Technical Report: Tamper Detection in Academic Credentials

Role: AI Development Intern
Name: Pratyush Kaushal
Submission Date: May 2025

1. Overview

Many fake academic documents are being created today. People change dates, names, grades, or even seals using editing tools. This project aims to build a simple system that can automatically check and detect if a document (like a degree certificate or transcript) has been tampered with.

2. Goal of the Project

To create a basic prototype that can:

- Check PDF **metadata** (like creation and modification dates).
- Use **image comparison** to detect layout changes.
- Use **OCR** (Optical Character Recognition) to read text and find differences.
- Flag anything that looks suspicious or inconsistent.

3. What Types of Tampering We Detect

The system is designed to find the following:

- **Metadata Changes**: If someone edits the author name, creation date, or adds a fake title.
- **Fake Updates**: If the document shows it was modified long after it was created, or has a date in the future.
- **Visual Layout Changes**: If someone edits the look of the certificate (e.g., changes the university logo or adds fake seals).
- **Text Differences**: If the name, grades, or dates are changed, OCR can help catch those.

4. How the System Works

a. Checking Metadata

- The code uses PyPDF2 to read PDF file metadata.
- It looks for important fields like /Author, /CreationDate, /ModDate.
- If any important field is missing or has a strange date (e.g., future year), it gives a warning.

b. Using OCR to Read Text

- pytesseract (Tesseract OCR tool) reads text from scanned PDF pages or images.
- The text is then compared with an original or expected version using Python's difflib.
- If names or grades don't match, the system points them out.

c. Layout Comparison

- OpenCV is used to visually compare the layout of documents.
- It can help detect changes like altered logos, seals, or signature sections.

5. Challenges Faced

- **Poor Scanned Quality**: OCR sometimes struggles to read low-quality or blurry images.
- **No Real Forged Dataset**: We used mock documents, which may not reflect real-world cases.
- **Time Zone Issues**: PDF dates may be stored in UTC, causing confusion.
- **Template Dependence**: Layout comparison works best when we already have a clean reference document.

6. How It Can Be Improved

- Add more sample documents to train the system on real tampering cases.
- Use AI models to learn from tampered vs original documents.
- Build a web interface to make the tool easier for others to use.
- Work with educational institutions to include a signature or digital hash for verification.

7. Conclusion

This project shows that it is possible to detect tampering in academic documents using simple tools. By checking metadata, layout, and text content, the system can raise red flags on suspicious files. With more data and feedback, this can be turned into a useful tool for schools and employers.