN apt-get update && apt-get install -y \
    poppler-utils \
    libmagic1 \
    && rm -rf /var/lib/apt/lists/*

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY app.py .
COPY formula_detector.py .
COPY index.html .
COPY results.html .
COPY floating-formulas.txt .

# .env placeholder that will be overridden by azure environment variables at runtime
RUN echo "Azure credentials will be set via environment variables" > .env

# expose port (azure container apps will use this)
EXPOSE 8000

HEALTHCHECK --interval=30s --timeout=10s --start-period=5s --retries=3 \
    CMD python -c "import urllib.request; urllib.request.urlopen('http://localhost:8000/')" || exit 1

CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

A Docker container is created containing all necessary dependencies and the pretrained pix2tex model. It's tested locally for functionality, then pushed as a public image to Azure Container Registry. Version management and future updates occur through new image pushes

````
Updating docker image + container app
```bash
 $TAG="2026-01-23-1"
docker build -t andialexandrescu/latexocr:$TAG .

docker push andialexandrescu/latexocr:$TAG

$RESOURCE_GROUP="latex-ocr"
$CONTAINER_APP_NAME="latex-ocr-container-app"
az containerapp update `
  --name $CONTAINER_APP_NAME `
  --resource-group $RESOURCE_GROUP `
  --image "andialexandrescu/latexocr:$TAG"
```
````


latex-ocr-container-app.bravepebble-b7cf780e.francecentral.azurecontainerapps.io
LaTeX OCR Formula Extractor, based on pix2tex
Upload an image or pdf and draw boxes to encapsulate separate formulas in order to extract LaTeX code
andialexandrescu and davide-perli on GitHub
Usage
1. Upload an image (.png, .jpg) or a pdf file
2. Draw boxes around the mathematical formulas you want to extract
3. View your results on the page you will be redirected to
Click to select file
.pdf, .png, .jpg (for pdf files, all pages will be processed)
Previous page
Page1 of 1
Next page
Extraction engine:
Local/ pix2tex LaTex-OCR model
Azure Document Intelligence (Form Recognizer Read)
Auto detect formulas
Remove last box
Clear all selected boxes
Extract LaTeX

A1 Taylor Series

A2
$$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n$$

A3
$$x^n+y^n=z^n$$

A4
$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}+\frac{(z-z_0)^2}{c^2}=1$$

Auto detected boxes (blue) User drawn boxes (red)

Extraction engine: Local/ pix2tex LaTex-OCR model Azure Document Intelligence (Form Recognizer Read)

Auto detect formulas | Remove last box | Clear all selected boxes | Extract LaTeX

Selected boxes regions:

Auto detected boxes (9)

Accept all | Reject all

Box A1: (270, 242) - 170×48px (confidence: 85%)

Box A2: (465, 263) - 294×85px (confidence: 85%)

Box A3: (526, 361) - 172×48px (confidence: 85%)

Box A4: (418, 421) - 390×76px (confidence: 85%)

Box A5: (524, 509) - 175×75px (confidence: 85%)

Box A6: (413, 596) - 401×74px (confidence: 85%)

Auto detected boxes (blue) User drawn boxes (red)

Extraction engine: Local/ pix2tex LaTex-OCR model Azure Document Intelligence (Form Recognizer Read)

Auto detect formulas | Remove last box | Clear all selected boxes | Extract LaTeX

Selected boxes regions:

- Box #1: (465, 263) - 294×85px Remove
- Box #2: (526, 361) - 172×48px Remove
- Box #3: (418, 421) - 390×76px Remove
- Box #4: (524, 509) - 175×75px Remove
- Box #5: (413, 596) - 401×74px Remove

Box #5 Page 1 AZURE

Original selection

$$\int \sec(ax)dx = \frac{1}{a}\ln\left|\tan\left(\frac{ax}{2}+\frac{\pi}{4}\right)\right| + c$$

Processed image sent to Azure (40px padding)

$$\int \sec(ax)dx = \frac{1}{a}\ln\left|\tan\left(\frac{ax}{2}+\frac{\pi}{4}\right)\right| + c$$

LaTeX Code:

```
\int \sec \left( a x \right) d x = \frac { 1 } { a } \ln | \tan \left(
\frac { a x } { 2 } + \frac { \pi } { 4 } \right) | + c
```

Copy LaTeX

Rendered formula:

$$\int \sec \left(a x \right) d x = \frac { 1 } { a } \ln | \tan \left(\frac { a x } { 2 } + \frac { \pi } { 4 } \right) | + c$$

Box #6 Page 1 AZURE

Original selection

$$e^{-at}\sin(\Omega t)u(t) \Leftrightarrow \frac{\Omega}{(s+a)^2+\Omega^2}$$

Processed image sent to Azure (40px padding)

$$e^{-at}\sin(\Omega t)u(t) \Leftrightarrow \frac{\Omega}{(s+a)^2+\Omega^2}$$

LaTeX Code:

```
e ^ { - a t } \sin \left( \Omega t \right) u \left( t \right)
\Leftrightarrow \frac { \Omega } { \left( s + a \right) ^ { 2 } + \Omega
^ { 2 } }
```

Copy LaTeX

Rendered formula:

$$e ^ { - a t } \sin \left(\Omega t \right) u \left(t \right) \Leftrightarrow \frac { \Omega } { \left(s + a \right) ^ { 2 } + \Omega ^ { 2 } }$$

```
LaTeX OCR Extraction Results

Total Boxes: 8
Extraction Date: 1/28/2026, 12:56:41 PM

Box #1 (Page 1):
LaTeX: f \left( x \right) = \sum _ { n = 0 } ^ { \infty } \frac { f ^ { \left( n \right) } \left( a \right) } { n ! } \left( x - a \right) ^ { n }
Region: (465, 263) - 294×85px
Box #2 (Page 1):
LaTeX: x ^ { n } + y ^ { n } = z ^ { n }
Region: (526, 361) - 172×48px
Box #3 (Page 1):
LaTeX: \frac { \left( x - x _ { 0 } \right) ^ { 2 } } { a ^ { 2 } } + \frac { \left( y - y _ { 0 } \right) ^ { 2 } } { b ^ { 2 } } + \frac { \left( z - z _ { 0 } \right) ^ { 2 } } { c ^ { 2 } } = 1
Region: (418, 421) - 390×76px
Box #4 (Page 1):
LaTeX: \varepsilon = \frac { \sqrt { a ^ { 2 } + b ^ { 2 } } } { a }
Region: (524, 509) - 175×75px
Box #5 (Page 1):
LaTeX: \int \sec \left( a x \right) d x = \frac { 1 } { a } \ln | \tan \left( \frac { a x } { 2 } + \frac { \pi } { 4 } \right) | + c
Region: (413, 596) - 401×74px
Box #6 (Page 1):
LaTeX: e ^ { - a t } \sin \left( \Omega t \right) u \left( t \right) \Leftrightarrow \frac { \Omega } { \left( s + a \right) ^ { 2 } + \Omega ^ { 2 } }
Region: (440, 682) - 344×76px
Box #7 (Page 1):
LaTeX: - \frac { \hbar ^ { 2 } } { 2 m } \frac { \partial ^ { 2 } \psi \left( x , t \right) } { \partial x ^ { 2 } } + U \left( x \right) \psi \left( x , t \right) = i \hbar \frac { \partial \psi \left( x , t \right) } { \partial t }
Region: (399, 770) - 426×76px
Box #8 (Page 1):
LaTeX: y = \frac { 1 } { \sqrt { 2 \pi } } e ^ { - \frac { x ^ { 2 } } { 2 } } = . 3 9 8 9 e ^ { - 5 z ^ { 2 } }
Region: (463, 857) - 297×75px
```

$$h(n)x(n) \Leftrightarrow \frac{1}{2\pi j}\oint_C H(v)X(z/v)v^{-1}dv$$

$$\lim_{x\to c}\frac{f(x)}{g(x)} = \lim_{x\to c}\frac{f'(x)}{g'(x)}$$

$$E_n = -R_H\left(\frac{1}{n^2}\right) = \frac{-2.178\times 10^{-18}}{n^2}\text{ joule}$$

$$y = \frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}} = 0.3989e^{-5z^2}$$

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} + \frac{(z-z_0)^2}{c^2} = 1$$

$$f(x) = \sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n$$

$$\int \sec(ax)\,dx = \frac{1}{a}\ln\left|\tan\left(\frac{ax}{2}+\frac{\pi}{4}\right)\right|$$

$$e^{-at}\sin(\Omega t)u(t) \Leftrightarrow \frac{\Omega}{(s+a)^2+\Omega^2}$$

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + U(x)\psi(x,t) = i\hbar\frac{\partial\psi(x,t)}{\partial t}$$

$$\varepsilon = \frac{\sqrt{a^2+b^2}}{a}$$

$$y(x,t) = A_n\cos(\omega_n t+\delta_n)\sin(k_n x)$$

$$\lim_{x\to-\infty}\tan^{-1}(x) = -\frac{\pi}{2}$$

$$A = \frac{F_0}{\sqrt{m^2(\omega_0^2-\omega^2)^2+b^2\omega^2}}$$

$$x^n + y^n = z^n$$

$$\Gamma(a) = \int_0^\infty s^{a-1}e^{-s}ds$$

$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$$