AI-Driven Accessibility Checker for UI Screenshots

Roja Shuruthika Kathiravan (G39829724)

M.S in Computer Science

A Report submitted to

Prof. Armin Mehrabian

George Washington University

School of Engineering and Applied Science

CSCI 6364-Machine Learning

05-11-2025

CONTENT

ABSTRACT 3

1. INTRODUCTION 3
2. DATASET 3
3. METHODOLOGY 4
   1. DATA PROCESSING 4
   2. RULE-BASED EVALUATION 4
   3. MACHINE LEARNING MODEL 4
4. RESULTS 5
5. BIAS-VARIANCE TRADEOFF 5
6. CHALLENGES 5
7. LEARNINGS 6
8. ADVANTAGES AND DISADVANTAGES 6
9. CONCLUSION 6
10. REFERENCE 6
11. APPENDIX 7

**ABSTRACT**

This project aimed to build an AI-powered accessibility checker for user interface (UI) screenshots, aimed at evaluating and enhancing digital accessibility based on WCAG 2.1 guidelines. The tool combines rule-based analysis with a machine learning model trained on a manually labeled dataset. The machine learning component was fully implemented from scratch using first principles and the NumPy and Pandas libraries, with no external ML libraries. A user-friendly web application was developed using Streamlit to provide real-time accessibility evaluation and downloadable reports.

1. **INTRODUCTION**

Digital accessibility is essential for ensuring equal access to information and services for all users, including those with disabilities. Manual evaluation of accessibility remains tedious and error-prone. This project introduces a semi-automated AI-driven approach to evaluate the accessibility of UI screenshots.

The initial project design included the option of either URL input or image upload for accessibility evaluation. Due to technical and security limitations related to URL parsing, the focus was shifted entirely to image-based evaluation using UI screenshots.

1. **DATASET**

A handmade dataset was constructed by manually labeling UI screenshots to train the machine learning model, Figma API for real-world design file extraction, Synthetic datasets created for accessibility pattern learning.

Screenshots were analyzed and scored based on the following WCAG-derived attributes:

* Contrast ratio
* Color dependency for information
* Text resize capability
* Presence of alternative text (alt text) for images
* Overall accessibility score out of 100

Features were extracted and paired with manually assigned labels to build the dataset for supervised learning

1. **METHODOLOGY**
   1. **DATA PROCESSING**

All screenshots were processed using image analysis techniques. Features such as edge density, grayscale contrast, presence of alt text, and color usage patterns were extracted. A total of 125 images were used. The dataset was split into 60% for training, 20% for validation, and 20% for testing.

* 1. **RULE-BASED EVALUATION**

Parallel to the machine learning model, a rule-based module was developed which applied four core accessibility checks:

1. **Contrast Ratio:** Minimum contrast ratio of 4.5:1 between foreground text and background.
2. **Use of Color:** Verifying that color is not the only method of conveying information.
3. **Text Resize:** Ensuring text remains readable when resized up to 200%.
4. **Alt Text Presence:** Verifying presence of alternative descriptions for non-text content.
   1. **MACHINE LEARNING MODEL**

A linear regression model was implemented from scratch using NumPy. No ML packages such as sklearn or tensorflow were used. Gradient descent optimization was applied over 1000 iterations with learning rate adjustment. Both training and test loss were tracked to assess generalization.

The custom function predict\_ml\_score() was created to:

* Accept an uploaded screenshot
* Extract feature vectors
* Predict an accessibility score using the pre-trained model

This machine learning evaluation ran alongside rule-based scoring within the Streamlit web application.

1. **RESULTS**

A dataset of **125 screenshots** was created and split into:

* **60% training set** (75 samples)
* **20% validation set** (25 samples)
* **20% testing set** (25 samples)

The machine learning model was trained using the training set, hyperparameter tuning was guided using the validation set, and final performance was measured on the testing set.

Model performance on the unseen test set:

| **Metric** | **Value** |
| --- | --- |
| MSE | 0.0103 |
| RMSE | 0.1014 |
| R² Score | -0.955 |

Both the training and testing loss curves were plotted and showed minimal variation, indicating underfitting.

1. **BIAS-VARIANCE TRADEOFF**

The model exhibited high bias and low variance. Both training and test loss remained stable but did not reduce significantly, confirming underfitting. This was primarily due to the small dataset size and the limited expressive capacity of the linear regression model. More complex algorithms such as decision trees or neural networks combined with larger datasets are recommended for future work.

1. **CHALLENGES**

* **Dataset Size:** Manual labeling of screenshots limited dataset scale.
* **Data Collection:** Ensuring screenshots met quality and dimension requirements.
* **Image Classification:** Separating webpage screenshots from arbitrary images using edge density and top-bar detection heuristics.
* **User Experience:** Fine-tuning validation checks to minimize false negatives in the UI image filter.

All these were overcome through iteration, strict dataset curation, and the development of custom filtering logic.

1. **LEARNINGS**

The project provided hands-on experience with:

* Data collection and dataset design
* Image feature extraction and preprocessing
* Implementing a regression model from scratch
* Building a hybrid evaluation system (rule-based + ML)
* Streamlit application development and real-time image processing

1. **ADVANTAGES AND DISAVANTAGES**

The accessibility checker offers a hybrid approach by combining rule-based checks with ML prediction for flexibility and provides real-time analysis through a user-friendly Streamlit interface.

However, the project was limited by a small dataset size of 125 samples, which led to underfitting, and the use of a simple linear regression model, which lacked the ability to capture complex non-linear patterns in UI screenshots.

1. **CONCLUSIONS**

The final product is a fully functional AI-driven accessibility checker that evaluates UI screenshots for accessibility compliance. Although the ML model underperformed compared to expectations, the system demonstrates the potential for hybrid approaches combining deterministic checks with data-driven predictions.

Future improvements include:

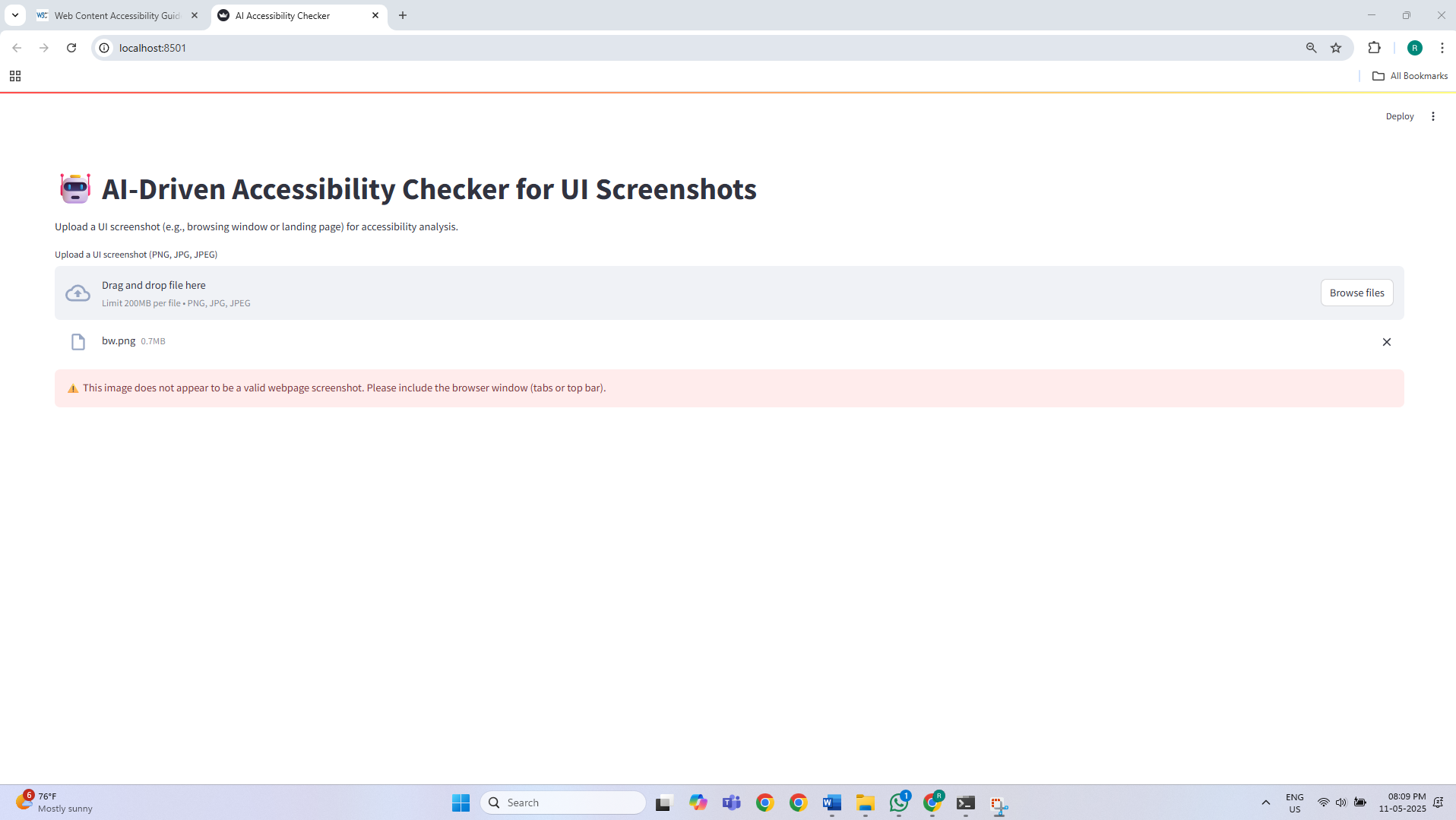
* Expanding the dataset
* Transitioning to more sophisticated ML models (e.g., decision trees, ensemble methods)
* Enhancing feature extraction for more detailed accessibility metrics.

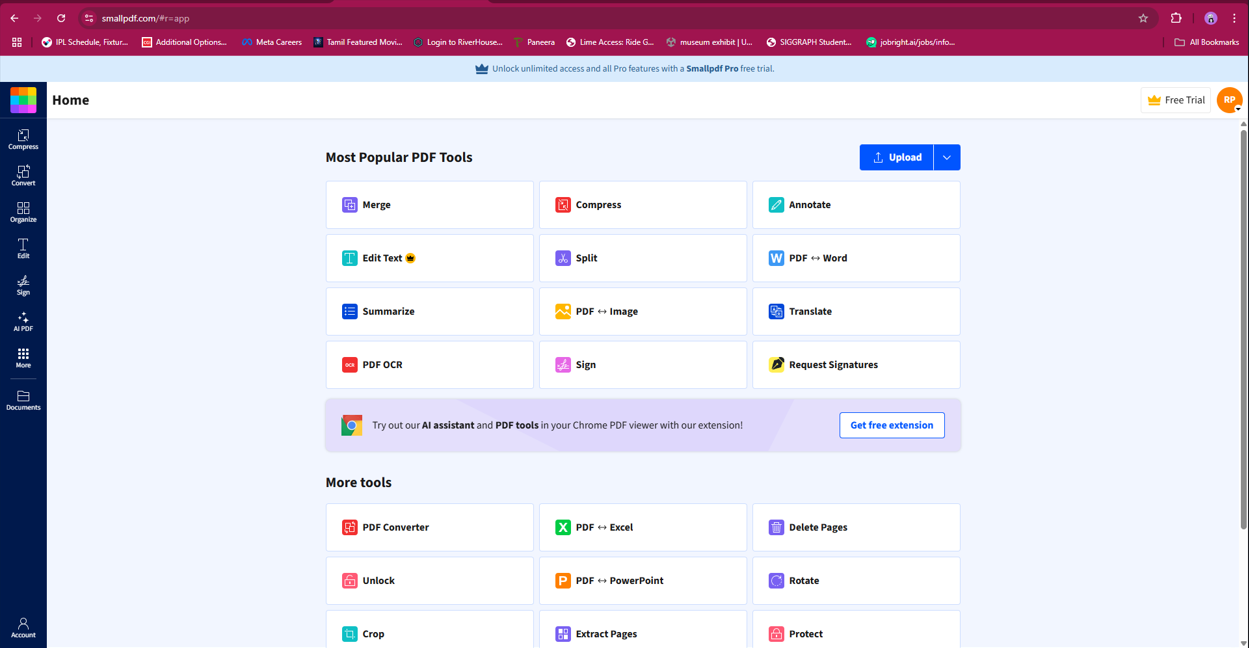
1. **REFERENCES**

WCAG 2.1 Guidelines - <https://www.w3.org/TR/WCAG21/>

1. **APPENDIX**

I/P 1:

O/P: 

I/P 2:

O/P: