# **CAPSTONE**

**Computer Vision for Real-Time Monkeypox Diagnosis on Embedded Systems**

# **Introduction**

The COVID-19 pandemic not only highlighted the global disparities in healthcare access, particularly in developing countries with limited resources and infrastructure, but also highlighted the critical importance of accurate and timely diagnostic tools in managing infectious disease outbreaks [[1]](https://www.zotero.org/google-docs/?nzyPTM). With over 7 million deaths and countless cases worldwide [[2]](https://www.zotero.org/google-docs/?2tA6bE), this crisis underscored the urgent need for robust public health systems and innovative diagnostic tools [[3]](https://www.zotero.org/google-docs/?VqkF0F). Similarly, the recent outbreaks of monkeypox (Mpox), including the 2022 outbreak [[4]](https://www.zotero.org/google-docs/?KSHZUa) and the new strain in 2024 which has already resulted in over 15,600 cases and 537 deaths [[5]](https://www.zotero.org/google-docs/?etEPAo), have further exposed vulnerabilities in our ability to quickly diagnose and contain emerging infectious diseases. The symptoms of Mpox, such as fever, headache, muscle pain, and blistering rashes [[6]](https://www.zotero.org/google-docs/?fjJBRE), are often challenging to diagnose accurately, particularly in resource-limited settings where advanced diagnostic tools are not readily available. Artificial Intelligence (AI) systems provide an approach to tackle this issue as they have become a staple with their applications in infectious disease detection and diagnosis.

Traditional diagnostic methods, such as the Polymerase Chain Reaction (PCR) test, have shown limitations in accurately diagnosing Mpox due to the short window of virus presence in the bloodstream and variability in infection stages [[7]](https://www.zotero.org/google-docs/?HK9oRB). Moreover, the PCR test can take a few days of processing to produce results, and their availability in rural or remote areas is greatly hindered. This has led to a pressing need for novel diagnostic approaches that can provide accurate and timely results, especially in areas with limited access to healthcare infrastructure. Our project aims to address this gap by leveraging the capabilities of artificial intelligence (AI) and computer vision to develop a reliable, fast, and deployable diagnostic tool for Mpox.

AI, particularly machine learning (ML) and computer vision has shown great potential in revolutionizing healthcare diagnostics by enabling the analysis of complex medical data with high accuracy and speed [[8]](https://www.zotero.org/google-docs/?EhC2jq). In the context of Mpox, AI-driven models can be particularly valuable in resource-limited environments, where access to expert medical professionals and advanced diagnostic tools is scarce. By utilizing models and deploying them on embedded systems, we can create a diagnostic tool that is not only accurate but also efficient and accessible, addressing the critical need for rapid and reliable Mpox diagnosis in vulnerable populations.

We propose developing an automated diagnostic tool using advanced computer vision techniques to meet the urgent need for accurate and rapid Mpox diagnostics. Our approach will leverage a model with a pre-trained deep-learning architecture known for its efficiency and lightweight structure [[9]](https://www.zotero.org/google-docs/?Dp2KiI). Then, we would apply transfer learning with a monkeypox skin lesion dataset to fine-tune the model specifically for Mpox detection. By integrating NVIDIA's TensorRT framework, we will optimize this model for deployment on the NVIDIA Jetson Orin, a compact yet powerful embedded system. This novel solution is designed to be both efficient and scalable, making it ideal for use in resource-constrained environments where timely and accurate diagnosis can significantly improve public health outcomes.

# **Problem Statement**

Access to fast and reliable diagnostic tools for infectious diseases is critically limited, particularly in rural and underserved areas where basic healthcare infrastructure is often insufficient. This gap in healthcare access hampers the ability to predict, diagnose, and contain infectious diseases, increasing the risk of localized outbreaks escalating into global pandemics. A pressing example is the emergence of the new Mpox variant, which is currently causing significant public health challenges in Africa and spreading to parts of Europe and Asia [[10]](https://www.zotero.org/google-docs/?fcwnvd). The limitations of the current PCR test, including varying accuracy based on the disease's life cycle and prolonged wait times for results, highlight the urgent need for innovative diagnostic methods that can overcome these constraints. Failure to address this need could result in delayed diagnoses, ineffective containment efforts, and worsened health outcomes on a global scale.

Our project targets the healthcare domain, with a specific focus on professionals operating in resource-limited settings such as rural clinics, mobile health units, and temporary healthcare facilities in outbreak zones. These healthcare providers often face significant challenges in diagnosing infectious diseases due to the lack of advanced diagnostic tools and limited access to specialist support. By equipping these professionals with a user-friendly, real-time diagnostic tool, we aim to empower them to make accurate and timely decisions, ultimately improving patient outcomes and reducing the spread of Mpox in vulnerable communities.

To address the limitations of current diagnostic methods, we propose leveraging artificial intelligence (AI) and computer vision technologies to develop a real-time diagnostic tool specifically designed for Mpox infection. This innovative approach involves using a computer vision model to analyze images of skin lesions, a common symptom of Mpox, to assess the likelihood of infection. Unlike traditional methods [[11]](https://www.zotero.org/google-docs/?1qqqas), our solution will employ a lightweight, efficient model that can be deployed on embedded systems, making it suitable for use in resource-constrained environments. This innovation is poised to significantly enhance the accessibility and accuracy of Mpox diagnosis, providing a critical tool for healthcare professionals in underserved areas where rapid and reliable diagnosis is most needed.

Our project aims to make a meaningful contribution to the healthcare domain by providing a cost-effective, efficient diagnostic solution that can be deployed in the most challenging environments. By focusing on Mpox, a disease with growing global relevance, we seek to enhance the accessibility of accurate diagnosis in underserved areas, thereby improving public health outcomes through early detection and timely treatment. Additionally, this project will advance the application of AI in healthcare, demonstrating the potential of computer vision models to transform the way infectious diseases are diagnosed, particularly in settings with limited resources.

# **Project Objectives**

**Project General Goal**

This project aims to develop an AI-powered diagnostic tool that accurately identifies monkeypox (Mpox) from skin lesion images. This tool will be deployable in resource-constrained environments, providing healthcare professionals with a reliable, real-time solution to improve disease detection and containment efforts.

#### **Objective 1: Identify a Relevant Dataset**

* **Specific**: Identify a Mpox skin lesion dataset suitable for training the computer vision model.
* **Measurable**: The dataset should be open-source and have a usability score of at least 7.0 on Kaggle.
* **Achievable**: Review and compare available datasets and relevant literature to ensure the selected dataset meets the usability requirements.
* **Relevant**: Selecting a high-quality dataset is crucial for effectively training the model and minimizing the risk of overfitting or underfitting.
* **Time-based**: This objective should be completed by September 6.

#### **Objective 2: Identify a Lightweight Pre-Trained Model**

* **Specific**: Identify an efficient pre-trained model that can be adapted for deployment on embedded systems.
* **Measurable**: The selected model should be capable of achieving a baseline accuracy of at least 85% on initial tests.
* **Achievable**: Utilize an existing state-of-the-art model, such as MobileNet or EfficientNet, as the foundation for this project.
* **Relevant**: Choosing the right model is crucial as it will significantly influence the final accuracy, efficiency, and deployment feasibility, thus reducing development time.
* **Time-based**: This objective should be completed by September 6.

#### **Objective 3: Model Development and Accuracy**

* **Specific**: Develop a computer vision model using a pre-trained architecture, fine-tuned with a Monkeypox Skin Lesion Dataset.
* **Measurable**: Achieve a model accuracy of at least 90% in correctly identifying Mpox from skin lesion images during tests.
* **Achievable**: Leverage data augmentation and transfer learning techniques to enhance model performance, ensuring it meets the accuracy requirement.
* **Relevant**: The accuracy of the model is crucial for the tool’s effectiveness in real-world scenarios, particularly in resource-limited settings where misdiagnosis can have serious consequences.
* **Time-based**: Complete model development and achieve the accuracy benchmark by September 15.

#### **Objective 4: Model Optimization for Deployment**

* **Specific**: Optimize the trained model using NVIDIA's TensorRT framework to ensure it is lightweight and capable of real-time processing on the NVIDIA Jetson Orin platform.
* **Measurable**: Reduce the model's average inference time to under 500 milliseconds while maintaining the accuracy threshold.
* **Achievable**: Utilize TensorRT's model compression techniques and hardware-specific optimizations to meet performance goals.
* **Relevant**: Ensuring the model is optimized for real-time deployment is critical for its use in environments where computational resources are limited.
* **Time-based**: Achieve this optimization and deploy the model on the Jetson Orin by October 30.

#### **Objective 5: Web Application**

* **Specific**: Create a web-based user interface (UI) to display the results of the AI inference generated by the Jetson Orin.
* **Measurable**: The application should display inference results (e.g., Mpox diagnosis probability) within 5 seconds of receiving data from the Jetson Orin.
* **Achievable**: Utilize common web frameworks like Flask or Django for backend integration, and HTML/CSS/JavaScript for the frontend, ensuring compatibility with the Jetson Orin’s system architecture.
* **Relevant**: This web application will serve as the primary interface between the AI model and the end-user, providing a clear and accessible way to view diagnostic results.
* **Time-based**: Achieve the web application by November 15.

#### **Objective 6: Wifi Access Point (AP) Hotspot**

* **Specific**: Set up the Jetson Orin as a WiFi AP to allow devices to upload skin lesion images directly to the model for analysis.
* **Measurable**: The system should successfully transfer skin lesion images from a device to the Jetson Orin within 5 seconds.
* **Achievable**: Use the built-in WiFi capabilities of the Jetson Orin and configure the device as an AP, ensuring seamless image transfer between a device and the embedded system.
* **Relevant**: Establishing a WiFi AP hotspot allows the system to function in remote areas without needing external network infrastructure, thus improving the deployability of the solution in resource-constrained settings.
* **Time-based**: Interface web application with Jetson Orin and run final tests by December 6.

# **Solution Approach**

We propose an AI-powered automatic diagnosis tool for detecting monkeypox (Mpox) through image analysis. This solution leverages a machine learning model, specifically utilizing computer vision techniques to classify images of skin lesions. By compressing the model and deploying it on an embedded system, such as the Jetson Orin, the diagnostic tool can perform real-time analysis of images captured via a phone camera or regular camera connected to the embedded system. This setup enables rapid, on-device diagnosis without the need for high computational resources or access to centralized labs. The solution is ideal for resource-constrained environments, such as low-income or rural communities, making it a highly accessible option for global healthcare systems.

For the model, we’re considering using either MobileNet or EfficientNet. They’re both models trained by Google, small in scale compared to other models, and have high accuracy. EfficientDet has good detection and accuracy capabilities, whereas MobileNet is more focused on accuracy [[12]](https://www.zotero.org/google-docs/?GsIAHE). The team is leaning towards MobileNet since, as the name suggests, it is made to be deployed on mobile devices. This will make it easier to compress and deploy in an embedded system.

We are inclined to use the **Jetson Orin** as the embedded system to deploy the machine learning model. One of the primary reasons for considering this platform is its integrated GPU, which allows us to leverage **TensorRT** for model compression, optimizing performance without requiring significant computational resources. The Jetson Orin is also well-suited for AI applications, making it an ideal choice for real-time image processing. However, there are risks associated with this approach, notably the possibility that the model may not perform as expected after compression. Despite utilizing TensorRT for optimization, there remains the chance that the compressed model's accuracy or performance might degrade to a point where it is no longer viable for reliable diagnosis.

To demonstrate the solution, we are exploring multiple options for how the system will interact with users. One approach is to develop a **server-based web application** that establishes a connection between the model and the user. In this scenario, a user, potentially accessing the system via a smartphone with a camera, would upload an image of a skin lesion. This image would be sent to the server, processed by the AI model, and the diagnosis result would be displayed on the web application. This solution provides flexibility and scalability, as it allows remote access to the diagnostic tool.

Alternatively, we are considering directly connecting the Jetson Orin Nano with **Wifi AP Hotspot** to create a more localized, real-time solution. Another possibility is to use Bluetooth. While Bluetooth communication does exist and has been tested for file transfer, we are still uncertain whether it can reliably support the volume and speed of data transfer needed for image diagnosis in this context. Given these uncertainties, a **direct connection to a compatible camera** might be the simplest and most straightforward approach for initial testing and demonstration, as it eliminates the potential complications of wireless communication.

We are also evaluating other hardware options, such as the **Arduino Nano**, but are not entirely sure of its efficacy due to its lack of a GPU. The Jetson Orin has a more established track record for handling machine learning tasks, particularly those requiring computer vision, making it a possible better fit for our use case.

Currently, the gold standard for monkeypox detection is the Polymerase Chain Reaction (PCR) test. However, the PCR method has significant drawbacks:

* **High cost per PCR test**: Each test can cost approximately $175 [[13]](https://www.zotero.org/google-docs/?9BNJ00), making it expensive, especially in low-resource settings where healthcare budgets are constrained.
* **Long wait times for results**: PCR test results take anywhere from 3 to 7 days, which can delay patient care, treatment, and isolation decisions [[14]](https://www.zotero.org/google-docs/?WQF3HV).
* **High false positive rates**: PCR tests often suffer from high false positives due to human error and the lack of universal standards on how to conduct the tests [[14]](https://www.zotero.org/google-docs/?nwpiF6). The variability in testing procedures across laboratories contributes to inconsistencies and inaccuracies.

Our solution addresses these shortcomings by offering a faster, more cost-effective method for diagnosing monkeypox with the potential to reduce human error through automated image-based analysis. The key value propositions of our solution are as follows:

* **Accessibility**: By compressing the model to run on an embedded system, this tool becomes accessible in resource-constrained environments, expanding its reach to areas with limited access to advanced healthcare facilities.
* **Reduced wait time**: Our AI-powered tool offers a significant reduction in the time required for diagnosis, providing results in real-time, as opposed to the 3-7 day wait time for PCR results. It could improve treatment outcomes and reduce the risk of spread in outbreaks.
* **Cost-efficiency**: The cost per test using this technology would be significantly lower than the PCR tests, making it an affordable solution for both healthcare providers and patients.
* **Embedded system integration**: The ability to communicate with a smartphone via Wifi AP hotspot or similar connectivity allows seamless integration with existing technology platforms, making it user-friendly for healthcare professionals.

These factors together make the proposed solution a more efficient, scalable, and cost-effective alternative to existing methods. However, it isn’t without its limitations and risks.

* **Model performance after compression**: Compressing the model for deployment on the embedded system may lead to reduced efficiency, resulting in a potential decrease in the accuracy of diagnosis.
* **Risk of incorrect diagnosis**: Although the solution will be subjected to rigorous standards to meet certain diagnosis accuracy, it is almost certain that the accuracy will not be perfect, which can lead to harmful false negatives.
* **Regulatory hurdles**: Achieving regulatory approval (such as from the FDA or other global health authorities) may be a challenging process. Compliance with medical standards for AI-driven diagnostics will be critical for market entry.

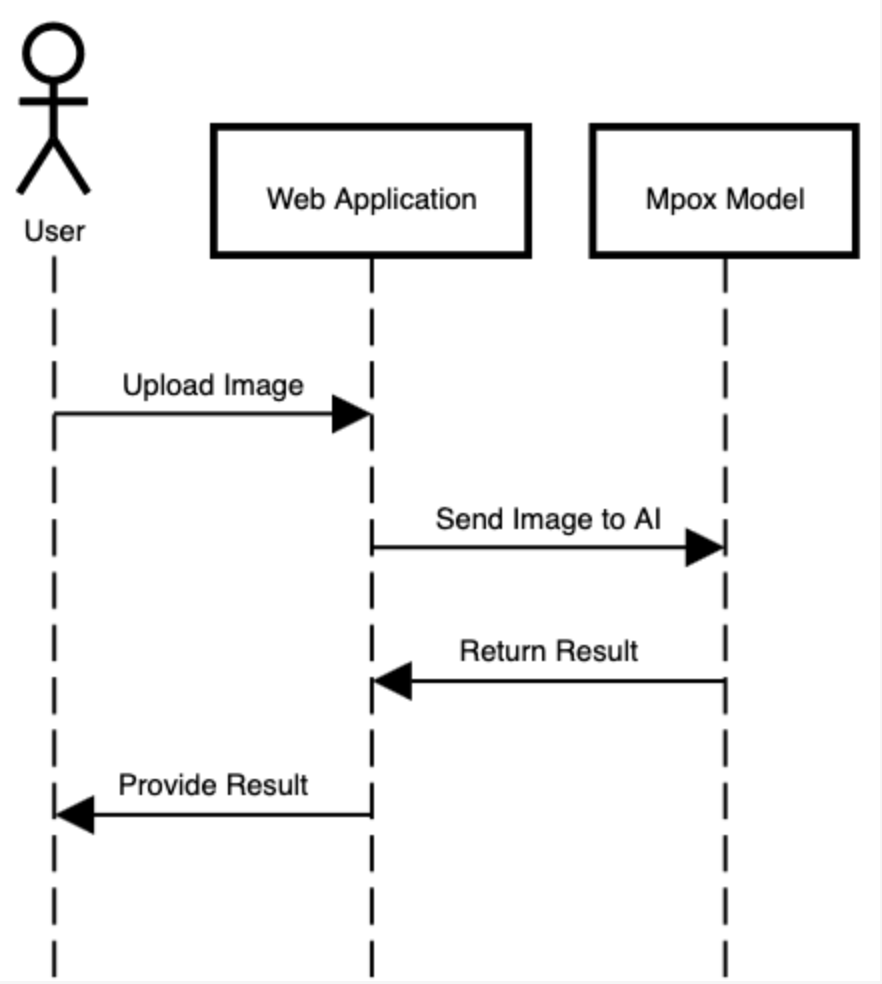
Monkeypox has seen a resurgence in recent years, with notable outbreaks, including one confirmed in the UK in May 2022 [[15]](https://www.zotero.org/google-docs/?Bxw1jL). This has highlighted the urgent need for reliable, rapid diagnostic tools. The growing demand for improved diagnostic solutions in healthcare provides a strong commercial opportunity for our AI-powered tool. As of December 2023, the PCR test remains the only available diagnostic tool for identifying monkeypox. However, its drawbacks, including high costs and long wait times, create a market gap that our solution can fill. By offering a faster, more affordable alternative, we can help healthcare providers diagnose and treat monkeypox more efficiently.

Our project has the potential to generate several intellectual property assets, including the model compression pipeline for deploying AI models on embedded systems and the web application interface designed for real-time monkeypox diagnosis. To manage intellectual property concerns, we will adhere to the University of Puerto Rico's intellectual property policies [16]. This will ensure that any IP generated during the project is appropriately documented and ownership rights are clearly defined. Additionally, in order to protect and potentially commercialize our work in the future, we will explore options for filing patents or other IP protections for project components that demonstrate significant innovation.

# **Technical Description**

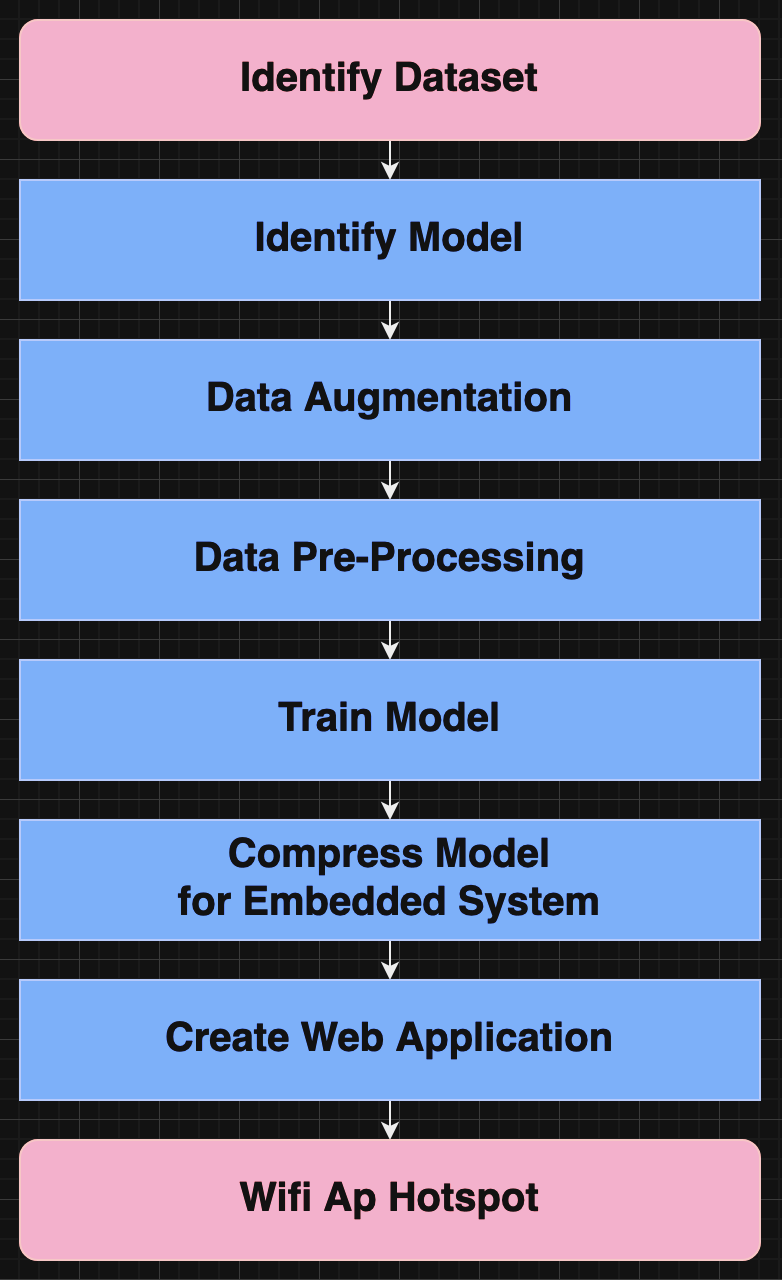
Previous work has produced AI models that can accurately predict disease based on skin lesion images with high precision. This is also true for MonkeyPox as there has been work utilizing different datasets and AI models for diagnosis. However, as far as we are concerned, this is the first work that will compress these models for deployment on embedded systems.

The architecture of the system will start by taking an image of the skin as input to the model. This image will be given by the user from the web application and be able to choose which image the user wants to verify if it has monkeypox or not. Then the model will determine if the image has Mpox by going through the compressed model in the embedded system that we choose. This will return the user if the image was found to be Mpox by the model. This architecture can be observed in Figure 1.



**Figure 1: System Architecture**

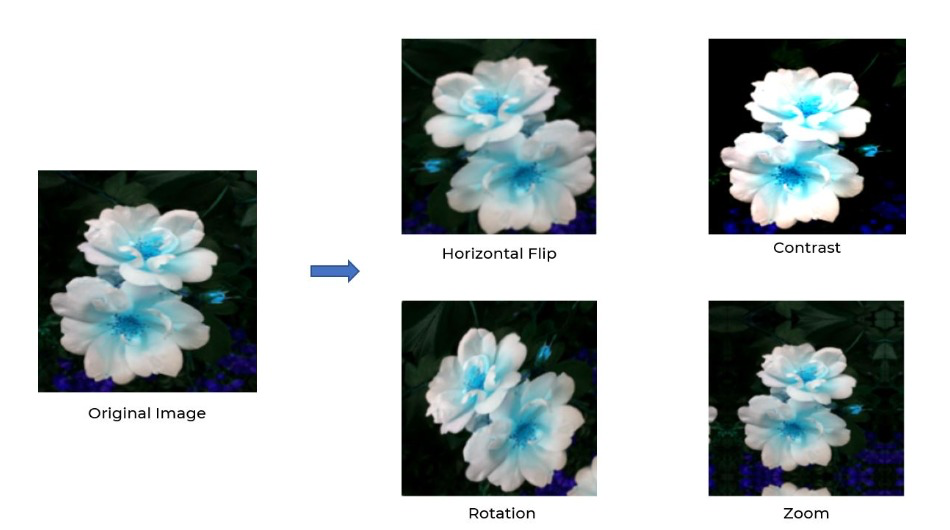
Multiple steps need to be completed to create this interface as described in Figure 2.



**Figure 2: Modules to Implement**

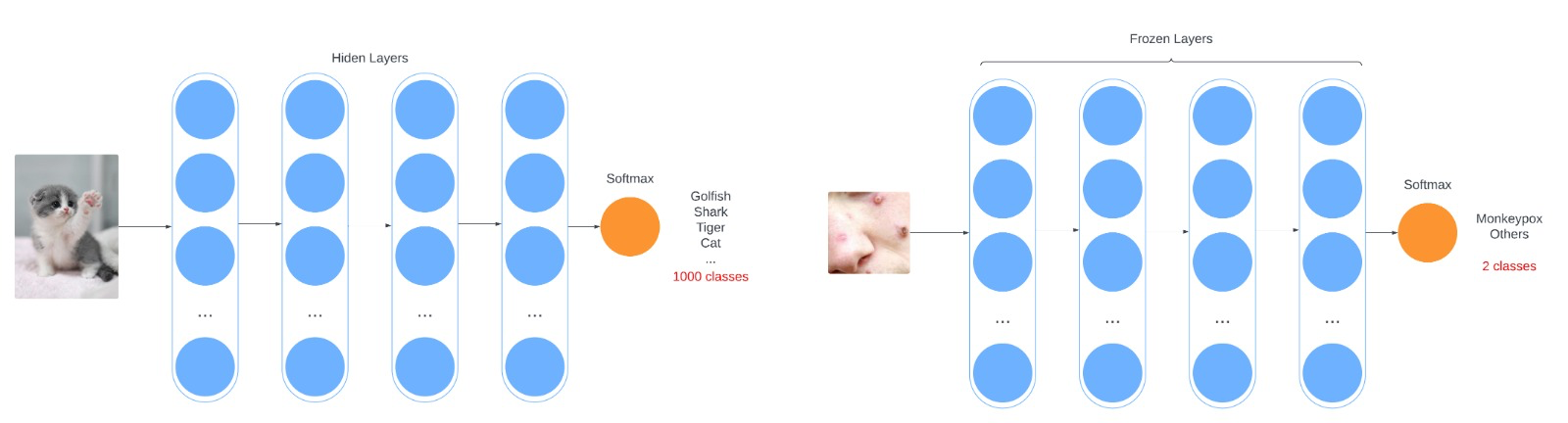
The first two steps involve selecting an appropriate dataset and identifying a pre-trained model for transfer learning. A major challenge is finding a high-quality dataset; failure to do so won’t allow the model to learn the appropriate features to detect Mpox. To ensure this, the dataset will be sourced from platforms like Kaggle or other reliable sources with high usability ratings. Once the dataset is chosen, we will proceed with selecting a pre-trained model suitable for transfer learning. Data augmentation techniques will then be applied to enhance the dataset. This is a critical step to ensure that the model can generalize well to different scenarios.

Data augmentation will involve manipulating the original dataset by applying various transformations, such as flips, rotations, zooms, and contrast adjustments, as presented in Figure 3. This process increases the dataset’s size and variability, allowing the model to better recognize images in various orientations and lighting conditions. These techniques help the model improve its ability to correctly classify images under different circumstances, thereby reducing the risk of overfitting. However, it is important to balance the use of data augmentation, as too many transformations will make the model learn unnecessary features and it won’t generalize well for real-world use.

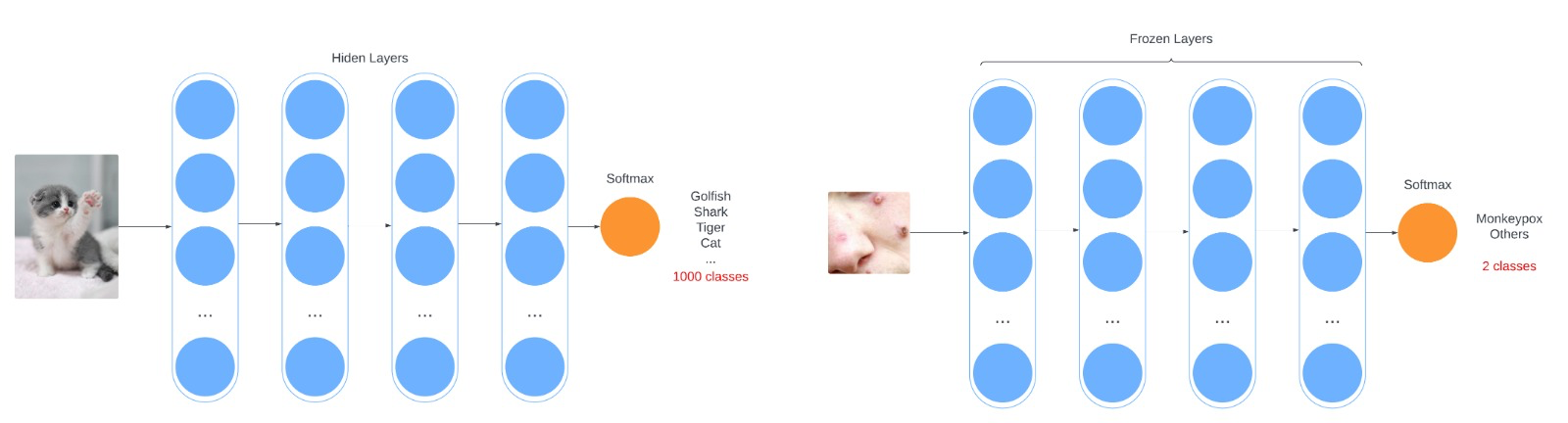
****

**Figure 3: Data Augmentation Example**

Before training, the data will undergo preprocessing, including scaling and normalization, to ensure consistency with the model’s input format. This preprocessing will be performed using Keras and TensorFlow libraries. Properly preprocessed data is essential for the model’s accuracy and performance. Then, transfer learning will be employed to fine-tune a pre-trained model using the selected dataset. This involves removing the final layer of the pre-trained model and retraining it with our dataset, allowing the model to adapt to monkeypox detection as described in Figures 4 and 5. This process will also be done using Keras and TensorFlow libraries. Transfer learning is an efficient way to train a model, leveraging existing knowledge to reduce the amount of training data and time required. Since we are tackling a medical application, the major hurdle is achieving high precision metrics after training, as misdiagnosis can be critical for patient health.

****

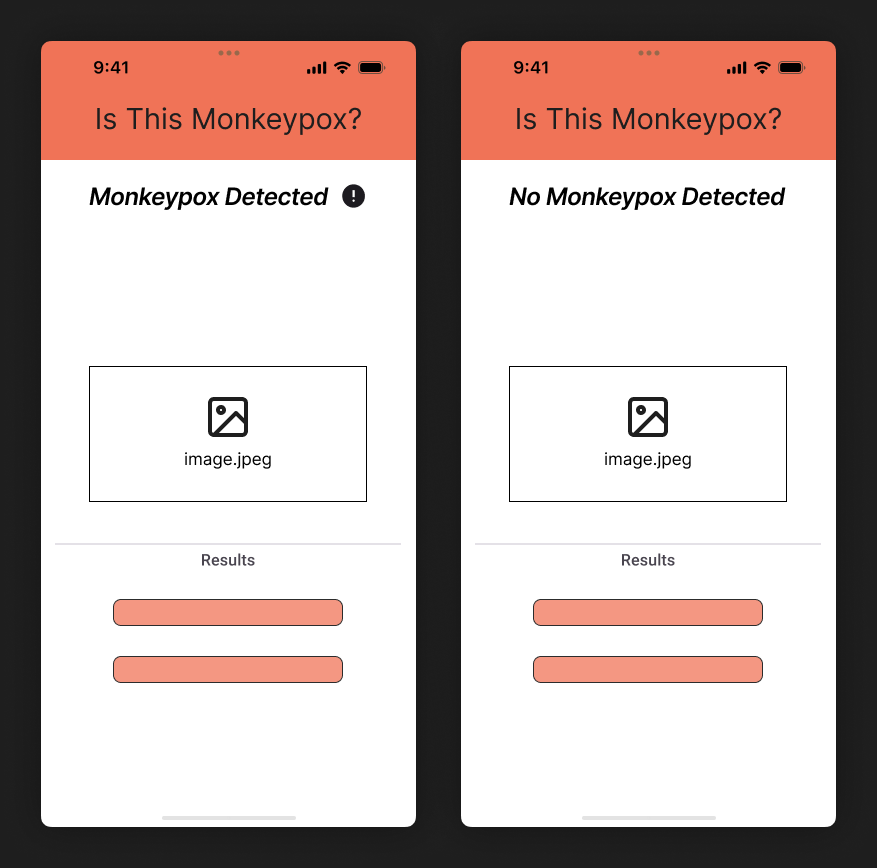
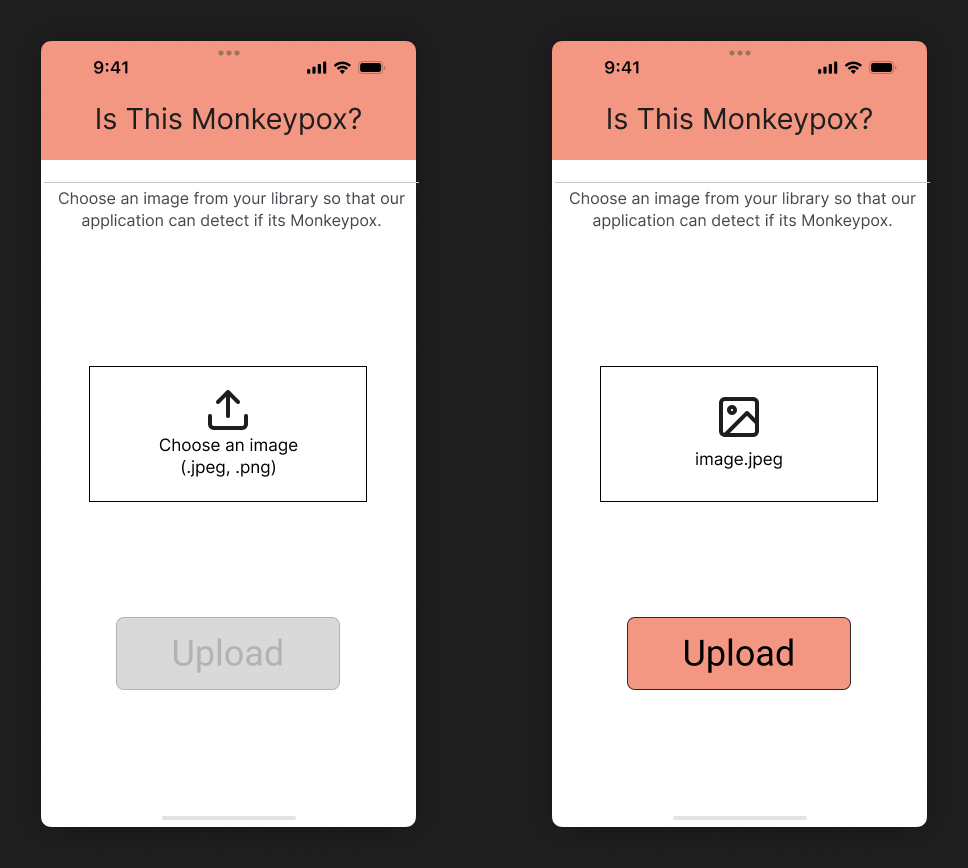
**Figure 4: Original Model**

****

**Figure 5: Transfer Learning**

Once trained, the model will be compressed using NVIDIA’s TensorRT framework to optimize it for real-time inference on embedded systems like the Jetson Orin. This compression is necessary to ensure the model can run efficiently on limited hardware. While TensorRT is well-integrated with the Jetson Orin, a potential constraint is that some models may not compress well due to their size, or compatibility issues may arise with other embedded systems. However, Jetson Orin is preferred due to its native support for TensorRT, making it an ideal choice for this project. Other alternatives like Arduino Nano were considered but may pose compatibility challenges. The possible constraint is the model precision deprecating too much after compression or the power consumption being too great, making it unsuitable for use in diagnosis. Additionally, even after compression, the model might require too many resources not available in the Jetson Orin, making it impossible to deploy.

After the model is deployed, a web application will be created to serve as the user interface. This interface will allow users to upload skin lesion images and receive predictions from the AI model. The web application will be developed using HTML, CSS, and JavaScript to ensure simplicity and responsiveness. The interface will display the model’s prediction in a clear, user-friendly format. Wireframes similar to Figure 6 will be designed to outline the layout of the web application, ensuring that the user experience is intuitive. A possible limitation is the web server not being able to handle the model even after compression, leading to long inference times or the web application crashing.



**Figure 6: Wireframes for Web Application Layout**

To facilitate image input, the Jetson Orin will be configured as a WiFi access point (AP) [[17]](https://www.zotero.org/google-docs/?UJDW6F), allowing mobile devices to connect and upload images directly. This approach enables real-time interaction between the mobile device and the AI model without requiring external network infrastructure. Alternative methods such as Bluetooth or an integrated camera in the embedded system were considered but were deemed less reliable due to potential connectivity issues. The success of the WiFi AP configuration will be tested to ensure it works seamlessly with the Jetson Orin. The challenge is that the Wi-Fi AP Hotspot might not be possible with the Jetson Orin, forcing us to find another alternative for hosting the web application.

Lastly, to ensure the project adheres to high-quality standards, we will follow several established engineering standards. Usability will be assessed based on **ISO 9241-11:2018**, which measures how well the solution meets real-world user needs. Development will follow the agile methodology in accordance with **ISO/IEC/IEEE 12207:2017**, ensuring iterative and flexible project management. For the web application, we will apply the **ISO 9241-210:2019** standard to guarantee simplicity, usability, and user-friendliness.

# **Project Plan**

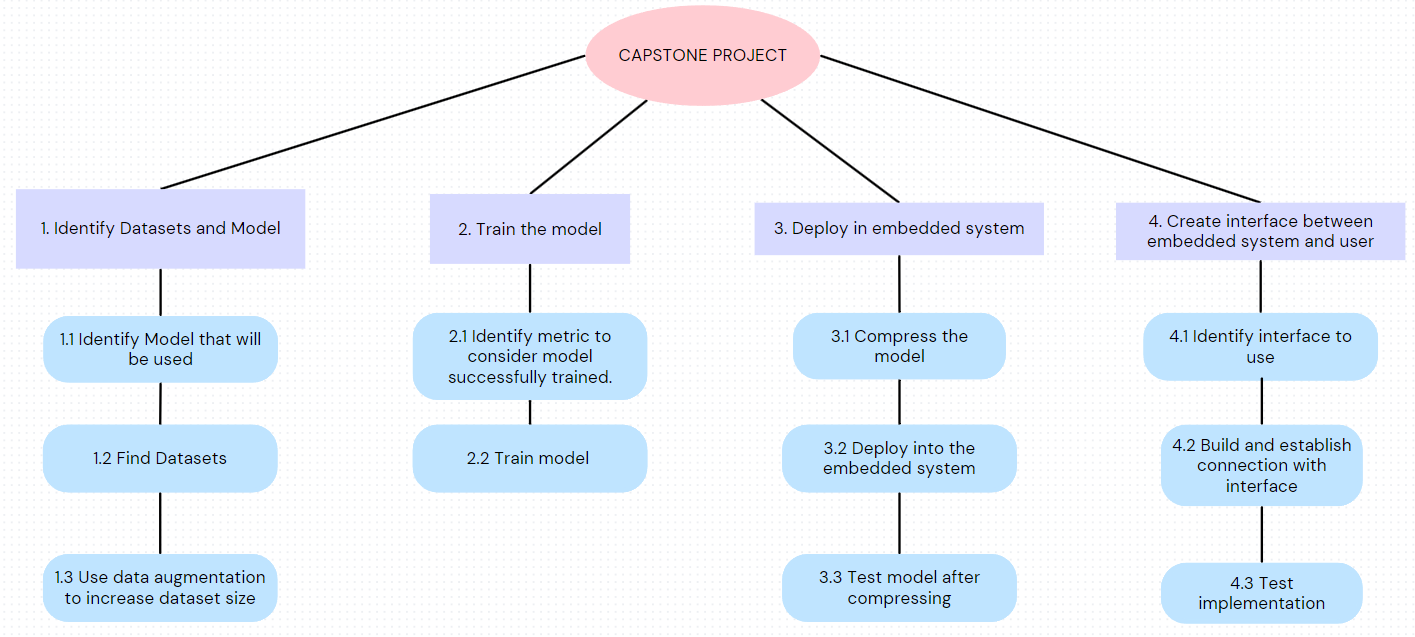
Our team will be working together throughout the project; therefore, our roles are best described as FullStack. Everyone will be part of every section of the project. However, there are sections of the project where a member will have primary responsibility:

**Jacob M Delgado:** Will be responsible for the identification of the model, as well as the augmentation of the dataset and training the model. Will ensure that the model is optimized and precise before and after compression.

**Sebastian O Espinosa:** Will work on the compressing of the model, deployment into the embedded system and creation of the web interface, making sure it satisfies the predefined requirements after compressing and runs as expected on the website.

**Ricardo A Morell:** Will set up the Wi-fi Hotspot for the chosen embedded system (Jetson Orin Nano), ensuring that the website and model runs smoothly on it. Will also monitor the power consumption when running the model and website to avoid excessive power usage.

The following Work Breakdown Structure (WBS) in Figure 7 shows what our milestones are, and the tasks that must be completed for each milestone for the project to be considered completed.



**Figure 7: Work Breakdown Structure (WBS)**

The second level, or purple level, outlines the milestones of the project. The third level, or the blue level, outlines the tasks that must be completed to consider the respective milestone as done. Important to note that the tasks are placed in order of getting completed since most of them are sequential.

The first milestone is about identifying the dataset and model that we will be using. As stated previously, we are considering using either MobileNet or EfficientDet. Once we choose the model, it’ll be easier to find a dataset more tailored to the model. For the dataset, we plan on searching through Kaggle and identifying one with at least a 7.0 rating. Once we have the dataset, we’ll use data augmentation to increase the size. Data augmentation is the practice of doing operations on the images in the dataset, such as adding zooms, rotations, blurs, etc. The main reason is that when we train the model, it isn’t too sensitive when identifying Monkeypox and it’s more broad when analyzing the image. We expect this to be done by **September 6, 2024.**

The second milestone is about training the model chosen. The team will be using Transfer Learning, which would take the pretrained model that we choose and train it to identify Monkeypox by image analysis. We would choose a metric to establish when the model will be considered trained. Ideally, we would choose among the known ML Metrics such as Recall, Accuracy, Precision and F1 Score. The decision on which metric to choose will depend on what our target demographic would value more to consider our solution viable. We expect to have the model fully trained by **September 15, 2024**.

The third milestone is focused on compressing the trained model to an embedded system. We expect a decrease in key metrics such as accuracy or precision after compression. Therefore, our main goal in this milestone is to successfully compress the model while losing the least amount possible of key metrics and still be a viable solution for Monkeypox detection. We expect to finish this milestone around **October 30, 2024**.

For the final milestone, we plan on building an interface between the embedded system model and the potential user. As stated previously, we are considering the creation of a web server application, a Wifi AP Hotspot connection between a phone and the system, or a simple camera connected to the embedded system. After deciding what implementation we’ll be doing, the team will begin to develop it. We expect that the interface should be able to capture an image and correctly identify if it’s Monkeypox or not. The milestone should be completed by **December 6, 2024**.

# **Updated Design Elements**

#### **Completed Objective 1: Identified Skin Lesion Dataset**

We sourced our dataset from Kaggle, a well-known open-source platform for high-quality datasets. During this process, we evaluated three potential datasets for training our model: the [Monkeypox Skin Lesion Dataset (MSLD)](https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset), [Monkeypox Skin Lesion Dataset version 2.0 (MSLD v2.0)](https://www.kaggle.com/datasets/joydippaul/mpox-skin-lesion-dataset-version-20-msld-v20), and the [Monkeypox Skin Image Dataset (MSID)](https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset).

After careful analysis, we selected MSLD, which had a usability score of 7.65. Although MSLD v2.0 offered a higher usability score of 9.41, it lacked supporting literature or peer-reviewed studies utilizing it, making it a less reliable choice. In contrast, MSLD had been referenced in multiple peer-reviewed publications, demonstrating its reliability and acceptance in the scientific community. This ultimately informed our decision to prioritize robustness over usability scores alone.

#### **Completed Objective 2: Identified Pre-Trained Model**

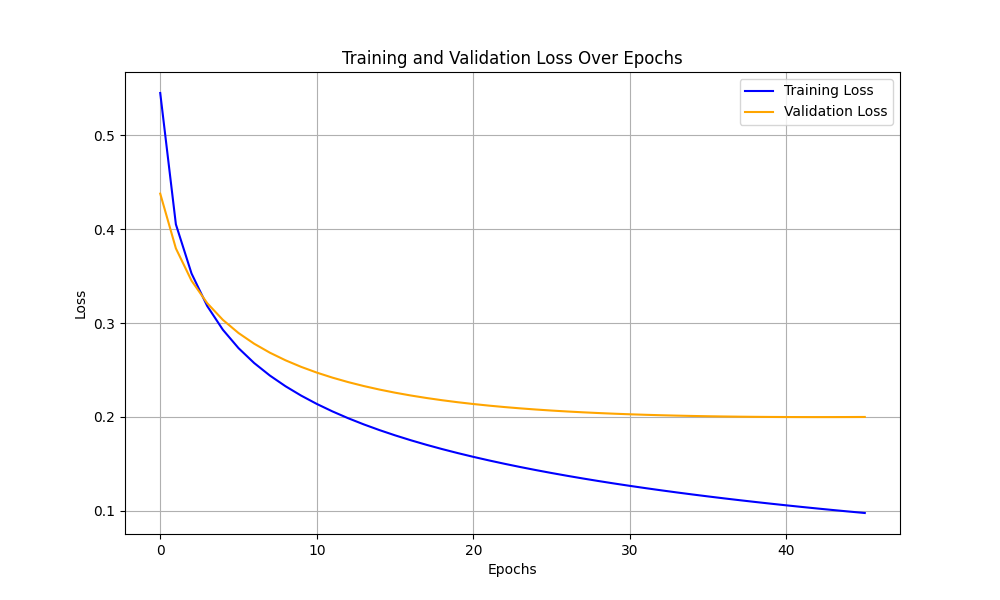
We conducted an extensive search for open-source models widely used in image classification, with a focus on models suitable for deployment on embedded systems. Among the models considered were MobileNet, ResNet, and EfficientNet, each offering distinct advantages in terms of accuracy, efficiency, and deployment capabilities.

After a thorough literature review, we selected MobileNet as the most appropriate model for our use case. MobileNet stood out due to its optimized architecture for mobile and edge devices, which aligns with our project’s goal of deploying the model on the Jetson Orin embedded system. Additionally, MobileNet has been proven in numerous real-world applications to balance computational efficiency and high accuracy, making it ideal for use in resource-constrained environments. Its relatively lightweight structure reduces both memory and processing power requirements, ensuring smooth operation on embedded devices without sacrificing performance.

#### **Completed Objective 3: Model Training**

We successfully trained the model using the pre-trained MobileNetV2 architecture, focusing on achieving a baseline model with high precision to maximize patient outcomes. Given the small and imbalanced nature of our dataset, we identified overfitting as a primary concern during initial trials. To mitigate this, we employed several key techniques. First, we used the Keras callback library to implement early stopping, which halted training when validation loss stopped improving, and model checkpoints, which saved the model with the best validation precision during training. These measures helped prevent the model from overfitting and ensured that we preserved the most performant version.

Additionally, we applied cross-validation to enhance the model’s generalization capabilities. This technique divided the dataset into multiple folds, allowing us to train and validate the model across different subsets of the data. By doing this, we ensured that our model’s performance was not limited to a specific partition of the data but was instead able to generalize well across the entire dataset. These strategies collectively helped us avoid overfitting and achieve a robust baseline as shown by the loss curve in Figure 8.

****

**Figure 8: Training and Validation Loss Over Epochs**

#### **Completed Objective 4: Model Optimization**

Our team optimized the model through TensorRT's post-training quantization process, implementing FP32, FP16, and INT8 precision formats. These optimizations significantly reduced the model size while enhancing both inference speed and throughput. The improvements proved essential for memory management within our Jetson system, particularly when handling multiple concurrent inference requests in our resource-constrained edge environment. By leveraging TensorRT's mixed-precision capabilities, we maintained consistent performance levels, a crucial factor for our healthcare application where accuracy cannot be compromised.

#### **Completed Objective 5-6: Web Application and Wifi Access Point (AP) Hotspot**

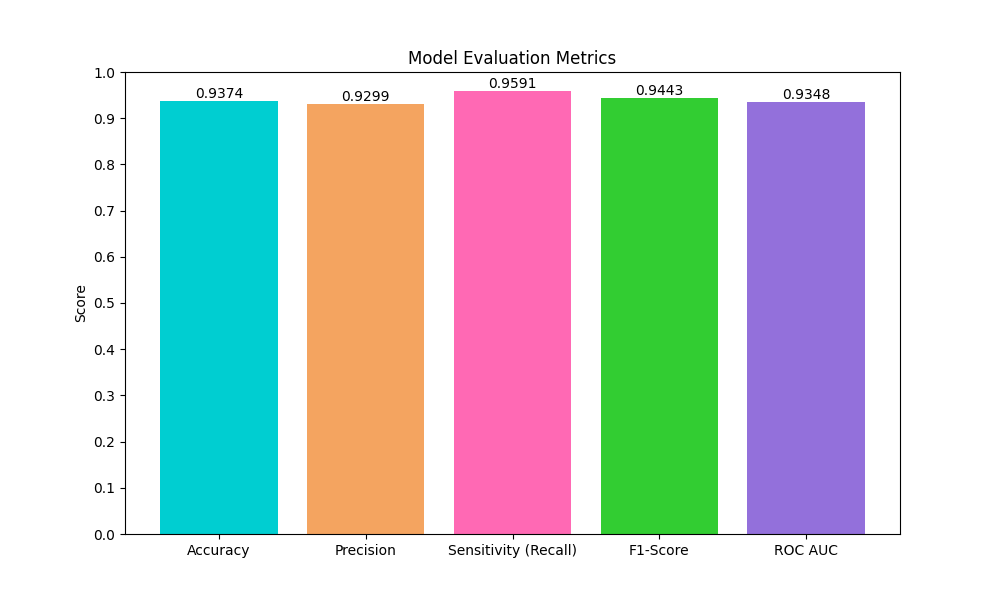
Our team developed and deployed a streamlined web application with an intuitive interface that prioritizes user accessibility and ease of use. To ensure universal access, we implemented a dedicated WiFi AP that enables users to connect seamlessly through their mobile devices without requiring complex network configurations. This mobile-first approach is particularly valuable for field operations, where users can quickly access the application from their smartphones or tablets without dependency on external networks. The combination of thoughtful interface design and reliable connectivity is geared towards achieving high user adoption rates and operational efficiency across our target user base.

# **Performance Results**

#### **Baseline Results**

The resulting baseline metrics, which are displayed in Figure 9, demonstrate the model’s ability to effectively balance precision and generalization. Specifically, the model achieves a precision score of 0.9299, a recall score of 0.9591, and an F1-score of 0.9443, these being our most important metrics.

* Precision: Our precision score of 0.9299 means that 92.99% of the instances identified as Mpox by the model are correctly classified. In the context of healthcare, a high precision score is essential to minimize false positives, ensuring that patients are not incorrectly diagnosed with the disease.
* Recall: With a recall score of 0.9591, the model can correctly identify 95.91% of the true Mpox cases in the dataset. High recall is critical in medical diagnostics because failing to identify a patient with Mpox could delay critical treatment and lead to further spread of the disease.
* F1-score: The F1-score of 0.9443 indicates that the model strikes an effective balance between precision and recall. This is particularly important in healthcare applications, where both false positives and false negatives carry significant consequences. In other words, the model avoids sacrificing precision for recall or vice versa.

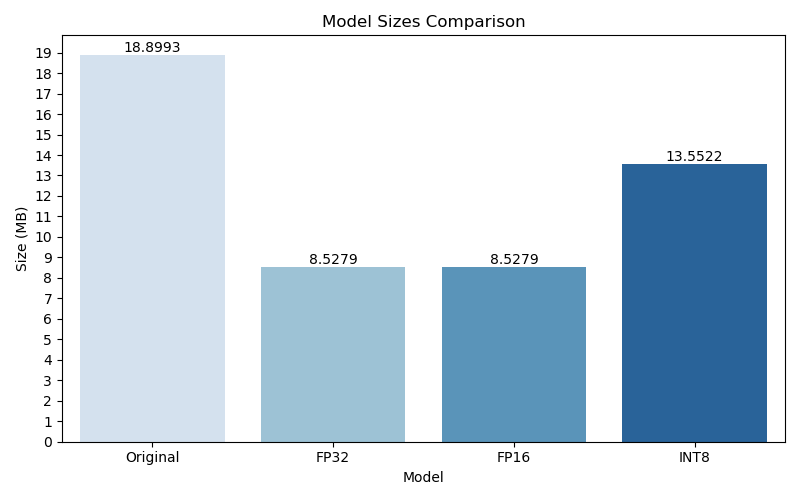
****

**Figure 9: Model Evaluation Metrics**

These results are promising for the early stages of model development, indicating that the model can both accurately diagnose Mpox and reliably detect true positive cases. This balance between precision and recall demonstrates the model’s potential to be an effective diagnostic tool as the cost of misdiagnosis can be high.

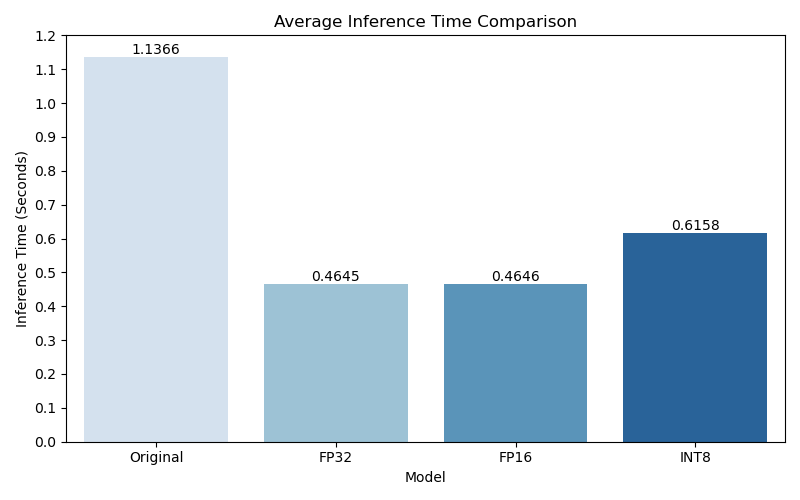
#### **Optimization Results**

To evaluate model optimization effectively, we established three key benchmarks focusing on model size, inference speed, throughput, and performance metrics. Our analysis through TensorRT demonstrated significant improvements across these dimensions.

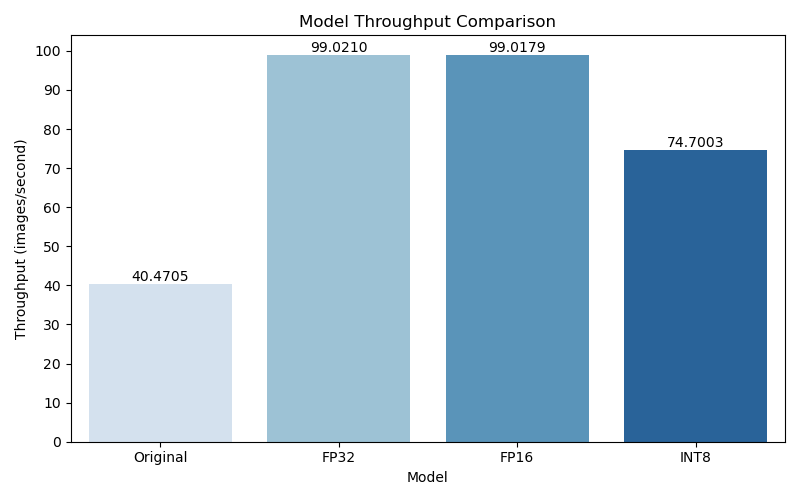
****

**Figure 10: Model Sizes**

The model size decreased substantially across all precision formats, demonstrating the effectiveness of our optimization approach, as illustrated in Figure 10. Specifically, both FP32 and FP16 formats achieved identical reduction factors of 2.22, effectively cutting the original model size by more than half. While the INT8 format showed a more modest reduction factor of 1.39, it still represented a significant improvement in storage efficiency. This optimization of model size proves particularly valuable for deployment in resource-constrained edge devices, where storage capacity and memory management are critical considerations.

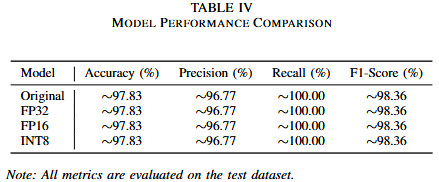
****

**Figure 11: Model Average Inference Time**

****

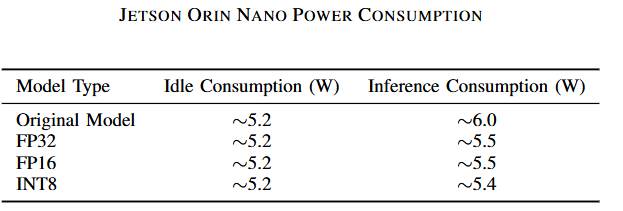
**Figure 12: Model Throughput Time**

The optimization yielded dramatic improvements in average inference speed and throughput, as shown in Figures 11 and 12. Notably, FP32 and FP16 formats produced identical values for both metrics, likely due to Jetson's hardware constraints. Similarly, the INT8 had slightly higher metrics than its FP32 and FP16 counterparts, further implying that the Jetson’s hardware impeded further optimization for this precision, most likely due to the 8GB RAM limitation during calibration, which prevented efficient optimization.



**Table I: Model Performance Comparison**

Despite these optimizations, performance metrics remained consistent across all models thanks to TensorRT's mixed-precision capabilities, as detailed in Table I. While this demonstrates successful preservation of model accuracy, we observed that INT8 quantization showed slightly lower performance improvements compared to FP32 and FP16. This limitation stems from TensorRT's selective tensor quantization process, where only certain tensors could be effectively converted to INT8 format.

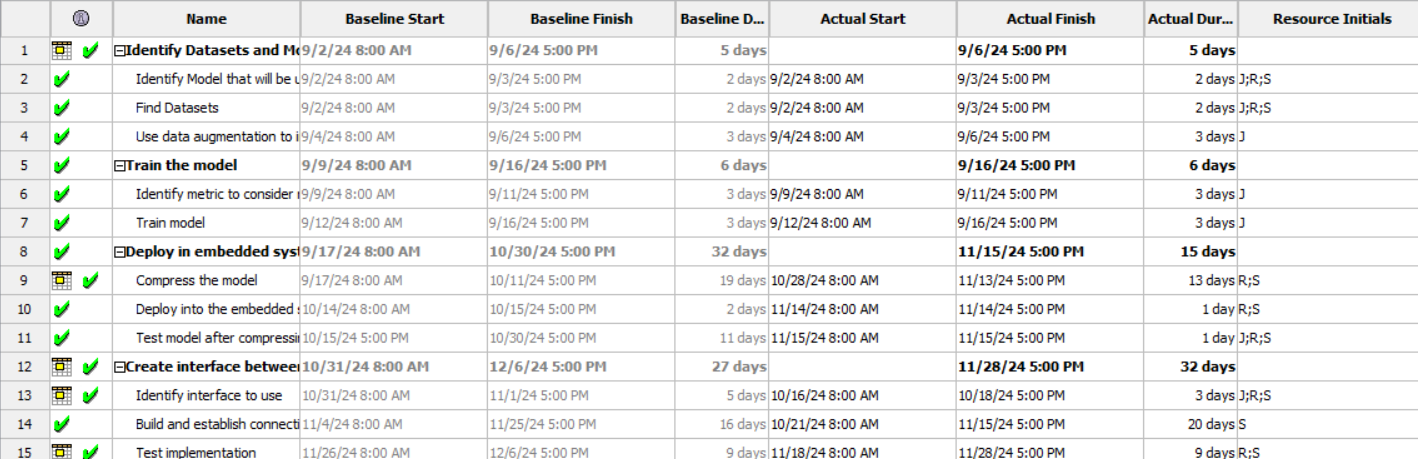


**Table II: Jetson Orin Nano Power Consumption**

Table II displays the power consumption data for the Jetson Orin Nano. This provides valuable insight into the energy efficiency of the device, highlighting how its power usage varies with the different optimized models and how it compares to the original. An analysis of power consumption reveals that the optimized models consume significantly less power compared to the original model. Model quantization, which simplifies the architecture using FP32, FP16, and INT8 precision formats, achieves this improvement. Specifically, we reduce power consumption by ratios of 1.2, 1.4, and 1.8 for FP32, FP16, and INT8, respectively. These results strongly support the original hypothesis, showing that quantization is a good way to make the Jetson Orin Nano use less power.

# **Project Conclusion**

We adhered closely to the established Project Plan; however, certain tasks required more time than initially anticipated, while others were completed with greater efficiency than expected. The table in Figure 13 outlines the milestones and their associated tasks, organized according to the structure defined in the Work Breakdown Structure (WBS) shown in Figure 7. For clarity, the Baseline Columns indicate the planned schedule established at the beginning of the project, serving as a reference point for comparison. In contrast, the Actual Columns capture the schedule as it unfolded in practice, providing a clear depiction of deviations from the original plan.

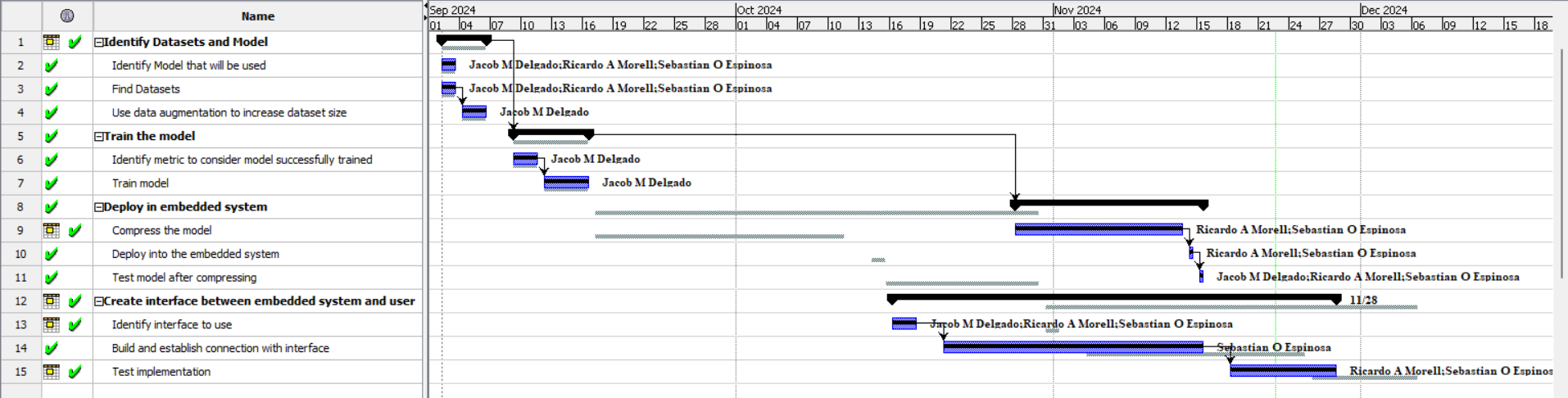


**Figure 13: Gantt Chart Table**

The most significant challenge in our project arose after completing the **Train the Model** milestone. We encountered unexpected delays while setting up the necessary tools for model compression, which took over a month before we could even begin the compression process. This setback impacted our timeline and required adjustments to the schedule. Fortunately, once the setup was complete, the compression process proved less complicated than anticipated. We successfully completed the milestone in half the time originally planned, compensating for some of the lost time.

Following the completion of the training phase, the team adopted a parallel workflow to realign with the planned schedule. This approach allowed **Sebastian** to start building the web interface ahead of schedule, while **Ricardo** focused on configuring the Jetson’s AP Wi-Fi Hotspot to facilitate integration between the components. By working concurrently, we ensured that the project regained momentum and that subsequent milestones were completed more efficiently.

Figure 14 displays the Gantt chart for our project, providing a clear visual representation of how the completed tasks align with the established timeline. Each task is depicted using a blue bar and a smaller, gray bar. The smaller gray bar represents the baseline schedule, which captures the originally planned start and end dates for each task as outlined at the project's outset. On the other hand, the blue bar illustrates the actual progress of each task. It reflects the actual start and end dates, as well as the assigned members for each task.

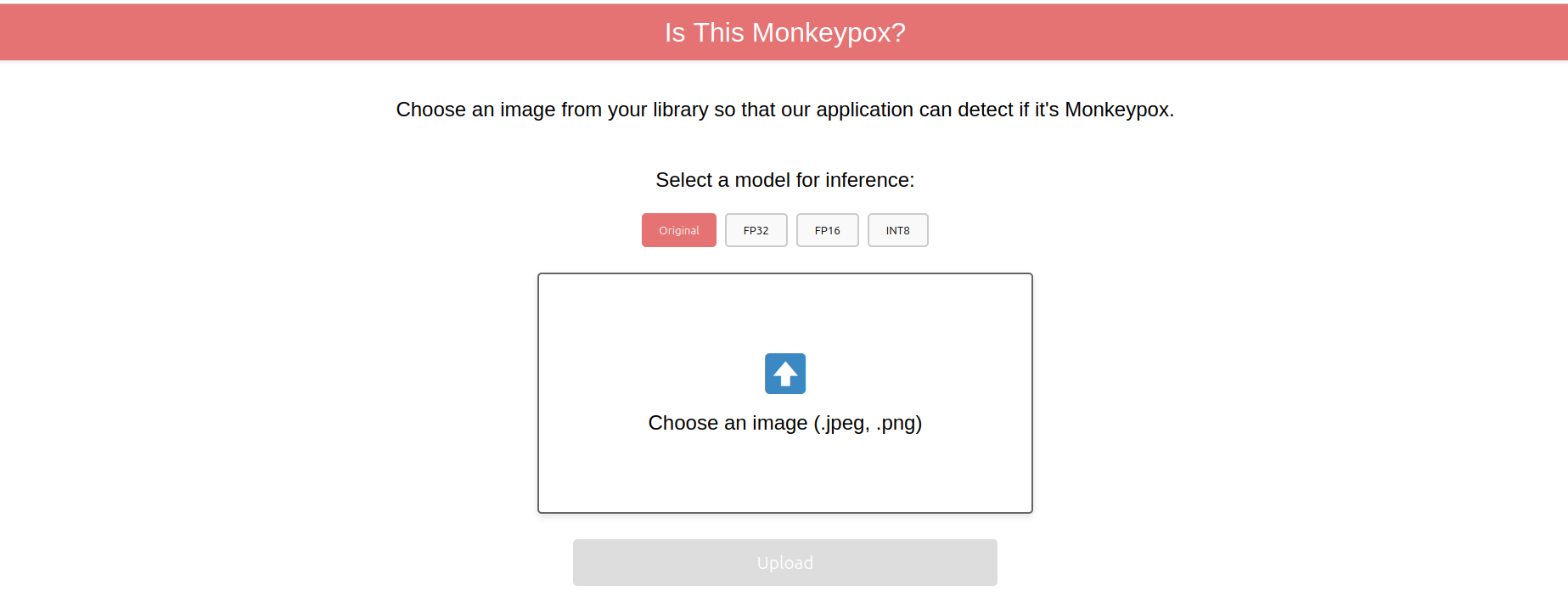


**Figure 14: Gantt Chart**

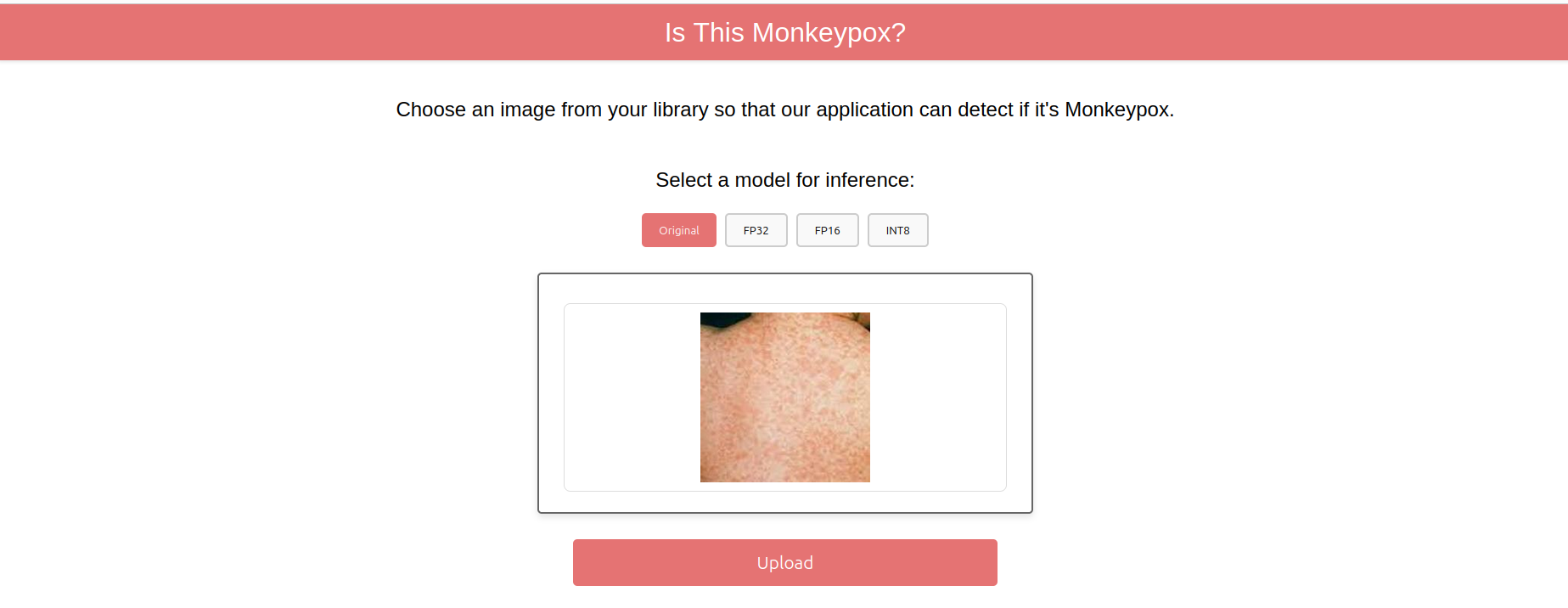
# **Project Documentation**

GitHub Repository: <https://github.com/Jacobdelgado1002/CAPSTONE>

The web application has three main pages. Figures 15.1 and 15.2 display the landing page, which provides inference options for the original model and the FP32, FP16, and INT8 compressed models. These options allow users to compare inference times and output probabilities. The page also includes an upload box for users to take or select a picture.

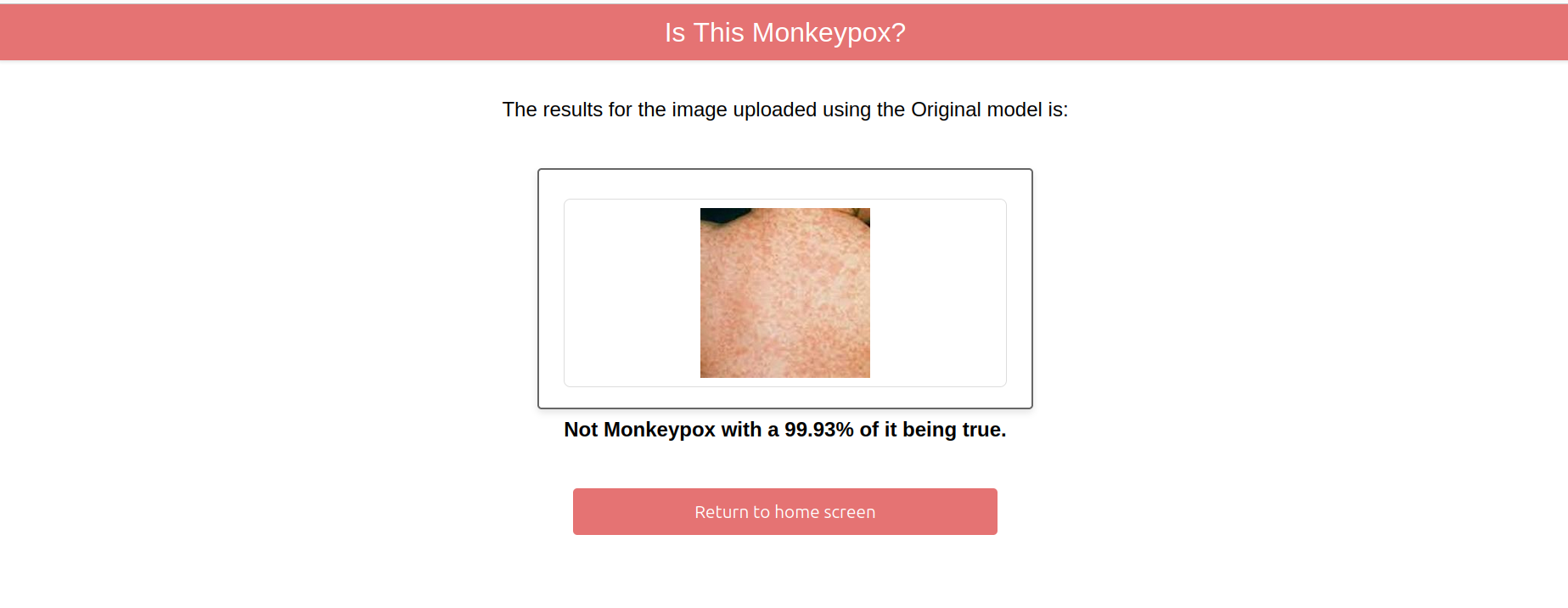


**Figure 15.1: Landing Page - Before Choosing Image**



**Figure 15.2: Landing Page - After Choosing Image**

Figure 15.1 represents the landing page prior to selecting an image for upload, during which the upload button remains disabled. Conversely, Figure 12.2 represents the landing page subsequent to selecting an image for upload, displaying the rendered image and enabling the upload button for prediction.



**Figure 16: Upload Successful Page**

Figure 16 is the page for a successful upload. The image is rendered again, and the page includes the model used for the inference and the result returned by the model. The model classifies the data and calculates the certainty percentage for each classification. The user can then return to the landing page with the button on the bottom to repeat the process with another image or model.

# **References**

[[1] A. T. Levin *et al.*, “Assessing the burden of COVID-19 in developing countries: systematic review, meta-analysis and public policy implications,” *BMJ Glob Health*, vol. 7, no. 5, p. e008477, May 2022, doi: 10.1136/bmjgh-2022-008477.](https://www.zotero.org/google-docs/?tyaU4J)

[[2] “COVID-19 deaths reported (2024 global),” datadot. Accessed: Aug. 25, 2024. [Online]. Available: http://data.who.int/dashboards/covid19/circulation](https://www.zotero.org/google-docs/?tyaU4J)

[[3] “Coronavirus disease (COVID-19) pandemic.” Accessed: Aug. 25, 2024. [Online]. Available: https://www.who.int/europe/emergencies/situations/covid-19](https://www.zotero.org/google-docs/?tyaU4J)

[[4] M. Hraib, S. Jouni, M. M. Albitar, S. Alaidi, and Z. Alshehabi, “The outbreak of monkeypox 2022: An overview,” *Ann Med Surg (Lond)*, vol. 79, p. 104069, Jun. 2022, doi: 10.1016/j.amsu.2022.104069.](https://www.zotero.org/google-docs/?tyaU4J)

[[5] “WHO Director-General declares mpox outbreak a public health emergency of international concern.” Accessed: Aug. 25, 2024. [Online]. Available: https://www.who.int/news/item/14-08-2024-who-director-general-declares-mpox-outbreak-a-public-health-emergency-of-international-concern](https://www.zotero.org/google-docs/?tyaU4J)

[[6] “Mpox (monkeypox).” Accessed: Aug. 25, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/monkeypox](https://www.zotero.org/google-docs/?tyaU4J)

[[7] A. Paniz-Mondolfi *et al.*, “Evaluation and validation of an RT-PCR assay for specific detection of monkeypox virus (MPXV),” *J Med Virol*, vol. 95, no. 1, p. e28247, Jan. 2023, doi: 10.1002/jmv.28247.](https://www.zotero.org/google-docs/?tyaU4J)

[[8] H. Habehh and S. Gohel, “Machine Learning in Healthcare,” *Curr Genomics*, vol. 22, no. 4, pp. 291–300, Dec. 2021, doi: 10.2174/1389202922666210705124359.](https://www.zotero.org/google-docs/?tyaU4J)

[[9] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” Apr. 16, 2017, *arXiv*: arXiv:1704.04861. doi: 10.48550/arXiv.1704.04861.](https://www.zotero.org/google-docs/?tyaU4J)

[[10] “Mpox is still being transmitted across the world, says UN health agency | UN News.” Accessed: Aug. 26, 2024. [Online]. Available: https://news.un.org/en/story/2024/08/1153141](https://www.zotero.org/google-docs/?tyaU4J)

[[11] H. Lindroth *et al.*, “Applied Artificial Intelligence in Healthcare: A Review of Computer Vision Technology Application in Hospital Settings,” *J Imaging*, vol. 10, no. 4, p. 81, Mar. 2024, doi: 10.3390/jimaging10040081.](https://www.zotero.org/google-docs/?tyaU4J)

[[12] E. Kim, “Difference between SSD Mobi, EfficientNet and Faster R-CNN ResNet 50,” Medium. Accessed: Sep. 05, 2024. [Online]. Available: https://medium.com/@elvenkim1/difference-between-ssd-mobi-efficientnet-and-faster-r-cnn-resnet-50-928f6600a1fa](https://www.zotero.org/google-docs/?tyaU4J)

[[13] “Mpox (Monkeypox) virus PCR,” Canterbury Health Laboratories. Accessed: Sep. 03, 2024. [Online]. Available: https://www.chl.co.nz/test/monkeypox-virus-pcr/](https://www.zotero.org/google-docs/?tyaU4J)

[[14] J. F. Huggett, D. French, D. M. O’Sullivan, J. Moran-Gilad, and A. Zumla, “Monkeypox: another test for PCR,” *Euro Surveill*, vol. 27, no. 32, p. 2200497, Aug. 2022, doi: 10.2807/1560-7917.ES.2022.27.32.2200497.](https://www.zotero.org/google-docs/?tyaU4J)

[[15] M. Hraib, S. Jouni, M. M. Albitar, S. Alaidi, and Z. Alshehabi, “The outbreak of monkeypox 2022: An overview,” *Ann Med Surg (Lond)*, vol. 79, p. 104069, Jun. 2022, doi: 10.1016/j.amsu.2022.104069.](https://www.zotero.org/google-docs/?tyaU4J)

[[16] upr\_admin, “Propiedad intelectual,” Universidad de Puerto Rico. Accessed: Sep. 15, 2024. [Online]. Available: https://www.upr.edu/investigaciones/propiedad-intelectual/](https://www.zotero.org/google-docs/?tyaU4J)

[[17] “Creating Wifi AP Hotspot on Jetson Nano - Jetson & Embedded Systems / Jetson Nano,” NVIDIA Developer Forums. Accessed: Sep. 10, 2024. [Online]. Available: https://forums.developer.nvidia.com/t/creating-wifi-ap-hotspot-on-jetson-nano/72982/2](https://www.zotero.org/google-docs/?tyaU4J)