# Software Requirements Specification

**for**

# VeilTech

## A Steganographic technology implemented using ML

**Prepared by**


| | | |
|---|---|---|
| **VV Prabhath** | **227Z1A05I0** | **vprabhat.v@gmail.com** |
| **S Shashank** | **227Z1A05G9** | **suramshashank@gmail.com** |
| **Sk. Rayees** | **227Z1A05G4** | **skrayees@gmail.com** |

**Instructor:** Dr. Samuel Chepuri

# Contents

# 1 Introduction

## 1.1 Document Purpose

The purpose of this Software Requirements Specification (SRS) is to describe the functional and non-functional requirements of the VeilTech application. This document helps developers, testers, and stakeholders understand the system's design and behaviour clearly. VeilTech uses Machine Learning (ML) and Steganography to securely hide secret data inside images while maintaining image quality.

## 1.2 Product Scope

VeilTech is a mobile-based steganography system that allows users to hide and extract confidential data within digital images using a machine learning model. The app performs the hiding and extraction offline, directly on the device, ensuring user privacy. However, an internet connection is required when users want to send or receive these images through the application or any online platform. The system is trained using the ImageNet Mini dataset and evaluated with PSNR, SSIM, and BER metrics to measure accuracy and image quality.

## 1.3 Intended Audience and Document Overview

This document is for:

- Developers – to implement and maintain the app.
- Testers – to verify and validate the system.
- Project Stakeholders – to understand the system's purpose and operation.

It provides details on system functions, interfaces, performance, and requirements to guide development and testing.

## 1.4 Definitions, Acronyms, and Abbreviations

- ML – Machine Learning
- PSNR – Peak Signal-to-Noise Ratio

- SSIM – Structural Similarity Index Measure
- BER – Bit Error Rate
- Encoder–Decoder Model – ML model used to hide and extract data from images
- Stego Image – The image containing hidden data

## 1.5 Document Conventions

- The document follows the IEEE Software Requirements Specification (SRS**)** format for section numbering and organization.
- The font style used throughout the document is Times New Roman with a font size of 12 for all text content.
- Mandatory requirements are written using the word “shall”, while optional requirements use “may” or “should.”
- Numbered and bulleted lists are used to present requirements, functions, and features clearly and concisely.

## 1.6 References and Acknowledgements

### 1.6.1 Acknowledgements


- Special thanks to faculty mentors and the Department of Computer Science for their guidance and support.
- Gratitude to all researchers and open-source contributors whose work inspired and supported the development of **VeilTech**.


### 1.6.2 References


[1] Design of an Efficient Multimodal Image Steganography Framework with Multi-Domain Feature Analysis and Secret Sharing Operations – E. Ekta & A. Singh (2024) ijisae.org

[2] TSCL: Multi-party Loss Balancing Scheme for Deep Learning Image Steganography based on Curriculum Learning – F. Liu, T. Zhang & C. Zhang (2025)

# 2 Overall Description

## 2.1 Product Overview

VeilTech is a mobile application that enables users to conceal secret data within images using an ML encoder–decoder model. The encoder embeds the data into the image, and the decoder extracts it later. The process is completed locally, without requiring internet access. When users want to share their stego images, the app uses the internet for transmission. This setup provides both privacy and flexibility.

## 2.2 Product Functionality

- Hide secret text or image inside a normal image.
- Extract hidden data from stego images.
- Encrypt secret data before hiding.
- Show quality metrics (PSNR, SSIM, BER) after processing.
- Save stego images securely on the device.
- Send or receive stego images using the internet.
- Provide a simple mobile interface for easy use.

## 2.3 Design and Implementation Constraints

- The available hardware resources may limit model training and testing, as processing high-resolution images or large datasets could exceed memory and computation capacities.
- The project will rely on specific technologies such as Python, TensorFlow, and OpenCV, which restrict development to compatible environments and versions.
- Strong encryption methods and secure key management will be required to protect hidden data, which may limit the choice of open-source tools to those with verified security.
- The system will initially support only common image formats like PNG and JPEG, reducing interoperability with other media types or external platforms.

- The software must follow consistent coding and documentation standards for maintainability, though advanced UI features and cross-platform compatibility may be deprioritised due to time and resource constraints.

## 2.4 Assumptions and Dependencies

- The user has a stable internet connection for sending or receiving images.
- The user provides valid image files for processing.
- Device storage and ML model files are available locally.
- The application runs on Android-based devices.

# 3 Specific Requirements

## 3.1 External Interface Requirements

- User Interface: Simple screens for image upload, data hiding, extraction, and sharing.
- File Storage: Local storage used for saving stego images and extracted data.
- Hardware Interface: Camera and file access for selecting or capturing images.
- Network Interface: Internet required for image sharing and receiving.

## 3.2 Functional Requirements

- The system shall allow the user to select an image and secret data.
- The system shall hide secret data inside the image using the ML model.
- The system shall extract hidden data when requested by the user.
- The system shall save processed images securely on the device.
- The system shall encrypt secret data before embedding.
- The system shall allow users to share or receive images online.
- The system shall show PSNR, SSIM, and BER after processing

# 4 Other Non-functional Requirements

## 4.1 Performance Requirements

- Hiding and extraction should complete within 5 seconds on average devices.
- Image quality loss (PSNR) should remain low.
- System accuracy (SSIM) should remain above 90%.

## 4.2 Safety and Security Requirements

- All data hiding and extraction must occur locally on the device.
- The system should encrypt data before embedding.
- Internet use is limited only to image sharing to protect privacy.
- Prevent unauthorized access to stored stego images.

## 4.3 Software Quality Attributes

- Usability: Simple and easy-to-understand interface.
- Reliability: Consistent results for multiple uses.
- Portability: Works across different Android devices.
- Maintainability: Easy to update or retrain models.
- Scalability: Can support new features like cloud backup later.

# 5 Other Requirements

- The system will require a structured database to store metadata related to images, encryption keys, and user activity logs securely.
- All stored and transmitted data must comply with basic data protection and privacy guidelines to prevent unauthorised access or misuse.
- The project should maintain modular and reusable code components to allow future upgrades, such as extending support for video or audio steganography.
- The application interface and model outputs should use clear, language-neutral labels to ensure ease of understanding and potential internationalisation.
- Proper documentation and version control must be maintained throughout development to facilitate debugging, collaboration, and scalability.
- The software should include logging mechanisms to monitor embedding and extraction activities for security auditing purposes.
- External open-source libraries used must comply with licensing terms to avoid legal or usage conflicts.

.