SOFTWARE ENGINEERING PROJECT

Protractor

BY

Nantawan Paramapooti
Pichayoot Tanasinanan

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
KASETSART UNIVERSITY

Academic Year 2026

Protractor

BY

Nantawan Paramapooti
Pichayoot Tanasinanan

This Project Submitted in Partial Fulfillment of the Requirement for Bachelor Degree of Engineering (Software Engineering)
Department of Computer Engineering, Faculty of Engineering KASERTSART UNIVERSITY
Academic Year 2026

Approved By:

Advisor . . . . . **Date** . . . . . .
(Punpiti Piamsa-nga)

Co-Advisor . . . . . **Date** . . . . . .
(Prof. Chantana Chantrapornchai)

Head of Department . . . . . **Date** . . . . . .
(Punpiti Piamsa-nga)

Abstract

Currently, Generative AI has significantly advanced in producing high-quality video content, but it also introduces risks such as deepfake misuse, copyright infringement, and malicious manipulation. The Protractor project presents an AI-driven video poisoning processor designed to counteract the threats posed by Generative AI video models. By adding imperceptible perturbations to videos, Protractor ensures that while the video remains unchanged to the human eye, AI models misinterpret and degrade their outputs when trained on poisoned data.

The system leverages Breaking Temporal Consistency (BTC-UAP) and Spatially Transformed Adversarial Attacks (stAdv) to disrupt both frame-by-frame spatial details and motion-based temporal consistency, preventing AI from accurately learning patterns from poisoned videos. Additionally, the project implements adversarial noise embedding, perceptual similarity loss, and automated AI pipeline optimizations to maintain high fidelity for human viewers while corrupting AI training datasets.

The Protractor system is built for content creators, artists, and copyright holders who wish to protect their work from unauthorized AI training. Experimental results show that video poisoning significantly disrupts AI-generated outputs, making it a practical defense against AI exploitation and data misuse.

Keywords: Video Poisoning, Generative AI, Adversarial Attack, Deepfake Protection, AI Security

Acknowledgement

We would like to express our sincere gratitude to the Department of Computer Engineering, Kasetsart University for providing us with invaluable resources, technical knowledge, and support throughout the development of this project. Their guidance has been instrumental in shaping the Protractor system.

We are deeply thankful to our advisor, Assoc. Prof. Dr. Punpiti Piamsa-nga, for his expertise and mentorship in the field of image processing and adversarial attacks, which have been critical to the project's success. We also extend our heartfelt appreciation to Prof. Chantana Chantrapornchai, Ph.D., for her insightful guidance on parallel computing and optimization, enabling us to refine the efficiency of our AI pipeline.

Additionally, we would like to acknowledge the research communities and open-source contributors whose work on adversarial robustness, video processing, and AI security provided essential knowledge and tools that contributed to this project's implementation.

Finally, we would like to thank our faculty members, colleagues, and everyone who has supported and encouraged us throughout this journey. Their insights, discussions, and feedback have been invaluable in developing a meaningful and impactful solution to counter AI exploitation in video generation.

Nantawan Paramapooti
Pichayoot Tanasinanan

Table of Contents

List of Tables

Page

List of Figures

Chapter 1

Introduction

1.1 Background

Video Generative AI[0] refers to artificial intelligence models that create videos based on textual descriptions, images, or existing video inputs.

The widespread use of this AI has led to concerns about copyright infringement[1], as AI models[2] often rely on vast datasets[3] scraped from the internet[4], including copyrighted content[5], without explicit permission from original creators[6]. Many artists and content creators advocate for stricter regulations to protect their works from being used without consent.

One of the most alarming consequences of AI-generated video technology is impersonation[7], often referred to as "deepfakes[8]." AI can create realistic videos of individuals, making them appear to say or do things they never did. This poses risks in identity theft[9], misinformation[10], fraud[11], and political manipulation[12]. The ability to create hyper-realistic fake videos raises concerns about trust in digital content and calls for advanced detection methods to counteract malicious use.

Data poisoning is a method of corrupting AI models by injecting misleading or harmful data into their training sets[13], ensuring that generative models[14] cannot easily exploit original artistic content. However, it would be considered an aspect of data security[15], and restrict malicious actors from exploiting your data against your interests.

1.2 Problem Statement

Problems from the invention of Video-output Generative AI

1. Deepfake[8] 1.1. Identity theft[9] 1.2. Forgery of False Evidence of crimes 1.3. Defamation from non-consensual explicit generated video 2. Grifters 2.1. Copyright Infringement 2.2. Dead internet theory

Technical Problems

1. There is currently no existing video poisoning processor, but there is research on video poisoning tactics. 2. Frame-by-Frame poisoning with static Image poisoning processor as an alternative. 2.1. Manually poisoning frame by frame is inconvenient for production use. 2.2. Processing time scales horribly with video duration and fps. 2.3. Static Image poisoning tactics is less effective against Video generative AI.

1.3 Solution Overview

This software seeks to simplify the process of video poisoning to be as easy as a few clicks. We'd only need the user to inputing their video, set some preferences, start the process and wait for the poisoned video output in their designated folder. While being effective against generative ai and efficiently optimize hardware resources to process larger video; ranging from 5 minutes to 2 hours, to be processed fast and reliable enough for our target users such as filmmakers, content creators, and studios to incorporate this in their workflow.

1.3.1 Features

1. Video Input: Input your video to poison

2. Perturbation Settings: Set predefined Parameters such as perturbation weights or quality to set the perturbation strength and output quality. More parameters may be added depending on the available parameters of the system's poisoning methods.

3. Output folder: User can select where the output will be stored when the video poisoning process has finished.

4. Start Poisoning: Click to start the poisoning process. The process cannot be stopped while it is running until the process finishes.

5. Hardware optimization: Optimize the available hardware to minimize processing time duration. This would be done automatically but may allow users to set hardware themselves if deemed appropriate.

1.4 Target User

- **Digital Content Creators & Video Artists:** They have had their creations used as training data without their permission to replicate their work, making their creative, unique, curated work being buried amongst their AI copies that hurt their profits and fame.

- **Industry Professionals in Media & Entertainment:** Animation studios are at risk of having their creative works being exploited to create lower quality but faster animations. This could result in the death of the Animation industry altogether as Animator and other creatives being laid off after their works had been trained on AI and the audience ends up with an incoherent meaningless repetitive mediocre slop because the company thought that was good enough for the audience and artist become more distrustful of sharing their works online.

- **Anti-AI social media platforms:** Cara, BlueSky, Teezr, VGen are against AI content on their platform. This could be part of their feature to protect their userbase's video against being used to train on AI.

- **Individuals who do not want their videos to be used to train generative AI:** From the dangers of deepfakes, regular people do not want their face to be used to train generative AI in General, but data scraping was done without their consent.

1.5 Benefit

The Protractor system protects video content from unauthorized AI training by applying adversarial techniques that disrupt AI perception while remaining imperceptible to humans.

- It prevents AI from accurately processing videos, reducing the risk of deepfake generation and unauthorized replication.
- It enhances intellectual property protection for creators and safeguards the creative industry from AI-driven content theft.
- Its easy-to-use implementation allows content creators and researchers to apply AI poisoning without requiring advanced technical expertise.

1.6 Terminology

1.6.1 Background

[0]AI : artificial intelligence
[1]copyright infringement : violating copyright law over a content
[2]AI models : AI programs consisting of complex mathematical and computational techniques to process vast amounts of data and extract meaningful insights.
[3]datasets : collections of data used to train AI models.
[4]scraped from the internet : automatically collecting data from online sources, often using web crawlers or scrapers.
[5]copyrighted content : Any creative work (e.g., videos, images, music) legally protected under copyright law, requiring permission for use.
[6]original creators : The individuals or entities who produce and hold the legal rights to creative content.
[7]Impersonation : The act of fraudulently imitating a person, often using AI-generated media, to deceive others.
[8]deepfakes : AI-generated videos that convincingly replace a person's likeness or voice with another, often for deceptive purposes.
[9]identity theft : The unauthorized use of someone's personal information to commit fraud or other crimes.

[10]misinformation : False or misleading information spread unintentionally or deliberately, often amplified by AI-generated content.
[11]fraud : Deceptive actions intended to achieve financial or personal gain, sometimes involving AI-generated media.
[12]political manipulation : The use of deceptive tactics, such as deepfakes or AI-generated propaganda, to influence public opinion or elections.

Deepfake

Grifters

Dead internet theory genuine human interaction is overtaken by AI slop

Data Poisoning Data Security datasets scraped

Adversarial Attack: A method to manipulate AI models by adding small, unnoticeable changes that cause errors.

Breaking Temporal Consistency (BTC-UAP): A technique that disrupts AI's ability to track motion across video frames.

Spatially Transformed Adversarial Attack (stAdv): Alters video structure to mislead AI while keeping it unchanged for humans.

Deepfake: AI-generated fake media that manipulates video content realistically.

AI Poisoning: Modifying data to mislead AI models and prevent accurate learning.

Perceptual Similarity Metrics (LPIPS & SSIM): Measures that compare AI and human perception of video similarity.

Universal Adversarial Perturbation (UAP): Small changes in visuals that significantly affect AI recognition without altering human perception.

Chapter 2

Literature Review and Related Work

2.1 Competitor Analysis

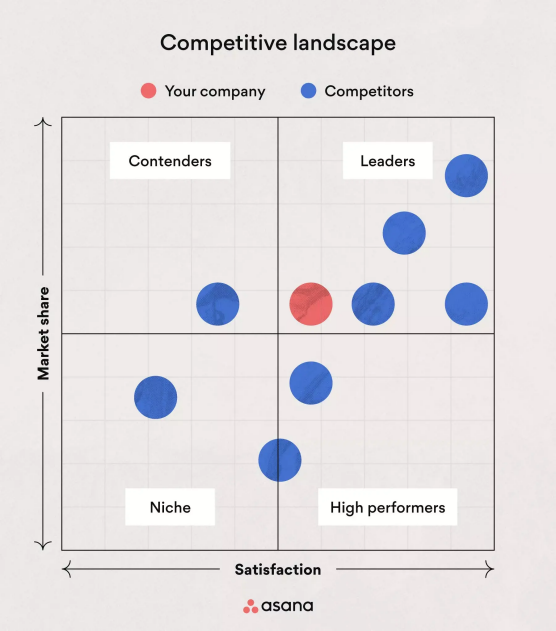

Figure 2.1: Competitive Landscape by Asana

2.2 Literature Review

Add a literature review section if it fits with your project.

Chapter 3

Requirement Analysis

3.1 Stakeholder Analysis

<TIP: List your stakeholders for your project here/>

Stakeholders are individuals, groups, or entities that have an interest, concern, or stake in a particular project, decision, organization, or system. These are individuals or groups who can affect or be affected by the outcomes of your project.

3.2 User Stories

<TIP: Write user stories for each of your stakeholders here./>

User stories are a technique used in agile software development to capture and describe functional requirements from an end user's perspective. They are a way of expressing software features or functionality in a simple, non-technical language that can be easily understood by both developers and stakeholders.

3.3 Use Case Diagram

<TIP: Write a use case diagram for your project here. Refer to an article "What is a use case diagram?" by Lucidchart for help./>

3.4 Use Case Model

A use case is a detailed description of how a system interacts with an external entity (such as a user or another system) to accomplish a

specific goal. Use cases provide a high-level view of the functionality of a system and help in capturing and documenting its requirements from the perspective of end users.

<TIP: Write use cases for your project here. Make sure to use the appropriate type of use case for each scenario (brief, casual, and fully-dressed use case)./>

3.5 User Interface Design

<TIP: Put the initial design of your application here. You can showcase a detailed design of a specific page or a sitemap of your application. See an example below./>

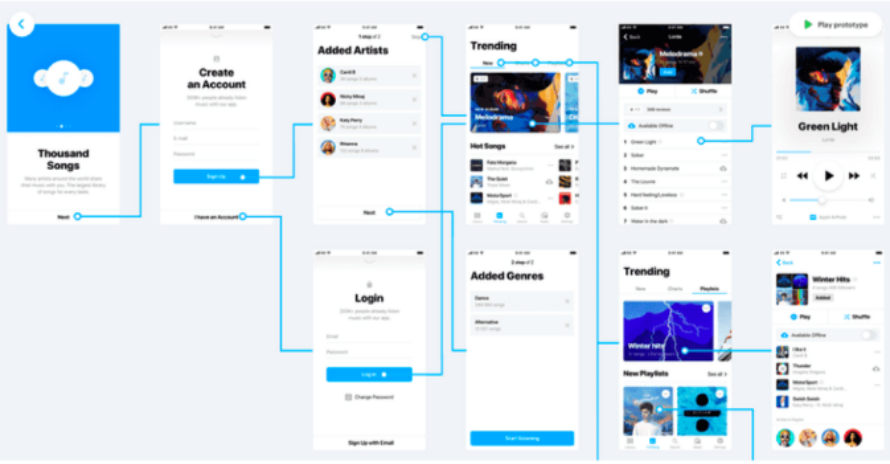

Figure 3.1: User Interface Design

Chapter 4

Software Architecture Design

<TIP: Describe how you design your application using Unified Modelling Language (UML). There should be at least two diagrams that describe the software architecture. You may add additional or remove unnecessary diagrams. However, there needs to be a coherency between them at the end./>

4.1 Domain Model

<TIP: Describe the business concept of your project. Showcase a domain model that captures the said concept./>

4.2 Design Class Diagram

<TIP: Showcase a design class diagram for your project and explain how it works here. You can group classes into packages or layers to communicate your design better./>

4.3 Sequence Diagram

<TIP: Sequence diagrams describe how the software runs at runtime. You do not have to create a sequence diagram for every scenario. However, there should be one for all the main ones./>

<ChatGPT: Creating a sequence diagram for every use case is not strictly necessary, but it can be a valuable tool in certain situations. Sequence diagrams are particularly useful for illustrating the interactions

between different components or objects in a system over time, showcasing the flow of messages or actions between them./>

4.4 Algorithm

<TIP: Optional, If you are working on a research project that proposes a new algorithm, you can describe your algorithm here. It can be in the form of pseudocode or any diagram that you deem appropriate./>

Chapter 5

Software Development

5.1 Software Development Methodology

<TIP: Describe your software development methodology in this section. />

5.2 Technology Stack

<TIP: Describe your technology stack here. See the following example from ThaiProgrammer.org />

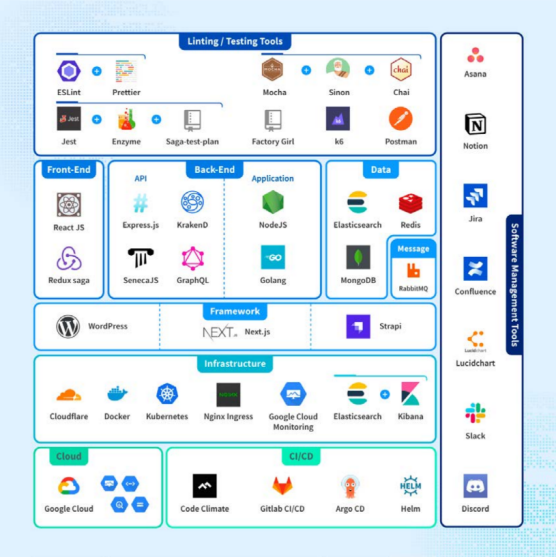

Figure 5.1: Example technology stack

5.3 Coding Standards

<TIP: Describe your coding standard for this project here. />

5.4 Progress Tracking Report

<TIP: Show that you have been working on this project overtime. It can be in the form of a burndown chart or a contribution graph from GitHub./>

Chapter 6

Deliverable

6.1 Software Solution

<TIP: Share a link to your Github repository. Showcase screenshots of the application and briefly describe each page here. />

6.2 Test Report

<TIP: Describe how you test your project. Place a test report here. If you use continuousintegration and deployment (CI/CD) tools, describe your CI/CD method here. />

Chapter 7

Conclusion and Discussion

<TIP: Discuss your work here. For example, you can discuss software patterns that you use in this project, software libraries, difficulties encountered during development, or any other topic. />

We both need to develop a better base understanding of our AI generation, Data poisoning and many other topics to understand how we could poison video best, and as hardware efficient as possible. Thus, we currently are doing AI workshop labs in our freetime, advised by our project overseer Dr. Punpiti, starting since 7 January 2025.

Reference

Bibliography

[1] Overleaf, “Learn latex in 30 minutes,” https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes.

Appendix A

Appendix A: Example

<TIP: Put additional or supplementary information/data/figures in appendices. />

Appendix B

Appendix B: About LaTeX

LaTeX (stylized as LaTeX) is a software system for typesetting documents. LaTeX markup describes the content and layout of the document, as opposed to the formatted text found in WYSIWYG word processors like Google Docs, LibreOffice Writer, and Microsoft Word. The writer uses markup tagging conventions to define the general structure of a document, to stylize text throughout a document (such as bold and italics), and to add citations and cross-references.

LaTeX is widely used in academia for the communication and publication of scientific documents and technical note-taking in many fields, owing partially to its support for complex mathematical notation. It also has a prominent role in the preparation and publication of books and articles that contain complex multilingual materials, such as Arabic and Greek.

Overleaf has also provided a 30-minute guide on how you can get started on using LaTeX. [1]