Project Plan

Project Overview

Project Title: **REACH** (**R**etrieval **E**ngine for **A**rticles and **C**ommon **H**ousehold items)

Project Overview:

1. Objective:
    a. **Goal:** Our project aims to assist the user in finding misplaced household items, such as keys and wallets, that are kept around the house through an AI-based system. For example, imagine you are getting late to an online meeting and you couldn't find your laptop charger. With REACH you can get the image where your charger is located.

    b. **Problem to be solved:**

    - In our ordinary life, we have some designated places almost for everything which we keep in our daily use. One drawer is for wallets and watches, and one basket is for car keys, etc. We might be a bit careless at times and put our things in the wrong spots

    - Our muscle memory constantly looks in the same spot, so sometimes it's an oversight or we're rushing to get somewhere else. But regardless, the lost items lead to stress, irritation, and waste of time. The misplaced article is in the house. To address this issue, we are working on mapping the location and detecting objects in real-time. Such that, this system eventually becomes a part of your daily life, helps in maintain in the inner peace and reduces stress in finding the misplaced objects.

    c. **Expected Outcomes**:
        i. **Object location:** Snapshot and the location of the place where the object has been placed.
        ii. **User-friendly Interface:** Website with friendly UI, providing complete access to laymen. This includes easy navigation inside the interface and clear visual representations for identifying the detected locations.

iii. **Enhanced convenience:** Users can find their misplaced items at their fingertips.

2. Scope:

a. What will the AI system do?

i. This AI system identifies the objects through YOLO models with the help of user home CCTV video footage and stores the location of objects.
ii. It identifies the location of the object returns the snapshot of it.
iii. After identification of the desired object, it provides the snapshot along with the various objects it is surrounded by.
iv. Through API's we display the image on the UI to identify the objects and get alerts.
v. Provides access to historical data based on past object detection.

b. What data will it use?

i. Video or image data that is generated by CCTV in a home environment.
ii. User input that can be text or image from UI.
iii. Pre-trained AI models for object detection such as YOLO.

c. What are the limitations?

i. The accuracy of identifying varies on quality of image, position of object.
ii. We need high performance hardware while dealing with video footage.
iii. Ensuring the user data is secure and private.
iv. Pre-trained models may not contain the objects in their home.
v. The improper placement of CCTV will affect the accuracy of the model.
vi. There will be some accuracy issues of low image quality, we will handle it by using adaptive learning techniques.

3. Ai techniques and tools:

- YOLO (You Only Look Once) – Real time object detection that identifies multiple objects in a single frame.

- CNN Model - Used for the classification of the detected objects in the various categories like person, vehicle, etc.

- RNN model - to detect unusual patterns of events.
- Framework - TensorFlow, Keras – Used for building and training AI models
- Libraries - OpenCV, Pandas, NumPy – Essential libraries for building the model
- UI - React, Bootstrap, Django – Helps in creating responsive user interface.

Stakeholders:

1. Project Team:
    a. Project Manager:
        i. **Role:** One has to keep track of all the stages from planning the project to delivery.
        ii. **Responsibilities:**
            - Check if the project objectives and goals are met and the projects are completed on time and on budget.
            - Engage with stakeholders to collect their input and illustrate them to the group in a concise manner.

    b. AI/ML Specialist:
        i. **Role**: Design and build the algorithm.
        ii. **Responsibilities**:
            - Train the model with appropriate processed dataset.
            - Utilize frameworks and libraries to make easier and optimized code.

    c. Data Scientist:
        i. **Role:** Analysis and processing the data.
        ii. **Responsibilities**:
            - Create new features or eliminations to the model so as to improve the model.
            - Build report based on the data performance.

    d. Software Engineer:
        i. **Role:** Developing frontend and backend application to integrate the AI system.
        ii. **Responsibilities**:
            - Creating a system on the server-side, controlling the data storage and executing API.
            - Building and maintaining the UI.

    e. UI/UX Designers:
        i. **Role:** Design an attractive and user-friendly interface.
        ii. **Responsibilities**:
            - Conduct various researches to understand user needs.
            - Creating prototypes to visualize and test the needs.

f. DevOps Engineer:

i. **Role:** Managing and maintaining the application for deploying and running AI.

ii. **Responsibilities**:

- Implementing CI/CD pipelines which helps in testing and deployment of systems.
- Monitor performance of the machine and identify issues occurred by it and provide regular updates to fix them.

g. Privacy and Security Experts:

i. **Role:** Ensure the data is secure and meets data protection regulations.

ii. **Responsibilities**:

- Implement and maintain protocols for data protection and integrity.
- Conduct risk assessments and manage vulnerabilities.
- Allowing users to flag misidentifications and improve the system overtime in terms of security.

h. Testing Team:

i. **Role:** Testing the system for bugs, issues in performance.

ii. **Responsibilities**:

- Validate whether the system satisfies the needs of the user and work accordingly.
- Documenting the results and providing feedback for improvement.

i. Customer Service Representative:

i. **Role:** Provide service to end users regarding issues and troubleshooting.

ii. **Responsibilities**:

- Helping customers to understand how to use the system effectively.
- Collect user experience for better improvement.

j. Legal Advisor:

i. **Role:** Providing timely guidance on all legal aspects.

ii. **Responsibilities**:

- Provide advice regarding data privacy, intellectual property and compliances.
- Draft and review legal documents and privacy policies.

k. Business Analyst:

i. **Role:** Gather and document the requirements provided by stakeholders.

ii. **Responsibilities**:

- Analyze market trends and assist in developing strategies for marketing and scaling the system.

2. End Users:
    a. **Household members:** Family members or friends in a house who always tend to misplace the items. They can check for them through a smartphone.

    b. **Elder Users:** Our system is mainly useful for elderly people who might have memory-related issues and commonly misplace items. With the help of their younger ones, they can track their items.

    c. **Busy Professionals:** These are the individuals with tight schedules and have a higher chance of disorganization of things. they can check the location of their belongings with a single click.

    d. **Smart Home Enthusiasts:** These people are usually tech geeks and through this system, they can navigate through their smartphone for the position of their things.

    e. **Frequent Travelers:** They can check their important items quickly, such as chargers and passports, via smartphones before leaving the house.

    f. **Caretakers:** They can monitor the essentials of those they take care of. Giving them time to take care of the person instead of searching for the items.

3. Other Stakeholders:
    a. **Cloud Providers:** Cloud services like AWS, Azure to store and access the data remotely.

    b. **IoT manufacturers:** Hardware devices like cameras and sensors like infrared in low light and ultrasonic for distance calculation, etc.

    c. **Home automation companies:** Companies providing solutions to automate the home might be interested in integrating our idea into their systems.

    d. **AI developers and cybersecurity engineers:** Teams that are continuously contributing in the field of AI for better models and cybersecurity engineers who can secure the system from cyber-attacks.

    e. **Privacy Authorities:** Data protection authorities who look after privacy laws such as GDPR and CCPA.

Computing Infrastructure

Project Needs Assessment:

1. Tasks:

a. The primary objective of the project is **Object Detection** through pre-trained models like **YOLO or EfficientDet**, which helps in identifying daily use objects in live video streams.

b. Processing the image with **OpenCV** libraries for **edge detection** and **image enhancement** for better input to the model.

c. Our main data types would be the images captured using the camera positioned around the house.

2. Performance Benchmark:

a. **Latency:** Our system must detect the object, retrieve the location, and display the location image on the interface in a minimum amount of time **(less than 500 ms).**

b. **Throughput:** The system should be able to support **15-30 frames per second (FPS)** for the smooth flow of operations.

c. **Accuracy: 95% or higher accuracy** is required in identifying and locating the objects.

3. Deployment Constraints:

a. **Environment:** Lightweight hardware integration on edge devices to perform on-device operations to reduce the latency. Cloud architecture for heavy tasks.

b. **Power:** The edge devices should work with minimal energy consumption.

c. **Network Conditions:** The system should work in low network bandwidth.

Hardware Requirements Planning:

1. GPU/CPU needs:
    a. For model training, cloud-based high-performance GPUs like NVIDIA A100 or V100 would be ideal, or according to the availability, Hypergator would work well to train and run our models.

    b. For interpretation inside the house, devices like the NVIDIA Jetson or Raspberry Pi with a camera module can be used as edge devices.

2. Memory and Storage:
    a. Training Environment:
        i. A minimum of **32-64 GB of DDR4 RAM** is required to prevent bottlenecks during model training
        ii. Up to **1 TB of SSD** is required for storage of models, weights, and logs.

    b. Inference Environment:
        i. Minimum of **4GB RAM** for inference on edge devices.
        ii. **32-64 GB** of microSD on board for storing and deploying the model.
        iii. An additional USB driver to store the data that is stored so far before any power cut.

Software Environment Planning:

1. Operating systems:
    a. **Windows Server** provides a better platform as it is used by a maximum number of people; it becomes easier to share with others at the time of development.

    b. **Ubuntu** is recommended for model training due to its compatibility with AI frameworks like TensorFlow and PyTorch.

2. AI Frameworks:
    a. **Pytorch or TensorFlow** will be used to develop and train the object detection models.

    b. Libraries like OpenCV and OpenVINO for inference environment on the edge devices.

    c. **Flask** will help object detection models for easier integration with the user interface.

d. For edge devices, we consider TensorFlow Lite for optimized inference which reduces the latency and lead in improving the performance.

3. Virtualization tools:
   a. **Docker** ensures the consistent development and deployment of our project.
   b. **Kubernetes** helps us in scaling and managing the containerized AI workloads.

Cloud Resource Planning:

1. Cloud Services:
   a. **AWS SageMaker** assists with hyperparameter tuning, deployment, and end-to-end object detection model training.
   b. **Azure Machine Learning** for training and **Azure IOT** for integrating the edge devices.
   c. **Google AI Platform** for managing model training and **TPUs** for accelerating the model training time.
2. Storage Resources:
   a. **AWS S3, Azure Blob Storage, and Google Cloud Storage** for storing the training data, model, and logs.
3. Cloud-native AI resources:
   a. **AWS Lambda** to trigger specific models without managing the inference. For instance, we could trigger cloud-based models when the system identifies low-confidence predictions.

Scalability and Performance Planning:

1. Scalability:
   a. Use of **Kubernetes** to autoscale the inference workloads and train these dynamically based on demand.
   b. **Auto-scaling** the cloud instances to manage the varying workload during model retraining or increased object detection tasks.
2. Performance Optimization:
   a. **Model Quantization:** Converting the models from FP32 (32-bit floating point) to INT8 (8-bit integer), as FP32 uses 32 bits whereas INT8 uses only 8 bits, which reduces the inference time and computation required on edge devices.

b. **Model Pruning:** Pruning neurons in the network to reduce the model size and improve the speed.

3. Performance Monitoring:

a. Use of **Prometheus and Grafana** to monitor real-time system performance, including GPU latency and memory usage across devices.
b. To optimize the performance of GPU and CPU during model training, **NVIDIA Nsight Systems** will be used to ensure low latency and efficient hardware.

Security, Privacy and Ethics (Trustworthiness)

Problem Definition:

1. Goals:

a. Defining objectives and needs that an AI system should satisfy and identifying goals, problems, and risks that affect the system.

2. Strategies:

a. Stakeholder Engagement:

i. Engaging stakeholders such as homeowners, privacy advocates, security experts to understand their perspective on the system and privacy.

ii. **Techniques:**

1. **Interviews:**

a. Conduct one-to-one interviews with end users to get insights and understand their perspectives in depth.

b. In our case, we conduct one-to-one interviews with homeowners, and security experts by asking questions and getting insights for better problem-defining.

2. **Focus Group:**

a. Gather diverse stakeholders for discussion to explore feedback and know issues.

b. In our case, we let stakeholders such as homeowners, security experts, and privacy advocates discuss among themselves, and this is physically monitored by a person or recorded from which they gather issues and insights.

iii. **Tools**:

1. Focus group software to create or organize virtual focus groups using **FocusGroupIt**.

2. Conducting interviews through **Zoom** to interact more with users.

b. **Risk Assessment:**

i. Analyzing to identify and manage the risks.
ii. This information helps generate a powerful problem definition.
iii. **Techniques**:

1. **Brainstorming**:
   a. Grouping up a different group of stakeholders to identify risks through open or general discussion.
   b. Through this we can gather more and innovative insights.
   c. In our case, this helps in identifying new challenges and solutions to the problem, such as blurring user faces for more security purposes.
   d. Tools:

   - **Miro**: A collaborative online whiteboard platform where teams can brainstorm, create mind maps, and visualize ideas in real-time.
   - **XMind**: A powerful mind-mapping software for creating structured diagrams and visualizing complex ideas.

2. **SWOT Analysis**:
   a. Analyze strengths, weaknesses, opportunities, and threats related to the project.
   b. This helps in identifying both internal and external risks.
   c. In our case, we identify a problem and from it we get threats and weakness that need to be converted into strength and opportunities.
   d. Tools:

   - **Lucid chart**: A diagramming tool that enables users to create professional-looking SWOT analysis charts easily.
   - **XMind**: A powerful mind-mapping software that helps structure SWOT analyses and explore relationships between elements.

Data Collection:

1. Goals:

a. Gather high-quality, representative, and privacy-conscious data for model training.

b. Proper data collection can lead to a high-performing model.

2. Strategies:

a. Data Anonymization:

i. Implementing techniques such as **blurring faces** and files to protect the important or sensitive information captured in video feeds.

ii. Techniques:

1. Data Masking:

a. Replacing sensitive information with realistic information.

b. In our case, we replace the user images and sensitive data into a blur mode or change to color surrounded by it.

b. Bias Detection and Correction:

i. This ensures the system is fair like treating all individuals in the same way, accurate by using a general balanced dataset which works balanced in any situation and trust is achieved with fairness and transparency.

ii. Implement statistical tests and bias correction methods during data preprocessing

iii. **Techniques**:

1. **Confusion Matrix Evaluation:** Use confusion matrices to assess the rates of false positives and false negatives to analyze the matrix for different demographic groups to identify bias in predictions.

AI Model Development:

1. Goal:

a. Selecting algorithms and preprocessing techniques in a way that handles the biases in the data and ensures equal model performance across all setups.

2. Techniques:
    a. **Algorithm Fairness:**
        i. Use of techniques like **adversarial debiasing and reweighting** during model training to ensure the object detection models perform consistently across all setups.
        ii. Implementing **Data Augmentation** strategies such as various lighting conditions and backgrounds to increase fairness across different scenarios.
        iii. Use of **Fairlearn** to evaluate and mitigate biases in object detection predictions.
    b. **Explainability Tools:**
        i. Use of frameworks like **LIME (Local Interpretable Model-agnostic Explanations)** and **SHAP (SHapley Additive exPlanations)** to provide insights into model decisions that ensures the users understand how the system identifies the object.
        ii. **Grad-CAM (Gradient-weighted Class Activation Mapping)** highlights the areas of the image that the model focuses on during object detection.
        iii. These transparency implementations aid the user in trusting the system on how it works by taking their house as input. This also addresses the privacy concerns of the user.
    c. **Robustness Testing:**

        i. Through **Foolbox,** we will stimulate adversarial conditions and test how our model is performing under difficult conditions.

        ii. **CleverHans** to check the benchmark and defend the object detection model against adversarial like noise or occlusions.

3. Tools:
    a. **SHAP (SHapley Additive exPlanations):** To understand the contribution of image features to the object detection.
    b. **LIME (Local Interpretable Model-agnostic Explanations):** For explaining local interpretable approximations of the object detection decisions.
    c. **Grad-CAM:** Highlights the parts of the image that are influenced by the model prediction.
    d. **Foolbox:** To test the object detection robustness by simulating adversarial attacks.

AI Deployment:

1. Goal:
    a. Deploying the AI model in real-world and making sure it operates as expected.
2. Techniques:

    a. **Continuous Integration/Continuous Deployment (CI/CD):**

        i. Implementing CI/CD pipeline for effective updates, rollbacks, and automation.

ii. Clear and efficient CI/CD pipelines that automate testing, building, and deploying applications.

iii. **Tools:**

1. **Kubernetes**: A tool that helps manage containerized applications and can enforce security policies, such as network segmentation and role-based access control.

2. **Docker**: Containerizes models and their dependencies, ensuring that they run in isolated environments.

b. Secure Model Serving:

i. Essential to protect AI models from unauthorized user access and misuse.

ii. **Strategies**:

1. Authentication: Use strong authentication such as API keys and others to ensure unauthorized users.
2. Access Control: Implement role-based access control to restrict correct users from accessing correct data.
3. Model encryption: Encrypt the model file to protect against unauthorized access and reverse engineering.

iii. **Tools**:

1. **OAuth2 Providers (e.g., Auth0, Okta):** These platforms provide robust authentication mechanisms for securing API access.

Monitoring and Maintenance:

1. Goal:

a. Continuous monitor of the system to detect any anomalies and rectify them right away.

2. Techniques:

a. Continuous Improvement:

i. Implement a mechanism to gather feedback on maintenance practices to refine the system continuously.

ii. Opportunities can be identified through feedback or comparing with existing products.

b. **Performance and Drift Monitoring**:

i. This involves tracking the operations and their functionality and checking whether they are optimal or not and also whether they meet the requirements.

ii. Techniques:

1. Application Performance Monitoring (APM):

a. It monitors the application performance, which provides insights into response times and transaction times.

b. Tools:

i. New Relic, AppDynamics

2. **Monitoring Model Performance:**
    a. It assesses model performance or accuracy metrics such as precision, recall, and F1 score to identify any decline in performance due to data.

Human-Computer Interaction (HCI)

Understand User Requirements:

1. Monitoring daily activities:

a. Monitoring users on how they are performing their daily activities on both working and non-working days with their consent. By this, we will get a gist of how they are handling their house-held objects and the key areas where one searches for objects around the house.

2. Focus Groups:

a. By organizing distinct discussion groups with individuals of varying ages, we can get crucial information about the needs and requirements. We will also get to know about the important places and items to focus on.
b. With these details we will address all these needs and implement all those that are missing in our system, and if some objects are not included, we will train the system with those objects for better usage.

3. Observational Interviews:

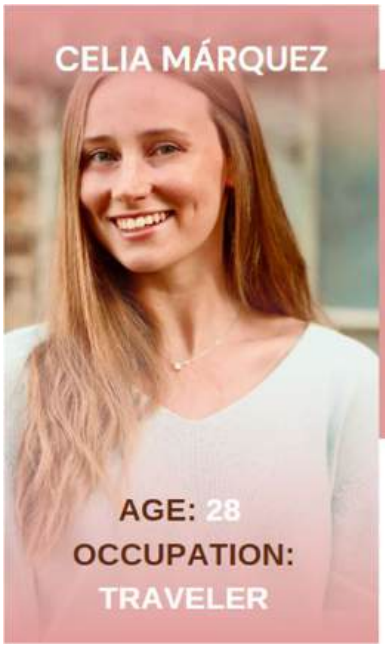
a. I want to watch people and see what they bring with them and how they find things when they're on the go, rather than doing formal interviews. Our comprehension will expand as a result of these contextual questions, which will assist us in comprehending the needs unique to each user.
    i. **For example (User is a working professional):**
        1. Where do you keep your car keys?
        2. How often do you misplace your car keys?
        3. Where do you find the misplaced car keys?
        4. Have you ever felt the need for a tracking system?
        5. How organized are you?

4. Tools:

a. **Otter.ai:** Helps us in recording and transcribing the interviews to analyze the user insights.
b. **Zoom/Google Meet:** This enables us to have interviews regardless of the location and will act as a platform to conduct our focus groups.
c. **Toggl:** This helps track and log user activities according to the time.

Create Personas and Scenarios:

1. Personas:



ABOUT CELIA MARQUEZ

Celia is a traveler who loves exploring places and vlogging. She spends weeks or months traveling. She always returns home to reset before her next trip. She often misplaces her important items like travel documents, keys, and camera accessories as she unpacks in a hurry.

FRUSTRATIONS

Post-trip disorganization which leads to the misplacement of important items and feels stressed as she wants to relax after a long trip but couldn't do so as she gets sidetracked in finding her belongings.

GOALS

- Quick recovery after trips
- Staying organized at home
- Maximising the rest time

SCENARIO:

After the trip she is exhausted and unpacks her gear quickly and takes rest. After a while she is looking for her essentials like camera, memory card. She couldn't find them then instead of searching the whole house she uses our application and gets to know where the items are located.

ABOUT DANIEL GALLEGO

Daniel is a senior software developer who spends most of his time in coding session, virtual meetings and debugging. He enjoys his off time in home to relax. He always tends to leave things around like his headphones, laptop charger and even his headphones. After a work session he always tend to search these things

FRUSTRATIONS

Sometimes during an important meeting some essential items gets misplaced like a USB or a mouse and gets lot frustrated with unorganized work environment which leads in loosing valuable time.

GOALS

- Reduced Clutter
- Efficiency in the work environment
- Staying focused

SCENARIO:

After an intense coding session, there is a meeting scheduled with his team and he couldn't find his wireless headphones to attend the meet. So, instead of searching for them he uses our application and finds them to get ready for his call.

2. Scenarios:
    a. Monitored Environment:

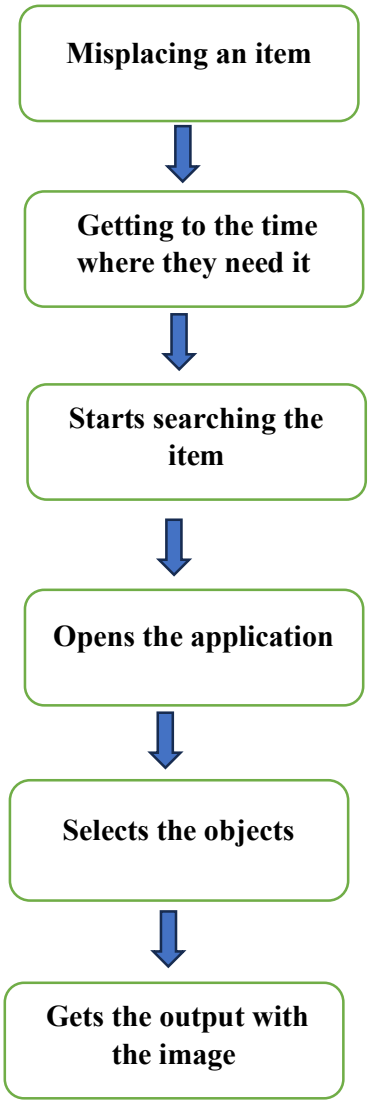
i. Instead of creating a virtual environment, we take the consent of the user to monitor their environment and track how they find their misplaced household objects.

b. Controlled Environment:

i. Here we create a scenario where the user faces the task of finding their daily household objects in a limited amount of time just to create a sense of hurriedness. This allows us to observe user behavior.

Conduct Task Analysis:

1. Flow Chart of finding things:



2. Behavioral Task Analysis:

a. **Observe:** Observe how the user interacts with the system's UI and how useful this system is for them.
b. **Record:** Documenting all the actions taken by the user and making notes on all the places where the user was stuck.
c. **Develop:** Improvising the areas after analyzing the areas where the users struggled.

3. Reverse Identification:
   a. Instead of going from misplacing the items to identification. If we go in reverse order, it helps us to find the key areas where the users usually either lose or find their items.

Identifying accessibility requirements:

1. Less Complex UI: Making the interface user-friendly, such that even laymen can access it without any hassle. This enables our end users to be more relaxed while using our interface.

2. Inclusive Design: Involving people with disabilities to understand their requirements and add elements that assist them.

3. Visual Displays: Displaying the picture of the place on the interface where the lost item is located. This helps the users to get a visual representation of located place.

Outline usability goals:

1. Emotional Stability: We want the customer to feel less anxious about their misplaced belongings by using this AI system. Most misplaced home things are just within the house we just don't know where we kept them. As a result, we are able to detect and track items while also reducing user stress and promoting emotional stability.
2. Time-Saving: With this application, users can track their misplaced objects with a click instead of searching the whole house. This helps them to save time and allows them to focus on other tasks.
3. Incorporating maximum number of items: Adding a wide range of items aids us in detecting maximum household objects and helps to scale our product even better.

4. User-Satisfaction: We will create a post-deployment survey that focuses on user involvement and privacy aspects of the app. Through this, we aim to see a significant increase in user satisfaction in the curve on managing the household items.

Risk Management Strategy

Project Overview:

1. **Key Risks:**
    a. **Misalignment with objectives:** The list of detectable objects may not reach the expectations of the user which can cause to misunderstandings and frustration to the user.

    b. **Ethical Risks:** Since household clients are our primary focus, constantly monitoring their homes raises a number of privacy concerns.

    c. **Stakeholder Exclusion:** There are wide range of users and excluding some potential diverse needs might be possible such as elders and disable persons.

    d. **Legal/Regulatory Compliance:** There is risk of non-compliance with privacy laws as households are included.

2. **Mitigation Strategies:**
    a. Including diverse range of people from homemakers, business people to sports persons etc.
    b. Conducting focus groups with advocates and AI specialists.

3. **Technical Mitigation strategies:**
    a. Use of Lucidchart for detailed work flow through we can get continuous feedback from the users.
    b. Blurring out people faces in order to protect privacy.

Data Collection:

1. **Key Risks:**
    a. **Data Quality:** Prediction accuracy depends on the quality of image we take and the quality of the images also effects the model training a lot.

    b. **Bias in data:** Performance of the data might vary by different light conditions and sizes of the images.

c. **Data Privacy:** If fallen into wrong hands there might be a huge misuse of these data as it might contain personal data of the user.

2. Mitigation Strategies:
   a. Conducting regular bias assessments through which we can check wide range of lighting and camera placements.
   b. Implementing diverse data collection to include images under various conditions.
   c. Retaining only the required amount of data and deleting the unnecessary data to maintain privacy regulations.

3. Technical Mitigation Strategies:
   a. Implementing OpenCV methods to blur the faces of the user to protect user privacy.
   b. Using transformers to resize the image before feeding them for model training.

AI Model Development:

1. Key Risks:
   a. **Algorithm Bias:** There's a risk on the detection of objects may be biased on certain environments.

   b. **Explainability:** If users cannot understand why the objects are getting detected or missed can lead to mistrust.

   c. **Overfitting or Underfitting:** Sometimes models might fail to generalize among diverse household if training data or fine tuning isn't handled properly.

2. Mitigation Strategies:
   a. Representing underrepresented scenes using methods like data rebalancing or oversampling to avoid bias
   b. Integrating feedback mechanism where users can report misclassified objects or undetected scenarios.

3. Technical Mitigation Strategies:
   a. Using L2 regularization or dropout layers to prevent overfitting.

b. Use of SHAP, LIME, and Grad-CAM for model transparency which solves the risk of explainability.
c. Scikit-Learn for cross validation, hyperparameter tuning etc.
d. Foolbox to test robustness of the model for adversarial attacks.

AI Deployment:

1. Key Risks:
    a. **Integration Issues:** We might face integration challenges with the edge devices and hardware placement inside the house for maximum view.

    b. **Security Breaches:** Risk of unauthorized access to local camera feed.

2. Mitigation Strategies:
    a. Conducting A/B testing to check different configurations of model and test its performance in different user environments.
    b. Implementing CI/CD pipeline to automate the build, test and deployment stages while reducing human errors.

3. Technical Mitigation Strategies:
    a. Using devices like Nvidia Jetson, to reduce the latency by running the model locally instead of running them on cloud.
    b. In terms of security usage of Auth0 or Okta for API authentication, and SSL/TLS and AES for data encryption.

Monitoring and Maintenance:

1. Key Risks:
    a. **Model Drift:** Environmental changes can lead to model degradation over time.

    b. **Emerging Security Threats:** New vulnerabilities might arise these days due to rapid rise in technologies.

2. Mitigation Strategies:
    a. Using tools like **Prometheus and Grafana** to track the system health metrics such as CPU, GPU and memory usage etc.
    b. Maintaining detailed system logs of system access, changes in API etc.

3. Technicial Mitigation Strategies:
    a. Scheduling regular updates of libraries and dependencies used to prevent vulnerabilities from the outdated components.

Residual Risk Assessment:

1. Identify Residual Risks:
    a. **Algorithm Bias:** There is a chance that object detection could be skewed in particular settings.

    b. **Data Privacy Issues:** There is a risk of capturing personal or sensitive information.

    c. **System Reliability Issues:** Possibility of hardware or software crashes.

    d. **User Misuse:** Users might misuse or misunderstand the system, which could have unexpected effects.

Residual Risk	Likelihood	Impact	Matrix Category
Algorithm Bias	Possible	Tolerable	Yellow
Data Privacy Issues	Possible	Tolerable	Yellow
System Reliability Issues	Improbable	Tolerable	Green
User Misuse	Possible	Acceptable	Green

2. Mitigation Strategies:
    a. Low Risk:
        i. **User Misuse:** Educate the users on system limitations, data privacy and encourage feedback from the users to guide them further.
        ii. **System Reliability:** Conducting weekly maintenance and monitoring the connection. Enabling backup to secure the data.

    b. Moderate Risk:
        i. **Algorithm Bias:** Including diverse data through which the training data covers broader range of aspects and environments.

ii. **Data Privacy Issues:** Blurring out the faces of the user which helps maintain the privacy of the user.

Data Collection Management and Report

Data Type:

1. Type of Data:
    a. **Unstructured Data:** In this project of object detection, we need to retrain the current model using image data. To identify household items, we retrain the YOLO (You Only Look Once) model using unstructured object image data.
    b. **Data Granularity:** The data that we take initially is raw and it needs to be processed with various things like labeling and image resizing.

2. Challenges and Adjustments:
    a. **Data Quality:** There are few cases where we get affected by blurry and improper images with more lighting which led to noise data and improper trained model.
    b. **Data Size:** As there are many household items to train, we may get large datasets with various sizes.

Data Collection Methods:

1. Source of Data Collection:
    a. **Public Datasets:** We are using a public dataset from Kaggle which is named as "**Home Office Dataset for Domain Adaptation**". This dataset contains various images in the household.
    b. **Reliability:** The Kaggle website is reliable and there are few issues such as the requirement of internet accessing the website and downloading the data and also have few issues in data quality.

2. Methodologies Applied:
    a. **Direct Download and Batch Processing:** The dataset is downloaded directly from Kaggle in forms of batches that allow staged processing. It also ensures manageable storage and resources.

3. Ingestion for Training:
    a. **Data Loader with Batch processing:** This Data Loader helps images to load in batches which optimize the GPU memory and processing time. The images that are input are converted in a form that fits the YOLO model and leads to efficient training.

4. Ingestion for Deployment:
    a. **Real Time Streaming:** In deployment time the real time images are captured from CCTV footage or any other sensors to detect the objects. We use messaging queues such as RabbitMQ for real time processing and predictions.

Compliance with Legal Frameworks:

1. Applicable laws and Standards:
    a. **GDPR Compliances:** The project will adhere to GDPR principes such as minimization data and limiting the identical features.

2. Compliance Strategy and Results:
    a. **Anonymization and Consent:** All the individuals using this are informed about the CCTV monitoring within the premises. We use Anonymization methods such as mage masking such as blurring the human image to protect privacy. Regular audits ensure GDPR compliance, particularly regarding data storage and retention.

Data Ownership and Access Right:

1. Ownership and Access Control:
    a. **Data Ownership:** As we get data from public dataset that can be accessed easily by anyone. The Kaggle dataset usage follows Kaggle's terms, while any proprietary processing data belongs to the project team.

    b. **Access Logging and Authentication:** Authentication mechanisms, including two-factor access and secure logging, ensure that all data interactions are monitored. Regular access audits reinforce data security.

2. Lesson Learned:
    a. **Security Enhancements:** Implementing a role-based access approach has proven valuable in controlling access to sensitive data, improving overall data protection.

Metadata Management:

1. Metadata management and Management System:
    a. **Metadata Attributes:** Each image is tagged with relevant metadata, including source, timestamp, image resolution, and version information. Consistency in metadata standards is maintained to streamline data management.

2. **Challenges:** Initial metadata variations were standardized to improve data handling efficiency across the system.

Data Versioning:

1. Version Control System and Strategy:
    a. **System Used:** DVC (Data Version Control) tracks dataset updates, tagging each version for easy reference.

2. **Strategy:** Each batch download or dataset modification is versioned, ensuring transparency and enabling rollback to previous versions if necessary.

Data Preprocessing, Augmentation and Synthesis:

1. Preprocessing Techniques:
    a. **Normalization:** Pixel values are normalized to a range of 0 to 1 to ensure uniformity, enhancing model convergence.
    b. **Resizing:** Images are resized to 416x416 pixels to align with YOLO's input requirements, optimizing training and storage.
    c. **Dimensionality Reduction:** Reducing unnecessary data dimensions, like converting images to grayscale when color is not essential, decreases data size and improves model efficiency.

2. Data Augmentation:
    a. **Image Transformation:** Augmentation techniques (rotation, flipping, brightness adjustments) enhance dataset diversity, improving model generalization.

3. Challenges and solutions:
    a. **Quality Management:** Excessive resizing and augmentation led to image degradation initially. Experimenting interpolation methods helped balance image diversity and quality.

Data Management Risks and Mitigation Strategies:

1. Identified Risks and Mitigation Strategies:
    a. Data Resizing:
        i. Resizing images for model compatibility can lead to loss of resolution, which impacts feature clarity in smaller objects.
        ii. Mitigation strategies involve experimenting with interpolation methods to retain important details and resizing only to the extent necessary for model requirements.

Fig. 1: Original Image and Transformed Image

b. Data Augmentation:

i. Over-augmentation can lead to degraded image quality, making it harder for the model to generalize.

ii. This risk is managed by balancing augmentation techniques (e.g., rotation, scaling) to prevent excessive transformation, ensuring images remain useful for training.

Data Management Trustworthiness and Mitigation:

1. Data Privacy:

a. Blurring the human face helps it from privacy issues and also routine audits and restricted access protocols have proven effective. Access is restricted to authorized personnel, and records are audited to maintain accountability.

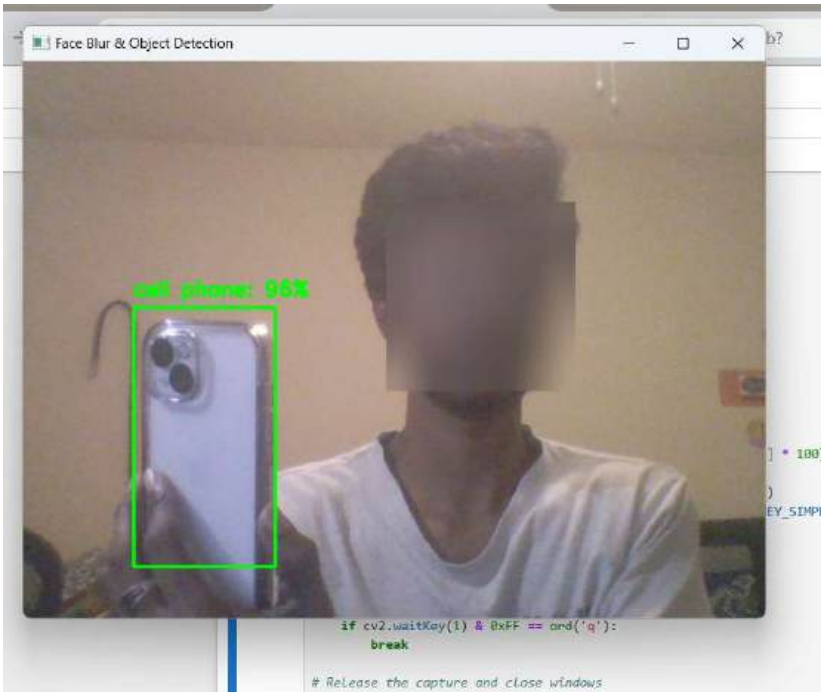


Fig. 2: Face blur while detecting objects

Model Development and Evaluation

Model Development:

1. Algorithm Selection:
    a. Chosen Algorithm:
        i. YOLO (You Only Look Once), in particular YOLOv8, was chosen for this project because of its superiority in speed and precision in real-time object detection tasks.
        ii. YOLO's structure makes it possible for it to classify many objects at once, which makes it advantageous for home environments with more than one likely item in a single frame.
    b. Rationale:
        i. YOLO is basically a real-time detector designed for edge devices that have constrained resources. As a single-stage detector, YOLO is capable of fast real-time performances, while also preserving high localization accuracy, which was concurrent with the project's aims of real-time supervised item learning.

2. Feature Engineering and Selection:
    a. Feature Engineering:
        i. The YOLO model auto-generates corresponding features, such as object edges, textures, and shapes, from unprocessed input image frames.
        ii. It also applies some suitable algorithms like resizing, normalization, and data augmentation (e.g., rotation, color adjustment) that strengthen invariance to lighting conditions and mastering diverse orientations of the environment among different users.
    b. Feature Selection:
        i. YOLO utilizes Convolutional Neural Networks (CNNs) and thus, hierarchies of representations of features are learned from image data, which means no feature selection is needed. Nevertheless, the model's ability to adapt to a wide variety of inputs makes its performance more generalizable, which is crucial for accuracy improvement.

3. Model Complexity and Architecture:
    a. YOLO Architecture:

i. **Backbone:** The backbone is CNN-based feature extractor (e.g., CSPDarknet53 for YOLOv8), which learns high-level features from the input images such as shapes and edges.
ii. **Neck:** The neck of the model uses feature pyramid networks (FPN) to combine features at different scales, which adds the model to be able to capture the objects of various sizes.
iii. **Head:** The head comprises the detection layers which are used to estimate the bounding boxes, the confidence levels, and the class probabilities of each object in the image.

b. **Model Complexity:**
   i. YOLO has designed its architecture to be optimized for real-time performance, thus making the network as complex as possible will not lead to slower inference.
   ii. YOLOv8's highly efficient architecture allows it to achieve high mAP and at the same time, it is operating with fast inference both on GPU as well as CPU, making it greatly suitable for home cameras processed data entry.

Model Training:

1. Training Process:

a. **Dataset Preparation:** The training dataset was the one which was drawn from a camera of the household, and objects were displayed by the manual drawing of the rectangular areas (bounding boxes) by the AI. The images were annotated with the labels of the objects, which were the class names and the bounding box coordinates of the objects in the frames.
b. **Data Augmentation:** The training data was altered using techniques such as rotation, noise adding, brightness adjustments, and blurness. These augmentations taught the model to recognize objects under different lighting conditions, orientations, and positions that are usually present in home environments.
c. **Training Duration:** This procedure, which focuses on image classification, was repeated 10 times with the use of the GPU, while the training process was supervised for the improvements in mean Average Precision on the validation set to avoid overfitting.

2. Hyperparameter Tuning:

a. Tuned Hyperparameters

i. **Learning Rate:** Several different levels of learning rates for the model were explored to find out the most optimal rate which is basically a trade-off between the learning speed, and the stability of learning. A lower rate of learning was the best outcome, nevertheless, it helped eliminate the possibility of overshooting during training.

ii. **Batch Size:** The batch size was extended to sixteen, following the tend to be the case principle, to make the gradients during the update stable, which allowed the Mean Average Precision (mAP) to be increased without substantial memory burdens.

iii. **IoU Threshold:** To optimize the accuracy of bounding boxes, the NMS was adjusted to a different IoU threshold, which ensured that the overlapping detections were properly suppressed, whereas the detections that were relevant were kept.

b. Tuning Process

i. Performed a grid search for the initial setting of the hyperparameters by actually trying out the different possible settings of the learning rate, batch size, and IoU threshold.

ii. Fine-tuned hyperparameters according to the mAP improvement trends of the validation set and also preferred the settings that were faster in inference but at the same time did not compromise the model performance.

Model Evaluation:

1. Performance Metrics:

a. Selected Metrics:

i. **MAP (Mean Average Precision):** We examine the model's detection performance at various Intersections over Union (IoU) criteria.It provides an accurate assessment of the model's ability to dependably discover and categorize things, making it very valuable for object detection tasks.

ii. Reliable detection indicates mAP@0.5 (IoU threshold of 0.5) measures object detection when the bounding box overlaps the genuine item by at least 50%.

iii. Strict assessment of the model's localization accuracy offers mAP@[0.5:0.95] averages mAP over IoU thresholds ranging from 0.5 to 0.95.

iv. **IOU (Intersection over union):** This measures the overlap between predicted and ground truth bounding boxes, which indicates how successfully the model localises items. A greater IoU implies improved detection accuracy, making it an important parameter for evaluating exact localization in real-time.

v. **Latency:** Latency evaluates how quickly the model can identify and detect the object location that is crucial for user experience in real-time.

2. Holdout Validation:

a. This type of validation consists of dividing the dataset into one part for training and one part for validation. Then, in the first one, we correctly train the model, and only after that, we assess its performance on the only validation set, which is fundamentally different.

b. Holdout validation is a simple and very often used technique for getting an initial estimate of model performance. Nevertheless, since it only uses one part of the data for testing, it might miss the model's performance on other possible variations of the dataset, which cross-validation, in contrast, averages out performance over multiple subsets.

Implementing Trustworthiness and Risk Management in Model Development

1. Risk Management Report:

a. Identified Risks and Mitigation Strategies:

i. **Data Privacy:** Risk of exposing personal data from household and right now we are not saving all the data that is being recorded.

ii. **Mitigation:** Data preprocessing techniques, such as resizing and augmentation, were applied to minimize personal data exposure while enhancing model generalization.

1. **Resizing:** Standardized the frame dimensions to ensure uniform model input, reducing potentially sensitive detail.
2. **Data Augmentation:** Data Augmentation helps us in creating new training scenarios with the help of shearing, rotating and noise adding.

iii. **Model Bias:** Unreliable performance in different aspects such as lighting and environment conditions we are achieving this by capturing images in different conditions.

iv. **Mitigation:** Adaptive learning and diverse data collecting across illumination configurations are used to increase accuracy in a range of environments.

2. Trustworthiness Report:
    a. Trustworthiness Considerations and Mitigation Strategies:
        i. **Face Blurring:** Individuals on the camera stream are not comfortable and do worry about their privacy when using Cameras.
            1. **Mitigation:** Implemented automated face blurring to anonymize individuals while preserving the integrity of the object detection task.
            2. **Outcome:** This way the user could be successfully safeguarded against the privacy issues.
        ii. **Fairness:** Equal model performance for all household setups.
            1. **Mitigation:** Collected a balanced dataset with diverse lighting and object placements.
            2. **Outcome:** The approach of transparency and fairness has rightfully been the most successful one with the same mAP score appearing under all the test cases. However, the future strategies could be incorporated into the AI training process which now needs less frequent training.

Apply HCI Principles in AI Model Development:

1. Develop Interactive Prototypes
    a. **Tool:** Gradio Library, to implement a simple web interface.
    b. **Strategy:** We working on a interface where the user can select an object and the system checks the environment with the camera and if it is not detected then it returns the image of the object when of its last seen.
    c. **Outcome:** The components that allow interaction were used by the users to provide useful insights, which helped the designer of the user interface to add functions that make the interaction between the user and the system more flexible and usable.

2. Design Transparent Interfaces
    a. **Tool:** Matplotlib and Plotly to visualize confidence scores of the detected objects and the mean average precision (mAP).
    b. **Strategy:** Additional visual feedback is included in the interface, with confidence scores and IoU metrics that each detected object has, thus users can judge the detection reliability.
    c. **Outcome:** The customers revealed that the transparent feedback time enabled them to understand the accuracy of the model quickly and, thus, they were more confident in it making the system reliable.

3. Create Feedback Mechanism

a. Strategies:

- Thumbs up/down and a text box where users express their opinions are added in the interface.
- Direct collection of feedback was a tool for the users to inform the detections by their feedback of which the feedback was also relieved for the retraining and improvement in future.

b. Outcome:

- This feedback implies the consumers think the new technological innovation, which has varied the way we interact with it, is very cool. Actually, they express that the model is not only useful but also exciting and interesting to them.

Deployment and Testing Management Plan

Deployment and testing form a great part of the AI life cycle that ensures the object detection system effectively, securely, and reliably functions in a real-world home setting. This phase includes selecting the deployment platforms, using all kinds of security measures, and performing rigorous testing for validating system performance. Stand behind a strong deployment and testing process; keep it as close to real-world operational requirements as possible and ensure seamlessness from development into production.

1. Deployment Environment Selection:

a. Deployment Type:

i. Local Deployment with Edge Capabilities

b. Platform:

i. Stream lit for user interaction on a home server or a dedicated device, such as Raspberry Pi or NVIDIA Jetson Nano.

c. Explanation:

i. Real-time object detection requires low latency, which is best supported by performing the processing locally.

ii. Streamlit allows for intuitive interfacing with detected objects and snapshots interactively.

iii. Augmented datasets through Roboflow are processed online to ensure privacy and efficiency when deployed locally.

2. Deployment Strategy:

a. Strategy:

i. The deployment is done locally using Docker to containerize the application, thus ensuring consistency and portability.

b. Tools:

i. **Docker** - Package applications into standardized units called containers.

ii. **TensorFlow Lite -** Provide the ability to perform predictions on an already trained model.

iii. **Streamlit -** Create interactive web applications for visualizing and exploring data.

iv. **RoboFlow -** To simplify the process of preparing the data.

c. Scalability:

i. Modular architecture allows easy addition of new devices or cameras without redesigning the system.

d. Additional Features:

i. **Snapshot Storage:** Latest detected object snapshots saved in the database, accessible through Streamlit UI.

ii. **Local Alerts:** Integrated notifications via Streamlit to alert users of new detections or flagged objects

3. Security and Compliance in Deployment

a. Security Measures:

i. Sensitive home images are stored locally, while data is encrypted both in rest and in transit.

ii. Provided secured login credentials for restricted access to the Streamlit interface.

iii. Provide audit trails in place to record system access and changes.

b. Compliance:

i. Compliance with data privacy laws by avoiding unnecessary cloud transfers.

ii. Audit trails implemented for user interactions and system changes.

4. CI/CD for Deployment Automation

a. Strategy:

i. Integrated GitHub Actions to automate builds, tests, and deployments for fast deployment cycles.

ii. Store previous stable versions in a repository so that rollbacks could be done quickly in case of deployment errors or performance issues.

iii. Continuous Integration guarantees that new model updates from Roboflow datasets are well-tested and deployed.

b. Tools:

i. Docker, MLFlow, Rollback mechanisms

c. Potential Enhancements

i. **Canary Deployment**: Introduce the model updates gradually by testing on a single camera and then deploy the changes across all devices.

5. Testing in the Deployment Environment

a. Testing Strategies:

i. **Functioning Testing:** Verify object tracking and snapshot saving through unit tests

ii. **Performance Testing:** Ensure real-time detection with latency below 0.2 seconds per frame using JMeter

iii. **Integration Testing:** Validate end-to-end operations from detection to UI interaction through Selenium

iv. **Scenario-Based Testing:** Create specific test cases based on potential home use scenarios, such as tracking moving objects, detecting overlapping objects, or low-light conditions.

b. Summary of Results:

i. System achieves efficient object detection for various household items

ii. Identified occasional false detections for similar-looking objects, requiring augmentation tuning in Roboflow

Evaluation, Monitoring, and Maintenance Plan

It's important to keep the system reliable after deployment. The tasks include model performance monitoring, collecting user feedback through the Streamlit UI, and retraining the model with fresh data from Roboflow.

1. System Evaluation and Monitoring

a. Metrics Tracked:

i. Accuracy Detection across object categories.

ii. Latency parameter for each detection event.

iii. Frequency of false positives/negatives logged through the UI.

b. **Tools:**

i. To track and live monitor the system's health, integrate a local dashboard using Grafana.

ii. Setup Alert notifications for detecting anomalies.

2. Feedback Collection and Continuous Improvement

a. **Mechanisms**:

i. Streamlit-based feedback forms allows users to flag incorrect detections.

ii. Feedback logs analyzed monthly to identify trends in detection errors.

b. **Continuous Improvement:**

i. Leveraging Roboflow's augmentation capabilities for better generalization.

ii. Combining YOLO with other lightweight models to improve detection accuracy in different difficulty scenarios.

iii. Retrain the model with new object categories requested by users to enhance functionality.

3. Maintenance and Compliance Audits

a. **Maintenance Schedule**

i. Weekly Review detection logs and retrain on flagged images.

ii. Monthly cleanups of the database and software updates.

iii. Quarterly system audit to validate security compliance and reliability.

b. **Compliance**

i. Periodic checks to ensure local storage remains secure and free of unauthorized access.

4. Model Updates and Retraining

a. **Process**

i. Data pipeline with Roboflow ensures monthly model updates, retraining, and seamless CI/CD redeployment.

b. **Tools Used:**

i. MLflow: Tracking model performance across retraining iterations.

DONE BY:

Jugal

Naveen