A Comprehensive Web Application for Art Showcasing Utilizing Generative AI for Art Therapy Applications

Athena Bi[1], Yu Sun[2]

[1]Los Alamitos High School, 3591 W Cerritos Ave, Los Alamitos, CA 90720
jinlili134@gmail.com
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768
softcom.lab.cpp@gmail.com

ABSTRACT

Art therapy is a proven method for fostering self-expression and supporting mental health, yet traditional approaches often require resource-intensive, manual analysis by therapists, limiting accessibility [1]. Our project addresses this gap by creating an AI-powered art therapy web application that enables users to upload artwork and receive personalized, therapeutic insights. By combining Firebase for user management, Python Flask for backend processing, and OpenAI for generating art interpretations, our platform bridges the expressive power of art with the analytical capabilities of AI [2].

Key challenges included integrating diverse technologies seamlessly and designing prompts that guide AI to interpret abstract and symbolic art effectively. These challenges were addressed through standardized API protocols and iterative prompt refinement. Experiments tested the AI's accuracy and consistency, showing strong alignment with expert interpretations for symbolic art and reliable outputs across multiple analyses.

Our application democratizes access to art therapy, offering scalable, non-verbal pathways to self-discovery and mental wellness for a broader audience.

KEYWORDS

Comprehensive, Web App, Generative AI, Art Therapy Applications

1. INTRODUCTION

Our project aims to create an AI-powered art therapy web application that not only serves as a platform for users to upload and share artwork but also provides a unique analysis of their creations through AI [3]. This analysis helps individuals gain insights into their mental and emotional states. Art therapy has a history of revealing personal truths, aiding in self-reflection, and facilitating healing. Through colors, patterns, and shapes, art can communicate aspects of mental health that may be challenging to express verbally.

Art therapy has shown promising results in supporting individuals with mental health challenges [4]. For example, a study published in the Journal of the American Art Therapy Association found that art therapy significantly reduced symptoms of depression and anxiety in patients. AI's ability to analyze art enhances this therapeutic effect by identifying principles of composition, such as balance, harmony, contrast, and symbolism, which can indicate underlying emotions and perspectives.

This project addresses the need for accessible, AI-driven mental health tools. By using an AI to analyze artwork, the platform provides users with a meaningful interpretation of their art, promoting self-awareness and healing. This solution is particularly beneficial for individuals who

may feel uncomfortable sharing their thoughts in words, offering them a non-verbal path to self-understanding and growth.

Expressive Writing and Sentiment Analysis:

This methodology analyzes textual data from expressive writing using sentiment analysis to gauge emotional states. It tracks emotional trends over time, providing quantifiable insights. However, it excludes non-verbal expression, struggles with nuanced emotions, and relies on language proficiency [5]. Our project improves this by analyzing visual artwork, offering non-verbal therapeutic insights, and capturing symbolic and abstract elements.

Image Classification for Emotion Recognition:

Using CNNs, this method categorizes artwork into predefined emotional labels based on visual features. It is effective for structured datasets with distinct emotional markers but oversimplifies complex emotions, ignores contextual meaning, and struggles with abstract art. Our project addresses these gaps by focusing on art principles and user-specific analysis for deeper, personalized interpretations.

Manual Art Interpretation in Psychotherapy:

Therapists analyze artwork collaboratively with patients for tailored insights. While effective, it is resource-intensive, inconsistent, and inaccessible to many. Our project automates the process, ensuring scalability, consistent interpretations, and affordable, real-time access for a broader audience.

Our project integrates the strengths of these methods while addressing their shortcomings.

Prompt for OpenAI designed to extract Principal Components of Art and Design, which will be used to synthesize a more catered response regarding the deeper-meanings of the artwork [6]. Prompt are specifically designed to analyze contextual elements of the art image. Response gets sent and outputted back to the web application so that the user can interpret the results.

Our solution is an art therapy platform that uses AI to analyze uploaded artwork, offering users personalized insights into their mental and emotional states based on their creations. This system uses a combination of Firebase for user account management, Render for deploying Python Flask servers, and OpenAI for generating text-based analyses of the uploaded artwork [7].

The AI analysis leverages a carefully designed prompt for OpenAI's language model, focusing on extracting key principles of art and design, such as composition, color balance, contrast, and symbolism. By identifying these elements, the AI synthesizes an interpretation of the deeper meanings within the artwork. The resulting analysis offers users a unique perspective on their creative expression, potentially revealing emotions or themes that contribute to their overall mental health and self-awareness.

Our solution stands out from traditional art therapy methods because it leverages the power of AI to provide immediate, personalized feedback on artwork. Unlike conventional therapy, where interpretations may require professional guidance, this system allows users to explore their artwork's meaning independently, making art therapy more accessible. The combination of Firebase, Render, and OpenAI ensures a seamless experience, from secure account creation and data storage to the generation of insightful analyses. By integrating these technologies, we create an engaging and reflective tool for self-discovery, ideal for individuals seeking alternative approaches to personal growth and mental well-being.

In the first experiment, we aimed to test the accuracy of the generative AI model in providing therapeutic interpretations of artwork. The experiment used well-known artworks, such as Starry Night, with established emotional and psychological analyses as control data. AI-generated outputs were compared to expert interpretations using cosine similarity scores. The most significant finding was the variability in accuracy, with higher scores for artworks containing distinct symbolic elements (Persistence of Memory) and lower scores for more abstract pieces (Starry Night). This suggests the AI struggles with abstract and emotive art, likely due to limitations in training data.

In the second experiment, we tested the consistency of the AI model by analyzing the same artworks multiple times under identical conditions [8]. The experiment revealed high cosine similarity scores across runs, indicating stable outputs. Slight inconsistencies, especially for complex pieces like Mona Lisa, were attributed to the inherent randomness in generative models. Both experiments highlighted the model's strengths and areas for improvement.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. User friendly experience

Understanding how to use Google Cloud Services, Python, Flask, Render, APIs in tandem in creating a functional web page (lots of different programming syntax), to make a seamless experience for users interacting with the platform. (ux)

One major challenge is integrating Google Cloud Services, Python, Flask, Render, and APIs to create a cohesive, user-friendly experience on the platform. Each service and technology has its own syntax and setup requirements, making it crucial to harmonize their interactions smoothly. For instance, managing API requests between Firebase and Render in real-time can lead to latency or compatibility issues if not handled properly. I could use standardized API protocols to streamline communication, ensuring that user requests and responses flow seamlessly. Additionally, error-handling mechanisms would be essential to catch and address any issues that disrupt the user experience.

2.2. The generative AI component

Creating the generative ai component was particularly difficult because it required specific domain knowledge regarding art analysis and how to make interpretations of artwork. This was necessary since we needed to create a prompt that understood the task we were looking to accomplish. (ai)

The generative AI component posed a unique challenge due to the specific domain knowledge required for art analysis and interpretation. Crafting a prompt that accurately directs the AI to analyze and interpret the artwork's visual elements necessitates an understanding of art principles, such as composition and symbolism. This specialized prompt must instruct the AI to focus on these aspects and produce a meaningful response. To address this, I could leverage art analysis frameworks to build prompts that capture the complexity of art interpretation, guiding the AI to generate insights that align with users' expectations of an art therapy platform.

2.3. Data consistency and tracking

Syncing the data across all different stages of the webpage, and across different users. Needed to keep track of account information, as well as information connected to that account (e.g., liked artworks). (data storage and tracking)

Syncing data across different stages of the webpage and for various users presented a data consistency and tracking challenge. User data, such as account details and artwork preferences, must be reliably stored and easily retrievable, ensuring a consistent experience across sessions. Potential problems include data conflicts when multiple users update information simultaneously or when syncing fails due to connectivity issues. To resolve this, I could implement Firebase's real-time database with appropriate data structure design, ensuring efficient data retrieval and updates. Additionally, employing caching mechanisms would help maintain data consistency during connectivity interruptions, providing a stable experience for all users.

3. SOLUTION

Therapart is an art therapy web application built using HTML, JavaScript, and CSS, with Firebase and Python Flask as supporting technologies. It connects three major components: the frontend web application, the Firebase backend, and the Flask-based generative AI hosted on Render to form a seamless system for users to interact with and analyze artwork.

One principal component is the frontend, where users interact through a web interface. Users can create accounts with CAPTCHA verification, log in, and manage their profiles, including updating their profile picture, display name, and other personalized information, such as username, email, and passwords [9]. The frontend also provides the functionality to upload artwork images for other users to see. Users can then communicate to artists to purchase artworks through the platform. These inputs are sent to Firebase, which handles user authentication, data storage, and real-time updates.

The system also gives users the option to request an art analysis from artworks which are uploaded. The image is forwarded to the Flask-based AI module hosted on Render, which will then process the image, and then convert into a descriptive text-based art analysis. The result is then returned to the frontend and displayed to the user so they can receive an analysis of the artwork through the context of art therapy.

The program uses Firebase Authentication and Storage for secure user management and data storage. It leverages Python Flask for AI integration, deployed on Render, and communicates between the components via REST API calls.

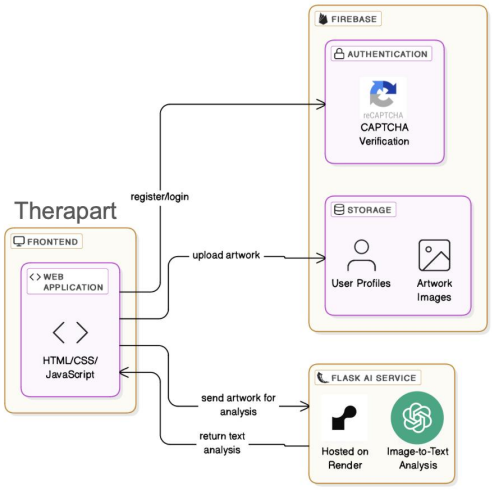

Figure 1. Overview of the solution

The Firebase backend is responsible for user authentication, profile management, and artwork storage. It uses Firebase Authentication to secure accounts with CAPTCHA verification and Firebase Storage to manage user data and uploaded images. This component ensures real-time data updates and smooth interaction between the frontend and other components via event listeners and REST API calls.

Figure 2. Sign up page

```
document.getElementById('signupForm').addEventListener('submit', async function (event) {
  event.preventDefault();

  const nickname = document.getElementById('name').value;
  const email = document.getElementById('email').value;
  const password = document.getElementById('password').value;
  const confirmPassword = document.getElementById('confirmPassword').value;
  const recaptchaToken = document.getElementById('g-recaptcha-response').value;

  if (password !== confirmPassword) {
    alert('Passwords do not match');
    return;
  }

  try {
    console.log('Verifying reCAPTCHA token:', recaptchaToken);  // Log token before verification
    const response = await fetch('https://us-central1-therapart-4134.cloudfunctions.net/app/verify-recaptcha'
    method: 'POST',
    headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
    body: new URLSearchParams({ 'g-recaptcha-response': token })
});
```

Figure 3. Screenshot of code 1

The code snippet shown handles user authentication and profile management using Firebase. This code runs during account creation, login, and profile updates. For example, when a user creates an account, it calls a Firebase function to register the user and verify credentials via CAPTCHA. Upon successful authentication, the program initializes a session for the user, and the user's profile data (e.g., display name, profile picture) is stored in Firebase Realtime Database or Firestore.

Methods in the code interact with Firebase's Authentication API to validate user credentials [10]. Variables such as email and password store user input, while user is an object returned upon successful authentication. Event listeners update the user interface in real time when data changes. If the code communicates with the backend, it ensures that profile updates and artwork uploads are securely linked to the authenticated user.

The Flask-based generative AI provides an art analysis feature. Hosted on Render, this component uses image-to-text processing through neural networks to analyze uploaded artwork. It accepts artwork from the frontend via REST API, processes the image to generate a descriptive text analysis, and returns the result to the frontend.

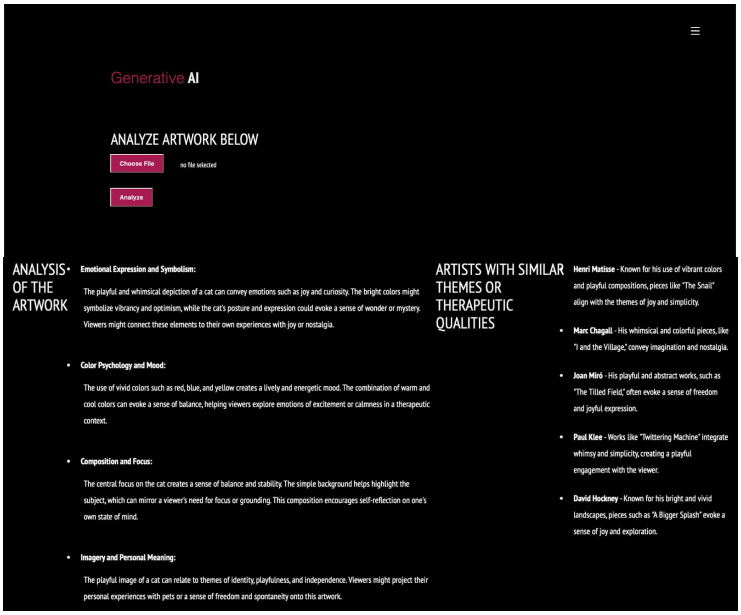

Figure 4. Generative AI

```python
@app.route('/analyze_image', methods=['POST'])
def analyze_image_endpoint():
    try:
        print("Received request to /analyze_image")
        api_key = get_openai_api_key()
        print("API key retrieved.")

        if 'image' not in request.files:
            print("No image file provided in request.")
            return jsonify({'error': 'No image file provided'}), 400
        file = request.files['image']
        print(f"Image file received: {file.filename}")

        image_url = upload_image_to_firebase(file)
        print(f"Image uploaded to Firebase. URL: {image_url}")

        assistant_content = analyze_image(file, api_key)
        print("Image analyzed by OpenAI API.")

        print("Returning assistant's message content to the frontend.")
        return jsonify({'result': assistant_content})
    except Exception as e:
        LOGGER.error("Error processing and rendering assistant's message content", exc_info=True)
        error_msg = str(e)
        print(f"Error in analyze_image_endpoint: {error_msg}")
        return jsonify({'error': error_msg}), 500

if __name__ == '__main__':
    print("Starting Flask app...")
    app.run(debug=True)
```

Figure 5. Screenshot of code 2

The Flask code processes artwork by receiving an HTTP POST request from the frontend containing the image file. This triggers the Flask server to use a neural network model for image-to-text analysis. The model interprets the image and generates a descriptive text output based on predefined parameters (e.g., composition, color usage, emotional tone).

Key methods include process_image(image), which preprocesses the image for the neural network, and generate_analysis(image), which runs the model and returns the text analysis. Variables include image_data, which stores the raw image, and analysis, which stores the model's output. The server then sends the text back to the frontend in a JSON response. This process enables seamless interaction between the frontend and backend, providing real-time art analysis to users.

The frontend web application acts as the user interface, built with HTML, CSS, and JavaScript. Its purpose is to enable account management, artwork uploads, and display art analysis results. It uses AJAX for asynchronous communication with Firebase and the Flask server, ensuring a responsive and interactive experience for users.

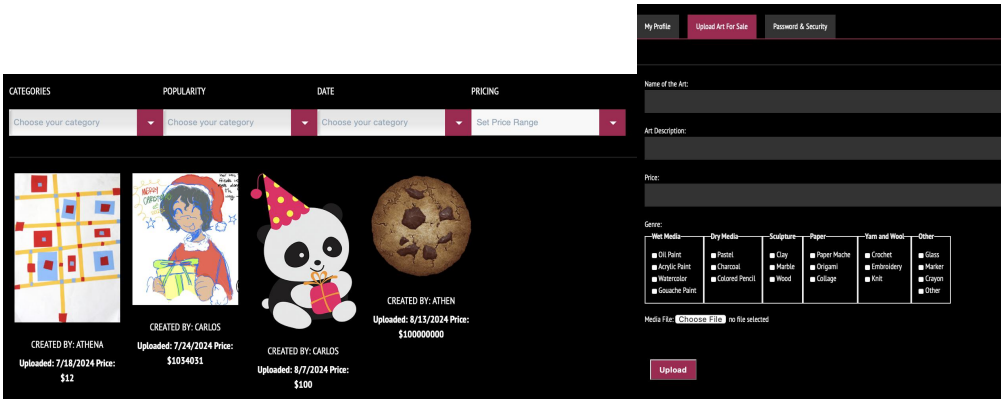

Figure 6. Screenshot of the artworks

```
document.getElementById('artworkForm').addEventListener('submit', async function(event) {
    event.preventDefault();

    const name = document.getElementById('name').value;
    const description = document.getElementById('description').value;
    const price = document.getElementById('price').value;
    const genreCheckboxes = document.querySelectorAll('input[name="genre"]:checked');
    const genres = Array.from(genreCheckboxes).map(checkbox => checkbox.value);
    const file = document.getElementById('file').files[0];
    const fileName = new Date().getTime() + '_' + file.name;

    try {
        const storageReference = storageRef(storage, 'artwork/' + fileName);
        await uploadBytes(storageReference, file);
        const fileURL = await getDownloadURL(storageReference);

        const user = auth.currentUser;
        const email = user ? user.email : 'unknown';
        const displayName = user ? user.displayName : 'unknown';
        const datePosted = new Date().toISOString();

        const artworkId = new Date().getTime();
        await set(ref(database, 'artwork/' + artworkId), {
            name: name,
            description: description,
            price: price,
            genres: genres,
            email: email,
            uid: user.uid,
            artist: displayName,
            datePosted: datePosted,
            fileURL: fileURL,
            likeCount: []
        });

        alert('Artwork uploaded successfully!');
        document.getElementById('artworkForm').reset();
    } catch (error) {
        console.error('Error uploading artwork:', error);
        alert('Error uploading artwork. Please try again.');
    }
});
```

Figure 7. Screenshot of code 3

Key methods include uploadArtwork() to handle the image upload and getAnalysis() to retrieve the AI-generated text analysis. Variables such as file store the uploaded image, and response stores the server's returned text. The server's JSON response is parsed and displayed in the UI. Event listeners tied to buttons or form submissions trigger these methods. This code ensures a seamless flow where users can upload images and view results without refreshing the page, enhancing interactivity and user experience.

4. EXPERIMENT

4.1. Experiment 1

One thing we wanted to investigate in more detail was the accuracy of the generative ai model in providing comprehensive therapeutic interpretations of various different artworks. This is the focal point of our project, so proper testing is imperative.

We will firstly gather a set of world-renowned artworks that have already been extensively studied, particularly from an emotional and psychological perspective (e.g., Starry Night by Vincent Van Gugh). We will then collect control data for each of these artworks, which will serve as benchmarks that the generative ai should aim to detail in its response. Collect multiple samples, and identify core elements that the AI should address, like emotional resonance, symbolic meaning, and psychological themes.

To test the AI's accuracy, we'll collect a dataset of well-known artworks, such as Starry Night by Vincent Van Gogh, which have documented emotional and psychological interpretations. This

will serve as a baseline or "control" data for comparison. Each artwork will have core interpretative elements related to emotional resonance, symbolic meaning, and psychological themes. We ' ll prompt the AI to analyze each artwork, comparing its output to established analyses to measure alignment. We will gather multiple outputs per artwork to assess consistency. This setup allows us to identify how closely the AI aligns with expert interpretations and its ability to provide meaningful insights.

Artwork	Cosine Similarity Score
Mona Lisa	0.7132202128786071
Guernica	0.7276938080788308
Starry Night	0.615788439898422
Persistence of Memory	0.7988995027162855
The Scream	0.6503974244947017

Figure 8. Figure of experiment 1

The mean cosine similarity score of 0.7012 indicates a moderate to high alignment between the AI-generated and human interpretations across the artworks. The median score is close to the mean at 0.7132, suggesting a relatively even distribution of scores without extreme outliers.

The lowest similarity score of 0.6158 for Starry Night was somewhat surprising. Given Starry Night's rich emotional and symbolic content, I expected the AI to align closely with human interpretations. This lower score might indicate that the AI struggled with certain abstract or expressive elements, such as the emotional impact conveyed through Van Gogh's use of swirling patterns and intense colors, which may not have been captured as effectively in a purely textual analysis.

The highest score of 0.7989 for Persistence of Memory reflects the AI's better alignment with the human interpretation. This might be due to the surrealist elements of Persistence of Memory, which involve more identifiable symbolic objects (like melting clocks) that the AI can more readily interpret. The presence of distinct visual symbols in Dali's work likely made it easier for the AI to identify themes of time and decay, aligning well with human analysis.

Factors Affecting Results:

Art Style and Composition: Artworks with clearer symbolic elements (like Persistence of Memory) appear to yield higher similarity scores. This suggests that the AI performs better when it can identify distinct, easily interpretable symbols.

AI's Training Data: The AI may have been trained more extensively on certain types of symbolism and compositional analysis, making it more effective at interpreting surrealism than abstract expressionism. This could lead to higher scores on works with conventional symbolic representation and lower scores on more abstract, emotive pieces.

Conclusion: The results suggest that the AI performs variably depending on the artwork's style and the presence of clear, recognizable symbols. Improving the AI's alignment with human interpretation on expressive or abstract works may require additional training on non-literal, emotional, and dynamic elements in art, enhancing its capacity to interpret the nuanced aspects of artworks like Starry Night.

4.2. Experiment 2

A potential blind spot is the consistency of the generative AI model's responses when analyzing the same artwork multiple times. Consistency is crucial because users need to trust that the AI provides reliable insights rather than random or inconsistent interpretations. Inconsistent results could undermine the credibility of the analysis, as users might receive differing interpretations each time they upload the same image.

To test consistency, we'll conduct multiple runs of the AI model on the same artwork images and compare the similarity of the resulting interpretations.

Select Artworks: Use a subset of artwork images that were previously analyzed, such as Starry Night, Mona Lisa, and The Scream.

Multiple Runs: For each selected artwork, prompt the AI model to generate an interpretation multiple times (e.g., 5 runs per artwork) under the same conditions.

Control Data: Use a reference interpretation from the human analysis as a baseline to assess any changes in focus or tone in each AI-generated response.

Metrics for Analysis

Cosine Similarity: Measure the similarity between each AI-generated interpretation and the initial AI interpretation for each artwork. Higher cosine similarity scores across runs indicate more consistency.

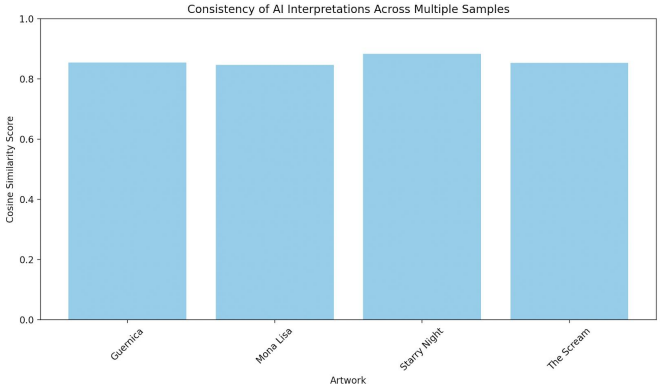

Figure 9. Figure of experiment 2

The data shows a relatively high consistency between the two AI-generated interpretations for each artwork, with cosine similarity scores ranging from 0.846 to 0.884. This indicates that the AI model generates fairly stable interpretations, as each score is close to 1, reflecting a high degree of overlap in the textual content between the main and alternate versions for each artwork.

The similarity score for Starry Night (0.883) is slightly higher than the others, which suggests that the model may be more consistent with certain well-known artworks that have distinctive and widely recognized characteristics. The slightly lower similarity for Mona Lisa (0.846) is unexpected, as one would expect a similarly iconic artwork to yield consistent interpretations.

This variability could be due to the subtleties in the Mona Lisa's artistic elements, which might lead the AI to focus on different interpretative aspects in each run.

The biggest effect on these results likely comes from the model's underlying randomness and sensitivity to small variations in processing. Generative models, especially those based on language, can produce slightly different outputs even for identical inputs due to inherent randomness or differences in emphasis on various descriptive aspects. This could explain the minor variations in consistency across artworks. The results overall suggest that the model performs consistently but may vary slightly based on the complexity or ambiguity of the artwork, which affects the depth and focus of its interpretation.

5. RELATED WORK

Expressive writing combined with sentiment analysis uses natural language processing (NLP) to analyze textual content from therapy participants, identifying emotional trends through sentiment classification. This method provides quantifiable data on emotional states, aiding therapists in tracking patients' mental health progress over time. While effective in capturing broad emotional trends, it is limited by its reliance on verbal communication, difficulty handling nuanced emotions, and cultural bias in language models. Non-verbal expressions, such as visual art, remain unexplored. Your project improves on this methodology by analyzing visual artwork, offering deeper, personalized insights into emotions, symbolism, and abstract themes [11].

Image classification for emotion recognition employs Convolutional Neural Networks (CNNs) to analyze visual features in artwork, such as color, texture, and shapes, and categorize them into predefined emotional labels (e.g., happy, sad, calm). This method is effective for automating emotional analysis of structured datasets, achieving high accuracy with distinct visual-emotional correlations. However, it oversimplifies complex emotions, ignores symbolic or contextual elements, and struggles with abstract art. Your project builds on this by incorporating deeper art principles and user-specific analysis, providing nuanced interpretations beyond emotional categories and accommodating diverse artistic styles for more inclusive therapeutic insights [12].

Manual art interpretation involves trained therapists analyzing patients' artwork to uncover emotional, cognitive, and psychological states. By examining visual elements like color, shape, and composition, therapists offer personalized insights and foster self-reflection through collaborative discussions. While effective for deep and nuanced understanding, this approach is resource-intensive, subjective, and limited in accessibility, making it unsuitable for large-scale or remote use. Your project addresses these limitations by automating the analysis process, ensuring consistent and scalable interpretations while retaining depth and personalization. It democratizes access to therapeutic art analysis, offering real-time feedback and expanding support to underserved communities [13].

6. CONCLUSIONS

One limitation of this project is the variability in the generative AI's interpretations. While generally consistent, the AI can produce slightly different outputs for the same artwork due to inherent randomness [14]. This inconsistency may affect user trust in the platform's feedback. To address this, implementing a fine-tuned model with a higher emphasis on stability or using a

multi-pass approach (where the AI generates multiple analyses and the most consistent one is chosen) could improve reliability.

Another limitation is the dependency on external services like Firebase and Render, which could lead to latency issues or service disruptions [15]. Integrating a caching mechanism could reduce delays by storing recent analyses and user data locally for quicker access.

Finally, the platform lacks customization in the art analysis for different therapeutic needs. Adding configurable settings to allow users to select different types of interpretations or focus areas (e.g., emotional vs. symbolic analysis) could enhance the user experience, making it more adaptable to individual needs.

This art therapy platform leverages AI to offer users insightful interpretations of their artwork, promoting self-reflection and emotional growth. By combining Firebase, a web interface, and a Flask-based AI model, the platform provides a unique tool for self-discovery, bridging art and technology for mental wellness.

REFERENCES

[1] Geue, Kristina, et al. "An overview of art therapy interventions for cancer patients and the results of research." Complementary therapies in medicine 18.3-4 (2010): 160-170.

[2] Mufid, Mohammad Robihul, et al. "Design an mvc model using python for flask framework development." 2019 International Electronics Symposium (IES). IEEE, 2019.

[3] Luo, Xuexing, et al. "Emergence of Artificial Intelligence Art Therapies (AIATs) in Mental Health Care: A Systematic Review." International Journal of Mental Health Nursing (2024).

[4] Frank, Richard G., and Thomas G. McGuire. "Economics and mental health." Handbook of health economics 1 (2000): 893-954.

[5] Phutela, Deepika. "The importance of non-verbal communication." IUP Journal of Soft Skills 9.4 (2015): 43.

[6] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." Future Internet 15.6 (2023): 192.

[7] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[8] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." Future Internet 15.6 (2023): 192.

[9] Shu-Guang, Huang, et al. "A CAPTCHA recognition algorithm based on holistic verification." 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control. IEEE, 2011.

[10] Moroney, Laurence, and Laurence Moroney. "Using authentication in firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 25-50.

[11] Tausczik, Yla R., and James W. Pennebaker. "The psychological meaning of words: LIWC and computerized text analysis methods." Journal of language and social psychology 29.1 (2010): 24-54.

[12] Saleh, Babak, et al. "Toward automated discovery of artistic influence." Multimedia Tools and Applications 75 (2016): 3565-3591.

[13] Lusebrink, Vija B. "Art therapy and the brain: An attempt to understand the underlying processes of art expression in therapy." Art Therapy 21.3 (2004): 125-135.

[14] Luo, Xuexing, et al. "Emergence of Artificial Intelligence Art Therapies (AIATs) in Mental Health Care: A Systematic Review." International Journal of Mental Health Nursing (2024).

[15] Cheng, Chelsea, et al. "Drawing on emotions: the evolving role of art therapy." Irish Journal of Psychological Medicine 40.3 (2023): 500-502.