Configuration Manual

MSc Research Project
MSCDAD_C

Shreyas Sawji
Student ID: x23281065

School of Computing
National College of Ireland

Supervisor: Yalemisew Abgaz

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shreyas Sawji
Student ID:	x23281065
Programme:	MSCDAD_C **Year:** 2025
Module:	Research Practicum Config file
Lecturer:	Yalemisew Abgaz
Submission Due Date:	11/08/2025
Project Title:	Hybrid SHAP-LRP for Adversarial Debugging in Vision Transformers: A Layer-Wise Explainability Approach
Word Count:	736 **Page Count:** 4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shreyas Sawji

Date: 10/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Hybrid SHAP-LRP for Adversarial Debugging in Vision Transformers: A Layer-Wise Explainability Approach

Shreyas Mukund Sawji
Msc. Data Analytics of MSCDAD_C
National College of Ireland
x23281065@student.ncirl.ie

1. Purpose of This Document

This configuration manual provides detailed steps to set up and run the Hybrid SHAP–LRP Vision Transformer adversarial debugging system.
It is aimed at ensuring anyone with a compatible system can reproduce results, interact with the Streamlit dashboard, and test the model's robustness and explainability features.

2. System Requirements

Minimum Hardware (tested on HP Spectre x360 Convertible 13-ae0xx)

- CPU: Intel Core i7-8550U, 4 cores / 8 threads
- RAM: 16 GB
- Storage: 5 GB free space
- Internet connection for library downloads and model weights
- (Optional) GPU acceleration – recommended for faster SHAP/LRP computations

Operating System

- Windows 11 Home (Version 10.0.22631 Build 22631)
- UEFI boot with Secure Boot enabled

3. Required Software

- **Python** 3.9+ (Anaconda or standalone installation)
- **pip** – Python package manager
- **Streamlit** – for the dashboard interface
- **PyTorch** – for deep learning model execution
- **timm** – pretrained Vision Transformer architectures
- **torchvision** – dataset utilities
- **Captum** – SHAP and LRP explainability library

- **Matplotlib** & **Pandas** – visualization and data handling

4. Installation Instructions

Step 1 – Install Python

If Python is not installed:

- Download from python.org or
- Install Anaconda from anaconda.com

```
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>python --version
Python 3.10.9

C:\Users\hp>
```

Step 2 – Set Up Project Folder

- Create a new directory, e.g., vit_shap_lrp_project
- Place all project files in this directory:
  - app.py (main dashboard file)
  - Model scripts and helper functions (e.g., research_practicum_sem3_x23281065.py)
  - Any required data or configuration files

```
project-folder/
|  app.py
|  requirements.txt
|
└──assets/
└──modules/
```

Step 3 – Create a Virtual Environment

```
python -m venv vit_env
vit_env\Scripts\activate  # Windows
# or
source vit_env/bin/activate  # Mac/Linux
```

```
C:\Users\hp>cd C:\Users\hp\Desktop\vit-debugger-streamlit
The system cannot find the path specified.

C:\Users\hp>cd Desktop\vit-debugger-streamlit
The system cannot find the path specified.

C:\Users\hp>cd Desktop\vit_debugger_streamlit

C:\Users\hp\Desktop\vit_debugger_streamlit>python -m venv venv

C:\Users\hp\Desktop\vit_debugger_streamlit>venv\Scripts\activate

(venv) C:\Users\hp\Desktop\vit_debugger_streamlit>pip install streamlit torch t
orchvision matplotlib numpy pillow
Collecting streamlit
  Downloading streamlit-1.47.1-py3-none-any.whl (9.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.9/9.9 MB 9.0 MB/s eta 0:00:00
```

Step 4 – Install Required Libraries

pip install streamlit torch torchvision timm captum matplotlib pandas
Optional (if you want to run in Jupyter/Colab):

pip install notebook

1 Step 5 – Verify Installation
Run the following to check library versions:

python -c "import torch, timm, streamlit, captum, pandas, matplotlib; print('All packages imported successfully!')"

5. Running the Application
Step 1 – Navigate to Project Folder

cd path\to\vit_shap_lrp_project

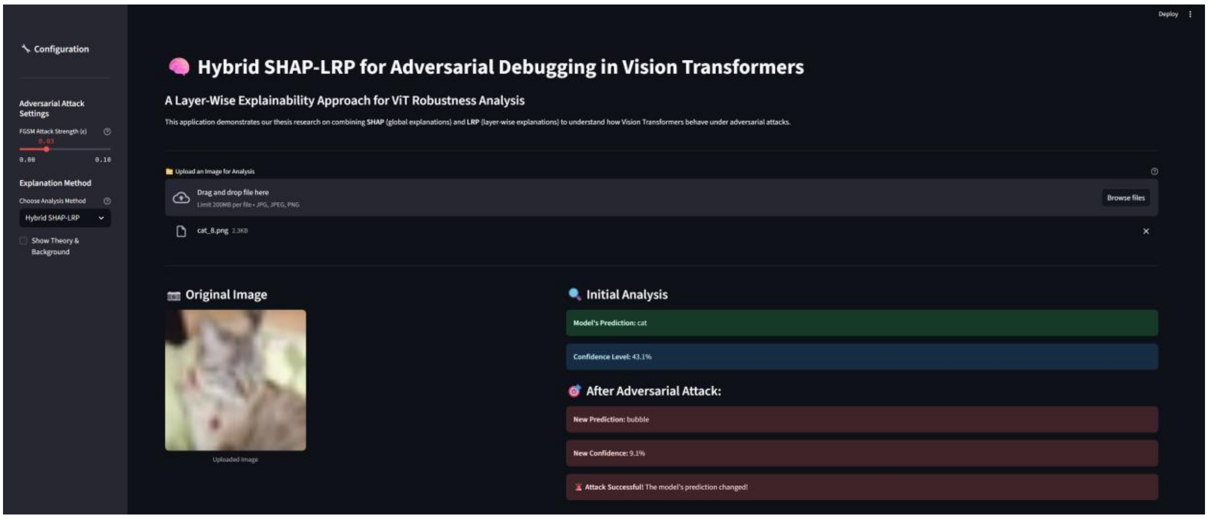
Step 2 – Launch the Streamlit Dashboard
streamlit run app.py

This will open the dashboard in your default web browser.

```
(venv) C:\Users\hp\Desktop\vit_debugger_streamlit>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.0.138:8501
```

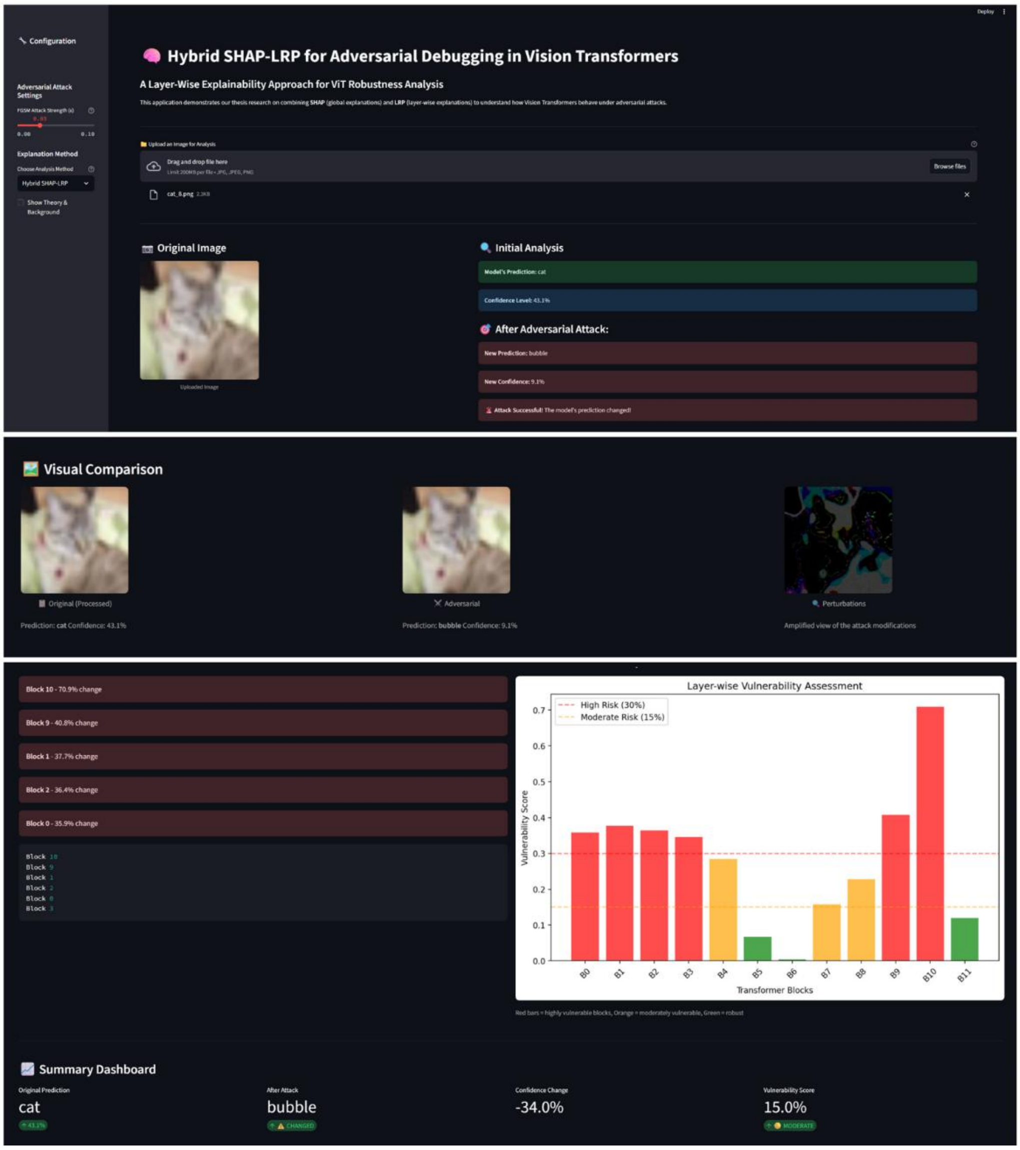


6. Using the Dashboard

The interface contains multiple tabs:

1. **Home/Introduction** – overview and instructions
2. **Upload & Prediction** – upload CIFAR-10 or custom images for classification
3. **Attribution Visualization** – view SHAP and LRP heatmaps
4. **Adversarial Detection & Alerts** – detection status and vulnerable layer report
5. **Model Internals & Metrics** – detailed SHAP/LRP score breakdowns
6. **About & Help** – extra resources and explanations

7. Demonstration of App



Demonstration of the interactive interface showing original and adversarial images, SHAP and LRP visualizations, vulnerability assessment, and detection a

1. Home / Introduction Tab

The introduction tab establishes the foundation for the entire application. It provides:

- Project overview – a concise summary of the research motivation, encapsulating the vulnerabilities of Vision Transformers to adversarial attacks and the reasons for using SHAP and LRP together in a hybrid framework.

- Contextual background – concise overviews of ViTs, what adversarial attacks are, and why explainability is essential in high-stakes usage.

- Usage instructions – unambiguous step-by-step guidance on how to use the dashboard and leverage all its functionality.

This page serves as an introduction for first-time users and a reference page for demonstrations to clarify before more vigorous exploration.

2. Upload & Prediction Tab
The tab allows the user to supply an input image, either the CIFAR-10 dataset or uploaded as a bespoke file. When entered, the system:

- Processes the image through the Vision Transformer model.
- Returns the class label prediction and the accompanying confidence score.
- If also an adversarially induced perturbed version of the image (e.g., obtained via FGSM) is provided or calculated, it displays side by side the original and the perturbed image for convenient comparison.

Immediate feedback loop of this kind allows users to observe model processing of clean and adversarially tampered inputs, grounding vulnerabilities more intuitively.

3. Attribution Visualization Tab
The visualization module facilitates the core explainability features of the framework:

- SHAP heatmaps placed over the original image show the most important regions that make the largest contribution to the model's prediction. The heatmaps can be toggled between clean and adversarial for comparison side-by-side.
- LRP layer-wise relevance maps, presented as heatmaps or bar plots, reveal where each transformer block contributed to the final prediction. This involves flagging blocks that exhibit surprisingly high or shifted relevance under attack.
- Adjustable visualization parameters like heatmap opacity are used to let the user subtly control the overlays' legibility and more easily understand differences between conditions.

With the use of SHAP and LRP perspectives, this tab provides the advantages of both global input-level knowledge and local, layer-level explanations.

4. Adversarial Detection & Alerts Tab
This tab addresses the framework's adversarial detection feature:

- Detection summary – a clear indication of whether an adversarial attack has been detected, based on a combination SHAP–LRP scoring measure.
- Vulnerability reporting – a vulnerability report of transformer blocks showing abnormal LRP relevance patterns, specifying the specific layers most impacted by the perturbation.
- Real-time alerts – SMS messages such as
  "Adversarial attack detected! Vulnerable blocks: Block 0, Block 1."
- provide immediate situational awareness.
- Threshold controls – operators have the ability to calibrate the detection criterion sensitivity to reach an optimal balance between false negatives and false positives.

This tab bridges the gap between raw attribution data and actionable security diagnosis, making it simple to interpret even for non-technical people.

5. Model Internals & Metrics Tab
For those interested in a more technical breakdown, this page shows high-level quantitative outputs:

- Classification confidence scores for clean and adversarial inputs.
- SHAP and LRP score distributions, layer-wise and sample-wise.
- Visual insights such as bar plots showing relevance magnitudes layer-wise and batch-wise, revealing patterns of recurring weaknesses.

These measures not only constitute research evidence but also enable targeted debugging of the Vision Transformer design.

6. About & Help Tab
This final section serves as a knowledge source with:

- Extended description of SHAP and LRP techniques, theoretically grounded and with an explanation of how they counterbalance one another within this system.
- Descriptions of adversarial attack techniques (e.g., FGSM [9]) and their impact on model performance.
- Reading instructions and explanation of the various visual outputs and warning indicators so the dashboard is comprehensible to technical or non-technical users.

8. Troubleshooting

- **Error: torch not found** → Reinstall with pip install torch torchvision
- **Streamlit not launching** → Ensure virtual environment is activated
- **App stuck loading** → Some SHAP computations may be slow on CPU; try smaller batch or use GPU
- **Browser not opening automatically** → Copy the localhost:xxxx URL printed in the terminal into your browser

Video for demonstration link is here - x23281065_research_video.mp4