

VisionClarity

Mitigating Object Hallucination for
Accurate Product Descriptions and Reliable Visual Search
([GithubLink](#))

Project Overview

Project Title:

- **VisionClarity-** Mitigating Object Hallucination for Accurate Product Descriptions and Reliable Visual Search

Project Overview:

Objective:

The VisionClarity project aims to create a robust system that focuses on mitigating object hallucinations to ensure accurate product descriptions and reliable visual search. Object hallucination occurs when a model mistakenly identifies or describes objects that are irrelevant or absent from an image. VisionClarity addresses this challenge through two primary goals:

1. **Generate Accurate Product Descriptions:** VisionClarity ensures that product descriptions align closely with actual product images, minimizing errors caused by object hallucination, where inaccurate details might be described.
2. **Provide Reliable Visual Search Results:** The system supports precise visual search by analyzing product images in depth, capturing relevant features to ensure correct matching and reduce the likelihood of hallucinated details.

These improvements aim to reduce mismatched product information, reduce product returns, enhance user satisfaction and lower the operational costs in e-commerce,retailers, and customer service.

Scope:

The project focuses on minimizing object hallucinations in product descriptions and visual search results, ensuring the model outputs are grounded in real, visible data. VisionClarity will use datasets that combine product images with verified descriptions, maintaining consistency between visual and textual data.

Challenges in this scope include managing ambiguous images, where hallucinations are more likely, and refining the model's ability to avoid unnecessary or misleading details in both descriptions and visual search results.

- **AI Techniques and Tools:**

The VisionClarity system will be built using a combination of multimodal fusion techniques and object recognition algorithms, specifically focusing on:

1. **Deep Learning Models for Product Descriptions and Image Analysis:**

- **Image Processing:** Convolutional Neural Networks (CNNs) and Vision Transformers handle image details, detecting relevant features (e.g., color, texture) that ensure accuracy in product representation while avoiding hallucinations.
- **Language Understanding:** Language transformers manage text generation to ensure that descriptions are grounded in the actual visual data, aligning product descriptions closely with the image content.

2. **Multimodal LLM Models (e.g., BLIP or CLIP):**

- **CLIP(Contrastive Language-Image Pre-training):** This model's strength lies in its contrastive learning approach, which helps the model to distinguish between relevant and irrelevant information. This capability should help minimize incorrect descriptions that don't match the visual content.

3. **Adversarial Training and Minimum Risk Training:**

- **Generative Adversarial Networks (GANs)** is used to expose the model to challenging, misleading examples. GANs generate realistic yet deceptive images that simulate difficult scenarios, such as minor variations in product details (e.g., color shifts, shape alterations) or subtle additions that could lead to object hallucinations.
- **How It Works:** The GAN model creates these adversarial samples, which the primary model (VisionClarity) then learns to identify and avoid hallucinations. By continuously training with these "tricky" images, VisionClarity improves its ability to resist producing hallucinated descriptions or misinterpreting visual elements. This process strengthens the model, making it more resilient, precise, and effective at generating accurate descriptions and reliable visual search results.

Key Tools and Libraries

- **PyTorch:**

I'll be using PyTorch because it's flexible and user-friendly.

- **OpenCV:**

OpenCV is essential for image processing tasks.

- **Hugging Face Transformers:**

This library is useful for fine-tuning models for my hallucination mitigating tasks, where I can boost the model's language understanding, making its descriptions much more accurate.

Stakeholders:

Project Team:

Team Member 1:

Role: Frontend Developer

Responsibilities:- UI/UX design, data visualization and testing

Team Member 2:

Role: Backend Developer

Responsibilities:-

Database Management, Model development & Backend Optimization & Scalability

End Users:

E-commerce platform users: Customers who use visual search to find specific products on ecommerce platforms benefit from VisionClarity's precise image matching. Accurate descriptions and visual search results ensure they find products that match their preferences, improving their shopping experience and reducing returns due to mismatched items.

Retailers and Merchandisers: Use the system to automate and validate product descriptions, ensuring that the information is accurate and aligned with the product visuals. This might lead to reducing customer complaints due to product misrepresentations.

Customer Service Representatives: These agents rely on VisionClarity's accurate product descriptions to assist with customer inquiries, troubleshooting, and recommendations. Reliable image-to-text descriptions and visual search results help them respond accurately, reducing mismatches and improving customer satisfaction.

Other Stakeholders:

- Industry experts or regulators ensuring the AI complies with ethical guidelines.
- Government agencies or organizations overseeing AI development and deployment.

Computing Infrastructure

1. Project Needs Assessment

- **Primary Objective:**

- The goal of VisionClarity is to mitigate object hallucinations in multimodal LLM models by ensuring the accurate generation of product descriptions that align closely with actual product images. The system will facilitate image-to-text conversion and reliable image generation, guaranteeing that descriptions are grounded in the visual data. This approach minimizes errors and avoids irrelevant or incorrect object descriptions.

- **Tasks:**

- Image classification for object identification.
- Natural Language Processing (NLP) for generating text descriptions from visual inputs.
- Object hallucination detection in text outputs.
- **Reliable Image Generation:** Design processes for generating high-quality product images that reflect the attributes and characteristics of the described items, ensuring visual fidelity in output.

- **DataTypes:**

- Images: Visual data from multimedia datasets.
- Text: Descriptions paired with corresponding images to train the model.
- Multimodal fusion: Combining both text and image for training and output generation.

- **Performance Benchmarks:**

- **Latency:** The system should generate real-time image descriptions with minimal delay. Given that the focus is on educational institutions and medical fields, it is crucial to ensure that the delay between image input and text generation is negligible, ideally within **50 to 100 ms**.
- **Throughput:** The system must handle multiple concurrent image descriptions. For example, in educational settings, multiple images might be uploaded at once (for course materials), and the system should generate descriptions for these images simultaneously. A reasonable throughput goal would be processing **20 to 30 images per second in a batch**, allowing the system to scale to larger workloads.
- **Accuracy:** The model should produce highly accurate image descriptions and produce reliable relevant search, aiming for **minimal object hallucinations**. This should be benchmarked against existing models such as **CLIP**. The goal is to have a high accuracy score, ensuring descriptions are as faithful as possible without introducing non-existent objects.

Deployment Constraints:

- **Environment:** Primarily cloud-based for scalability, though it should also be capable of running on local hardware for smaller-scale tests.
- **Power and Network Conditions:** Efficient enough to run on standard cloud instances but optimized for environments with varying network conditions.
- **Resources:**

For my project, we referred to the following journals and research papers to further understand the scope of the project, evaluate the performance of existing models concerning object hallucinations, and assist with data collection:

- **Papers with Code:**
 - [MITIGATING HALLUCINATION IN LARGE MULTIMODAL MODELS VIA ROBUST INSTRUCTION TUNING](#) | [code](#)
 - [Object Hallucination in Image Captioning](#) | [code](#)
 - [Visual Hallucinations of Multi-modal Large Language Models](#) | [code](#)
 - [Hallu-PI: Evaluating Hallucination in Multi-modal Large Language Models within Perturbed Inputs](#) | [code](#)
- [Mitigating Object Hallucination via Data Augmented Contrastive Tuning](#)

2. Hardware Requirements Planning

- **GPUs:** Focus on A100 or V100 GPUs from NVIDIA for training due to their efficiency in handling large multimodal datasets.
- **Hardware Specifications:**
 - GPU/CPU Needs: Training will require multiple GPUs for distributed processing. Inference can be handled by a single GPU instance.
 - Memory: 64GB to 128GB of RAM is recommended to handle large datasets and training tasks.
 - Storage: SSDs storage for faster data throughput and efficient handling of multimedia datasets.
- **Resources:**
 - Leverage the High-Performance Computing resources offered by the University of Florida's HiPerGator.

3. Software Environment Planning

- **Operating system:** Windows Server
- **Software stack:**
 - **Frameworks:**
 - **PyTorch** for building and training deep learning models, Hugging Face Transformers for multimodal model access and fine-tuning.
 - **Version Control and Collaboration:** Tools like Git and GitHub can be used to manage code versions and support collaboration.
 - **Libraries:**
 - **Pandas, NumPy:** For data processing.
 - **OpenCV:** For handling image preprocessing tasks.
 - **Hugging Face's Transformers library** :will provide access to multiple multimodal LLMs, such as BLIP or CLIP.
 - **Virtualization: Docker** for containerizing and scaling workloads during development and deployment.

- **Resources:**

- **Microsoft Docs for Windows Server:** Detailed guides on setting up AI environments on Windows.
 - [Microsoft Docs](#)
- **Official TensorFlow and PyTorch Sites:** Installation guides, tutorials, and performance tuning tips.
 - [PyTorch](#)
- **Containerization Tools:**
 - **Docker Documentation:** Comprehensive guides for setting up Docker containers for AI applications.
 - [Docker Docs](#)

4. Cloud Resources Planning

- Google Cloud offers GPU instances (e.g., A100, V100) for high-performance training tasks.
- **AI Services:** Google AI Platform could simplify the process of training and deploying models at scale, reducing infrastructure management overhead, especially with Pytorch support.
- **Data Storage:** Google Cloud Storage provides scalable, secure storage for large multimodal datasets. These services also support rapid access to data during model training and testing.
- **Cost Considerations:** Cloud pricing models (e.g., pay-as-you-go) should be evaluated based on the frequency and scale of training/inference.
- **Resources**
 - **Google AI Platform:** Managed services for training, tuning, and deploying AI models.
 - [Google AI Platform](#)
 - **Google Cloud Pricing Calculator:** Provides cost estimates for GCP resources and services.
 - [GCP Pricing Calculator](#)

5. Scalability, and Performance Planning

- **Scaling resources dynamically based on workload demand:**

- **Cloud Auto-Scaling:** For my project I require cloud auto-scaling services to handle unpredictable spikes in demand without manual intervention.
- **Performance optimization techniques :**
 - Model Pruning: Reduce the size of the multimodal models without losing significant accuracy, which can enhance both performance and inference time.
 - Grafana can monitor system performance, track model accuracy, and ensure that the infrastructure scales effectively under different loads.
 - We can use Profiling tools as well to optimize the GPU and CPU performance
- **Resources:**
 - **Grafana:** Enables to monitor system performance
 - [Download Instruction](#) | [Grafana Documentation](#)
 - **NVIDIA Nsight Systems:** Profiling tools to optimize GPU and CPU performance.
 - [NVIDIA Nsight](#)

Security, Privacy, and Ethics (Trustworthiness)

1. Problem Definition

- **Goal:** Define the ethical and societal impacts of mitigating object hallucinations in multimodal models.
- **Strategies:**
 - **Stakeholder Involvement:**
 - Conduct a series of semi-structured interviews with educators, content creators, and healthcare professionals who use the system.
 - Organize groups with potential end-users to gather diverse perspectives on how object hallucination might impact their work or decision-making.
 - Create an online survey to reach a broader audience and collect quantitative data on concerns and expectations.
 - **Ethical Impact Assessments:**
 - Utilize the IEEE Ethically Aligned Design framework to systematically evaluate the ethical implications of VisionClarity.
 - Apply the EU's Ethics Guidelines for Trustworthy AI to ensure the project aligns with principles of human agency, fairness, and transparency.
 - Conduct a thorough Privacy Impact Assessment to identify and mitigate potential privacy risks associated with processing visual data.
 - **Risk Analysis Frameworks:**
 - Implement the NIST AI Risk Management Framework to identify, assess, and manage risks throughout the AI system's life cycle.
 - Use the AI Fairness 360 toolkit to assess and mitigate bias in the model's outputs, especially concerning diverse representation in image descriptions.
 - Apply the MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems) framework to evaluate security risks specific to systems.
- **Tools and Approaches:**
 - **AI Blindspot:** This toolkit helps in identifying potential blind spots related to hallucinations in multimodal models and encourages mitigation strategies.

- **AI Explainability 360 toolkit** :to enhance the transparency of the model's decision-making process, particularly in cases where hallucinations are detected and mitigated.
- **Deon (an ethics checklist for data scientists)**: to ensure ethical considerations are integrated throughout the development process.

2. Data Collection

- **Goal:** Gather high-quality, representative, and unbiased data to support reliable AI model development for the VisionClarity project.
- **Strategies:**
 - **Multi-Source Data Integration:**
 - Combine existing datasets to create a more comprehensive and diverse dataset.
 - Develop data integration pipelines to harmonize annotations and metadata across sources and Implement cross-validation techniques to ensure consistency and quality across integrated data.
 - **Synthetic Data Generation and Augmentation:**
 - Use advanced image generation models to create synthetic images, especially for underrepresented scenarios.
 - Implement text augmentation techniques to diversify image descriptions.
 - **Adversarial Example Creation:**
 - Generate challenging examples that are prone to hallucinations, using techniques like adversarial attacks on image-to-text models.
 - **Human-in-the-Loop Verification:**
 - Employ expert annotators to verify and refine image descriptions, especially for complex or ambiguous images.
 - **Privacy-Preserving Data Collection:**
 - Implement differential privacy techniques when dealing with sensitive or personal images.
- **Tools and Libraries:**
 - **Libraries:-**

- **Weights & Biases (wandb):**

- For experiment tracking, dataset versioning, and visualizing model performance on different subsets of data.

- **ModAL** - An active learning framework for Python.

- **Approach:** Implement uncertainty sampling to identify images where the model's predictions are least confident.

- **Tools:-**

- **NLTK:** For text processing and augmentation of image descriptions.

- **Faker:** To generate synthetic metadata if needed.

- **Albumentation:** For image augmentation tasks

- **Technical Focus:** Utilize **ModAL** to use it to identify the most uncertain images from an unlabeled pool. These images would then be prime candidates for human annotation, helping to improve the model's performance in areas where it's least confident, potentially reducing hallucinations in image descriptions.

3. AI Model Development

- **Goal:** Develop fair, interpretable, and robust AI models that perform reliably across diverse scenarios, focusing on mitigating object hallucinations in image-to-text descriptions.

- **Strategies:**

- **Ensemble Learning:**

- Combine multiple models to leverage the system's strengths and mitigate individual weaknesses.

- Use boosting techniques to improve overall model robustness.

- **Explainable AI Techniques:**

- Integrate methods like LIME or SHAP to provide explanations for model predictions.

- Implement attention visualization techniques to show which parts of the image the model focuses on for each generated word.

- **Attention Mechanisms:**

- Incorporate advanced attention mechanisms to help the model focus on relevant image features when generating descriptions.
 - Use cross-attention between visual and textual features to enhance alignment
- **Tools and Libraries:**
 - **SHAP (SHapley Additive exPlanations):**
 - A game theoretic approach to explain the output of any machine learning model.
 - Helpful in identifying which image features contribute most to certain descriptions.
 - **Robust-ML:**
 - A library for training models that are robust to adversarial attacks. Useful in implementing adversarial training techniques.
 - **Fairlearn:** A Python library for assessing and mitigating unfairness in multimodal models.
 - **Tools:-**
 - **AIF360 (AI Fairness 360):** A comprehensive toolkit developed by IBM that provides metrics to check for biases in datasets and models, and includes algorithms to mitigate bias.
 - **InterpretML:** A toolkit for creating human-interpretable models and explanations.
- **Technical Focus:**
 - Integrate **SHAP** values to explain which image features contribute most to each generated word in the description.
 - Implement attention visualization, showing heatmaps of where the model focuses when generating each word.

4. AI Deployment

- **Goal:** Deploy the VisionClarity model, ensuring real-world reliability and accountability
- **Strategies:**
 - **Secure Model Serving:**

- When deploying the VisionClarity model, need to ensure that the model is served securely. Use secure frameworks that manage API access control, authentication, and encryption to prevent unauthorized access to the model's predictions and outputs.
 - This will protect the system from potential adversarial inputs or attacks that could cause the model to generate hallucinated or inaccurate image descriptions.
- **Continuous Integration/Continuous Deployment (CI/CD):**
 - Set up CI/CD pipelines to ensure that model updates are automatically tested before deployment. This helps in monitoring model drift, allowing updates if the model begins generating hallucinations or biased outputs after deployment.
 - Automating these processes also ensures consistency, reduces human error, and improves the overall reliability of the model after deployment.
- **Robustness Testing in Real-World Conditions:**
 - Test the model in different environments and conditions before deployment to ensure robustness. This involves deploying the model in both controlled environments (where data is clean and representative) and in noisy, real-world scenarios where ambiguous or low-quality images may be fed to the system.
 - By stress-testing the model before full-scale deployment, potential issues with object hallucination or incorrect descriptions can be identified early
- **User Feedback Loops:**
 - Implement a feedback loop mechanism where users can provide input on incorrect or hallucinated descriptions. This feedback can help fine-tune the model post-deployment and improve the system's accuracy over time.
 - Implement anomaly detection, where flagged descriptions are automatically analyzed for discrepancies between the image content and generated text.
- **Tools and Libraries:**
 - **Alibi:** Alibi is an open-source library that provides model explanations in production, allowing for more accountability during deployment. This tool can help explain why certain hallucinated outputs were generated and provide transparency to users and stakeholders.
 - **BentoML:** A flexible, high-performance framework for serving, deploying, and monitoring machine learning models in production. BentoML also provides APIs for real-time model feedback, which can be critical for gathering user input on hallucinated descriptions.

- **ONNX Runtime:** can be used to optimize the deployed VisionClarity model across different platforms. It provides efficient inference, reducing latency and computational costs, while ensuring that performance remains consistent across various environments.
- **MLflow:** MLflow is a tool to manage the lifecycle of machine learning models, including deployment, monitoring, and tracking metrics. It supports versioning and can trigger automatic rollbacks if the model begins generating inaccurate or hallucinated outputs. Additionally, MLflow integrates well with CI/CD pipelines.
- **Technical Focus:** In VisionClarity, the deployment pipeline can use **BentoML** for serving the model while ensuring security and scalability. Also, integrate **Alibi** for explainability, so stakeholders/developers can understand how the model produces each image description and identify when hallucinations occur.

5. Monitoring and Maintenance

- **Goal:** Continuously monitor model performance and detect issues like hallucinations, bias or data drift.
- **Strategies:**
 - **Performance and Drift Monitoring:** Set up alerts for model performance drops or data distribution changes to proactively address model drift.
 - **Bias Detection Tools:** Use bias monitoring tools to regularly evaluate predictions for fairness across different user groups.
 - **Retraining Pipelines:** Automate retraining and updating processes to keep models relevant as new data or scenarios arise.
- **Tools and Libraries:**
 - **NannyML:** A library for detecting data and concept drift in machine learning models during monitoring.
 - **Grafana with Prometheus:**
 - Grafana provides real-time monitoring and visualization of performance metrics, while Prometheus can track drift, latency, and hallucination rates. This combination enables proactive monitoring of system health, ensuring that any anomalies are detected early.
 - **Uncertainty Toolbox:** A tool for measuring, visualizing, and improving model uncertainty estimation, essential for maintaining reliable performance.

- **Technical Focus:** Use **Grafana** to continuously track key metrics such as hallucination rates, accuracy, latency, and user feedback. By integrating **Prometheus**, data can be collected and monitored in real-time to detect performance degradation.

Human-Computer Interaction (HCI)

- **Objective**
 - Align **VisionClarity** with the needs and expectations of users in fields like education and content creation ensuring the system generates accurate and contextually relevant image-to-text descriptions.
- **Understanding User Requirements:**

- To understand user requirements for **VisionClarity**, I will focus on both usability and usefulness. I will organize structured interviews with AI researchers, professors, and content creators. During these interviews, I will ask open-ended questions to explore their experiences with existing MLLMs, specifically focusing on the hallucination issues they encounter.

- User Surveys:-

- Created a Sample Google form to understand the users' perspective and gather insights. Below is the link:-

- [Survey](#)

- **User Interviews:**

- Semi-structured interviews can be conducted in person or online using Zoom, Google Meet, or any online platform that is convenient for the user.
- The following are the questions that can be used for the interviews:

User Interview Questions

- 1. User Interaction**
 - How do you typically interact with LLMs when generating image descriptions? (List the Applications)
 - What steps do you take to verify the accuracy of the descriptions generated by these models?
- 2. Alternatives and Preferences**
 - Have you tried alternative tools or models for image description generation? If so, what were your experiences compared to LLMs?
 - What specific qualities do you value the most in an image description generation tool? (e.g., speed, accuracy, ease of use)
- 3. Context of Use**
 - In what contexts do you primarily use image description generation? (e.g., academic research, content creation, accessibility)
 - How critical is the accuracy of image descriptions in your specific use case?
- 4. Feedback Mechanisms**
 - How do you currently provide feedback on inaccurate descriptions generated by LLMs? Is there a process in place for this?
 - Would you find it helpful to have a built-in feedback feature within the image description generation tool?
- 5. User Education and Training**
 - Do you feel adequately educated about the limitations of LLMs, including the potential for hallucinations? What additional information would be helpful?
 - How do you think user training could play a role in mitigating the impact of hallucinations in image descriptions?

- **Create Personas and Scenarios:**

Persona 1:



ETHAN CRUZ

CONTENT CREATOR

ETHAN CRUZ
Specializes in visual storytelling and aims to produce high-quality, engaging content for digital platforms.

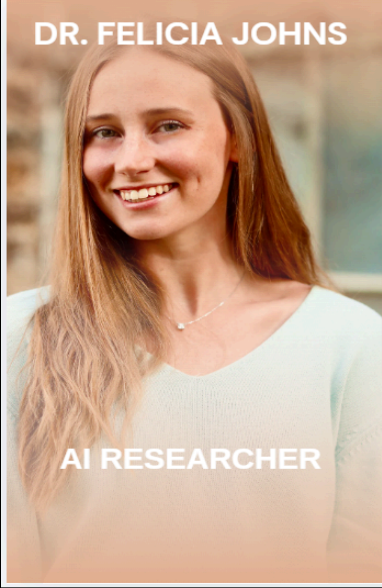
GOALS:
Generate engaging and accurate visual content descriptions for social media or blogs.

MOTIVATIONS
Strive for high-quality content that resonates with audiences and enhances brand visibility.

CHALLENGES
Overcoming the limitations of existing LLMs that may produce inaccurate or irrelevant descriptions and managing time constraints.

SCENARIOS
Ethan Cruz uploads nature images to VisionClarity to generate accurate descriptions for his educational videos. He reviews the AI-generated text and identifies instances of hallucination, providing feedback for improvements. After editing, he confidently incorporates the revised descriptions into his video script.

Persona 2:



DR. FELICIA JOHNS

AI RESEARCHER

DR. FELICIA JOHNS
Focused on improving language models and mitigating biases in machine learning.

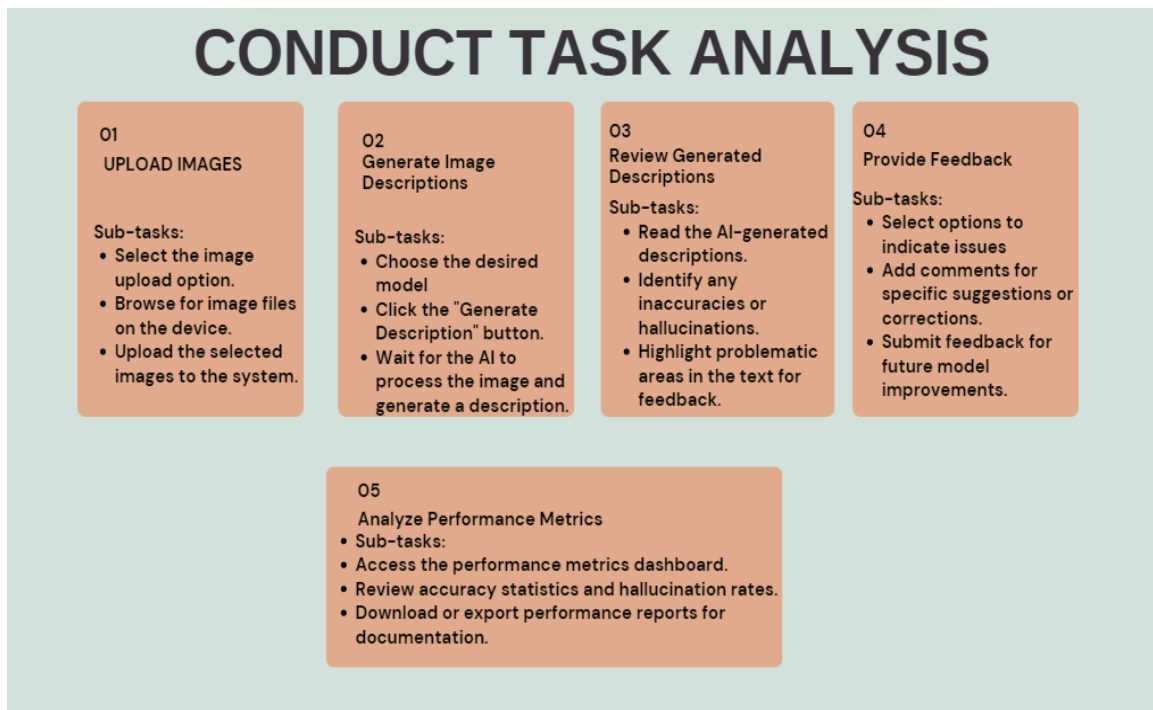
GOALS:
Enhance the accuracy of AI models, contribute to academic research, and publish findings on hallucination issues in LLMs.

MOTIVATIONS
Passion for advancing AI technologies, interest in practical applications, and the desire to improve existing models.

CHALLENGES
Dealing with data quality issues, understanding complex model behaviors, and addressing the limitations of current systems.

SCENARIOS
FELICIA JOHNS uploads images into the VisionClarity system to analyze and mitigate object hallucinations in generated descriptions. She explores performance metrics and fine-tunes model parameters to enhance accuracy. After running experiments, she documents her findings using the integrated reporting tool for her research seminar.

- **Conduct Task Analysis:**



- **Identify Accessibility Requirements:**

- **Screen Reader Compatibility**

Need to ensure that all critical information, such as generated image descriptions and confidence levels regarding hallucinations, is available in text form. This allows screen readers to convey the data effectively to visually impaired users.

- **Keyboard Navigation**

Design the entire VisionClarity system so that users can operate features like image upload, description generation, and result review fully through keyboard shortcuts, assisting users with mobility impairments.

- **High Contrast Mode**

Implement a high-contrast mode within the interface to support users with visual impairments, ensuring that they can easily distinguish text and graphical elements.

- **Font Size Adjustment**

Include functionality that allows users to adjust font sizes within the application, enhancing readability for those with low vision or reading difficulties.

- **Outline Usability Goals:**

- **Reduce User Errors**

Aim to decrease the frequency of incorrect image descriptions generated by the system. Target a reduction of user-reported errors by at least 30% compared to existing MLLMs.

- **Minimize Task Completion Time**
Strive to lower the average time users spend on tasks such as uploading images and generating descriptions. Set a goal to complete these tasks in under 2 minutes, improving efficiency and workflow.
- **Improve User Satisfaction**
Enhance overall user satisfaction with the VisionClarity system by targeting a satisfaction score of 85% or higher on user surveys. Collect feedback through Google Forms to collect user experiences and areas for improvement.
- **Increase Task Success Rate**
Aim for a task success rate of 90% or higher for users successfully generating accurate image descriptions without guidance. Monitor this metric to ensure the system meets user needs effectively.
- **Enhance Usability Across User Types**
Ensure that both novice and expert users can navigate the system with ease. Use user feedback to refine the interface and establish a usability rating of at least 4 out of 5 in user testing sessions.

Risk Management Strategy

1. Problem Definition

- **Key Risks:**
 - **Misaligned Objectives:** The project's goals might not align with end-user expectations, leading to irrelevant or inaccurate product descriptions.
 - **Ethical Risks:** Inaccurate image descriptions could result in misleading information, causing user dissatisfaction or financial loss.
- **Mitigation Strategy:**
 - **Involve stakeholders** in the early stages to define clear objectives and ensure alignment with stakeholder needs.
 - **Performance Metrics:** Establish measurable success criteria, such as accuracy rates and F1 score for product descriptions and the reduction of object hallucination instances.
- **Technical Implementation:**
 - Can use AI Blindspot to detect ethical risks in project framing, ensuring the problem statement is unbiased and properly scoped.
 - Use Marvel for prototyping requirements and aligning objectives visually with stakeholders.

2. Data Collection

- **Key Risks:**
 - **Data Quality:** Incomplete, noisy or biased data may lead to inaccurate product descriptions.
 - **Data Privacy:** Handling sensitive product or user data could result in privacy violations if not properly managed.
- **Mitigation Strategy:**
 - **Data Validation:** Use correct and incorrect image-descriptions data for data validation process to check for completeness, quality, and accuracy, ensuring datasets represent real-world e-commerce scenarios accurately.
 - **Bias Mitigation:** Regularly need to perform bias checks to ensure diverse product categories are represented fairly, minimizing risks of skewed descriptions.
 - **Privacy Measures:** Need to use anonymization techniques like data masking, tokenization and randomization for sensitive data and comply with GDPR standards when applicable.
- **Technical Implementation:**
 - Use Pandas for data cleaning, removing noise, and ensuring that datasets contain relevant product features.
 - Implement Differential Privacy using PySyft for secure handling of sensitive product or user information.

3. Model Development

- **Key Risks:**
 - **Algorithmic Bias:** Risk of biased predictions due to skewed training data, leading to inaccurate or unfair product descriptions.
 - **Overfitting:** The model may become too specialized, failing to generalize to unseen product images, affecting the reliability of visual search results.
- **Mitigation Strategy:**
 - **Fairness Checks:** Integrate fairness-aware algorithms such as reweighting and equalizing the odds to detect and mitigate biases during model training. Conduct bias assessments periodically to ensure diverse product categories are handled equitably.
 - **Regularization Techniques:** Use regularization (e.g., dropout or early stopping) to prevent overfitting, ensuring the model generalizes well across varied product categories.
 - **Explainability Tools:** Use tools like LIME or SHAP to enhance transparency in model decision-making, clarifying why certain product features were highlighted.
- **Technical Implementation:**
 - Use Fairlearn to monitor and address algorithmic biases.
 - Use SHAP for interpretability, showing which image features contribute most to specific product descriptions.

4. AI Deployment

- **Key Risks:**
 - **Integration Challenges:** Potential integration difficulties with existing e-commerce platforms, affecting the accuracy of visual search and product descriptions.
 - **Security Vulnerabilities:** Risk of exposure to cyber threats, potentially leading to compromised product descriptions or inaccurate search results.
- **Mitigation Strategy:**
 - **A/B Testing:** Conduct multivariate testing during deployment to compare performance with existing systems and validate the impact on e-commerce platforms without disrupting the current workflow.
 - **Security Best Practices:** Implement API access control such as Role-Based Access Control (RBAC) and JWT, and for encryption, use AES method. Finally, regular security testing to ensure the deployment environment is protected against potential threats.
- **Technical Implementation:**
 - Use **CI/CD Pipelines** with GitHub Actions for automated deployment, ensuring integration is smooth and secure or
 - Deploy **BentoML** for secure, scalable serving of the VisionClarity model, controlling access and monitoring security threats.

5. Monitoring and Maintenance

- **Key Risks:**
 - **Model Drift:** The data distribution may change over time, leading to a decline in the accuracy of product descriptions.
 - **Emerging Security Threats:** New security vulnerabilities may arise post-deployment, exposing the system to cyber-attacks.
- **Mitigation Strategy:**
 - **Drift Detection:** Establish drift detection mechanisms using performance metrics to identify when the model's accuracy decreases. Schedule regular retraining to handle changing product trends.
 - **User Feedback:** Implement feedback loops to gather user input on the accuracy of descriptions, helping to refine the system post-deployment. Enable reporting of inaccuracies directly through the user interface.
 - **Regular Security Audits:** Conduct periodic security assessments and apply updates to ensure system resilience to emerging threats.
- **Technical Implementation:**
 - Use **Prometheus** and **Grafana** for real-time monitoring, tracking model accuracy, latency, and drift rates.
 - Integrate **MLflow** to manage the lifecycle of the VisionClarity model, enabling automatic rollbacks if performance degrades.

6. Residual Risk Assessment:-

LIKELIHOOD	IMPACT				
		Acceptable	Tolerable	Unacceptable	Intolerable
	Improbable	Low risk	Moderate risk	High risk	Critical risk
	Possible	Low risk	Moderate risk	High risk	Critical risk
	Probable	Low risk	Moderate risk	High risk	Critical risk

Critical Risks

- **User Notification Issues:** Miscommunication or failure to effectively notify users when accurate results aren't available could lead to significant customer dissatisfaction and mistrust in the system.

Moderate Risks

- **Algorithmic Bias:** Subtle biases in model predictions could affect specific product categories, requiring ongoing monitoring and actions to ensure fairness and accuracy.
- **Data Drift:** Changes in product image styles and content over time can gradually decrease the accuracy of descriptions and search results in the model.

Low Risks

- **Minor Inaccuracies in Product Descriptions:** Occasional errors in descriptions due to challenging or noisy data inputs are manageable and can be corrected with user feedback.

Data Collection Management and Report

1. Data Type

- **Type of Data:**

- VisionClarity uses multimodal data, consisting of images (unstructured data) and text descriptions (structured data). The images come from product photos, while text data includes metadata such as product descriptions, categories, and attributes.
- **Data Granularity:** We use both raw images and processed textual data to train our models. The raw image data undergoes preprocessing to standardize dimensions, and textual data is processed for tokenization and embedding.
- **Challenges:** Handling large, high-resolution images requires significant memory, and some images may lack corresponding descriptions, affecting the data consistency for training.

2. Data Collection Methods

- **Source of Data:**

Public Datasets: We use public datasets like Kaggle, Open Images, and COCO to train our model with a variety of product images and descriptions. These datasets help the model learn general patterns between images and descriptions.

Proprietary Product Data: detailed images, specific descriptions, and metadata owned by retailers would make the model even better. This data includes precise product details, which would help our model create accurate, real-world descriptions and reduce object hallucination. Accessing this data would improve the model's performance by aligning it more closely with actual products in the industry.

Challenges: Proprietary data is consistently accurate but limited, while public datasets sometimes have quality inconsistencies.

- **Methodologies Applied:**

- **Data Downloads from Repositories:**

- Public datasets, such as Kaggle, Open Images, and COCO, are downloaded directly from online repositories. These datasets serve as the primary source for training general image-to-text mappings and are processed to ensure compatibility with VisionClarity's training pipeline.

- **Manual Collection and Curation:**

- Proprietary product data may require manual curation, where specific items and descriptions are selected to ensure high accuracy and relevance. This process involves human review to align proprietary data with our model's requirements for accuracy in image-to-text mappings.

- **Adjustments:** Batch processing and data cleaning steps are implemented to ensure completeness and accuracy. Additionally, data quality checks help standardize diverse data sources.

Ingestion for Training:

- **Techniques:** DataLoader classes in PyTorch and batch processing methods handle large datasets efficiently, allowing faster data loading without memory overload. Additionally, use Apache NiFi, an ETL tool for preprocess and standardize data from various sources before it enters the training pipeline.

Optimizations:

- We optimize data handling by implementing caching for frequently accessed data and minimizing data transformation steps within the training pipeline, reducing latency and improving throughput during training.

Ingestion for Deployment:

- **Process:** For deployment, data is ingested in real-time using APIs and queued in message systems like RabbitMQ to maintain low-latency responses.

3. Compliance with Legal Frameworks

Applicable Laws and Standards:

- The project adheres to GDPR for user data privacy and CCPA due to its potential use in e-commerce. NIST standards guide cybersecurity practices.

Compliance Strategy and Results:

- **Data Anonymization** and secure data storage strategies ensure privacy compliance. All data handling includes consent protocols, and third-party audits verify compliance.
- **Challenges:** Data anonymization reduced some detail in metadata, but alternative metadata fields were added to maintain training quality.

4. Data Ownership and Access Rights

- **Ownership and Access Control:**

- Public datasets are open-source licenses but in case proprietary product data is owned by third partners, licenses. Access to proprietary data is secured with two-factor authentication and logged access.
- Permissions: Team members have equal ownership in accessing the datasets.
- Lessons Learned:
 - Initial access permissions were too restrictive, slowing workflow.

5. Metadata Management

- **Metadata Attributes:** Metadata for images includes data source, timestamp, format, dimensions, and annotation quality. Text data is tagged with language, length, and category.
- **Management System:** MongoDB (database) stores metadata, allowing flexible querying and updates. Issues with incomplete metadata were resolved by implementing automated field checks.

Challenges: Missing metadata affects data retrieval, so automated validation scripts were added to ensure completeness.

6. Data Versioning

- **System:** Git is used for tracking data versions. Changes in data are tagged by version, enabling rollbacks if needed.
- **Strategy:** Different versions are maintained for training, validation, and testing datasets, ensuring traceability and enabling updates without disrupting the training pipeline.

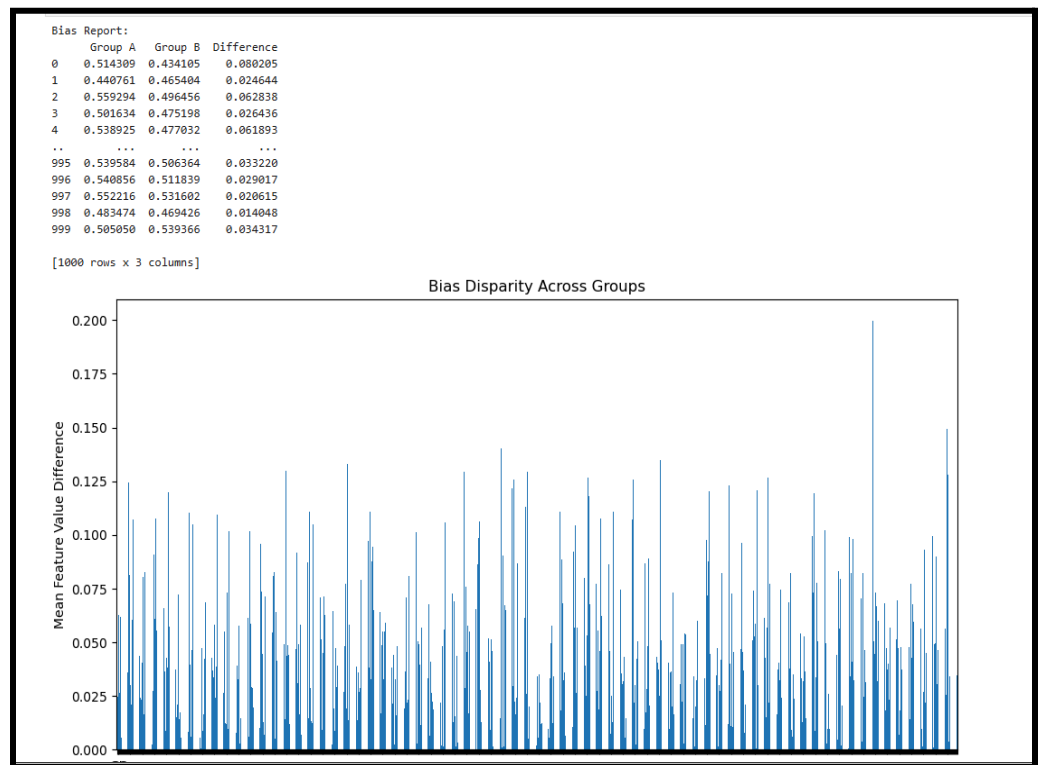
7. Data Preprocessing, Augmentation, and Synthesis

- **Preprocessing Techniques:**
 - **Normalization:**
 - Pixel values are normalized to a range of [0,1] using min-max scaling for image data, improving convergence during training.
 - **Challenges:**
 - Handling outliers that can skew the normalization process.
 - Managing different data ranges that may affect the model's ability to learn.
 - **Resizing:**
 - Images are resized to (256,256) pixels to meet model requirements while minimizing quality loss using bicubic interpolation.
 - **Challenges:**

- Potential quality degradation due to resizing, which may lead to a loss of important information.
 - **Scaling:**
 - Applied to numerical data in descriptions for consistency.
 - **Challenges:**
 - Features with widely varying ranges may dominate the learning process, leading to biased model performance.
 - **Dimensionality Reduction:**
 - Principal Component Analysis (PCA) is used for feature selection in high-dimensional text data.
 - **Challenges:**
 - There is a risk of information loss if dimensionality reduction is improperly applied, potentially impacting model performance.
 - **Feature Selection:**
 - Though we use PCA for feature selection, we will also use Random Forest because PCA focuses on variance and dimensionality reduction without considering feature importance, whereas Random Forest provides insights into feature significance based on predictive performance. Using both can enhance model performance
- **Data Augmentation and Synthesis:**
 - **Image Transformations:**
 - Includes rotations, flips, and color adjustments to increase diversity and robustness.
 - **Text Data Augmentation:**
 - Use controlled vocabulary lists to ensure that replacements maintain context and meaning.
 - Test for coherence and relevance of the augmented sentences by evaluating them with human reviewers or additional NLP checks.
 - **Synthetic Data Generation:**
 - GANs (Generative Adversarial Networks) are used to create additional images for underrepresented classes and also train the system with incorrect datas so that the system can avoid mistakes in real world situations . By regularly evaluating the generated data,we avoid introducing biases and hallucinations.
- **Impact on Model Performance:**
 - Reducing overfitting: By introducing variability, the model is less likely to memorize the training data and more likely to generalize well to unseen data.
 - Enhancing robustness: The model becomes more resilient to variations and noise in the input data, leading to better predictions in real-world scenarios.

8. Data Management Risks and Mitigation

- To mitigate risks in data collection performed following strategies applied :
 - Check-Bias :A report on wheather any categories are underrepresented



Outcome:-

From the above output,it can be inferred that it shows inconsistent bias patterns across features and while most features exhibit relatively low bias levels, there are clear outliers that may require attention.

- Perform error handling: Implementing error handling(Exception-handling) in my project ensures that unexpected issues are managed properly, enhancing system reliability and user experience. This proactive approach provides clear feedback for debugging and improves overall application stability.

9. Data Management Trustworthiness and Mitigation

- To show **trustworthiness** in data collection performed following strategies applied :-
 - **Monitoring dashboard** :provides real-time visibility into system performance metrics (like network traffic, disk usage, CPU threads, and memory) which helps ensure data is being collected reliably without system bottlenecks or failures. Additionally, the steady patterns in network traffic combined with consistent disk utilization rates (shown in the graphs) validate that data collection is proceeding smoothly and resources are being managed efficiently.



(Used weights and biases software tool for the system performance)

Outcomes:-

Increase in disk utilization from 123.02 GB to 123.12 GB over time demonstrates systematic data accumulation and storage. The controlled decrease in CPU threads from 42 to 40 while maintaining stable network traffic suggests efficient resource management during the data collection process.

Model Development and Evaluation

- **Model Development:**

- **Algorithm Selection:**

- In our project, Vision Clarity we will be implementing the following models:-

1. VisionTransformers- To analyze the image's features in depth
2. CLIP model- To pair the image and text description and provide an accurate text description that matches the image.
3. GAN model:- to train on incorrect results and reduce irrelevant text descriptions of the corresponding images.

- **Feature Engineering and Selection Strategies:-**

- We will be implementing the following:-

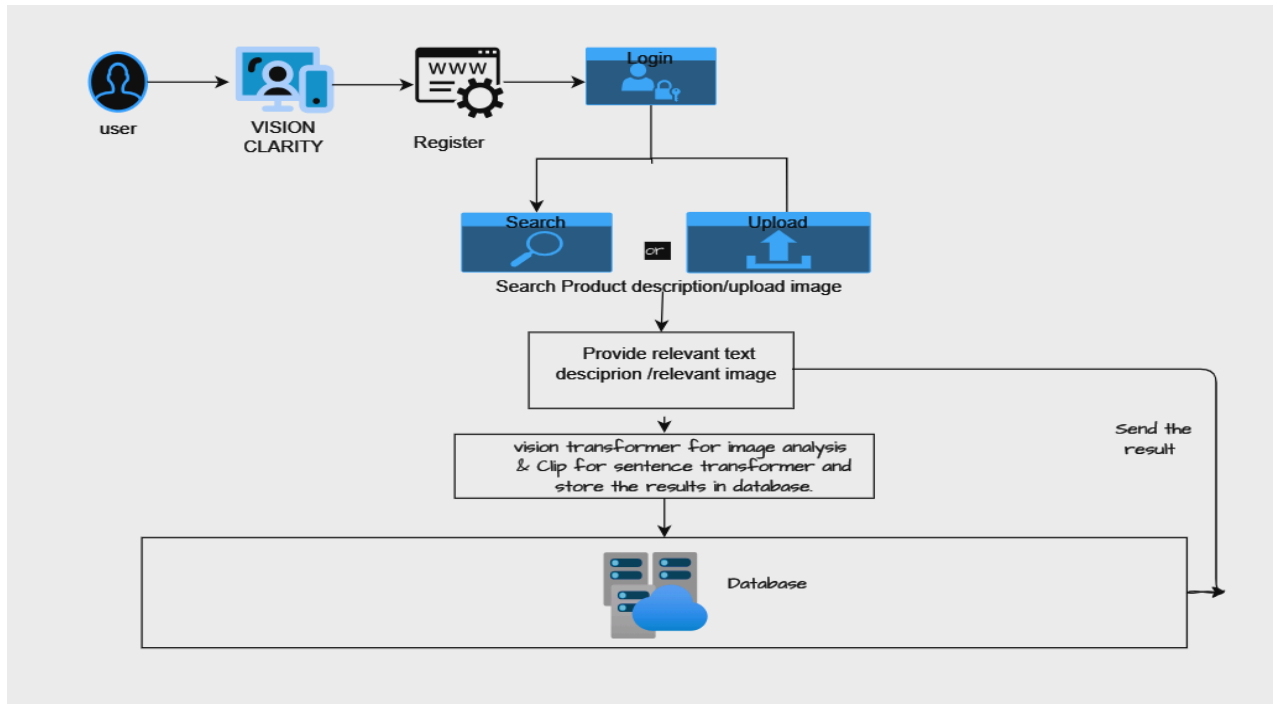
- 1. Text Preprocessing and Tokenization: We will implement text preprocessing steps such as lowercasing, removing special characters, and tokenization. This will help standardize the text input for the CLIP model, ensuring that the text descriptions are formatted and can be paired with the image features.
 - 2. Feature Scaling using TF-IDF scaling: Transform the text descriptions into weighted numerical vectors. This method ensures that more relevant features are given more importance, providing accurate results.
 - 3. Dimensionality Reduction using PCA: To reduce the dimensionality of the feature sets.
 - 4. Attention Mechanism-Based Feature Selection: We will use self-attention weights from the VisionTransformer to identify and emphasize the most relevant parts of the image

- **Model complexity/Architecture:**

- The following are the model's model's complexity and architecture:

- 1. Layered Architecture with Multi-Model Integration: We train using three models and the combination of those models makes the architecture more complex as it involves ensuring these models communicate effectively and share data seamlessly.
 - 2. High Parameter Count with Computational Demands: Each of the models we're using has a lot of parameters, which helps them learn complex relationships between images and text. However, this also means they require powerful computational resources, like high-performance GPUs, to train and run.

- **Model Architecture:**



- First, a user enters the system through Vision Clarity verification, followed by registration and login steps. After logging in, users can either search for product descriptions or upload images to the platform. The system then processes their input by providing relevant text descriptions and matching images. At the core of the processing, a vision transformer handles image analysis while CLIP technology manages sentence transformations, working together to understand and match the content. All this processed information moves into a database for storage, and finally, the system returns the results to the user. This streamlined workflow combines visual and textual analysis to deliver relevant product information to users.

- **Model Training:**

- **Training Process:**

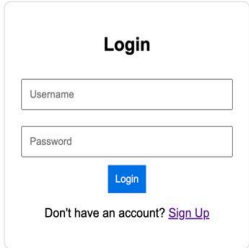
- Model training involves multimodal datasets, synthetic data augmentation, and adversarial training to handle ambiguous inputs, reducing hallucination risks.

- **Hyperparameter Tuning:**

- We will apply techniques like dropout and L2 regularization to reduce the chances of overfitting, helping the model generalize better to new data.
- During training, we will monitor the model's performance on a validation set and stop training when the performance starts to decline(early stopping method), preventing it from fitting too closely to the training data.

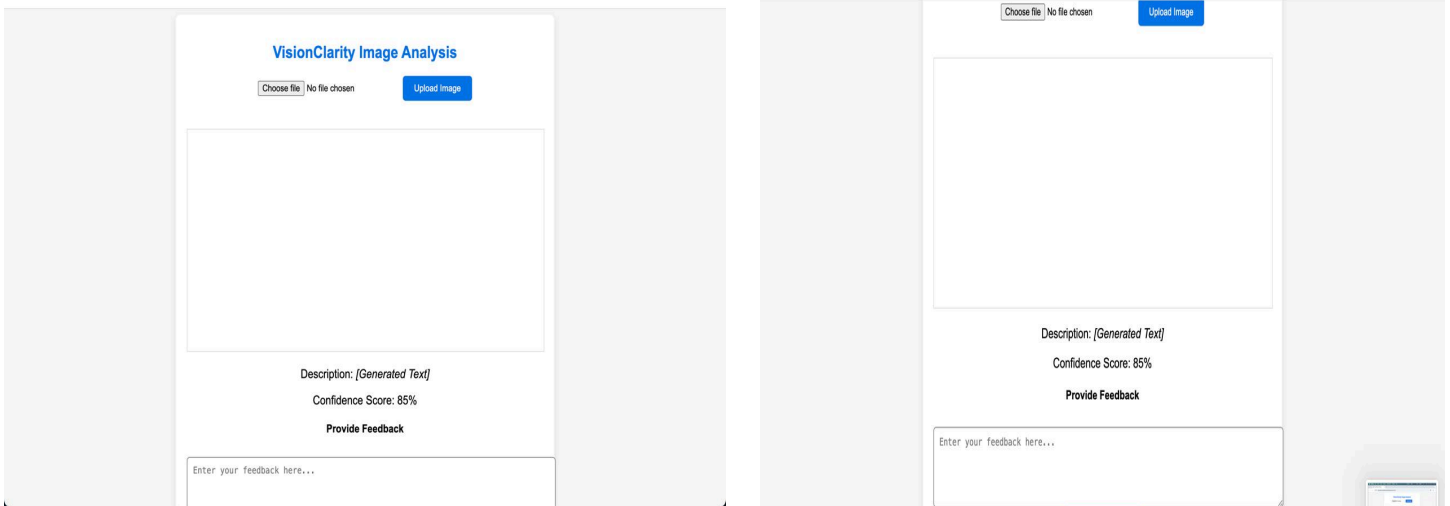
- **Model Evaluation:**
 - **Performance metrics:**
 - Will be implementing the following metrics :-
 - CF-Score(Content Fidelity Score) considers missing content that is important for completeness.
 - HRI (Hallucination Rate Index) focuses specifically on hallucination rather than overall accuracy
 - **Cross-validation**
 - We will implement stratified k-fold cross-validation where this method ensures that each fold is representative of the overall dataset, helping to maintain the distribution of classes and providing a more reliable estimate of the model's performance on unseen data.
- **Risk Management Report :**
 - The AIF360 toolkit provides fairness metrics and algorithms to mitigate bias in models.
- **Trustworthiness Report :**
 - Using InterpretML and SHAP for Explainability and Interpretability
 - To ensure the model's trustworthiness, we can use InterpretML for global and local interpretability and SHAP values to understand which features contribute most to predictions.
- **Apply HCI Principles in AI Model Development:**
 - **Prototype & Feedback Mechanisms :**

Login page :



The image shows a login page with a central form. The form has a title 'Login' at the top. Below the title are two input fields: 'Username' and 'Password'. Below the 'Password' field is a blue button labeled 'Login'. At the bottom of the form, there is a link that says 'Don't have an account? [Sign Up](#)'.

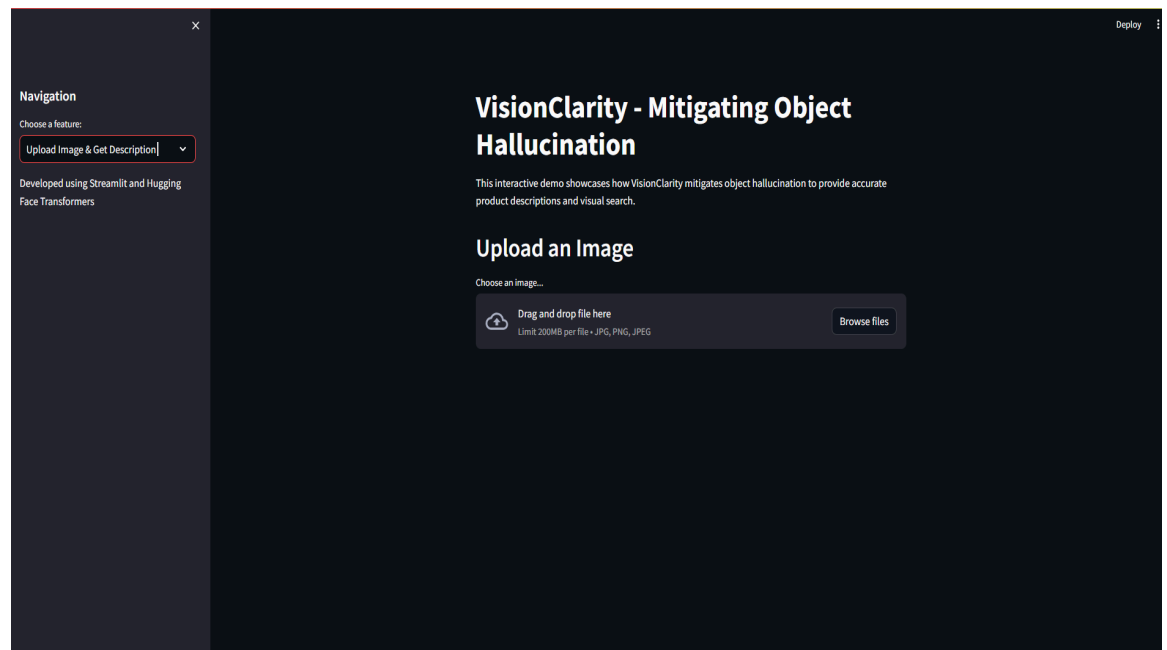
After login:

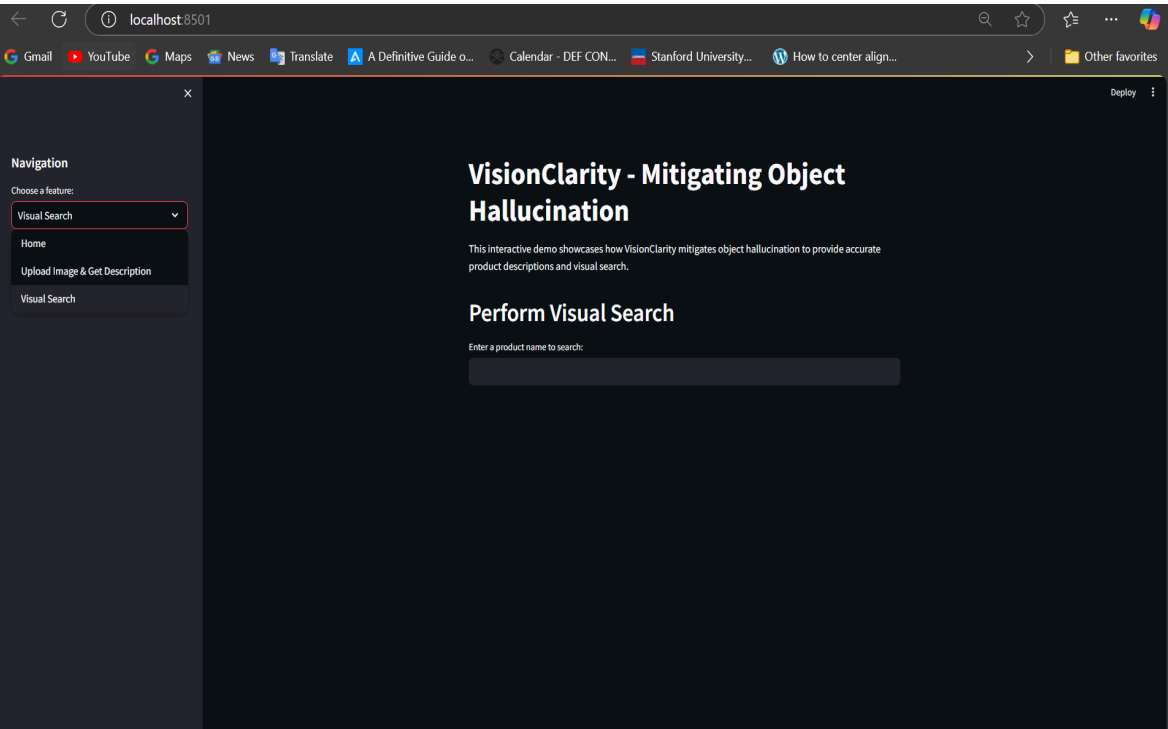


We ask for feedback from the user regarding the description through providing feedback after the results.

- **Design Transparent Interfaces :**

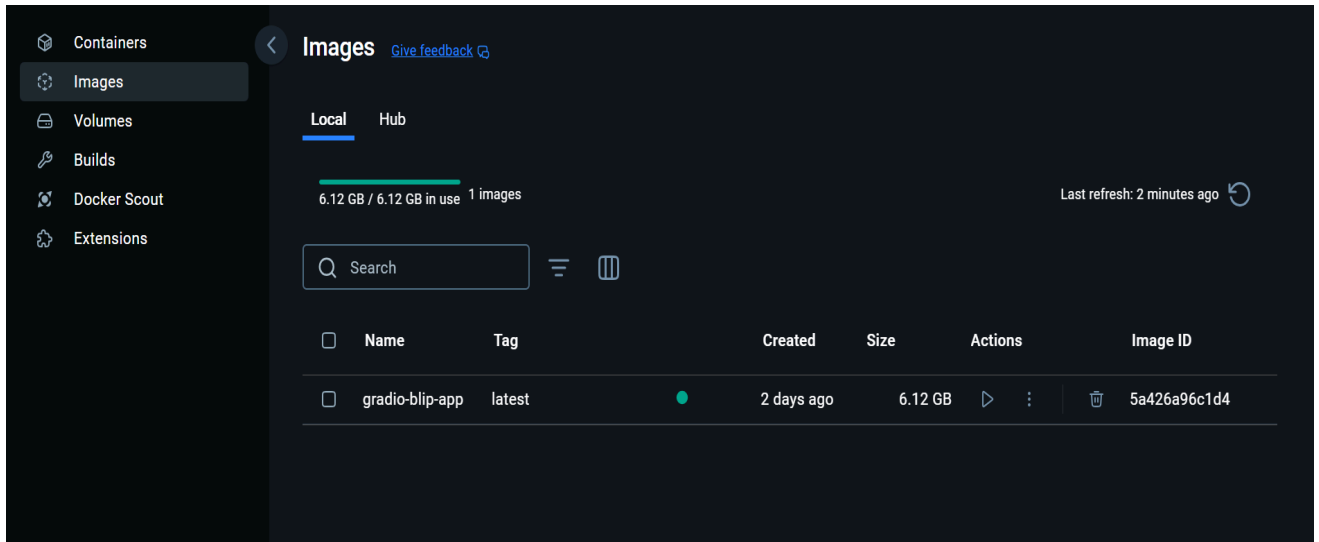
Used streamlit as interface and there are three navigations which are home, upload image & get descriptions and visual search.





Deployment Plan

- **Deployment Environment Selection:**
 - Using Docker for the deployment



Components within the Docker Environment:

- **Primary Files:**
 - app.py: Core application logic.
 - Pre-trained model weights in .pkl format.
 - requirements.txt: Specifies Python dependencies(torch, transformers, gradio, joblib pillow)
 - Dockerfile:

```
1 # Use an official Python runtime as the base image
2 FROM python:3.9-slim
3
4 # Set the working directory inside the container
5 WORKDIR /app
6
7 # Copy the local files into the container
8 COPY . /app
9
10 # Install necessary system packages
11 RUN apt-get update && apt-get install -y \
12     libgl1-mesa-glx \
13     libglu1-mesa \
14     && apt-get clean
15
16 # Install Python dependencies
17 RUN pip install --no-cache-dir torch torchvision transformers gradio joblib pillow scikit-learn
18
19 # Expose the port Gradio will use
20 EXPOSE 7860
21
22 # Command to run the application
23 CMD ["python", "app.py"]
24
```

- **Monitoring Tools:**
 - Integrated **Prometheus** for performance monitoring and logging.
 - **Grafana dashboards** to visualize metrics such as model latency, accuracy, and memory usage.
- **Security and Compliance in Deployment (Trustworthiness and Risk Management):**
 - **Container Hardening:**
 - Use minimal base images like `python:3.9-slim` to reduce vulnerabilities.
- **Logging and Monitoring:**
 - Prometheus tracks system metrics such as response times, memory usage, and error rates.
 - Anomaly detection triggers alerts for deviations.

Compliance Measures:

- Ensure all containerized environments comply with GDPR and CCPA through:
 - Data Masking: Obfuscate sensitive fields before processing.
 - Differential Privacy: Safeguard against potential data leaks.

Monitoring and Maintenance

- **System Evaluation and Monitoring:**
 - We will utilize Grafana for real-time system monitoring and visualization, integrated with Prometheus for metric collection. Key metrics to be tracked include:
 - **Accuracy:** Ensures descriptions closely match the product images.
 - **Latency:** Measures the time taken for generating descriptions and visual search results.
 - **Hallucination Rate Index (HRI):** Tracks instances of object hallucination to ensure precision.
 - **Drift Detection:**
 - To detect data drift, we will deploy NannyML for monitoring shifts in data distribution. If the distribution deviates significantly, retraining will be triggered to restore performance. Outcomes will include detailed reports on drift patterns and their mitigation.
- **Feedback Collection and Continuous Improvement:**
 - **Feedback Mechanisms:**
 - We will integrate a feedback form directly into the VisionClarity interface using Gradio. Users can report inaccuracies or provide suggestions post-usage.
 - **Advanced Analytics:**
 - User interaction data will be captured and analyzed through Qualtrics to evaluate behavioral trends and identify common errors in generated descriptions. This will provide actionable insights for refining the system.
- **Maintenance and Compliance Audits**
 - **Weekly Updates:** Review accuracy metrics and apply minor adjustments.
 - **Quarterly Audits:** Perform comprehensive evaluations of system trustworthiness and compliance with GDPR and CCPA standards.
 - **Maintenance Tools:**
 - **Apache Airflow:** Automates model retraining workflows based on monitored drift or feedback signals.
 - Adjustments to model parameters and infrastructure will be documented in a version-controlled repository using **MLflow**, ensuring transparent change tracking.

- **Model Updates and Retraining**

- **Retraining Process:**

New data will be gathered continuously from user feedback and interaction logs.

- **Manual Retraining:** Occurs every six months using updated datasets.
 - **Automated Pipelines:** Tools like **DVC** and **MLflow** will handle dataset versioning and pipeline automation
 - **Version Control:**

Each retrained model will be tagged with a version number, and testing results will be compared against the current version to ensure improvements. Challenges, like addressing drift or bias, will be logged and mitigated before deployment.