

Quantifying Copyright Infringement in AI-Generated Images

Student: Gabriel Arraño

Mentor: Wenbin Zhang, PhD

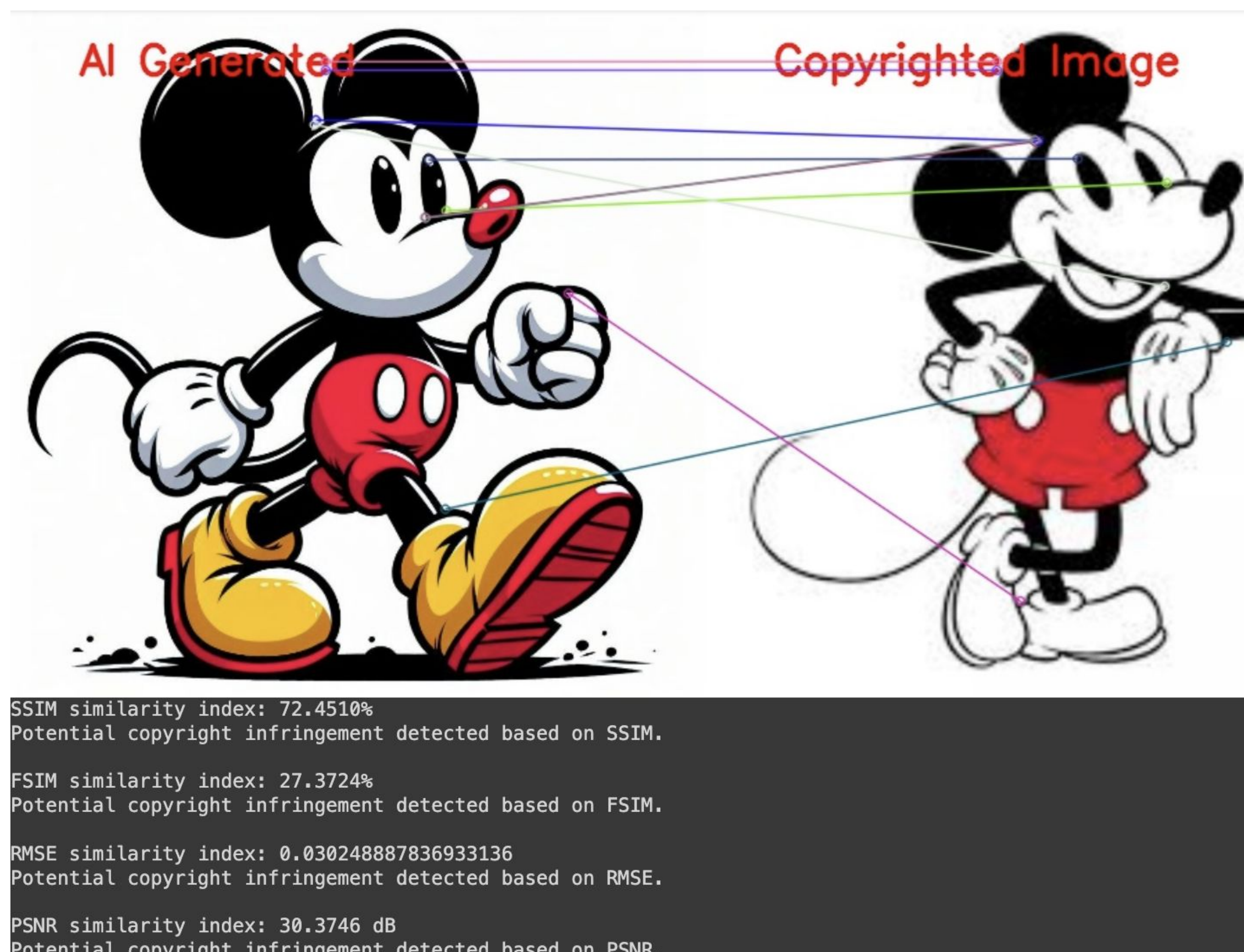
Instructor/Faculty: Masoud Sadjadi, PhD, Florida International University

SUMMARY

In today's era, being able to differentiate between AI generated images and real images poses a new challenge, which is AI generating copyrighted content. Therefore, for our project, "Quantifying Copyright Infringement in AI-Generated Images," we created a tool, which uses math methodologies in a simple python program to quantify copyright on AI-generated images, and helpm stakeholders to make decisions in the landscape of AI generated content.

AI IMAGE GENERATION

To generate images using AI we used a tool called BingImageCreator. We set up a list of prompts. Then, configured the tool to save the images in a designated folder labeled with the date they were created. This tool utilizes Bings image generation capabilities based on the input prompt and a specified cookie value, for authentication. After running the tool BingImageCreator creates images. saves their URLs. The final step involves getting these URLs as references to the generated images, for analysis.



MATH PYTHON METHODS

To quantify image similarity, of already known ai generated images and copyrighted images, I created different mathematical methods in python. I started by using the Structural Similarity Index Measure (SSIM) to assess how similar images are based on brightness, contrast and structure giving a look, at how alike they are. Then the Feature Similarity Index Measure (FSIM) looks closely at similarities in image features to capture details. In addition the Root Mean Squared Error (RMSE) and Peak Signal to Noise Ratio (PSNR) offer quantitative evaluations. RMSE focuses on the squared difference between images while PSNR measures the ratio of signal strength to noise level. These methods help us enhance our accuracy and precision when measuring similarity between images.

METHODOLOGY

Our project commenced with a comprehensive review of literature on image similarity and copyright in AI-generated content. We then benchmarked existing methodologies, guided by insights from a key paper, to understand and set standards for our tool. This was followed by exploring transformative language models to enhance our comprehension of AI's image processing capabilities. Utilizing these insights, we created a unique tool that compares AI-generated images with copyrighted ones, employing sophisticated metrics like SSIM, FSIM, RMSE, and PSNR to quantify their similarity.

REFERENCES

- Image Similarity Measures Library:
<https://pypi.org/project/image-similarity-measures/>
- Bing Image Creator API:
<https://github.com/beratcmn/bing-image-creator-cli>
- AI and Challenges for Copyright Law:
<https://doi.org/10.1093/ijlit/aaaa023>
- :DecodingTrust: Comprehensive Assessment of Trustworthiness in AI Repo:
<https://decodingtrust.github.io/>

SAMPLE IMAGES OF PYTHON PROGRAM

```

Structural Similarity Index (SSIM)

the SSIM score, which for reference is defined by
SSIM(x, y) = (2μxμy + c1)(2σxy + c2) / (μx2 + μy2 + c1)(σx2 + σy2 + c2)

where:
• x and y are the two images being compared.
• μx and μy are the means of x and y, respectively.
• σx and σy are the standard deviations of x and y, respectively.
• σxy is the covariance of x and y.
• c1 and c2 are constants added to avoid instability when the denominator approaches zero.

SSIM is designed to measure the similarity between two images. The SSIM index is a decimal value between -1 and 1, where 1 indicates perfect similarity. Thus, a higher SSIM value means greater similarity between the two images. It assesses visual impact based on luminance, contrast, and structure.

[ ] import cv2
from image_similarity_measures.quality_metrics import ssim

def calculate_ssim(image_path1, image_path2):
    image1 = cv2.imread(image_path1)
    image2 = cv2.imread(image_path2)
    image2 = cv2.resize(image2, (image1.shape[1], image1.shape[0]))

    ssim_value = ssim(image1, image2)
    print("SSIM similarity index: {(ssim_value * 100):.4f}%")
    if ssim_value > 0.5:
        print("Potential copyright infringement detected based on SSIM.\n")
    else:
        print("No potential copyright infringement detected based on SSIM.\n")
    return ssim_value

```

```

# Run BingImageCreator with user input prompt
image_creator = BingImageCreator(prompt,
    save_folder=save_folder_with_date,
    cookie_value=bing_cookie)

image_creator.run()
print(image_creator.urls)

time to wait seconds: 30
the following are the contents
['https://th.bing.com/th/id/OIG4.mDt19fn0syo45B_Dh0Vr?w=270&h=270&c=6&r=0&o=5&pid=ImgGn', 'https://th.b
Image saved to "/content/exportedFiles/2024-04-05_01-43-02/bingOutput/Generate an image of a classic bl
Image saved to "/content/exportedFiles/2024-04-05_01-43-02/bingOutput/Generate an image of a classic bl
Image saved to "/content/exportedFiles/2024-04-05_01-43-02/bingOutput/Generate an image of a classic bl
Image saved to "/content/exportedFiles/2024-04-05_01-43-02/bingOutput/Generate an image of a classic bl
Prompt saved to "/content/exportedFiles/2024-04-05_01-43-02/bingOutput/prompts/Generate an image of a c

[ 'https://th.bing.com/th/id/OIG4.mDt19fn0syo45B_Dh0Vr?pid=ImgGn', 'https://th.bing.com/th/id/OIG4.KyFr

Reverse Image Searching the Generated Results

```

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by XXXXX (name of the project sponsor: federal, state or local government, or corporate partners, where applicable). We/I thank XXXX for his/her/their assistance and mentorship that I/we received throughout the senior design project.