

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Quantifying Copyright Infringement in AI-Generated Images

Team Members:

Gabriel Arrano
Alejandro Adorno
Uriel Juarez
Gabriel Igliozi

Product Owner(s):

Wenbin Zhang

Mentor(s):

Wenbin Zhang
Jocely Dzuong

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document presents a comprehensive overview of the "Quantifying Copyright Infringement in AI-Generated Images" project. Our team has developed a software tool that uses advanced image analysis techniques to determine the similarity between AI-generated images and copyrighted works. This tool aims to assist legal professionals, content creators, and other stakeholders in effectively managing copyright issues related to digital content generated by artificial intelligence.

Table of Contents

INTRODUCTION	5
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM	5
USER STORIES	
IMPLEMENTED USER STORIES	7
PENDING USER STORIES	10
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	12
SPRINTS PLAN	
.....	13
<i>Sprint 1</i>	
.....	13
<i>Sprint 2</i>	
.....	13
<i>Sprint 3</i>	
.....	14
<i>Sprint 4</i>	
.....	15
<i>Sprint 5</i>	
.....	16
<i>Sprint 6</i>	
.....	17
<i>Sprint 7</i>	
.....	18
SYSTEM DESIGN	

ARCHITECTURAL PATTERNS	20
SYSTEM AND SUBSYSTEM DECOMPOSITION	21
DEPLOYMENT DIAGRAM	22
DESIGN PATTERNS	22
...	22
SYSTEM VALIDATION	23
GLOSSARY	37
APPENDIX	38
APPENDIX A - UML DIAGRAMS	38
<i>Static UML Diagrams</i>	38
<i>Dynamic UML Diagrams</i>	40
APPENDIX B - USER INTERFACE DESIGN	52
APPENDIX C - SPRINT REVIEW REPORTS	69
APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	74
REFERENCES	80

INTRODUCTION

The rise of artificial intelligence in digital content creation poses new challenges for copyright enforcement. Traditional methods of copyright analysis are often ill-suited to address the nuances of AI-generated images. Our project addresses this gap by introducing a robust, reliable tool designed to quantify the similarity between images, aiding stakeholders in making informed decisions regarding potential copyright infringements.

Current System

The current system consists of traditional copyright enforcement methods, which primarily rely on human analysis and lack the capability to handle the vast amount of digital content produced by AI. These methods are time-consuming, subjective, and often fail to keep pace with the rapid advancements in AI technology.

Purpose of New System

The new system introduces a technological solution to enhance the efficiency and accuracy of copyright analysis. By automating the similarity assessment process, our tool reduces the reliance on manual labor and introduces a standardized approach to copyright infringement detection. This not only speeds up the process but also provides a scalable solution as the volume of AI-generated content continues to grow.

USER STORIES

- * User Story #5: Document changes and developments in international copyright laws on AI
- * User Story #6: Document Case Studies and Real World Examples
- * User Story #7: Test Generative Pre-trained Transformer using Copyright Prompts
- * User Story #8: Review github repo sent by Product Owner
- * User Story #9: Think through how to adjust repo program into our own project
- * User Story #10: Prepare follow up presentation for Product Owner
- * User Story #11: Install necessary dependencies and prepare the environment for running the code.
- * User Story #12: Execute the code from the paper in the prepared environment.
- * User Story #13: Analyze and document the execution results, comparing them with the paper's claims.
- * User Story #14: Update the project repository with detailed findings and analysis.
- * User Story #15: Develop criteria for benchmarking AI copyright watermarking methods.
- * User Story #16: Compile a benchmarking dataset based on execution results.
- * User Story #17: Draft a survey paper that outlines the state of the art, based on literature review and practical findings.
- * User Story #18: Plan future research directions based on gaps identified in the survey.
- *
- * User Story #20: Create Slide presentation for our detailed findings on research papers
- * User Story #21: Create Google Doc for our summarized findings on research papers
- * User Story #22: Run the Fairness benchmark tests
- * User Story #23: Run the Privacy benchmark tests
- * User Story #24: Run the Machine Ethics benchmark tests
- * User Story #25: Run the Harmfulness benchmark tests
- * User Story #26: Confirm GitHub invite from Jocelyn
- * User Story #27: Create account for OpenAI's ChatGPT to test prompts
- * User Story #28: Review prompt creation on Bing's Copilot platform
- * User Story #29: Review the image-similarity-measures Python package
- * User Story #30: Contribute to copyright infringing prompt list
- * User Story #31: Update the GitHub documentation
- * User Story #32: Test prompts on image generating LLMs
- * User Story #33: Identify the prompts which generate copyright images
- * User Story #34: Create prompt criteria
- * User Story #35: Identify quantification parameters
- * User Story #36: Create flow for reverse image search

- * User Story #37: Create project infrastructure in Python
- * User Story #38: Start developing project in Python
- * User Story #39: Fix a couple bugs in our program related to saving web scrapped images
- * User Story #40: Create a better user friendly UI to make it easier to present
- * User Story #41: Add a more intuitive picture comparison that highlights similarities on both pictures.
- * User Story #42: Look into other APIs that don't have a daily usage limit for reverse image searching.
- * User Story #43: Create a summarized report of our existing Sprint Planning Meeting Minutes and Sprint Review Meeting Minutes.
- * User Story #44: Incorporate boundary boxes to compare attributes in output.
- * User Story #45: Start planning our showcase poster.
- * User Story #46: Implement FSIM image similarity measure.
- * User Story #47: Apply fix to ISSM image similarity measure.
- * User Story #48: Re-implement image saving functionality after testing.
- * User Story #49: Review the quantification comparison paper provided by Product Owner
- * User Story #50: Define copyright infringement criteria
- * User Story #51: Refine image similarity function to more accurately draw comparison vectors
- * User Story #52: Create individual posters for project recap
- * User Story #53: Look into free alternative reverse image search API
- * User Story #54: Explain the image similarity measures
- * User Story #55: Explain the threshold measurements
- * User Story #56: Explain logic behind OpenCV drawing match function
- * User Story #57: Create final report for Product Owner
- * User Story #58: Design Individual Poster
- * User Story #59: Prepare Group Information for Presentation
- * User Story #60: Produce Introduction Video
- * User Story #61: Produce Demo Video
- * User Story #62: Prepare for Showcase Presentation
- * User Story #63: Compile Final Deliverable Code
- * User Story #64: Document Daily Scrum Meetings
- * User Story #65: Create Installation Guide Video
- * User Story #66: Record Video on Software Use
- * User Story #67: Develop Video on Maintaining and Revising the Software
- * User Story #68: Create Video Highlighting Software Shortcomings
- * User Story #69: Create Presentation Slides

Implemented User Stories

All of the above.

Pending User Stories

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware:

To manage and process the extensive datasets involved in image similarity analysis, our project utilizes high-performance servers. These servers are equipped with powerful CPUs and GPUs to handle the computationally intensive tasks of image processing and machine learning model execution efficiently. The use of robust servers ensures that the system can manage high workloads and perform complex computations in real-time, which is critical for delivering accurate results swiftly.

Software Resources:

Python: At the core of our project is Python, a programming language known for its versatility and robustness, making it an ideal choice for complex computational tasks. Python's extensive library ecosystem allows for effective handling of image data and mathematical operations, which are essential for our image analysis.

image-similarity-measures Library: This Python library is crucial for our project as it provides various algorithms to measure the similarity between images. It includes implementations of several metrics such as Structural Similarity Index (SSIM), Feature Similarity Index (FSIM), and more. By using this library, we can efficiently quantify similarities and differences between AI-generated images and copyrighted materials, supporting the tool's core functionality.

Bing AI Image Generator: We utilize the Bing AI Image Generator for creating comparative images in testing scenarios. This tool helps in generating images based on textual descriptions, which allows us to produce a diverse set of images for validating the effectiveness of our image similarity metrics.

Jupyter Notebook: For interactive development and testing, we use Jupyter Notebook. It provides an excellent environment for experimenting with data, prototyping image processing algorithms, and visualizing the effects of various image transformations. Jupyter Notebooks are instrumental in documenting our exploratory analysis and presenting findings in a format that is accessible to both technical and non-technical stakeholders.

Sprints Plan

Sprint 1

- Formed team and selected project topic.
- Established meeting schedules and set up documentation.

Sprint 2

- Documented developments in copyright laws and AI.
- Began adapting a GitHub repo for our project's needs.
- Prepared a presentation on our research findings.

Sprint 3

- Analyzed a pivotal AI copyright watermarking paper.
- Faced and documented technical challenges with code execution.

Sprint 4

- Attempted to run Decoding Trust benchmarks.
- Explored solutions for technical difficulties, including seeking external support.

Sprint 5

- Initiated the development of a program to compare two pictures and quantify similarities.
- Focused on creating a solid project infrastructure and preliminary testing.

Sprint 6

- Polished general code of the program.
- Set thresholds for what would be considered copyright.
- Drafted paper required by the product owner.

Sprint 7

- Present our project for Senior Design Showcase
- Prepared and turned in Final Deliverables Folder

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

Microservices Architecture: The system is built using loosely coupled services, enhancing maintainability and scalability.

Client-Server Model: The frontend (client) interacts with the backend (server) where image processing and analysis are performed.

Model-View-Controller (MVC): Separation of concerns is achieved by dividing the application into the model (data), view (user interface), and controller (logic).

System and Subsystem Decomposition

The system is decomposed into three main components:

Image Input Module: Handles the input of images for comparison.

Analysis Engine: Utilizes image processing algorithms to compare images and quantify similarities. Based on SSIM, FSIM, RMSE, and PSNR

Output Generator: Displays the similarity score and detailed comparison results.

Deployment Diagram

Design Patterns

SYSTEM VALIDATION

GLOSSARY

APPENDIX

Appendix A - UML Diagrams

Appendix B - User Interface Design

Appendix C - Sprint Review Reports

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

REFERENCES

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2003). "Image Quality Assessment: From Error Visibility to Structural Similarity." Retrieved from <https://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>

Sankar, Vinu Sadasivan et al. (2023). "Can AI-Generated Text be Reliably Detected?" arXiv:2303.11156. Retrieved from <https://arxiv.org/pdf/2303.11156.pdf>

Kariyawasam, Kanchana. (2020). "Artificial intelligence and challenges for copyright law." International Journal of Law and Information Technology, vol. 28, no. 4, pp. 279–296. Retrieved from <https://doi.org/10.1093/ijlit/aaaa023>

Elkhatat, Ahmed M., Elsaid, Khaled, & Almeer, Saeed. (2023). "Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text." International Journal for Educational Integrity. Retrieved from <https://edintegrity.biomedcentral.com/articles/10.1007/s40979-023-00140-5>

Lancaster, Thomas. (2023). "Artificial intelligence, text generation tools and ChatGPT – does digital watermarking offer a solution?" International Journal for Educational Integrity. Retrieved from <https://edintegrity.biomedcentral.com/articles/10.1007/s40979-023-00131-6>

Cooley LLP. (2024). "Copyright Ownership of Generative AI Outputs Varies Around the World." Retrieved from <https://www.cooley.com/news/insight/2024/2024-01-29-copyright-ownership-of-generative-ai-outputs-varies-around-the-world>

Livieris, Ioannis E., Pintelas, Emmanuel, Kiriakidou, Niki, & Pintelas, Panagiotis. (2023). "Explainable Image Similarity: Integrating Siamese Networks and Grad-CAM." Journal of Imaging, vol. 9, no. 10, article 224. Retrieved from <https://www.mdpi.com/2313-433X/9/10/224>