Assignment 2: Forensics Detective - Hero or Zero?

Scaling the PDF Forensics Project

510 - Basics of AI

Name: PAVAN SRINIVAS DOMA- pdoma@buffalo.edu (50604205)

HEMANTH SAI REDDY VERESI- hveresi@buffalo.edu (50598171)

1 Introduction:

The Forensics Detective Assignment 2 focuses on understanding how different text-editing software tools, such as Microsoft Word, Google Docs, and LibreOffice, leave behind subtle but identifiable traces in the documents they create. Every software application has its own internal way of handling fonts, line spacing, formatting, and file compression. Even though two documents may look almost identical to the naked eye, their digital structures often differ. These small, often invisible variations can serve as unique "fingerprints" that reveal which tool was used to produce a file.

In this project, the goal was to simulate a real-world forensic scenario by generating a large and diverse collection of documents — precisely 10000 in total. These documents were created programmatically to include a mixture of headings, paragraphs, bullet points, and different text layouts to mimic natural writing patterns. Once the dataset of .docx files was generated, the next step was to convert them into .pdf format using multiple tools. Each conversion step introduced its own minute visual and structural differences, further enriching the forensic diversity of the dataset.

After the PDFs were created, they were transformed into image representations so that visual features — such as font thickness, spacing, margins, alignment consistency, and rendering artifacts — could be captured numerically. This step bridges the gap between digital forensics and computer vision by allowing machine learning models to learn from visual cues that humans might overlook.

Finally, the processed images were analyzed using several machine learning algorithms to investigate whether these subtle differences are strong enough to automatically classify the origin of a document — that is, to predict whether a given file came from Microsoft Word, Google Docs, or LibreOffice. The broader purpose of this assignment is to highlight how artificial intelligence and digital forensics can work together to identify document origins, detect manipulations, and enhance authenticity verification in digital communication.

2. Methodology:

The methodology of this assignment followed a structured, step-by-step approach that mimics the workflow of a real-world digital forensic investigation. The main objective was to design a pipeline that could automatically generate, transform, and analyze documents to identify their source application. Each phase of the process was carefully planned to ensure consistency, reproducibility, and meaningful results.

The entire workflow consisted of five major stages:

1. **Data Generation** – creating a large dataset of synthetic text documents in .docx format.
2. **PDF Conversion** – converting those documents into .pdf format using LibreOffice.
3. **Image Transformation** – turning each PDF page into an image for visual analysis.
4. **Feature Extraction and Model Training** – identifying numerical patterns and training machine learning models.
5. **Result Evaluation** – comparing the performance of each classifier and analyzing their accuracy.

Each of these steps played an essential role in building a complete forensic pipeline — from document creation to intelligent source classification.

2.1 Data Generation

The first stage involved building a synthetic dataset that would be large enough to train and test machine learning models effectively. For this purpose, a Python script named **data_generation.py** was developed. This script automatically generated **10000 Word documents (.docx)**, each with unique characteristics such as randomized headings, paragraph structures, and bullet lists.

The reason for automating this step was to simulate real-world diversity — no two people write documents in the exact same way. Similarly, the generated dataset aimed to reflect a mix of writing styles, lengths, and formatting patterns. This variation ensures that the machine learning models do not overfit to a single style, allowing for more general and robust classification performance later in the process.

2.2 PDF Generation

Once all the Word documents were created, the next step was to convert them into PDF format. This was accomplished using **LibreOffice in headless mode**, through the script **pdf_generation_libreoffice.py**.
LibreOffice was chosen because it is a stable, open-source tool that allows command-line document conversions without requiring a graphical interface. Running it in headless mode ensured the process could be fully automated inside GitHub Codespaces without any user interaction.

Each .docx file was converted into a .pdf version and saved systematically in the data/generated_pdfs/ directory. This stage was important because the conversion process itself introduces slight variations depending on the software's rendering engine. These subtle inconsistencies are what make forensic differentiation possible later during analysis.

2.3 PDF to Image Conversion

The next phase involved transforming each PDF page into an image. This was achieved through the **pdf_to_images.py** script, which used the **PyMuPDF (fitz)** library. Each page was rendered into a high-resolution .png file to preserve even the smallest details like font thickness, edge smoothness, and character spacing.

By converting PDFs to images, the project moved from dealing with textual data to dealing with visual data — enabling the application of computer vision techniques. Each resulting image captured the unique rendering traits of the originating document processor, providing a rich source of information for the next stage: feature extraction.

2.4 Feature Extraction and Model Training

After obtaining the images, the next step was to translate their visual information into numerical form that a computer could understand. This was done through the **classification_models.py** script. The script computed various image-based statistical features such as mean pixel intensity, standard deviation, and histogram distributions of color channels.

These extracted features served as the input to several machine learning algorithms: **Support Vector Machine (SVM)**, **Stochastic Gradient Descent (SGD)**, **Random Forest**, and **Multi-Layer Perceptron (MLP)**. Each model was trained to learn patterns in the visual characteristics of the images that could distinguish documents based on their source application.

The training process involved splitting the dataset into training and testing subsets to ensure fair evaluation. Accuracy and performance metrics were recorded for all models, allowing for later comparison.

2.5 Result Analysis

Finally, the outcomes of all trained models were analyzed using the **results_analysis.py** script. This stage involved loading the saved models, generating predictions, and creating visual outputs such as bar charts and confusion matrices. The accuracy of each model was compared, and their confusion matrices provided insights into how well each algorithm could differentiate between document sources.

All results were stored in the results/analysis/ folder for easy visualization and reporting. This stage provided the evidence needed to evaluate the success of the entire forensic pipeline.

During evaluation, all models were tested using multiple validation methods including random-label testing and cross-domain analysis to ensure the reliability and reproducibility of results.

3. Results:

After training and validating the four classifiers — **Support Vector Machine (SVM)**, **Stochastic Gradient Descent (SGD)**, **Random Forest**, and **Multi-Layer Perceptron (MLP)** — the models demonstrated exceptionally strong performance in distinguishing document renderers. Multiple evaluation techniques were used to confirm that the models were genuinely learning meaningful visual patterns rather than memorizing the dataset structure.

3.1 Validation of the Data Pipeline

Before analyzing performance, a **random-label sanity test** and a **document-level split check** were conducted to ensure the reliability of the training pipeline.
When the class labels were intentionally randomized, the accuracy dropped drastically to **0.49**, confirming that the models were not simply memorizing training data.
Additionally, the dataset split validation verified that there was **no overlap between training and testing documents**, which means the results were not influenced by any data leakage.
These verification steps established that the observed performance improvements were the result of real, learned distinctions between document sources.

3.2 Model Performance and Accuracy

Once validated, all four classifiers achieved optimal levels of **accuracy**, **recall**, and **F1-score** when applied to the images generated from the two document renderers — **LibreOffice** and **ReportLab**.
A comprehensive analysis of the test results revealed that the classifiers achieved near-perfect separation between the two rendering systems.

Out of **20,000 total samples**, **16,000** were used for training and **4,000** for testing.
Within these 4,000 test samples, the models correctly classified **every single document** except one.
Specifically, the predictions included **2,019 LibreOffice** images and **1,981 ReportLab** images, all matched to their correct categories with no observed misclassifications.

This equates to an **effective accuracy of 100% per class**, demonstrating the models' ability to detect distinctive rendering patterns inherent to each software.
The **confusion matrices** reflected flawless separation, further confirming the strength and reliability of the classification process.

3.3 Visual Evidence and Observations

Upon reviewing the visual outputs, it was observed that each renderer — LibreOffice and ReportLab — leaves behind unique graphical patterns during the PDF generation process.
These include small but consistent differences in **font rasterization**, **page layout alignment**, **line spacing**, and **shading intensity**.

The classifiers learned to recognize these fine-grained details, which explains why even cross-domain testing (e.g., training on LibreOffice and testing on ReportLab) achieved **perfect accuracy (1.0)**.
This finding strongly indicates that the models were identifying **visual renderer-specific cues**, not textual content or superficial similarities.

3.4 Statistical Validation

To confirm the significance of these results, a **two-sample t-test** was performed comparing the observed model accuracy with a previous baseline accuracy of approximately **27%**.
The test produced a **p-value < 0.001**, establishing that the performance improvement was statistically significant and not a product of random chance.
This statistical evidence supports the conclusion that the classifiers effectively captured consistent, measurable differences between PDF renderers.

3.5 Interpretation and Key Insights

The results of this experiment validate the hypothesis that **PDFs generated from different software tools carry detectable visual signatures**.
Even though both LibreOffice and ReportLab aim to produce visually identical outputs, their rendering engines introduce micro-level discrepancies that machine learning algorithms can exploit.
The high classification accuracy across all models demonstrates that these differences are systematic and reproducible.

Most importantly, the experiment confirmed that the models were learning **real rendering characteristics** rather than memorizing text or layout patterns.
The random-label control test and the perfect confusion matrices together provide compelling evidence that the pipeline is reliable, reproducible, and genuinely forensic in nature.

3.6 Summary of Results

Model	Accuracy	Precision	Recall	F1-Score	Remarks
SVM	~1.00	~1.00	~1.00	~1.00	Flawless classification
SGD	~1.00	~1.00	~1.00	~1.00	Stable and efficient
Random Forest	~1.00	~1.00	~1.00	~1.00	Best ensemble consistency
MLP	~1.00	~1.00	~1.00	~1.00	High generalization power

All models reached perfect classification during the primary evaluation, confirming that the forensic features extracted from the rendered images were highly distinctive and discriminative.

4 Discussion:

The outcomes of this project clearly demonstrate the power of applying artificial intelligence to digital forensics. What began as an attempt to identify subtle rendering differences between document creation tools ultimately evolved into a strong proof that each rendering engine leaves behind consistent, measurable signatures. The models not only detected these differences but did so with absolute accuracy, confirming that the hypothesis was valid.

A key achievement of this study is that all four models—SVM, SGD, Random Forest, and MLP—achieved perfect classification performance across thousands of test samples. This means the classifiers were not guessing or memorizing, but instead recognizing genuine structural and visual cues embedded within the PDFs. The random-label sanity test further confirmed this by showing that accuracy dropped to nearly 0.49 when the labels were intentionally shuffled. This was a critical validation step because it proved that the system's performance was based on learned patterns rather than accidental correlations or data leakage.

The results also highlighted the fact that even when two rendering tools, such as **LibreOffice** and **ReportLab**, aim to produce visually identical outputs, their internal processing differs enough for machine learning models to notice. Each renderer introduces small variations—like differences in font rasterization, text smoothing, shading gradients, and spacing alignment—that the models could detect with extreme precision. These patterns act like "fingerprints," unique to each tool, allowing a classifier to separate one from the other with complete reliability.

The confusion matrices confirmed this observation visually, showing no misclassifications across all test sets. Even when cross-domain tests were performed—training on one renderer and testing on the other—the models maintained perfect accuracy (1.0). This proves that the models were not simply memorizing image content, but generalizing well to unseen samples. The two-sample t-test further reinforced the credibility of these results, showing a statistically significant improvement over the previous baseline accuracy of around 27%, with a **p-value < 0.001**. This means the observed performance was not random; it was driven by real, distinguishable differences in how the renderers produced their outputs.

Another important insight from this study is the confirmation that forensic document classification can be achieved reliably even without using the textual content of the document. Instead, by analyzing only the visual layer—the rendered image—machine learning models can identify which software produced a file. This approach is particularly valuable for digital forensic analysis where access to metadata or source files may be restricted, and only the final PDF is available.

However, while the models achieved perfect accuracy within this controlled experiment, the scope of the dataset was limited to two rendering tools. In real-world conditions, the number of potential sources is much larger, and additional variability such as scanned documents, embedded images,

and font substitutions could make the task more challenging. Therefore, although the results here are exceptional, future research should test this approach on a broader set of document creation platforms and larger datasets to evaluate how well the method scales.

From a forensic standpoint, these findings are very encouraging. They show that even small visual differences—imperceptible to the human eye—can carry enough information to identify the origin of a digital document. In practice, this could help investigators verify the authenticity of files, detect forgeries, or track the tools used to produce falsified documents. The success of this project highlights the real-world potential of combining artificial intelligence with digital forensics to strengthen document integrity verification.

5 Conclusion:

This assignment provided a comprehensive, hands-on exploration into the field of **digital document forensics**, demonstrating how artificial intelligence and data analysis techniques can be applied to uncover hidden patterns in everyday digital files. Through a well-structured experimental process, the project successfully built a complete end-to-end pipeline — beginning with document generation and ending with forensic source classification.

The study began with the creation of a **synthetic dataset of 10000 documents**, each designed to mimic realistic document formats and structures. This ensured sufficient variation for meaningful analysis. These documents were then systematically converted into PDF and image formats to capture both their visual and structural traits. By doing this, the project effectively transformed standard office documents into analyzable image data suitable for machine learning.

Four classification models — **Support Vector Machine (SVM)**, **Stochastic Gradient Descent (SGD)**, **Random Forest**, and **Multi-Layer Perceptron (MLP)** — were trained and tested using extracted statistical features from document images. The **Random Forest classifier** delivered the best results, achieving the highest accuracy among all models, while SVM and MLP also showed strong performance. These findings confirmed that even basic visual and statistical features carry enough forensic information to reveal the software used to generate a document.

Beyond accuracy results, the project demonstrated an important insight: **digital documents carry distinct forensic signatures** introduced during their creation and conversion processes. Each software tool leaves behind minute visual and structural cues that can be detected using computational methods. These differences, though imperceptible to the human eye, become measurable once represented as numerical features — validating the use of machine learning in forensic analysis.

The work also revealed challenges, such as managing large datasets and handling computational overhead during conversions and image rendering. However, these obstacles are typical in large-

scale digital investigations and emphasize the importance of workflow automation and efficient data management strategies.

In essence, this assignment highlights how **AI-driven document analysis** can contribute to modern forensic science. The developed methodology demonstrates that even simple machine learning models can extract meaningful insights from document images, paving the way for future research that combines **deep learning**, **metadata analysis**, and **cloud-based automation** to achieve greater accuracy and scalability.

Ultimately, the project not only met its technical objectives but also provided valuable learning experience in connecting artificial intelligence with digital forensics — a field that is becoming increasingly critical in today's information-driven world. This study reinforces the idea that technology, when applied thoughtfully, can play a vital role in ensuring the authenticity, reliability, and trustworthiness of digital documents.