

# **Software Design Document for Machine Learning For Network-Denied Environments**

**Version 1.0 approved**

**Prepared by** Sanjog Baniya, Jonathon M Dooley, Enrico Efendi, Wilson Gan,  
Xavier Lara, Kevin Maravillas, Howard Nguyen,  
Nisapat Poolkwan, Johnson Tan, Justin To, Alvin Yu

**National Security Innovation Network**

12/6/2023

Table of Contents.....	<pg 2>
Revision History.....	<pg 4>
1. Introduction.....	<pg 5>
1.1. Purpose.....	<pg 5>
1.2. Document Conventions.....	<pg 5>
1.3. Intended Audience and Reading Suggestions.....	<pg 5>
1.4. System Overview.....	<pg 5>
2. Design Considerations.....	<pg 6>
2.1. Assumptions and dependencies.....	<pg 6>
2.2. General Constraints.....	<pg 6>
2.3. Goals and Guidelines.....	<pg 6>
2.4. Development Methods.....	<pg 7>
3. Architectural Strategies.....	<pg 8>
4. System Architecture.....	<pg
10>	
4.1. ....	<pg
10>	
4.2. ....	<pg
10>	
5. Policies and Tactics.....	<pg
11>	
5.1. Specific Products Used.....	<pg
11>	
5.2. Requirements traceability.....	<pg
11>	
5.3. Testing the software.....	<pg
11>	
5.4. Engineering trade-offs.....	<pg
11>	
5.5. Guidelines and conventions.....	<pg
11>	
5.6. Protocols.....	<pg
11>	
5.7. Maintaining the software.....	<pg
11>	
5.8. Interfaces.....	<pg
11>	
5.9. System's deliverables.....	<pg
11>	
5.10. Abstraction.....	<pg
11>	
6. Detailed System Design.....	<pg
12>	
6.x Name of Module.....	<pg
11>	

11>	6.x.1	Responsibilities.....	<pg
11>	6.x.2	Constraints.....	<pg
11>	6.x.3	Composition.....	<pg
11>	6.x.4	Uses/Interactions.....	<pg
11>	6.x.5	Resources.....	<pg
12>	6.x.6	Interface/Exports.....	<pg
7.	Detailed Lower level Component Design		
13>	7.x	Name of Class or File.....	<pg
13>	7.x.1	Classification.....	<pg
13>	7.x.2	Processing Narrative(PSPEC).....	<pg
13>	7.x.3	Interface Description.....	<pg
13>	7.x.4	Processing Detail.....	<pg
13>	7.x.4.1	Design Class Hierarchy.....	<pg
13>	7.x.4.2	Restrictions/Limitations.....	<pg
13>	7.x.4.3	Performance Issues.....	<pg
13>	7.x.4.4	Design Constraints.....	<pg
13>	7.x.4.5	Processing Detail For Each Operation.....	<pg
8.	Database Design.....<pg		
14>			
9.	User Interface		
15>	9.1.	Overview of User Interface.....	<pg
16>	9.2.	Screen Frameworks or Images.....	<pg
19>	9.3.	User Interface Flow Model.....	<pg
10.	Requirements Validation and Verification.....<pg		
19>			

11.	Glossary.....	<pg 21>
12.	References.....	<pg 22>

## Revision History

Name	Date	Reason For Changes	Version
Justin To	12/6/2023	Initial draft	1.0

# **1. Introduction**

## **1.1 Purpose**

The purpose of this Software Design Document (SDD) is to describe the design and architectural structure for the image classification application. It outlines the system architecture, software components, interfaces, and data management strategies. The document is intended to guide the development team through the implementation phase.

## **1.2 Document Conventions**

There were no special conventions followed in the writing of this document.

## **1.3 Intended Audience and Reading Suggestions**

Intended audience for this SDD are:

- Software Developers: Developers use it as a guide for implementing the system's design
- Liaisons: To understand the technical aspects and ensuring adherence to the design specifications

For the reading sequence:

- System Architecture: High level understanding of the system's design
- Component Design: Crucial for developers, details individual components
- Database Design: Gives overall idea on how data is managed and stored
- User Interface: Helps to understand the interaction between system components and external entities

## **1.4 System Overview**

The image classification application is highly scalable and designed to be used offline. It includes an AI module to classify images and offer interactive tutoring. Other features include offline content management and sharing modules, as well as a synchronization module to update data when the internet connection is available. This design emphasizes user experience and data efficiency, as well as the ability to adapt to varying hardware requirements.

## 2. Design Considerations

### 2.1 Assumptions and Dependencies

- Users are expected to have a phone with a camera and a gallery application
- Users are expected to have an internet connection to download the application.

### 2.2 General Constraints

#### Web Client

- N/A

#### Server

- Model limitations
  - Storage: The model used for classification must not require a large file size to store.
  - Update Capability: The model must be incrementally updatable.
  - Mobile Deployment: Lightweight enough to deploy on mobile devices.
  - Classification Accuracy: Should possess a sufficiently high prediction accuracy.

#### Mobile Client

- Hardware Limitations
  - Mobile Device: The application must operate within the constraints of mobile devices, considering limitations in processing power, memory, and storage.
  - Camera: The image classification process relies on the capabilities of the mobile device's camera, such as image resolution and quality.
- Operating System Limitations:
  - The application design must be compatible with iOS and Android. This includes utilizing platform-specific development frameworks and guidelines.
- Network Limitations:
  - Internet Connectivity: The application requires a reliable internet connection primarily for communication with the server for data transmission.
  - Data Usage: Potential constraints on data usage for users with limited data plans.

### 2.3 Goals and Guidelines

- The Software has a mandatory delivery date that must be met near the end of Spring 2024
- The end product must be lightweight for deployment on mobile devices

### 2.4 Development Methods

The method of approach utilized for software design revolves around the distribution of work into different sub-teams. Each sub-team independently focused on a component of the project to work on and incrementally made progress every week. Weekly meetings showcasing progress of each sub-team were also conducted to receive feedback and determine the direction of the next implementation.



### **3. Architectural Strategies**

System to consist of three main components: cross-platform mobile clients, web client admins, and server. Mobile clients are designed with the intention to operate offline and connect to the server for select functionality and updates. Web clients will invoke some same functionalities on the server as mobile clients and some additional functionality. Flask server design selected to communicate with both client types regardless of platform or framework.

Mobile client application is required to work on both Android and iOS devices. For performance, cross-platform development, and ML libraries, the mobile application is being developed with React Native Expo.

## 4. System Architecture



- The user connects to the server through a mobile device and is able to send information such as images and text
- The server receives the data, using it to further train the classification model
- The user receives an updated version of the model to use for classification
- The server can be managed through the use of a web client on the desktop

## **5. Policies and Tactics**

### **5.1 Choice of which specific products used**

- Coding languages
  - Python
  - JavaScript
- PyCharm
- TensorFlow
- TensorFlow JS
- Keras
- Flask

### **5.2 Plans for ensuring requirements traceability**

Requirements traceability will be ensured by referencing this document at appropriate intervals of development and evaluating as appropriate.

### **5.3 Plans for testing the software**

Once key requirements of the system are completed, their functionality will be tested for correctness before being fully integrated.

## **6. Detailed System Design**

This section provides an in-depth discussion of the components outlined in the system architecture section. Each subsection will include detailed descriptions of system software components, their responsibilities, constraints, and composition.

### **6.1 Server Model**

#### **6.1.1 Responsibilities**

This component is responsible for preparing images for classification. It performs operations like resizing, normalization, and augmentation to make the images suitable for the model.

#### **6.1.2 Constraints**

The image processing must be efficient, ensuring minimal delay in feature extraction, crucial for real time applications.

#### **6.1.3 Composition**

TBA

#### **6.1.4 Uses/Interactions**

This component is primarily utilized by the Image Classification component for providing preprocessed and featured extracted images necessary for classification. It interacts with data storage components to retrieve raw images and store processed images. And with excessive usage can lead to increased memory consumption affecting other components.

#### **6.1.5 Resources**

TBA

#### **6.1.6 Interface/Exports**

TBA

## **7. Detailed Lower level Component Design**

TBA

## **8. Database Design**

No database is currently being used in this project.

## 9. User Interface

### 9.1 Overview of User Interface

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback Information that will be displayed for the user. This is an overview of the UI and its use. The user manual will contain extensive detail about the actual use of the software.

#### Mobile Client:

First the user is introduced with a login page, which they will have to make an account to enter the app. Once they have finished making an account, then they will be able to confirm their email address with the code that they received. Soon they will be able to enter the main page of the app. On the main page, they will see a Sign Out button which they can sign out if they want. Next they will have two main functionalities, the camera and the gallery functions. They can decide whether they want to take a picture or upload an image. Underneath the buttons is where the image will be rendered. Next, the user will need to choose a label, but if the label they want is not indicated in the drop down menu, they can simply use the custom label input function below it. This allows the imputed text to be appended to the drop down. Once they've chose a label, they can add a confidence value to help the server collect and store the data to train. When everything is finished, they can press the upload button which uploads the information to the server, but the classify button is still a work in progress.

#### Web Client:





When the user first lands on the website, they will see the navbar at the top of the screen. In the navbar, it contains an add label, upload images, and trained with verified buttons. With these buttons, the user can either add more labels to the tag to help match the images, upload their own images to the website to train, or send all the images that already verified to the server to train. A little bit below the navbar, the user will see three different elements. The first one is the option for the user to choose how many images per page they wanted to be display on the screen. Right below that we have four buttons next to each other, Fetch Data, Back, Next, and Send Verified. The Fetch Data button is basically fetch the images from the sever and shown as many as the user wanted from above element. Next and Back buttons are for the user to traverse through the previous or next images. The last button is to send all the images that you user had verified by selecting the tags. The last element is the counter of the fetched images, it will tell the user how many images left that's need to be verify and how many that's already verified.

### 9.2 Screen Frameworks or Images

These can be mockups or actual screenshots of the various UI screens and popups.

#### Mobile Client:

11:06

 65%

# Create an account






Register

By registering, you confirm that you accept our [Terms of Use](#) and [Privacy Policy](#)

Have an account? Sign In

[illegible]








5:52     83% 

## Confirm your Email



**Confirm**

**Resend Code**


**Back to Sign In**

9:10     71% 

**Sign out**

No image taken yet

Select item 

**Add Label**

Confidence Number

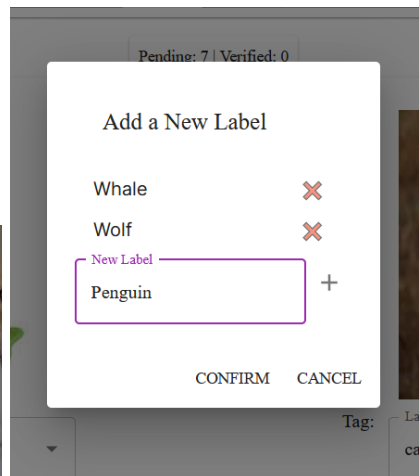
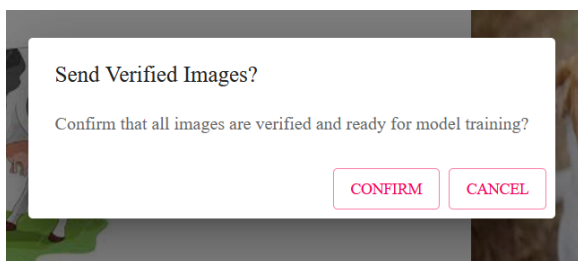
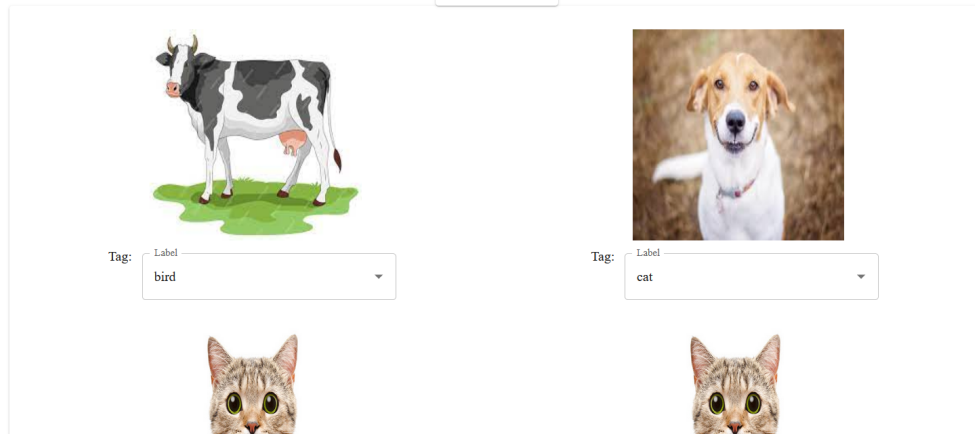
**Classify** **Upload**

Web Client:

Display 4 images per page

FETCH DATA BACK NEXT SEND VERIFIED

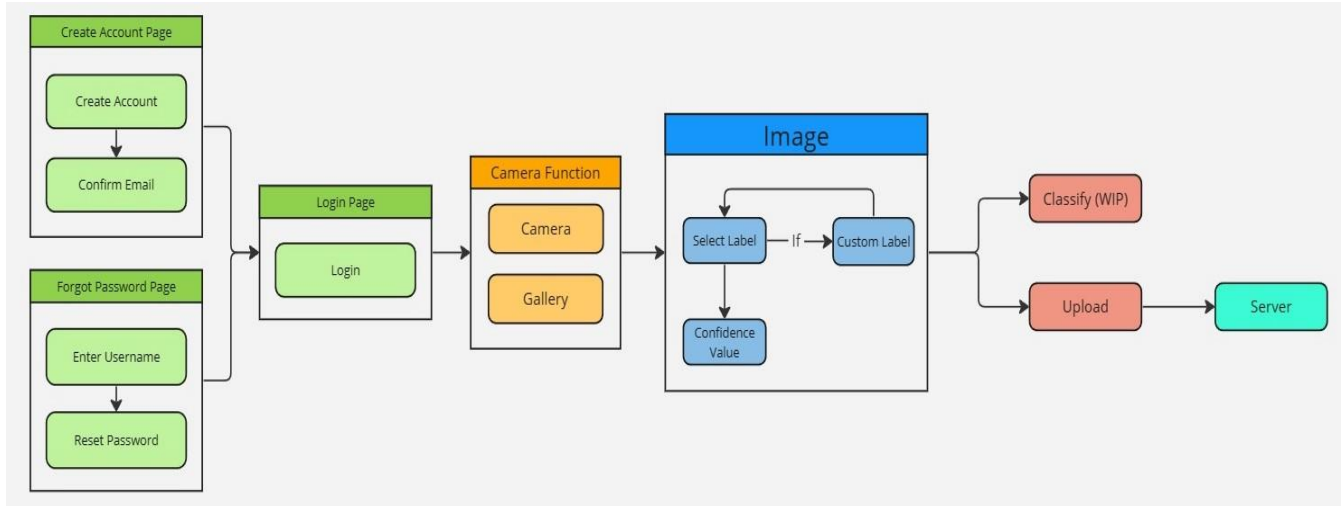
Pending: 7 | Verified: 0



### 9.3 User Interface Flow Model

A discussion of screen objects and actions associated with those objects. This should include a flow diagram of the navigation between different pages.

Mobile Client:



## 10. Requirements Validation and Verification

Functional Requirements	Component Modules/UI Elements That Satisfy Requirements	Method of Testing that Satisfy Requirements
The system shall receive data in the form of images from connecting devices		
The system shall incrementally train the model using the received images and their labels		
The system should possess the capability of adjusting the data used for the training set		
The system may possess security preventing the leak of information		

The system shall allow the user to select an image from the mobile device's photo gallery or take an image from the mobile device's camera.	ImagePicker module	The user should be able to press a button that opens the mobile device's photo gallery and select a single image or access the device's camera to capture a single image.
The system shall process the selected image for classification.	ImageProcessor module	The application will utilize the image selected/taken by the mobile device's photo gallery or camera and process said image into Tensor in order for the pre-trained machine learning model to be able to classify.
The system shall classify the selected image using a pre-trained machine learning model.	Classification module	The application will confirm that the system successfully classifies the image using the pre-trained machine learning model and will return the predicted class labels and confidence scores.
The system shall capture and display the top predictions for the classified image.	User Interface module	The application shall ensure that the top predictions and confidence scores are formatted properly and presentable to the user in the home page.
The system shall display the selected image in the user interface.	User Interface module	The application shall ensure that the correct, selected image will be displayed properly in the center of the home page.
The system shall handle errors related to image selection and classification.	Error Handling module	The application will clean up tensors and dispose of loaded models when classification is not in process to avoid memory leakage.
The system shall send the classified image data and predictions to the server.	Server module	The application will collect labels and confidence scores provided by the machine-learning model or user input and send such data to the server for model training purposes.

## 11. Glossary

- **Tensor** - Multi-dimensional arrays of numbers that represent complex data.
- **Artificial Intelligence (A.I.):** A technology aimed at getting machines to perform functions much in the same way as humans.
- **Machine Learning:** A field of study branching off of A.I. that utilizes data analysis to learn without explicit programming.
- **TensorFlow:** An open source framework developed by Google researchers to run Machine Learning, deep learning and other statistical and predictive analytics workloads.
- **Keras:** An open source library that provides a python interface for artificial neural networks. Acts as an interface for the Tensorflow library.
- **Flask:** A web framework; a python module to develop applications easily.
- **React Native:** An open source UI software framework created by Meta, used to develop applications for Android, iOS, Web, Windows by enabling developers to use the React framework along with native platform capabilities.
- **Intellectual Property:** A set of products protected under laws associated with copyright, patent, trademark, industrial design, and trade secrets.
- **GDPR:** Stands for General Data Protection Legislation, governs the way in which we can use, process, and store personal data.
- **Open Source:** Software where the creator allows other users direct access to the source code so the user can alter and distribute the software for their own purpose.
- **Interface:** The place at which independent systems meet and act on or communicate with each other.

## 12. References

- Tensorflow: [https://www.tensorflow.org/tutorials/images/transfer\\_learning](https://www.tensorflow.org/tutorials/images/transfer_learning)
- Keras: [https://www.tensorflow.org/js/tutorials/conversion/import\\_keras](https://www.tensorflow.org/js/tutorials/conversion/import_keras)
- Digit recognition in sklearn: [https://scikit-learn.org/stable/auto\\_examples/classification/plot\\_digits\\_classification.html](https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html)
- React Native: <https://github.com/thecodingmachine/react-native-boilerplate>
- Flask: <https://flask.palletsprojects.com/en/2.3.x/quickstart/>
- Python: <https://www.python.org/>