



RESEARCH ARTICLE

Analyzing why AI struggles with drawing human hands with CLIP

[version 1; peer review: 1 not approved]

Meghna Sarkar¹, Siddhartha Chatterjee², Sudipta Hazra ³, Anurag Sinha ⁴,
Md. Szaid Reza⁵, Mohd Asif Shah ⁶⁻⁸

¹Department of Information Technology, Kalyani Government Engineering College, Nadia, West Bengal, India

²Department of Computer Science and Engineering, College of Engineering and Management- Kolaghat, Kolaghat, West Bengal, India

³Department of Computer Science and Engineering, Asansol Engineering College, Asansol, West Bengal, India

⁴Department of Computer Science and Engineering, ICFAI University Jharkhand, Ranchi, Jharkhand, India

⁵Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh

⁶Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India

⁷Department of Economics, Kardan University, Kabul, Kabul, Afghanistan

⁸Division of Research and Development, Lovely Professional University, Phagwara, Punjab, India

V1 First published: 12 Feb 2025, 14:193
<https://doi.org/10.12688/f1000research.159688.1>

Latest published: 12 Feb 2025, 14:193
<https://doi.org/10.12688/f1000research.159688.1>

Abstract

Background

Artificial Intelligence (AI) has made significant strides in various domains, but generating realistic human hands remains a challenge. This study explores the limitations of AI in capturing the fine details and proportions of hands, using Contrastive Language Image Pretraining (CLIP) as a case study.

Methods

Our analysis reveals that CLIP struggles to accurately represent hands due to inadequate training data, anatomical complexities, and practical challenges. We conducted a series of tests and analyses to identify the primary causes of CLIP's difficulties.

Results

Our results show that CLIP's struggles stem from data biases and insufficient anatomical representation in training datasets. Specifically, we found distorted finger relationships, inaccurate

Open Peer Review

Approval Status 

1

version 1

12 Feb 2025


[view](#)

1. **Mingyong Cheng** , University of California
San Diego, San Diego, USA

Any reports and responses or comments on the article can be found at the end of the article.

proportions, and deviations from expected hand geometry.

Conclusion

This study aims to provide a comprehensive examination of the current limitations and propose possible directions for future research. By leveraging CLIP for evaluation, control algorithms for structure enforcement, DALL-E for generation, AR for gesture tracking, and 3D modeling for anatomical accuracy, we can overcome the challenges of generating realistic human hands and advance AI's capabilities in artistic creativity

Keywords

Contrastive Model, Control Algorithm, CLIP, AI, DALL-E, AR, 3D.

Corresponding author: Mohd Asif Shah (m.asif@kardan.edu.af)

Author roles: **Sarkar M:** Conceptualization, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Chatterjee S:** Resources, Validation, Writing – Review & Editing; **Hazra S:** Formal Analysis, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Sinha A:** Conceptualization, Formal Analysis, Investigation, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Reza MS:** Conceptualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Shah MA:** Data Curation, Formal Analysis

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2025 Sarkar M *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Sarkar M, Chatterjee S, Hazra S *et al.* **Analyzing why AI struggles with drawing human hands with CLIP** [version 1; peer review: 1 not approved] F1000Research 2025, 14:193 <https://doi.org/10.12688/f1000research.159688.1>

First published: 12 Feb 2025, 14:193 <https://doi.org/10.12688/f1000research.159688.1>

1. Introduction

This AI is like a human trapped in a museum from its birth. It learns perceiving things from limited resources, the web. It cannot roam in real world like human, so does not have the ability to analysis thing from real world scenario. AI knows how hands looks but not how hands work. The human hand is an intricate and versatile organ, capable of performing a wide range of tasks with precision and dexterity. Despite the advancements in AI and machine learning, accurately modelling and predicting the movements and interactions of human hands remain a significant challenge. This thesis delves into the reasons behind these difficulties and explores the implications for various applications, including robotics, virtual reality, and healthcare. By learning to associate images with text, an artificial intelligence model known as CLIP (Contrastive Language–Image Pretraining) is able to understand and generate visuals in response to textual inputs. When used in conjunction with generative models like DALL-E or diffusion models, CLIP produces images that closely resemble the input descriptions and performs well in AI drawing applications. This function allows for the production of images that follow specific prompts or styles. Although because of its large training set, CLIP might not be able to produce features that are very accurate or realistic, it can help by showing intricate items like human hands. we can systematically address these challenges by using CLIP for evaluation, control algorithms for structure enforcement, DALL-E for generation, AR for gesture tracking, and 3D modeling for anatomical accuracy.

2. Related work

In The ‘bad hands’ phenomena, which artificial intelligence is responsible for creating, emphasizes the value of human creativity and media literacy. Artificial intelligence has advanced to the point where these skills are no longer required, but educators and artists can still employ ‘bad hands’ to push the limits of machine learning and redefine humanity in algorithms.¹ Hand surgery requires precise techniques due to the hand’s complexity. Generative AI (GenAI) can enhance this by analyzing data, creating detailed simulations, and personalizing procedures, potentially reducing complications. This review explores how GenAI could improve hand surgery, leading to better patient outcomes and setting new standards in the field.² The Cascaded Deep Graphical Convolutional Neural Network (DCGCN) framework outperforms state-of-the-art models in accuracy and computational cost for 2D hand pose estimation in AI applications.³ Optimizing human hand gestures for AI systems reduces error rates and effort while maintaining the original gesture trajectory, improving interaction with AI systems.⁴ Human/computer control of dexterous remote hands presents unique challenges, including grasp stabilization and nonanthromorphic behaviour, but progress has been made in grasp planning and controlled slip techniques.⁵ This paper reviews current research in hand and finger modeling and animation, highlighting progress towards convincing, detailed motions for virtual characters in areas like manipulation and communication.⁶ The FF-SSD deep learning network effectively detects and localizes hands in space human-robot interaction, outperforming state-of-the-art methods.⁷ Pixelor is a competitive drawing AI agent that can achieve human-level performance in a Pictionary-like sketching game by learning optimal stroke sequencing strategies and achieving recognizable results faster than humans.⁸ A unified control framework for robotic hands can simplify and generalize their control, allowing for more advanced manipulation tasks in industries.⁹ A knowledge-based approach using a three- phased scheme can effectively simulate human hand motion and grasping of arbitrary objects, reducing search space and improving performance.¹⁰ This paper¹¹ presents an algorithm for hand-drawn interfaces that simplifies designs by replacing multiple strokes with a single stroke, rationalizing the designer’s creative intent. The hands-free human-computer interface using facial movements achieved high performance and accuracy, offering increased independence and confidence for patients with limited hand function.¹² The designed grasping control strategy effectively adapts an anthropomorphic robotic hand to object contours, achieving human-like behaviour and robustness.¹³ AI raises issues of responsibility attribution, including the problem of many hands and the temporal dimension of control, affecting transparency and explainability.¹⁴ Hand gesture recognition (HGR) is a research hotspot in HMI due to its high degree of differentiation, strong flexibility, and efficiency of information transmission.¹⁵ The AI edge computing-based system uses gesture tracking and recognition techniques to detect the correctness of stroke trajectory during writing or drawing.¹⁶ Combinatorial generalization and structured representations are key to achieving human-like abilities in AI, such as drawing human hands.¹⁷

3. Method

3.1 Mechanism of CLIP

There are various types of model that can be used for explain how the AI generates an image. One such fine example is of CLIP. It is a type of contrastive model that obtains knowledge of semantic information and contextual relationships by learning visual representations of large undefined text data. The following illustration represents a clearer view over this matter. The illustration in [Figure 1](#) is about a detailed overview of UnCLIP. Above the dashed line is the CLIP training process, through which we learn a unified presentation space for text and images. Our text-to image conversion process involves feeding the CLIP text into an autoregressive or diffusion system before embedding the image, followed by conditioning the diffusion decoder that generates the final image. Picture Note that the CLIP model is blocked during feedforward and decoder training. Success and careful manual guidance.

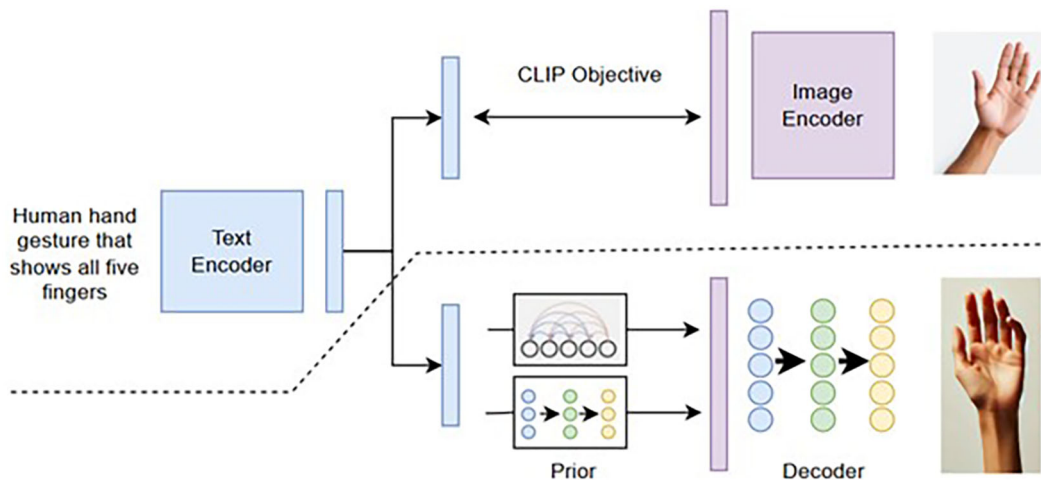


Figure 1. Mechanism of CLIP.

In addition, encoding and decoding images also gives us a tool to observe which image features are detected or ignored by CLIP. By integrating a CLIP image embedding decoder into an earlier model that creates a potential CLIP image embedded in arbitrary text, we can create generative image models from scratch. Our text-to-image system is compared to DALL-E and GLIDE by researchers, who observe that our samples are similar in quality to SLID, but our generations differ more. Our research demonstrates that propagation priors can be trained in hidden space while still performing at the same level of performance as autoregressive prior, with better computational efficiency. Due to the CLIP codec being flipped, the full-text conditional image generation stack is known as unclip. This methodology provides a technical and detailed approach to analyzing why AI models, particularly those leveraging contrastive learning such as CLIP, struggle with accurately drawing hands. The analysis involves model evaluation, latent space examination, quantitative metrics, and dataset scrutiny. The specific component flow works in the following way:

- i) **Text encoder:** The input is a textual description: “Human hand gesture that shows all five fingers” This text is passed through a text encoder, which converts the text into a latent vector representation.
- ii) **CLIP objective:** The CLIP (Contrastive Language-Image Pretraining) objective is used to align the text and image representations in the same latent space. This means that the text encoder and image encoder are trained together so that the latent vectors for matching text and images are close to each other in the latent space. The encoded text vector is compared to encoded image vectors using this CLIP objective. The goal is to ensure that the text encoding is close to the image encoding of the corresponding image (in this case, an image of a human hand with all of its five fingers being visible).
- iii) **Image encoder:** An image that corresponds to the text prompt is passed through an image encoder, producing an image latent vector. This vector representation of the image is used in conjunction with the text encoding to ensure alignment in the latent space via the CLIP objective.
- iv) **Prior network:** The prior network generates a distribution of potential image representations based on the text encoding. This step is crucial for creating diverse image outputs from the same text prompt. It takes the text latent vector and processes it to generate a set of potential latent vectors that could correspond to images matching the description. Two different model classes for the prior model are available:
 - **Autoregressive (AR) prior:** Based on the caption, the CLIP image embedding is transformed into a series of discrete codes and predicted autoregressively.
 - **Diffusion prior:** A Gaussian diffusion model conditioned on the caption is used to directly model the continuous vector.
- v) **Decoder:** The decoder takes the generated image latent vector from the prior network and converts it into a full-resolution image. This involves a generative model, such as a Variational Autoencoder (VAE) or a

Generative Adversarial Network (GAN), which can decode the latent representation back into a high-quality image. The result is the final image that visually represents the input text prompt. In this case, it would be an image of a human hand gesture with all its five fingers being exposed.

In the **Figure 2** a Model Architecture of CLIP model and its' interfaces is illustrated as the following, that shows how CLIP generates image from text in a very basic method.

- i) **Text Encoder:** This initial block takes the input text prompt (such as a description or caption) and converts it into a numerical representation. Think of it as translating words into a format that neural networks can understand.
- ii) **CLIP (Contrastive Language-Image Pretraining):** CLIP is a remarkable model that bridges the gap between language and vision. It learns to associate images and text by embedding them into a shared space. This allows it to understand both visual content and textual descriptions.
- iii) **Image Encoder:** Once we have an image, the image encoder processes it and generates a feature vector. This vector captures relevant information about the image, which can be used for subsequent steps.
- iv) **Prior and Diffusion Decoder Blocks:** These are critical for image synthesis. The “prior” refers to a learned distribution of latent variables (essentially, hidden factors), while the “diffusion decoder” reconstructs the image from these latent variables. Together, they enable controlled image generation.
- v) **Additional Conditioning Steps:** These steps refine the process. They might involve fine-tuning based on specific attributes mentioned in the text prompt. For example, if the prompt specifies “a sunny beach,” the conditioning steps adjust the generated image accordingly.

3.2 Calculation

A. Data preparation

Dataset selection: A dataset D consisting of paired text descriptions T and corresponding images I of hands is used. This dataset should cover a diverse range of hand poses, shapes, and contexts to ensure a comprehensive analysis. Using this method, an optimal solution can be found alongwith the following **equation (1)**.

$$D = \{(T_i, I_i)\}_{i=1}^N \quad (1)$$

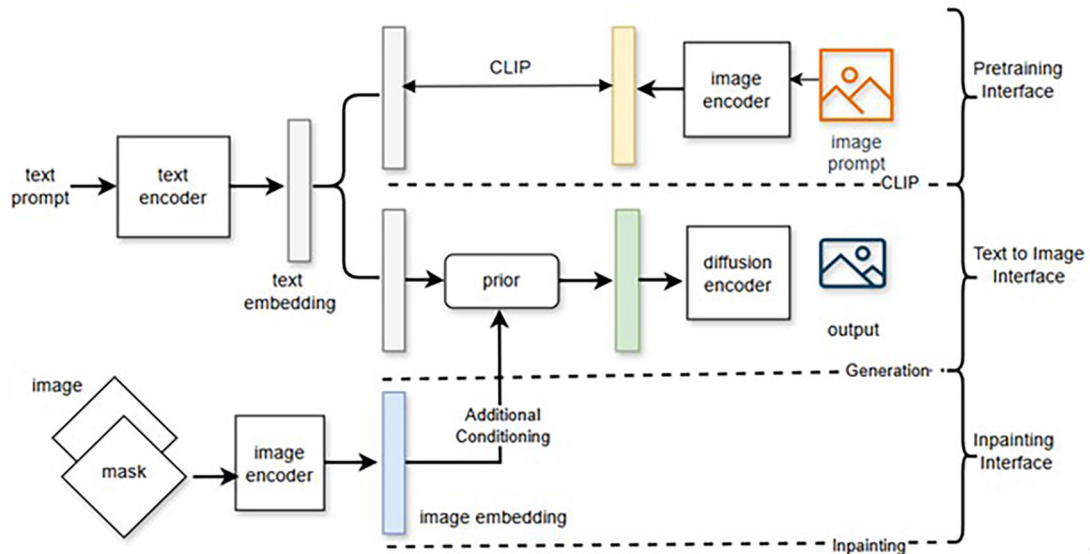


Figure 2. Model architecture of CLIP and its interfaces.

B. Model components

- i) **Text encoder:** A transformer-based text encoder ET is used to convert textual descriptions into latent vector representations zT . This can be obtained by the following [equation \(2\)](#).

$$zT = ET(T) \quad (2)$$

- ii) **Image encoder:** An image encoder EI is used to convert images into latent vector representations zI , which is gained by [equation \(3\)](#).

$$zI = EI(I) \quad (3)$$

- iii) **CLIP objective:** The CLIP (Contrastive Language-Image Pretraining) objective is employed to align the text and image representations in the latent space. The objective function for a batch of size N is defined as the following expression [\(4\)](#):

$$L_{\text{clip}} = -\frac{1}{N} \sum_{i=1}^N \left[\log \frac{\exp(\text{sim}(zTi, zIi)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(zTi, zIj)/\tau)} + \log \frac{\exp(\text{sim}(zIi, zTi)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(zIi, zTj)/\tau)} \right] \quad (4)$$

where sim = cosine similarity; τ (tau) = temperature parameter.

C. Analysis procedure

- i) **Model performance analysis:** Evaluate the performance of the text-to-image generation model specifically on hand images. The text encoder ET processes descriptions of hand images, and the image encoder EI processes the corresponding images. Analyze the alignment of text and image representations in the latent space. This whole analysis can be analyzed with the following [equation \(5\)](#),

$$\text{sim}(zTi, zIi) = \begin{cases} zTi = ET(Ti); \\ zIi = EI(Ii); \end{cases} \quad (5)$$

- ii) **Latent space visualization:** Use dimensionality reduction techniques like t-SNE or PCA to visualize the latent space of text and image encodings. The visualization helps in understanding the clustering of hand images and their textual descriptions in the latent space. This visualization follows the expression [\(6\)](#),

$$t-SNE(Z_T, Z_I) = \begin{cases} Z_T = \{Z_{Ti}\}_{i=1}^N; \\ Z_I = \{Z_{Ii}\}_{i=1}^N; \end{cases} \quad (6)$$

- iii) **Qualitative analysis:** Generate images from text descriptions of hands using the trained model. Visually inspect the generated images for common errors and patterns, focusing on aspects such as finger placement, proportions, and overall hand shape. This is expressed by [equation \(7\)](#)

$$\hat{i} = D(P(zT)) \quad (7)$$

- iv) **Quantitative metrics:** Employ quantitative metrics to assess the quality of hand images generated by the model. Metrics like Structural Similarity Index (SSIM) and Mean Squared Error (MSE) between generated and real hand images are used. Here these metrics are expressed by [equation \(8\)](#) and [\(9\)](#) for SSIM and MSE.

$$\text{SSIM}(I_g, I_r) = \frac{(2\mu_g\mu_r + C1)(2\sigma_{gr} + C2)}{(\mu_g^2 + \mu_r^2 + C1)(\sigma_g^2 + \sigma_r^2 + C2)} \quad (8)$$

$$\text{MSE}(I_g, I_r) = \frac{1}{N} \sum_{i=1}^N (I_{g,i} - I_{r,i})^2 \cdot z \quad (9)$$

where I_g and I_r are the generated and real images, respectively, μ and σ represent mean and variance, and $C1$, $C2$ are constants to stabilize the division.

- v) **Error analysis:** Perform detailed error analysis to categorize the types of mistakes made by the model. Errors can be classified into anatomical inaccuracies, unnatural poses, missing fingers, etc. ($E_{\text{anatomical}}$, E_{pose} , E_{missing}).
- vi) **Dataset evaluation:** Evaluate the dataset D to identify potential biases or gaps in the representation of hands. Assess whether the dataset includes a sufficient variety of hand poses, shapes, and contexts. Identify if the dataset lacks specific types of hand images that might contribute to the model's difficulties. This evaluation holds the [expression \(10\)](#).

$$\text{Variety}(D) = \frac{\sum_k^K \text{unique}(Tk, Ik)}{K} \quad (10)$$

4. Technical challenges

4.1 Complexity of hand anatomy

4.1.1 Challenges

4.1.1.1 Structural complexity

- i) **Bones and joints:** The human hand has 27 bones, including the phalanges (finger bones), metacarpals (palm bones), and carpal bones (wrist bones). Each joint, especially in the fingers, allows for a wide range of motion and poses.
- ii) **Movement dynamics:** The fingers can bend, twist, and rotate in various directions. Accurately capturing these movements and the transitions between them is challenging.¹⁸

4.1.1.2 Surface anatomy

- i) **Muscles and tendons:** The hand's surface anatomy includes muscles, tendons, and veins that change appearance based on hand movements and poses.¹⁹
- ii) **Skin texture and wrinkles:** The skin on the hand has unique textures, lines, and wrinkles, especially on the palms and knuckles. These details are crucial for realistic rendering.²⁰

4.1.1.3 Articulation and posing

- i) **Finger poses:** Each finger can independently move, creating countless possible poses. The AI must understand the natural range of motion and how fingers interact.²¹
- ii) **Hand gestures:** Hands can express a wide range of emotions and actions through gestures. Understanding and replicating these gestures adds complexity.²²

4.1.1.4 Perspective and proportion

- **Foreshortening:** Drawing hands from different angles, especially when fingers are pointed towards or away from the viewer, requires accurate foreshortening to maintain realistic proportions.²³
- **Relative size:** Each finger has a different length and thickness, and these proportions must be maintained from various perspectives.

4.1.1.5 Inter-hand interaction influenced hand grips and interactions: When hands hold objects or interact with other body parts, the AI must accurately depict the contact points and the resulting deformations in the skin and muscles.²⁴

4.1.1.6 Lighting and shadows

- i) **Light interactions:** The hand's complex structure creates intricate patterns of light and shadow, especially in the spaces between fingers and around joints. Capturing these details is crucial for realistic rendering.²⁵
- ii) **Reflective and translucent properties:** The skin of the hand has both reflective and translucent properties, which affect how light interacts with it.²⁶

4.1.1.7 Symmetry and asymmetry

- i) **Bilateral symmetry:** While hands are generally symmetrical, minor asymmetries due to individual differences and hand use must be considered.²⁷
- ii) **Dominance and wear:** The dominant hand often shows different wear patterns and muscular development compared to the non-dominant hand.²⁸

4.1.2 Quantitative measurements

4.1.2.1 Pose estimation accuracy:

a. Keypoint detection: Keypoint detection involves identifying specific points on the hand, such as joints and fingertip positions. To measure the accuracy of keypoint detection, the following metrics are commonly used:

iii) Mean Squared Error (MSE):

- **Description:** MSE is used to measure the average squared difference between the predicted keypoint coordinates and the ground truth coordinates. This can be expressed by [equation \(11\)](#) as well.

- **Formula:**

$$MSE = \frac{1}{N} \sum_{i=1}^N \left((x_i - x^i)^2 + (y_i - y^i)^2 \right) \quad (11)$$

where N is the number of keypoints, (x_i, y_i) are the ground truth coordinates, and (x^i, y^i) are the predicted coordinates.

- **Application:** MSE provides a straightforward indication of the overall prediction error. Lower MSE values indicate higher accuracy.

iv) Percentage of Correct Keypoints (PCK):

- **Description:** PCK measures the percentage of keypoints that fall within a certain threshold distance from the ground truth and is expressed by [equation \(12\)](#).

- **Formula:**

$$PCK = \frac{1}{N} \sum_{i=1}^N 1 \left(\sqrt{(x_i - x^i)^2 + (y_i - y^i)^2} < \alpha \right) \quad (12)$$

where α is the threshold distance, $1(\cdot)$ is the indicator function, and the rest are as defined above.

- **Application:** PCK is often used to assess model performance under varying thresholds, providing insight into the robustness of the keypoint detection.

b. Average Distance Error (ADE): ADE measures the average Euclidean distance between the predicted and ground truth keypoints, providing a more intuitive understanding of the prediction error.

- **Description:** ADE calculates the average Euclidean distance between predicted keypoints and their corresponding ground truth keypoints. This is expressed by the [\(13\)](#) no. formula.

- **Formula:**

$$ADE = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - x^i)^2 + (y_i - y^i)^2} \quad (13)$$

where N is the number of keypoints, (x_i, y_i) are the ground truth coordinates, and (x^i, y^i) are the predicted coordinates.

- **Application:** ADE gives a direct measure of the average error in prediction, making it easier to understand how far off the model's predictions are from the actual keypoints. Lower ADE values indicate higher accuracy.

4.1.2.2. Shape and proportion accuracy

- Procrustes analysis:** Use Procrustes distance to measure the similarity between predicted hand shapes and ground truth shapes after removing differences in scale, rotation, and translation.²⁹
- Aspect ratio consistency:** Measure the consistency of aspect ratios of fingers and the overall hand structure.³⁰

4.1.2.3. Surface detail and texture accuracy

The **Figure 3** gives an overview on texture analysis base on outlines the steps involved in image classification, emphasizing the role of textural features and the Random Forest algorithm on a selected image segment that follows the flowchart given below:

- Input data (Labelled images):** The process starts with a set of labelled images. These images have known class labels (e.g., “cat,” “dog,” “car,” etc.).
- Feature extraction (Textural features):** Next, we extract relevant features from these labelled images. These features capture the visual characteristics of the images. Textural features play a crucial role in image classification. They describe patterns, textures, and spatial relationships within the image.
- Engineered & learned features:** The flowchart mentions both “engineered” and “learned” features. Engineered Features: These are handcrafted features designed by domain experts. Examples include texture descriptors, color histograms, and edge-based features.

Learned features: These are automatically learned by neural networks or other machine learning models during training. Convolutional Neural Networks (CNNs) excel at learning hierarchical features from raw pixel data.

- Random forest algorithm:**The flowchart includes a “Random Forest” algorithm. Random Forest is an approach for ensemble learning that combines several decision trees. Each decision tree is trained on a set of attributes and data points. The final prediction is based on the majority vote of individual trees.
- Image labels (Classification):** Using the extracted features, the Random Forest predicts the class labels for unlabelled images. The function “ $=f(\text{Features})$ ” represents this classification process.
- Feature scores & classification accuracy:** The output of the Random Forest includes feature scores, which indicate the importance of each feature. Classification accuracy measures how well the model performs on unseen data.

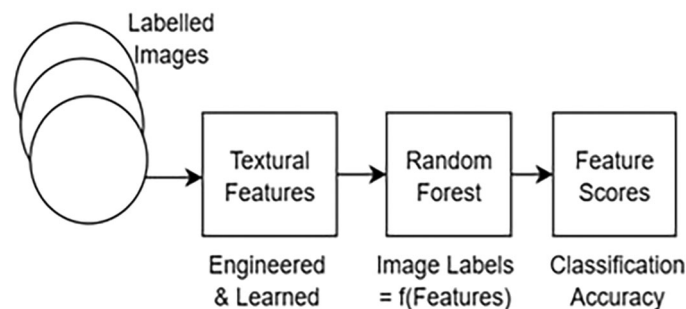


Figure 3. Texture analysis.

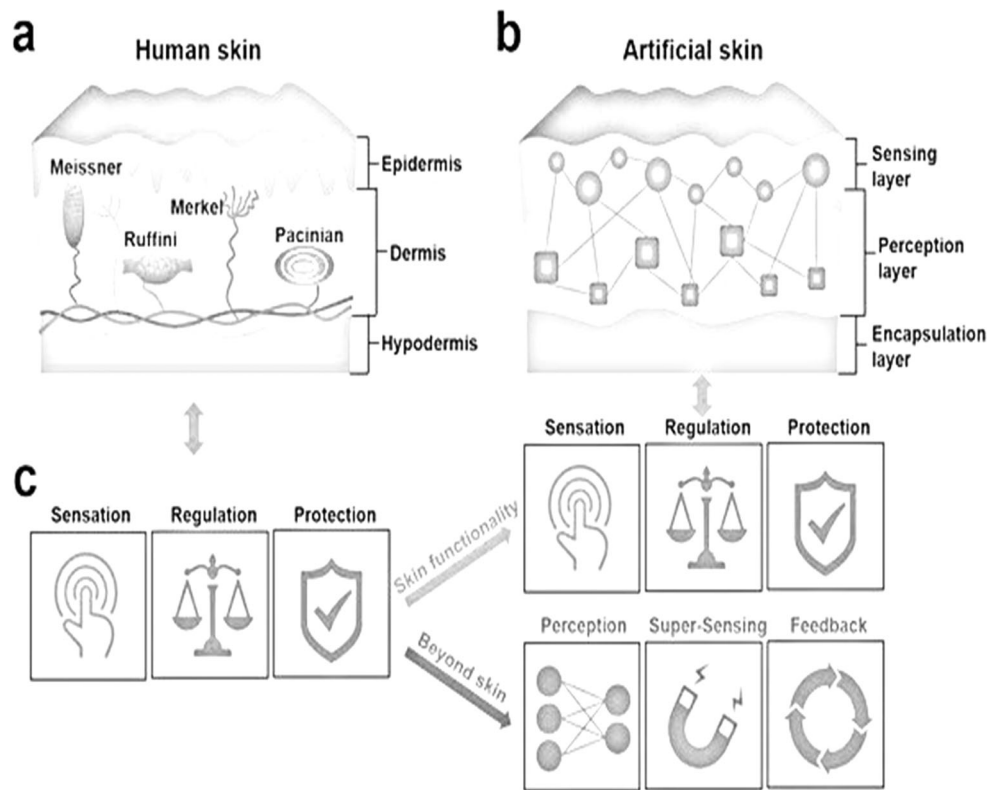


Figure 4. Surface analysis.

The image in Figure 4 depicts a comparison between human skin and artificial skin, highlighting their respective structures and functionalities.

1. **Human skin (Left side):** The cross-section of human skin reveals its layers: epidermis and dermis. Within these layers, various sensory receptors are labeled:
 - **Meissner's corpuscles:** Responsible for light touch and sensitivity.
 - **Merkel cells:** Involved in tactile discrimination.
 - **Ruffini endings:** Detect skin stretch.
 - **Pacinian corpuscles:** Detect pressure and vibration. These receptors contribute to our sense of touch and perception.
2. **Artificial skin (Right side):** The artificial skin structure consists of sensor nodes interconnected by lines, forming a network. An encapsulation layer covers these nodes. Icons below the illustrations compare functionalities:
 - **Sensation:** Human skin vs. artificial skin.
 - **Regulation:** Human skin maintains temperature; artificial skin aims to do the same.
 - **Protection:** Both provide protective functions.

Additional icons represent advanced features of artificial skin

- **Super-sensing:** Enhanced perception (depicted by an eye with circuit patterns).
 - **Beyond-skin perception:** Connectivity (depicted by a Wi-Fi symbol).
 - **Feedback:** Loop of information exchange.
- i) **Texture similarity metrics:** Use metrics like Structural Similarity Index (SSIM) or Peak Signal-to-Noise Ratio (PSNR) to compare the textures of AI-generated hands with ground truth images.³¹
 - ii) **Wrinkle and line detection:** Measure the presence and accuracy of skin details such as wrinkles and lines using edge detection algorithms.³²

4.1.2.4 Movement dynamics and articulation

i) **Temporal consistency:** For sequences of hand movements, measure the temporal consistency of keypoints and shapes across frames³³

Temporal consistency is crucial when working with video data, especially in the context of artificial intelligence. Applying spatial augmentations to video data, considering temporal consistency is essential. It helps maintain the coherence of the sequence and improves the quality of learned representations. In the image provided in [Figure 5](#) it shows various ways it impacts spatial augmentation.³³ It is shown with three rows demonstrating different approaches to spatial augmentation:³⁴



Figure 5. Temporal consistency.

- a) **Original video clip (Top row):** This row contains four frames showing a horse in various positions as it moves. These frames represent the natural progression of the video clip.
- b) **Frame-level spatial augmentation (Middle row):** In this row, we also have four frames, but each frame has undergone individual augmentations. These augmentations include changes in brightness, contrast, and color saturation. However, the key issue here is that these augmentations were applied independently to each frame, without considering the context of the previous or next frame. As a result, the appearance across the sequence lacks consistency. This lack of temporal consistency can be problematic for AI models that learn from video data because it disrupts the natural flow of movement.
- c) **Temporally consistent spatial augmentation (Bottom row):** The bottom row shows four frames where augmentations have been applied while maintaining temporal consistency. Temporally consistent augmentations smoothly transition from one frame to another. This ensures that the changes in brightness, contrast, and color saturation align with the video's natural progression. By preserving temporal consistency, AI models can learn more effectively from video clips.

ii) Joint angle accuracy: Compare the predicted joint angles with ground truth angles using angular error metrics.³⁵ The image in [Figure 6](#) illustrates a model architecture for predicting lower limb joint angles and moments during gait using artificial neural networks. It compares two approaches: a feed-forward neural network and an LSTM (Long Short-Term Memory) neural network. Although it suggests that both feed-forward and LSTM neural networks can be used for this prediction task but the LSTM is expected to perform better due to its ability to consider the temporal context of the IMU data. This process flows through following method.

- a) **IMU data:** The process starts with IMU (Inertial Measurement Unit) data, which likely captures information about acceleration and angular velocity during movement.
- b) **Neural networks:** It is consisted with two different paths.

1st. Feed-forward neural network: The IMU data is fed into a feed-forward neural network. This type of network processes data in one direction, from input to output, without forming loops or cycles.

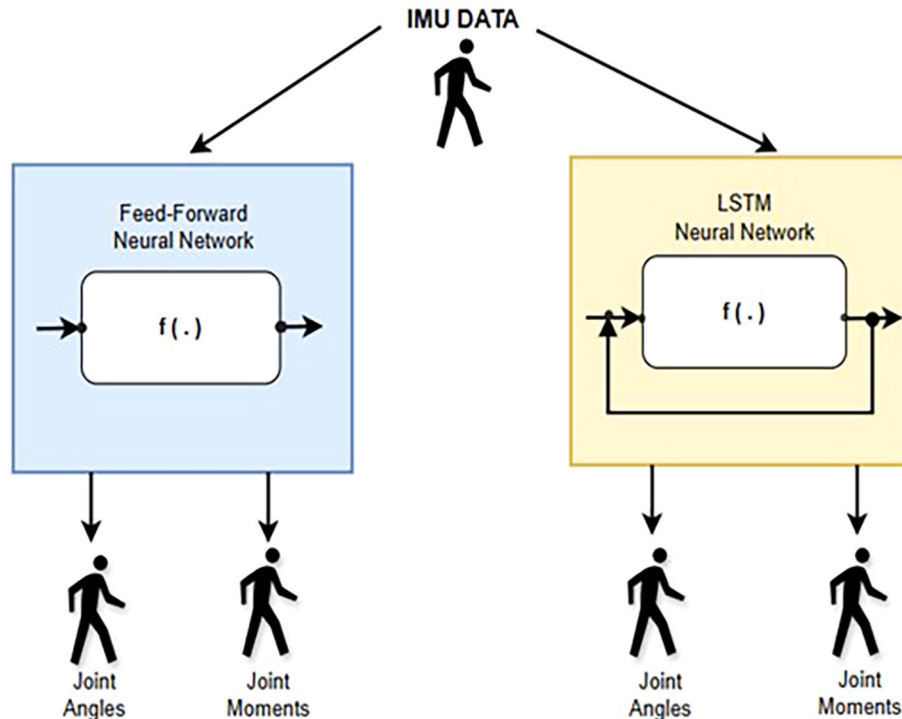


Figure 6. Joint angle accuracy using IMU.

2nd. LSTM neural network: Alternatively, the IMU data is fed into an LSTM neural network. LSTMs are specifically designed to handle sequential data like time series, allowing them to capture temporal dependencies in the data.

- c) **Output:** Both networks generate predictions for joint angles and moments. These represent the estimated positions and forces at the lower limb joints during gait.

4.1.2.5 Lighting and shadow realism

- i) **Light direction and intensity consistency:** Measure the accuracy of predicted lighting directions and intensities using photometric error metrics.³⁶
- ii) **Shadow accuracy:** Compare the predicted shadow patterns with ground truth shadows using metrics like Shadow Similarity Index.³⁷

In Figure 7 it is shown how ARShadowGAN-like training scheme in AI³⁷ generates realistic shadow (Lighting and Shadow Realism) in a picture works. This process ensures that the generated shadows blend seamlessly into the scene, enhancing visual realism.

- a) **Shadow-free image and mask:** Start with a shadow-free image (an image without any shadows) and a mask that highlights the object of interest.
- b) **Attention module:** The attention module analyzes the input and produces attention maps. These attention maps include a mask for neighboring objects and their shadows.

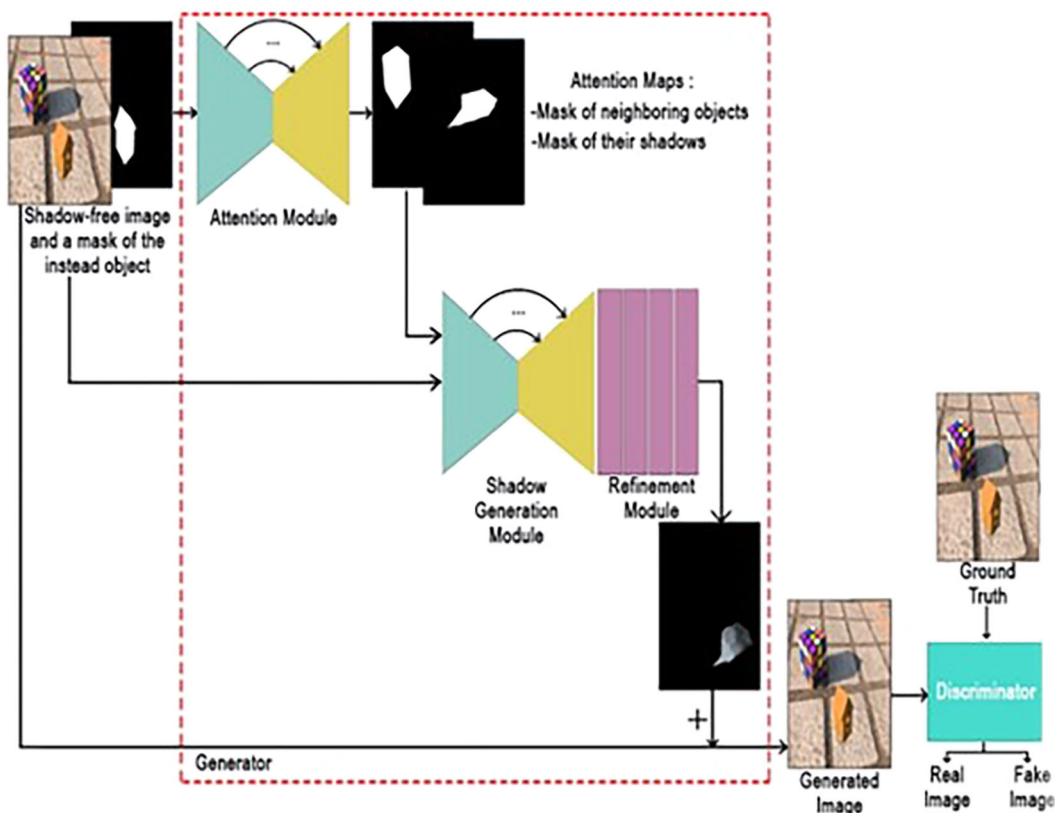


Figure 7. How ARShadowGAN-like training scheme in AI generates realistic shadow (Lighting and Shadow Realism) in a picture.

- c) **Shadow generation module:** Based on the attention maps, the shadow generation module creates a shadow for the object.
- d) **Refinement module ground truth:** The generated shadow undergoes further refinement to make it realistic.
- e) **Discriminator:** The discriminator compares the refined shadow with a real image to assess its authenticity.

4.2 Qualitative assessments

4.2.1 Expert evaluation

- i) **Human evaluators:** Have experts (e.g., artists, anatomists) assess the realism and accuracy of AI-generated hand drawings based on various criteria such as anatomical correctness, proportion, and movement.³⁸
- ii) **Visual turing test:** Evaluate if human observers can distinguish between AI-generated and real hand drawings.³⁹

4.2.2 User studies:

- i) **Perceptual studies:** Conduct studies with users to gather subjective feedback on the realism and accuracy of hand drawings.⁴⁰
- ii) **Preference tests:** Compare AI-generated hand drawings with human-drawn hands to see which one's users prefer.⁴¹

4.2.3 Comparative analysis:

- i) **Benchmarking against datasets:** Compare AI-generated hand drawings against established benchmarks and datasets (e.g., Human3.6M, RHD) to measure performance against known standards.⁴²
- ii) **A/B Testing:** Perform A/B tests with different versions of AI-generated hand drawings to determine improvements and preferences.⁴³

4.3 Tools and techniques

4.3.1 3D Hand models: Use 3D hand models and motion capture data to create accurate ground truth references for measuring AI performance. Employ 3D reconstruction techniques to compare predicted hand poses with 3D ground truth data.

4.3.2 Machine learning metrics: Utilize common machine learning metrics such as accuracy, precision, recall, F1 score, and area under the curve (AUC) for classification tasks related to hand gesture recognition.

4.3.3 Computer vision techniques: Implement computer vision algorithms for keypoint detection, segmentation, and texture analysis to evaluate the quality of AI-generated hand drawings.

4.4 Data collection and annotation

Collecting high-quality data on hand movements is a significant hurdle. Traditional motion capture systems can be cumbersome and expensive, while video-based methods often lack the necessary precision. Additionally, annotating hand movement data requires expert knowledge and can be time-consuming, leading to limited availability of large, annotated datasets that are essential for training AI models. Some detailed and specific breakdown of data collection and annotation for measuring the complexity and accuracy of AI-generated human hand drawings.

4.4.1 Data collection

4.4.1.1 Publicly available datasets

- i) **MPII+NZ Hand Pose Dataset:** Contains hand images with annotated keypoints and 3D poses.
- ii) **FreiHAND:** Includes color images, depth maps, and corresponding 3D hand models.

- iii) **Rendered Hand Pose Dataset (RHD):** Offers synthetic images of hands with keypoint annotations.
- iv) **CMU Panoptic Hand Dataset:** Provides multi-view images and 3D keypoints of hand poses.

4.4.1.2 Custom data collection

- i) **High-resolution imaging:** Capture images using high-resolution cameras to ensure detailed features of hands are recorded.
- ii) **Diverse subjects:** Include a variety of subjects with different hand shapes, sizes, skin tones, and ages to create a comprehensive dataset.
- iii) **Varied poses:** Ensure hands are captured in a wide range of poses, including open, closed, gripping objects, and interacting with other hands or objects.
- iv) **Lighting conditions:** Collect data under different lighting conditions to help the model learn how lighting affects hand appearance.

4.4.1.3 Extended data

Some resources can be adapted from OpenAI CLIP simple implementation that consists of CLIP models on Keras code from scratch in PyTorch.⁴⁴ Although OpenAI has open-sourced parts of CLIP, e.g., a dataset of OpenAI's CLIP model, VIT-LARGE-14-PATCH,⁴⁵ the code can be complex and overwhelming.

4.4.1.4 3D hand models

- i) **3D scanning:** Use 3D scanners like Artec Eva or Structure Sensor to capture high-resolution 3D models of hands in various poses.⁴⁶
- ii) **Synthetic data generation:** Create synthetic hand models using software like Blender or Unity. Apply different textures and poses to these models to augment the dataset.⁴⁷

4.4.2 Data annotation

4.4.2.1 Keypoint annotation

- i) **Manual annotation:** Manual Annotation includes tools like - Labelbox, VGG Image Annotator, or custom software and the process work by Annotating keypoints such as wrist, knuckles, and finger joints (21 keypoints: 4 per finger, 1 at the wrist). For example- Label the base, middle, and tip joints for each finger, and the wrist joint.
- ii) **Automated annotation tools:** Use pre-trained models like OpenPose to predict keypoints, then manually correct them for accuracy.⁴⁸

4.4.2.2 3D pose annotation

- i) **Motion capture:** Motion capturing can be achieved by systems like Vicon or OptiTrack. The process may work by the Record hand movements and generate 3D keypoints. Ensure accurate calibration for precise annotations.
- ii) **Multi-view stereo:** The Setup of multi view stereo Captures images from multiple angles using synchronized cameras. And the Reconstruction is done by Using stereo vision techniques to reconstruct 3D hand poses.

4.4.2.3 Surface detail and texture annotation: For manual Annotation tools or software like Adobe Photoshop or custom annotation tools are used. And the process follows by Annotation of fine details such as skin texture, wrinkles, and veins manually.

4.4.2.4 Shadow and lighting annotation is done by:

- **Tools:** Use software like Labelbox or custom annotation tools.
- **Process:** Annotate regions of shadows and light sources in the images.

4.4.3 Annotation tools and software are suggested as below:

- **Labeling software:** For this purpose, tools like Label Studio⁴⁹ can be used. Also VIA⁵⁰ and CVAT^{51,52} a good choice for analyzing as they have free source.
- **3D Modeling software:** Softwares like Maya⁵³ and Godot⁵⁴ are very adequate for 3D modeling and animation.
- **Motion capture systems:** OpenPose⁵⁴ for real-time human pose and key point detection for AI, VR/AR, and research, OpenMoCap⁵⁵ for motion capturing for 2D/3D tracking using cameras or video footage, or Kinovea⁵⁶ for simple 2D motion analysis for sports and rehabilitation satisfy for such systemic requirements.

4.4.4 Quality control is chosen on the basis of these two mentioned below:

- **Inter-annotator agreement process:** Have multiple annotators label the same data and calculate Cohen's Kappa to assess consistency.
- **Annotation validation process:** Review and correct annotations in a validation set by experts. Regularly update annotations to maintain high quality.

4.4.5 Data augmentation

- **Synthetic augmentation and transformations:** Apply rotations, scaling, translations, and color adjustments to existing images using libraries like OpenCV or imaging.
- **3D Augmentation and Software:** Use Blender⁵⁷ or Unity⁵⁸ to create new poses, textures, and lighting conditions for 3D hand models.

4.4.6 Documentation and metadata

- **Annotation guidelines and documentation:** Create detailed guidelines for annotators, specifying how to label key points, 3D poses, textures, and interactions. Include examples and edge cases.

4.4.7 Annotation guidelines

4.4.7.1 Keypoint annotation

- **Wrist:** The joint where the hand connects to the forearm.
- **Knuckles:** The joints at the base of each finger.
- **Finger joints:** Annotate the base, middle, and tip of each finger.

4.4.7.2 3D Pose annotation

- Use motion capture systems to record hand movements. Ensure accurate calibration for precise annotations.
- **Metadata:** Maintain metadata for each annotated image or 3D model, including information about the subject (e.g., age, gender), pose, lighting conditions, and annotation quality.

4.5 Real-time processing

Real-time processing in the context of AI-generated human hand drawings involves the rapid detection, analysis, and generation of hand images or movements. This is crucial for applications like virtual reality (VR), augmented reality (AR), and real-time interaction systems. Here's a detailed breakdown of the technical aspects involved in real-time processing:

4.5.1 Real-time hand detection and tracking

- i) Hand Detection is done by Object Detection Models such as,
 - **YOLO (You Only Look Once):** Efficient for real-time object detection, including hands.
 - **SSD (Single Shot MultiBox Detector):** Another real-time object detection framework.
- ii) Keypoint Detection is accomplished by Pose Estimation Models such as:
 - **MediaPipe hands:** A high-performance, real-time hand tracking solution by Google.
 - **OpenPose:** Multi-person key point detection including hand key points.

4.5.2 Real-time 3D pose estimation

- **Depth cameras:** Use depth cameras like Intel RealSense or Microsoft Kinect to capture depth information for 3D pose estimation.
- **Stereo vision:** Employ stereo cameras to calculate depth and reconstruct 3D hand poses.

4.5.3 Real-time Gesture Recognition with Gesture Classification Models: Use trained machine learning models to classify hand gestures in real-time. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks can be used for sequence prediction.

4.5.4 Real-time rendering and visualization

- **Graphics libraries:** Use OpenGL, DirectX, or Vulkan for rendering hand models and animations in real-time.
- **Game engines:** Unity⁵⁸ or Unreal Engine can be used for real-time rendering in VR/AR applications.

4.5.5 Real-time interaction and feedback

- **Haptic feedback:** Use haptic devices to provide real-time tactile feedback based on hand interactions.
- **Real-time collaboration:** Enable multiple users to interact with hand gestures in a shared virtual environment.

4.5.6 Performance optimization

- **Hardware acceleration:** Use GPUs or specialized hardware like NVIDIA Jetson for faster processing.
- **Model optimization:** Apply model compression techniques like quantization and pruning to reduce latency.
- **Parallel processing:** Use multi-threading or parallel processing to handle multiple tasks simultaneously.

5. Anatomical and biological considerations

5.1 Variability in hand shapes and sizes

Human hands vary widely in shape, size, and dexterity. AI models trained on a limited dataset may not generalize well to the diverse range of human hands. This variability necessitates the creation of more robust models that can accommodate different hand anatomies, which is a challenging task given the current state of AI technology.

5.1.1 Anatomical complexity on bone structure based 3D skeletal models: AI systems should use detailed 3D skeletal models that include the 27 bones of the hand, such as the carpals, metacarpals, and phalanges. These models can be created from 3D scans of real hands using tools like photogrammetry or depth sensors. This consists example of 3D scanning tools like Artec Eva to capture hand models and use software like Blender⁵⁷ or Autodesk Maya⁵³ to create and manipulate these models.

5.1.1.1 Joint flexibility based articulated hand models: Implement articulated hand models with kinematic chains to represent the finger joints. This allows each finger joint to move independently within physiological limits, using forward and inverse kinematics for realistic hand movements. This includes example of Using a physics engine like Bullet or PhysX to handle joint constraints and movements like- Load hand model URDF.

5.1.1.2 Musculature and tendons based muscle simulation: Simulate muscle contraction and tendon forces to produce realistic hand movements. This can be achieved with biomechanical models that translate muscle activations into joint torques. For example, of using the Finite Element Method (FEM) to simulate the deformation of muscles and tendons like system defining the mesh and function space for FEM simulation.

5.1.2 Biological diversity and dataset diversity comprehended data collection: Collect datasets with diverse hand shapes, sizes, and conditions, ensuring representation across age, gender, ethnicity, and hand condition. For example, the use of data augmentation techniques to artificially increase dataset diversity.

5.1.2.1 Adaptive algorithms influenced neural networks: Utilize models like Adaptive Resonance Theory (ART) or dynamic neural networks that can adjust to new data during inference. For example, the Implement online learning algorithms to continuously update the model with new hand data.

5.1.2.2 Hierarchical model based layered representation: Develop a hierarchical hand model where bones, joints, muscles, and skin are modelled separately but interactively. For example, the Use of a physics engine like PyBullet to simulate interactions between these layers like Simulate muscle contraction affecting joint angles

5.2 Sensory feedback and adaptation

Human hands rely heavily on sensory feedback from the environment to perform tasks. This feedback loop allows for continuous adjustment and adaptation, enabling precise control of hand movements. AI systems, however, lack this inherent sensory feedback mechanism, making it difficult for them to adapt to dynamic environments and perform tasks with the same level of precision as human hands.

5.2.1 Proprioception and tactile sensation for proprioception on sensor integration: Integrate sensors to capture hand position and movement data. Use IMUs (Inertial Measurement Units) and joint angle sensors. For example: Implement sensor fusion algorithms to combine data from multiple sensors for improved accuracy like - Kalman filter for sensor fusion.

5.2.1.1 Tactile sensation and haptic feedback: Use haptic devices to simulate tactile sensations. Devices like the Geomatic Touch provide force feedback to simulate touch. For example: Implement haptic rendering algorithms to convert virtual interactions into haptic feedback.

5.2.1.2 Real-time motor controlled algorithms: Use PID controllers or neural network-based controllers to adjust hand movements in real-time based on sensory feedback. Such proper example is of Implement PID control for precise hand movement adjustments.

5.2.2 Neural plasticity and learning and adaptive models: Develop models that can adapt to changes in sensory input over time. Use reinforcement learning or continual learning techniques to improve performance with experience. This consists example of Train agents using reinforcement learning to adapt to dynamic environments and varying sensory feedback.

5.2.2.1 Experience and learning (Supervised and unsupervised learning): Combine supervised learning for initial training with unsupervised learning to refine the model based on new data. For Example: Use self-supervised learning techniques to label data automatically and improve the model's performance without extensive manual labeling.

6. Practical implication and application

6.1 Robotics

6.1.1 Human-robot interaction: Grasping and Manipulation consists of Robots equipped with advanced AI hand models can perform complex tasks involving human-like dexterity, such as handling delicate objects, opening containers, or assembling intricate components. For example, a robot arm with an adaptive hand model can pick up various objects, from fragile glassware to irregularly shaped tools, by dynamically adjusting its grip based on real-time feedback. Another example is Load the robot model and configure the hand: Loading and configuring a robotic hand involves initializing models, calibrating sensors, and setting up real-time control and feedback mechanisms to ensure precise and adaptive functionality. This includes steps like: Load the Robot Model (Initialization, Hand Configuration), Sensor Calibration (Tactile and Position/Force Sensors), Real-time Feedback and Control (Control Algorithms, Feedback Loops), Object Interaction and Adaptation (Dynamic Grip Adjustment), Testing and Validation (Task Simulation).

6.1.2 Adaptive control systems: This includes properties like Real-time Adjustments. This includes Use real-time control algorithms and sensory feedback to adapt robot actions based on dynamic environments or varying object properties. For example- a robotic hand can use PID controllers to adjust its grip strength in response to changes in object texture or weight.

6.1.3 Collaborative robots: Collaboration with robots embodies Human-Robot Collaboration. This Be composed of Implementation robots that can work alongside humans, using AI-driven hand models to perform tasks that complement human abilities. For example- A collaborative robot (cobot) in a manufacturing line can assist human workers by handling heavy or repetitive tasks while adapting to the worker's movements and actions.

6.2 Virtual and augmented reality

6.2.1 Realistic interaction: This area is incorporated with phenomenon like Immersive Experiences. Here, AI-generated hand models in VR/AR can enhance user immersion by providing realistic and responsive interactions with virtual objects. For example- In VR, accurate hand models enable users to manipulate virtual objects with natural gestures and movements, improving the realism of the experience. Another example is Simulate hand interaction in a virtual environment

6.2.2 Haptic feedback: This Encompasses Enhanced Feedback. The use of AI to simulate tactile sensations in VR/AR environments, allowing users to feel textures, resistances, and forces. For example- Haptic gloves or controllers equipped with AI can provide feedback corresponding to virtual objects, enhancing the sense of touch and improving user interaction.

6.2.3 Training and simulation: This may include Skill Development. The use VR/AR for training scenarios that require precise hand movements or interactions, such as surgical simulations or mechanical repairs. For example- Surgeons can practice complex procedures in a virtual environment with realistic hand movements and haptic feedback, improving their skills without real-world consequences.

6.3 Healthcare and rehabilitation

6.3.1 Rehabilitation: This consists of Assistive Devices that includes AI-driven hand models which can be integrated into rehabilitation devices to assist patients in regaining hand function after injuries or surgeries. For example- Robotic exoskeletons with adaptive hand models can assist patients in performing exercises, adjusting the level of assistance based on real-time feedback from the patient. Load exoskeleton model and simulate rehabilitation exercises is the process where the involvement of initializing the exoskeleton model and simulating various rehabilitation exercises to evaluate and enhance the effectiveness of the exoskeleton in supporting patient recovery. This includes Load the Exoskeleton Model (Model Initialization, Configure the Model) and Simulate Rehabilitation Exercises (Defining Exercises, Run Simulations).

6.3.2 Prosthetics: This includes technology like Advanced Prosthetic Hands. This consists the Develop prosthetic hands that mimic the complexity of natural hand movements, offering improved functionality and user experience. For example, AI-driven prosthetics with adaptive hand models can provide more natural grasping and manipulation capabilities, allowing users to perform daily tasks with greater ease.

6.3.3 Diagnosis and monitoring: This forms concepts like Gesture Analysis. This includes the Use AI to analyse hand gestures and movements for diagnosing conditions or monitoring recovery progress. For example- AI systems can assess hand tremors or dexterity levels to help diagnose neurological conditions or track the effectiveness of rehabilitation interventions.

7. Conclusion

Due to several complex problems, AI finds it difficult to draw human hands, particularly when using models like CLIP. Human hands differ widely in size, shape, and movement, therefore obtaining a good approximation requires a huge and diverse training set. The intricate musculature, joints, and bones of the hands add to the difficulty of the work. Replicating movement and detail accurately requires complex models. Artificial intelligence (AI) models find it challenging to dynamically adjust to changing settings because they frequently lack the sensory feedback associated with human control. Even with recent improvements, real-time hand identification and keypoint tracking are still unable to fully capture the subtleties of hand interactions and motions. Applications like virtual and augmented reality, robotics, and healthcare all depend on high fidelity hand modeling. The current limitations of AI in this field highlight the need for continual advancements in data collection, model training, and real-time processing. These cutting-edge technologies could enhance AI's ability to more accurately and efficiently mimic and sketch human hands.

8. Future Scope

One of the biggest challenges in accurately modeling the human hand is capturing the body tissue. These improvements enable AI systems to better understand and reproduce simple movements and gestures, resulting in more precise and accurate gestures. Crowdsourcing platforms provide a low-cost, cost-effective way to collect this data and ensure that AI models can be integrated and transferred to diverse populations. Research on new neural network architectures such as Transformers and graph neural networks will provide powerful tools for modeling human muscles. These architectures improve vision and shape recognition, resulting in more accurate hand patterns. Training models on multiple data, including images, statistics, and motion, can provide a comprehensive understanding of hand movements, and the technology can improve the ability of artificial intelligence to repeat hand movements and daily behaviours. Developing real-time tracking and rendering software solutions can reduce downtime and improve the user experience in applications such as VR and AR. Develop highly optimized algorithms (low-latency algorithms) to analyze hand movements in real time to ensure that artificial intelligence systems can run smoothly on consumer devices. It could include additional tactile feedback, adaptive control systems, and more. Combined with an advanced tactile system, different emotions can be simulated, making the hand model generated by artificial intelligence more realistic and effective. These improvements can improve the user experience in virtual environments by providing an immersive and interactive experience. Developing control systems that adapt AI feedback based on real-time input will lead to more accurate manual interactions. This approach can improve the fidelity of hand movements generated by artificial intelligence and assistive applications such as medical simulation and prosthetic design. Analyzing the diversity of hand shape, size, and movement patterns across different populations will help create more inclusive and accurate AI models. For human-robot interactions, this can improve the grasping algorithms in robots to handle various objects of different shapes, sizes and textures, thus increasing the effectiveness of clever hand patterns. These avatars can provide users with a natural and immersive experience, making the virtual environment even more immersive. Using artificial intelligence to develop smart functions that can adapt to a user's unique movement patterns and provide sensory feedback improves performance and user experience. These prosthetics can provide natural and intelligent interactions and improve the lives of their users. The next innovation is the use of artificial intelligence to create interactive learning platforms that interpret hand movements to enable hands-on learning experiences in virtual environments. Other efforts include developing features to make interactive wearable technology more accessible to people with disabilities, which is important for inclusive design. This ability allows everyone to benefit from the advances in hand-sensing modeling, regardless of physical ability. Working with neuroscientists to better understand the brain's control of hand movements and incorporating these insights into artificial intelligence models can demonstrate the accuracy and precision of the hand. Developing cognitive models that simulate human cognitive processes related to hand movements can improve artificial intelligence to predict and repeat complex movements. These models provide a deeper understanding of human hand dynamics and improve AI performance. Other research suggests research and development of systems that allow humans and AI to adapt, learn from each other to improve manual interaction and control in over time, AI systems will become more intelligent and effective.

Ethics and consent

All data, figures, and diagrams used in this study were either generated by the author(s) or obtained from publicly available repositories on platforms such as Kaggle and GitHub.

The data used from these platforms are subject to the respective licensing terms provided by the original contributors. The author(s) confirm that:

- For data obtained from **Kaggle**, usage complied with the terms of the associated license specified by the dataset creator. Any restrictions or conditions set forth by the dataset provider have been respected.
- For code or resources obtained from **GitHub**, usage adhered to the terms of the repository's stated license (e.g., MIT License, Apache License, GPL). Proper credit has been provided to the original contributors where required.

No sensitive or personally identifiable information is included in the data. As the datasets and resources are publicly available and appropriately licensed, no additional ethical approval was required for their use in this study.

The author(s) affirm that all figures, diagrams, and outputs derived from these sources were created with due consideration of copyright, licensing, and usage rights. If requested, the detailed license information and attribution for any third-party data or code used can be provided.

No ethical approval needed as data used from online repository.

Data availability

O'Reilly: Deep Learning with TensorFlow and Keras - Third Edition <https://www.oreilly.com/library/view/deep-learning-with/9781803232911/>

OpenAI: CLIP: Connecting text and images <https://openai.com/index/clip/>

For code or resources obtained from GitHub, <https://github.com/openai/CLIP>, usage adhered to the terms of the repository's stated license MIT License

Some resources can be adapted from OpenAI CLIP simple implementation that consists of CLIP models on Keras code from scratch in PyTorch.⁴⁴ Although OpenAI has open-sourced parts of CLIP, e.g.- a dataset of OpenAI's CLIP model, VIT-LARGE-14-PATCH,⁴⁵ the code can be complex and overwhelming.^[<https://github.com/moein-shariatnia/OpenAI-CLIP>], MIT License

We have not used any or npor generated any extended data the reference link is as follows: <https://github.com/moein-shariatnia/OpenAI-CLIP>

References

1. Keyes OK, Hyland A: **Hands are hard: unlearning how we talk about machine learning in the arts.** *Tradition Innovations in Arts, Design, and Media Higher Education*. 2023; **1**(1): 4. [Publisher Full Text](#)
2. Rajaratnam V, May STS, Jerome JTJ: **Precision at hand: Revolutionising surgery with generative AI.** *J. Hand Microsurg*. 2024; **16**: 100090. [Publisher Full Text](#)
3. Salman S, Zakir A, Takahashi H: **Cascaded deep graphical convolutional neural network for 2D hand pose estimation.** 2023; **12592**: pp. 1259215–1259215-6.
4. Schneider J: **Optimizing human hand gestures for AI-systems.** *AI Commun*. 2022; **35**: 153–169. [Publisher Full Text](#)
5. Salisbury K: **Issues in human/computer control of dexterous remote hands.** *IEEE Trans. Aerosp. Electron. Syst*. 1988; **24**: 591–596. [Publisher Full Text](#)
6. Wheatland N, Wang Y, Song H, et al.: **State of the Art in Hand and Finger Modeling and Animation.** *Computer Graphics Forum*; 2015; **34**.
7. Gao Q, Liu J, Ju Z: **Robust real-time hand detection and localization for space human-robot interaction based on deep learning.** *Neurocomputing*. 2020; **390**: 198–206. [Publisher Full Text](#)
8. Bhunia A, Das A, Muhammad U, et al.: **Pixelor: a competitive sketching AI agent. so you think you can sketch?** *ACM Trans. Graph*. 2020; **39**: 1–166:15. [Publisher Full Text](#)
9. Gioioso G, Salvietti G, Malvezzi M, et al.: **Mapping Synergies From Human to Robotic Hands With Dissimilar Kinematics: An Approach in the Object Domain.** *IEEE Trans. Robot*. 2013; **29**: 825–837. [Publisher Full Text](#)
10. Rijpkema H, Girard M: **Computer animation of knowledge-based human grasping. Proceedings of the 18th annual conference on Computer graphics and interactive techniques.** 1991.
11. Wu J: **Research on rapid generation of 3D models based on art and design cognitive models.** *SHS Web of Conferences*. 2023.
12. Lu Z, Zhou P: **Hands-Free Human-Computer Interface Based on Facial Myoelectric Pattern Recognition.** *Front. Neurol*. 2019; **10**. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
13. Ficuciello F: **Synergy-Based Control of Underactuated Anthropomorphic Hands.** *IEEE Trans. Industr. Inform*. 2019; **15**: 1144–1152. [Publisher Full Text](#)
14. Coeckelbergh M: **Artificial Intelligence, Responsibility Attribution, and a Relational Justification of Explainability.** *Sci. Eng. Ethics*. 2019; **26**: 2051–2068. [Publisher Full Text](#)
15. Guo L, Lu Z, Yao L: **Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review.** *IEEE Trans. Hum. Mach. Syst*. 2021; **51**: 300–309. [Publisher Full Text](#)
16. Huang X, Chen H, Chen L: **An AI Edge Computing-Based Intelligent Hand Painting Teaching System.** *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*. 2022; pp. 942–943.
17. Battaglia P, Hamrick J, Bapst V, et al.: **Relational inductive biases, deep learning, and graph networks.** *ArXiv, abs/1806.01261*. 2018.
18. Mattar E: **A survey of bio-inspired robotics hands implementation: New directions in dexterous manipulation.** *Robot. Auton. Syst*. 2013; **61**(5): 517–544. [Publisher Full Text](#)
19. Nguyen CC, Wong S, Thai MT, et al.: **Advanced user interfaces for teleoperated surgical robotic systems.** *Adv. Sensor Res*. 2023; **2**(4): 2200036. [Publisher Full Text](#)
20. Ghosh P, Hazra S, Chatterjee S: **Future Prospects Analysis in Healthcare Management Using Machine Learning Algorithms.** *Int. J. Eng. Sci. Invention (IJESI)*. 2019; **23**: 6734.
21. Carfi A, Mastrogiovanni F: **Gesture-based human-machine interaction: Taxonomy, problem definition, and analysis.** *IEEE Trans. Cybern*. 2021; **53**(1): 497–513. [Publisher Full Text](#)
22. Nyatsanga S, Kucherenko T, Ahuja C, et al.: **A Comprehensive Review of Data-Driven Co-Speech Gesture Generation.** *Computer*

- Graphics Forum. 2023, May; **42**(2): 569–596.
[Publisher Full Text](#)
23. Schmitz C, Rösch C, Zingsheim D, et al.: **Interactive pose and shape editing with simple sketches from different viewing angles.** *Comput. Graph.* 2023; **114**: 347–356.
[Publisher Full Text](#)
 24. Balaji AN, Peh LS: **AI-On-Skin: Towards Enabling Fast and Scalable On-body AI Inference for Wearable On-Skin Interfaces.** *Proc. ACM Hum.-Comput. Interact.* 2023; **7**(EICS): 1–34.
[Publisher Full Text](#)
 25. Deng X, Zhang Y, Shi J, et al.: **Hand pose understanding with large-scale photo-realistic rendering dataset.** *IEEE Trans. Image Process.* 2021; **30**: 4275–4290.
[PubMed Abstract](#) | [Publisher Full Text](#)
 26. Chen MK, Liu X, Sun Y, et al.: **Artificial intelligence in meta-optics.** *Chem. Rev.* 2022; **122**(19): 15356–15413.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
 27. Chormai P, Pu Y, Hu H, et al.: **Machine learning of large-scale multimodal brain imaging data reveals neural correlates of hand preference.** *NeuroImage.* 2022; **262**: 119534.
[PubMed Abstract](#) | [Publisher Full Text](#)
 28. Mathew SP: **Predicting Functional Use of the Non-Dominant Hand using Machine Learning and Wearable Accelerometers.** Canada: University of Toronto; 2022. (Master's thesis).
 29. Hampali S, Rad M, Oberweger M, et al.: **Honnotate: A method for 3d annotation of hand and object poses.** *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020; pp. 3196–3206.
 30. Chevtchenko SF, Vale RF, Macario V, et al.: **A convolutional neural network with feature fusion for real-time hand posture recognition.** *Appl. Soft. Comput.* 2018; **73**: 748–766.
[Publisher Full Text](#)
 31. Deng J, Lin C, Zhao Z, et al.: **A Survey of Defenses against AI-generated Visual Media: Detection, Disruption, and Authentication.** *arXiv preprint arXiv:2407.10575.* 2024.
[Publisher Full Text](#)
 32. Naji S, Jalab HA, Kareem SA: **A survey on skin detection in colored images.** *Artif. Intell. Rev.* 2019; **52**: 1041–1087.
[Publisher Full Text](#)
 33. Hazra S: **Review on Social and Ethical Concerns of Generative AI and IoT.** Raza K, Ahmad N, Singh D, editors. *Generative AI: Current Trends and Applications. Studies in Computational Intelligence.* Vol 1177. Singapore: Springer; 2024.
[Publisher Full Text](#)
 34. Thien NP, Dang CN, Trinh TT, et al.: **Machine Learning for Enhanced Exercise Performance and Planning.** *International Conference on Future Data and Security Engineering.* Singapore: Springer Nature Singapore; 2024, November; pp. 249–263.
 35. Mundt M, Thomsen W, Witter T, et al.: **Prediction of lower limb joint angles and moments during gait using artificial neural networks.** *Med. Biol. Eng. Comput.* 2020; **58**: 211–225.
[PubMed Abstract](#) | [Publisher Full Text](#)
 36. Kruisselbrink T, Dangol R, Rosemann A: **Photometric measurements of lighting quality: An overview.** *Build. Environ.* 2018; **138**: 42–52.
[Publisher Full Text](#)
 37. Kruisselbrink T, Dangol R, Rosemann A: **Photometric measurements of lighting quality: An overview.** *Build. Environ.* 2018; **138**: 42–52.
[Publisher Full Text](#)
 38. Sai S, Gaur A, Sai R, et al.: **Generative ai for transformative healthcare: A comprehensive study of emerging models, applications, case studies and limitations.** *IEEE Access;* 2024.
 39. Pennanen N, Linkola S, Kantosalo A, et al.: **From Product to Producer: The Impact of Perceptual Evidence and Machine Embodiment on the Human Assessment of AI Creativity.** *PsyArXiv.* 2023 October, 6.
 40. Dey A, Billinghurst M, Lindeman RW, et al.: **A systematic review of 10 years of augmented reality usability studies: 2005 to 2014.** *Front. Robot. AI.* 2018; **5**: 37.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
 41. Lin L, Gupta N, Zhang Y, et al.: **Detecting multimedia generated by large ai models: A survey.** *arXiv preprint arXiv:2402.00045.* 2024.
 42. Lee J, Kim J, Kim SH, et al.: **Enhancing 3D hand pose estimation using SHaF: synthetic hand dataset including a forearm.** *Appl. Intell.* 2024; **54**: 9565–9578.
[Publisher Full Text](#)
 43. Wang J, Duan H, Zhai G, et al.: **Understanding and Evaluating Human Preferences for AI Generated Images with Instruction Tuning.** *arXiv preprint arXiv:2405.07346.* 2024.
 44. Kapoor A, Gulli A, Pal S, et al.: **Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models.** Packt Publishing Ltd.; 2022.
 45. Chen J, Yu Q, Shen X, et al.: **ViTamin: Designing Scalable Vision Models in the Vision-Language Era.** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2024; pp. 12954–12966.
 46. Redaelli DF, Barsanti SG, Biffi E, et al.: **Comparison of geometrical accuracy of active devices for 3D orthopaedic reconstructions.** *Int. J. Adv. Manuf. Technol.* 2021; **114**(1): 319–342.
[Publisher Full Text](#)
 47. Deng X, Zhang Y, Shi J, et al.: **Hand pose understanding with large-scale photo-realistic rendering dataset.** *IEEE Trans. Image Process.* 2021; **30**: 4275–4290.
[PubMed Abstract](#) | [Publisher Full Text](#)
 48. Jin S, Xu L, Xu J, et al.: **Whole-body human pose estimation in the wild.** *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16.* Springer International Publishing; 2020; pp. 196–214.
 49. Sharma G, Angleraud A, Pieters R: **Multi-label Annotation for Visual Multi-Task Learning Models.** *2023 Seventh IEEE International Conference on Robotic Computing (IRC).* IEEE; 2023, December; pp. 31–34.
 50. Prakiljačić S, Grbić R, Vranješ M, et al.: **Tool for image annotation in context of modern object detection.** *2024 Zooming Innovation in Consumer Technologies Conference (ZINC).* IEEE; 2024, May; pp. 48–53.
 51. Hansen US, Landau E, Patel M, et al.: **Novel artificial intelligence-driven software significantly shortens the time required for annotation in computer vision projects.** *Endosc. Int. Open.* 2021; **09**(04): E621–E626.
[Publisher Full Text](#)
 52. Hejazi P, Padte AK, Golazizian P, et al.: **CVAT-BWW: A Web-Based Video Annotation Platform for Police Body-Worn Video.** *International Joint Conferences on Artificial Intelligence Organization.* 2024, August.
 53. Wu WC, Lee MH: **The Influence of Applying Digital Toolkits to Assist 3D Software Development on Individual Work Performance and Perceived Stress.** *2024 10th International Conference on Applied System Innovation (ICASI).* IEEE; 2024, April; pp. 37–39.
 54. Malý J: **Real-time strategy videogame toolkit for Godot Engine.** 2024.
 55. Hachaj T, Ogiela MR: **RMoCap: an R language package for processing and kinematic analyzing motion capture data.** *Multimed. Syst.* 2020; **26**(2): 157–172.
[Publisher Full Text](#)
 56. Gupta S, Hazra S, Hazra S, et al.: **Mathematical Models of Heterogeneous Machine Learning Techniques for Ransomware Protection in Cyber-Physical Systems.** *2024 IEEE International Conference on Communication, Computing and Signal Processing (IICCCS).* India: ASANSOL; 2024; pp. 1–5.
[Publisher Full Text](#)
 57. Villar O: *Learning Blender.* Addison-Wesley Professional; 2021.
 58. Jeong-Shick Y: **Unity: A Powerful Tool for 3D Computer Animation Production.** *J. Korea Comput. Graphics Soc.* 2023; **29**(3): 45–57.
[Publisher Full Text](#)

Open Peer Review

Current Peer Review Status: 

Version 1

Reviewer Report 07 March 2025

<https://doi.org/10.5256/f1000research.175451.r367392>

© 2025 Cheng M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Mingyong Cheng 

University of California San Diego, San Diego, California, USA

This paper lacks scientific rigor, empirical evidence, and a coherent research structure. While it claims to analyze CLIP's ability to generate hands, it fails to present any dataset, experimental setup, or meaningful evaluation. Instead of offering concrete findings, it reiterates well-known limitations of AI-generated hands without substantiating them with data. The methodology section is vague and reads more like a generic overview of CLIP rather than a structured research process. Key elements such as dataset selection, baseline comparisons, control variables, and evaluation metrics are either absent or superficially mentioned without real application. The *Related Work* section is an unstructured compilation of papers, many of which are unrelated to the topic, with no synthesis of how they inform the study. Additionally, the writing is redundant, with concepts appearing multiple times in different sections without adding depth. The paper anthropomorphizes AI in an unhelpful manner, using phrases like "AI is like a human trapped in a museum," which add little technical value. The conclusion and future scope sections propose vague solutions without demonstrating how they could be implemented or validated.

Is the work clearly and accurately presented and does it cite the current literature?

Partly

Is the study design appropriate and is the work technically sound?

No

Are sufficient details of methods and analysis provided to allow replication by others?

No

If applicable, is the statistical analysis and its interpretation appropriate?

No

Are all the source data underlying the results available to ensure full reproducibility?

No

Are the conclusions drawn adequately supported by the results?

No

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Creative AI Application, Generative AI & Art

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research