Technical Countermeasures Against Training Generative AI on Images
(Technical Topic)

Identifying Sources of Negative Sentiment Toward Software Patents
(STS Topic)

A Thesis Project Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Nick Garrone

May 1, 2024

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed : _____

Advisor

Kathryn A. Neeley, Department of Engineering and Society

Introduction

As software has advanced, systems of intellectual property have failed to keep pace. This report will examine two major areas where the cracks are starting to show: first, in the use of copyrighted images to train artificial intelligence, and second, in the application of patent law to software that has caused many software engineers to lose faith in the patent system.

Text-to-image models, machine learning models that turn text prompts into high-quality images, are an emerging threat to the integrity of intellectual property. These models can even replicate specific artists' styles (Shan, Cryan et al., 2023). The prompting process is shown in Figure 1.

Figure 1. *Text Prompts passed to Stable Diffusion.* A text prompt passed to a text-to-image model such as Stable Diffusion yields a high-quality image (Created by author).

In the United States, the legality of training AI text-to-image generators on copyrighted works is uncertain. *Andersen v. Stability AI*, a lawsuit alleging that the training process violates copyright, is unresolved (Brittain, 2023). However, even if the training is found to be unlawful, artists can only be certain to protect their work by finding a solution not dependent on the legal system. The internet is global, so those in other jurisdictions may not be swayed by an American determination of legality. To protect their work, artists can apply technical countermeasures to

treat their images in order to make them resistant to being used as training data for text-to-image models. The technical work will be to research, compile, and analyze both basic and state-of-the art methods in order to determine their strengths, limitations, and future potential.

Where intellectual property and software clash again is in the patent system. The application of patents to software is controversial. Kamdar (2015), writing for the Electronic Frontier Foundation, asserts that "Patents—particularly software patents— have become a tool for intimidation and expensive litigation, chilling the very innovation the patent system was supposed to encourage" (1). There are several issues that cause this perception. First, they can be overbroad to the point that it is impossible to program anything substantial without violating some patent (Kamdar et al., 2015). Additionally, software patents have a history of being abused by non-practicing entities, otherwise known as patent trolls. These entities hold patents but do not create works based on them; instead, they sue those who they allege infringe on their patents (Appel et al., 2018). It is something of a truism in the software industry that software patents negatively affect innovation. However, not all evidence points to that fact. The STS work will be to examine the factors that cause software engineers to have animus toward software patents, as well as to identify positives of software patents that they may be ignoring.

Technical Topic: Technical Countermeasures Against Training Generative AI on Images

With as few as twenty images from an artist, anyone can replicate their style with the help of text-to-image models. Now, artists' unique styles, developed over years of work, can be imitated by anyone with access to a computer. Examples of style-transferred images are shown in Figure 2.

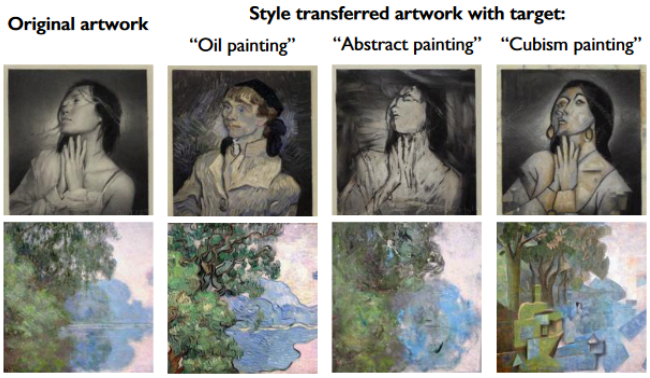

Figure 2. *Style-transferred images*. An image can be reimagined in a different style using text-to-image models (Shan, Cryan et al., 2023, p. 2191).

Given a pre-trained text-to-image model, users can fine-tune that model to allow it to imitate any artist's style. The user can then generate any image they desire in that style or transform existing images into that style. This fine-tuning process can be done with as few as twenty images from the target artist, orders of magnitude less than general training (Shan, Cryan et al., 2023). This process is shown in Figure 3. Fine-tuning puts many artists at risk, not only of their images being used to train commercial AI models but also of their unique style being copied. Without a solution, artists' works will continue to be infringed by this process, possibly worsening as text-to-image models advance.

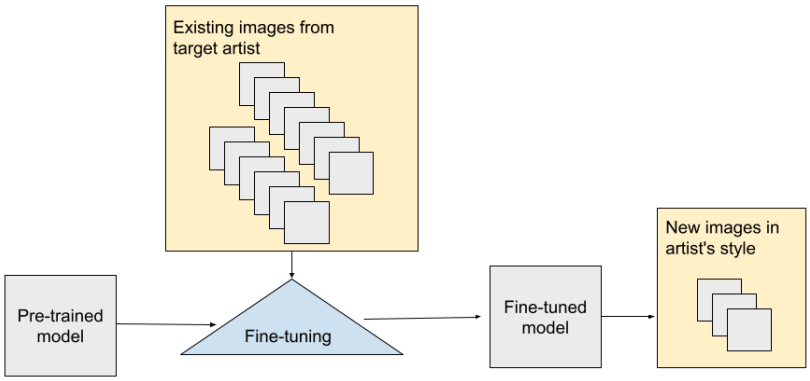

Figure 3. *Fine-tuning process.* Given a small number of images from a target artist, a pre-trained model can be fine-tuned to produce any image in the artist's style (Created by Author).

Text-to-image models have advanced quickly over the last three years. From the first text-to-image model, OpenAI's DALL-E, generative image AI has moved from a curiosity to a serious tool (Faber, 2022). A large technical breakthrough came with latent diffusion models, which drastically increase efficiency by working in latent, or compressed, space (Rombach et al., 2022).

Countermeasures exist that artists can use to protect their work from being used to fuel next-generation models (Shan, Cryan et al., 2023; Shan, Ding et al., 2024). Text-to-image models, like all machine learning models, work by ingesting a large quantity of training data. However, these models do not learn from data the same way humans do. This means that two images that appear the same to the human eye can look very different to the model. Countermeasures take advantage of this by modifying images such that they look the same to humans but act to trick models . Examples of poisoned images are shown in Figure 4.

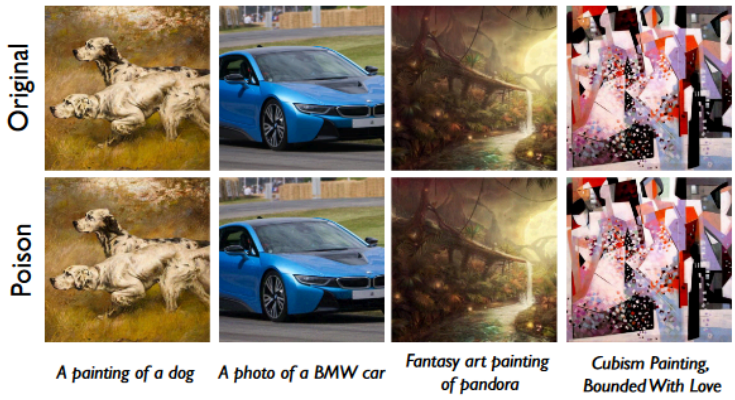

Figure 4. *Images poisoned by Nightshade.* To the human eye, these sets of images are indistinguishable, but the model is tricked (Shan, Ding et al., 2024, p. 5).

My technical work will analyze the most sophisticated systems– named Glaze and Nightshade– out of the University of Chicago. These systems work similarly: both apply subtle perturbations to an image so that it remains almost unchanged to the human eye but presents a radically different face to the AI model. Glaze is designed to protect styles in particular, while Nightshade is more general-purpose. They do this by using another machine learning model to solve an optimization problem that maximizes the distortion of the image in the target model's feature space– the mathematical representation of the image that is relevant to the model– while minimizing changes to the image overall (Shan, Cryan et al., 2023; Shan, Ding et al., 2024).

The technical work will also consider more formative techniques, like dirty-labeling data and text-based backdoor injection. Dirty-labeling data is the simplest technique– by mislabeling an image when it is ingested as training data, models will learn that the images are associated with the incorrect concept (Shan, Ding et al., 2024). Text-based backdoor injection allows users to produce unexpected outcomes when an uncommon character is included in the prompt. This method is more esoteric and requires direct access to a model (Struppek et al., 2023). My technical project will explain their workings, uses, and limitations, as well as considering their

viability in the future. It aims to make artists more informed about the options they have available to them so that they can use these tools to effectively protect their work.

STS Topic: Identifying Sources of Negative Sentiment Toward Software Patents

A survey found that more than 68% of software engineers believe that software patents should be abolished (Burton, 1996, p. 89). As a group, software engineers have lost faith in a patent system they feel has not adapted to their needs. First put into force in the United States with the Patent Act of 1790, the system was originally mostly used for agricultural improvements (Griesbach & Camora, 2016). As computers advanced, the patent system had to adapt. A 2014 US Supreme Court case, *Alice Corp. v CLS Bank International*, restricted the patentability of software by clarifying that abstract ideas could not be made patentable by simply implementing them as a software system (Saltiel, 2019). Software patents have proliferated despite those challenges. Electrical engineering patents, which include software, make up roughly 50% of new patents in the United States, up from 15% in 1975 (Chien, 2016, pp. 1673-1674). This rise is shown in Figure 5. Patent protections are increasingly being extended to non-physical inventions, which did not exist when the patent system was established.

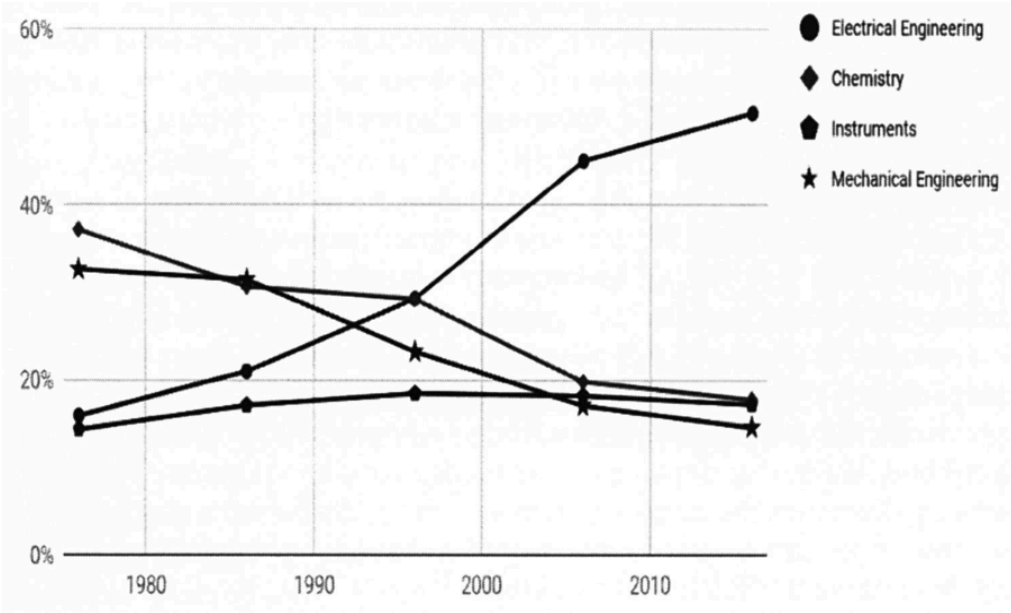

Figure 5. *Shares of U.S. Patents by Industry (1970-2016)*. The share of software patents, included here in the electrical engineering category, have risen sharply since 1970 (Chien, 2016, p. 1673).

As introduced earlier, a major issue with software patents is the prevalence of non-practicing entities. 60% of patent lawsuits are brought by non-practicing entities (Karakashian, 2015, p. 121). These groups hold patents but do not produce any technology with them. Rather, they make money primarily by licensing patents or suing alleged infringers. Sometimes they have valid claims, but other times they target small companies who settle rather than take a false claim to court (Karakashian, 2015). Patent trolls have had a measurably negative effect on startup employment. When states adopted anti-patent troll laws, employment in tech startups grew by 4.4% (Appel et al., 2019, p. 708). These laws include Vermont's legislation which targets unreasonable demand letters. These demand letters are typically sent by non-practicing entities en masse without having a genuine claim. (Appel et al., 2019). Patent trolls, initiating the majority of software patent lawsuits, have soured the reputation of software patent protection.

Another force in the software patent controversy is the patentability of software itself. Unlike mechanical inventions, there is no tangible physical artifact that is produced when new software is invented. Software lives mostly within an abstract space. This gets at the blurry line of what constitutes an invention. Because abstract ideas are not patentable, which aspects of software are patentable is not always clear (Zivojnovic, 2015). The rules governing the patentability of software are complex, but in general software that provides some concrete user experience or benefit tied to a specific implementation does qualify for patent protection (Zivojnovic, 2015). This uncertainty has led to problems– many software patents are overbroad as the patents lack clarity as to what exactly is covered. Kamdar (2015) argues that this uncertainty is also a result of a lack of standardization of the language used in the software field: "Software does not have a standardized lexicon. As a result, unscrupulous patent owners can insist that their patents cover a wide range of technologies." (5). A lack of clarity as to the patentability of software, as well a lack of standardization, has led to damaging over-broadness.

Software patents are not all negative. Software patents can and are used as genuine protection for genuine inventions. The majority of software patent licenses– that is, the acquisition of patent rights by one company from another– are actually instances of technology transfer. That means that, rather than just licensing patent rights, the purchaser is also receiving additional code or trade secrets (Chien, 2016). Chien (2016) claims that "[these] agreements supported the transfer of technology, rather than just transferring naked patent rights." (1679). This implies that the patent is protecting genuine technology in good faith. The alternative, licensing the patent alone, could mean that the company is just seeking legal cover. In this manner genuine software patents are useful in promoting innovative software. With the ease of replication of software, protecting an invention is much more difficult than in a field like

chemical manufacturing. There are typically zero start-up costs to writing a piece of software whereas innovations in manufacturing require expensive physical equipment. Because of this, software patents could play an even larger role in promoting innovation than in other fields.

Many of the negative effects of software patents come from the non-practicing entities that abuse them and not from companies operating in good faith. Still, there are legitimate concerns that software patents are often overbroad and burdensome for software companies. The STS work will examine how software engineers draw upon these factors to reach the conclusion that software patents should be abolished, as well as drawing attention to positives of software patents that they dismiss. This would provide insight into how to facilitate a more productive conversation between software engineers and the policymakers and industry leaders upholding the incumbent patent system.

Conclusion

The technical work will provide artists with a comprehensive overview of the current leading tools to protect their work from being used to train text-to-image models. This would make artists more aware of the options they have to protect their own work, reducing the number of artists whose works are infringed by AI models. The STS work will analyze the ways in which software engineers become disgruntled with the software patent system. It will analyze discourse in order to work towards more productive communication in bridging the gap between software engineers' opinions on patents and those of industry and government. This more productive discourse could lead to more effective action on patent reform in the future.

(2077 words)

References

Appel, I., Farre-Mensa, J., Simintzi, E. (2019). Patent trolls and startup employment. *Journal of Financial Economics, 133*(3), 708-725. https://doi.org/10.1016/j.jfineco.2019.01.003.

Brittain, B. (2023). Artists take new shot at Stability, Midjourney in updated copyright lawsuit. *Reuters*. https://www.reuters.com/legal/litigation/artists-take-new-shot-stability-midjourney-updated-copyright-lawsuit-2023-11-30/

Burton, D. (1996). Software developers want changes in patent and copyright law. *Michigan Telecommunications and Technology Law Review,* 2(1), 87-91. https://repository.law.umich.edu/mttlr/vol2/iss1/4/

Chien, C. V. (2016). Software patents as currency, not tax, on innovation. *Berkeley Technology Law Journa*l, 31(3), 1669-1724. https://www.jstor.org/stable/26381833.

Griesbach, R., Camarota, A. (2016). Putting down roots at the patent office. *United States Patent Office.* https://www.uspto.gov/learning-and-resources/newsletter/inventors-eye/putting-down-roots-patent-office

Faber, T. (2022) The golden age of AI-generated art is here. It's going to get weird. *Financial Times.* https://www.ft.com/content/073ea888-20d7-437c-8226-a2dd9f276de4

Kamdar A., Nazer D., Ranieri, V. (2015). Defend Innovation How to Fix Our Broken Patent System. *Electronic Frontier Foundation.* https://www.eff.org/document/defend-innovation-how-fix-our-broken-patent-system

Karakashian, S. (2015). Software patent war: the effects of patent trolls on startup companies, innovation, and entrepreneurship. *Hastings Business Law Journal*, 11(1), 119-156. https://repository.uclawsf.edu/cgi/viewcontent.cgi?article=1035&context=hastings_business_law_journal

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the 2022 EEE/CVF Conference on Computer Vision and Pattern Recognition* https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042

Saltiel, J. (2019). In the courts: five years after Alice - five lessons learned from the treatment of software patents in litigation. *World Intellectual Property Organization.* https://www.wipo.int/wipo_magazine/en/2019/04/article_0006.html

Shan, S., Cryan, J., Wenger, E., Zheng, H., Hanocka, R., Zhao, B (2023). Glaze: protecting artists from style mimicry by text-to-image models. *Proceedings of the 32nd USENIX Security Symposium.* https://www.usenix.org/system/files/usenixsecurity23-shan.pdf

Shan, S., Ding, W., Passananti, J., Wu, S., Zheng, H., Zhao, B. (2024). *Prompt-specific poisoning attacks on text-to-image generative models*. ArXiv. https://arxiv.org/pdf/2310.13828.pdf

Struppek, L., Hintersdorf, D., Kersting, K. (2023). Rickrolling the artist: injecting backdoors into text encoders for text-to-image synthesis. *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision.* https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00423

Zivojnovic, O. (2015). Patentable subject matter after Alice—distinguishing narrow software patents from overly broad business method patents. *Berkeley Technology Law Journal,* 30(4), 807–862. https://www.jstor.org/stable/26377742