Configuration Manual

MSc Research Project
Master of Science in Data Analytics

Saran Das Sivadasan Achary
Student ID: 23336030

School of Computing
National College of Ireland

Supervisor: Athanasios Staikopoulos

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Saran Das Sivadasan Achary
Student ID:	23336030
Programme:	Master of Science in Data Analytics **Year:** 2025
Module:	MSc Practicum Part 2
Lecturer:	Athanasios Staikopoulos
Submission Due Date:	28/01/2026
Project Title:	Risk Adaptive Triple Stream Vision Transformer for Dermatological Triage
Word Count:	604 **Page Count:** 6

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:

Date: 26/01/2026

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

Saran Das Sivadasan Achary
Student ID: 23336030

1 Introduction

This document describes the requirements and steps to be performed for replicating the Risk-Adaptive Triple-Stream Vision Transformer (TS-ViT) model for automated dermatological triage. It specifies the required tools, data processing steps, configuration parameters, and packages required for model development and evaluation.

2 Requirements

The main requirements for the model development are:

Google Colab Pro

Cloud-hosted platform for deep learning model development
NVIDIA Tesla T4 GPU (16GB VRAM) access
Preinstalled machine learning libraries
Eliminates environment setup complexity

DERM12345 Dataset

Published in Nature Scientific Data (2024) by Yilmaz et al.
12,345 dermatoscopic images
Hierarchical organization: 5 superclasses, 15 major classifications, 40 subclassifications
Available through Harvard Dataverse
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DAXZ7P

Python Environment

Python 3.10.12
PyTorch 2.2.2 framework
Dynamic computational graphs
CUDA acceleration for GPU training

3 Specification

The specifications of the system required for developing the model are given below:

Hardware

- GPU: NVIDIA Tesla T4 (16GB VRAM)
- RAM: 12GB system RAM

- Storage: 10GB for dataset and model checkpoints
- CUDA: Version 11.8 or higher

Software Specification

- Google Colab Pro with T4 GPU runtime
- Python 3.10.12
- PyTorch 2.2.2
- torchvision 0.17.2
- CUDA 11.8

Python Libraries Used

- PyTorch
- torchvision
- NumPy
- Pandas
- scikit-learn
- Matplotlib
- Seaborn
- OpenCV
- tqdm

4 Installation

Step 1: Setup Environment

- Access Google Colab website (colab.research.google.com)
- Create a new notebook
- Select Runtime → Change runtime type
- Hardware accelerator: GPU (T4 GPU)

Step 2: Verify GPU

```
# Device configuration
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(f"Device: {device}")
print(f"GPU: {torch.cuda.get_device_name(0) if torch.cuda.is_available() else 'None'}")

# Create output directories
os.makedirs('/content/outputs', exist_ok=True)
os.makedirs('/content/checkpoints', exist_ok=True)
os.makedirs('/content/visualizations', exist_ok=True)
```

```
Device: cuda
GPU: Tesla T4
```

Step 3: Install PyTorch

```python
# Core libraries
import os
import zipfile
import numpy as np
import pandas as pd
from PIL import Image
import matplotlib.pyplot as plt
import seaborn as sns
from tqdm import tqdm
import math

# PyTorch
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import Dataset, DataLoader, WeightedRandomSampler

from torchvision import transforms, models

import timm
```

Step 4: Mount Google Drive

```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Step 5: Create Directory Structure

```python
# Create output directories
os.makedirs('/content/outputs', exist_ok=True)
os.makedirs('/content/checkpoints', exist_ok=True)
os.makedirs('/content/visualizations', exist_ok=True)
```

Step 6: Set Random Seed

```python
def set_seed(seed=42):
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False

set_seed(42)
```

5 Data Preparation

Dataset Acquisition
Download DERM12345 dataset from Harvard Dataverse
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DAXZ7P

Extract dataset to /content/data/ directory
Verify 12,345 dermatoscopic images present

Dataset Organization

Total Images: 12,345
Training Set: 7,888 images (80%)
Validation Set: 988 images (10%)
Test Set: 2,485 images (held-out)

Risk Stratification

Three-tier clinical risk levels:
HIGH RISK: 7,365 images (59.7%)
 - Confirmed malignancies: melanoma, basal cell carcinoma, squamous cell carcinoma
 - High-risk precursor lesions
MODERATE RISK: 1,315 images (10.7%)
 - Dysplastic melanocytic nevi
 - Requires periodic surveillance
LOW RISK: 3,665 images (29.7%)
 - Benign lesions
 - Minimal transformation risk

Preprocessing Steps

Resize images to 256×256 pixels
Center crop to 224×224 pixels (ViT-B/16 input size)
Convert to PyTorch tensors
Normalize with ImageNet statistics: Mean [0.485, 0.456, 0.406], Std [0.229, 0.224, 0.225]

6 Model Building

Base Architecture (ViT-B/16)

Input size: 224×224 RGB images
Patch size: 16×16 (produces 196 patches)
Embedding dimension: 768
Transformer layers: 12
Attention heads: 12
Total parameters: 86.3 million
Initialization: Random (no ImageNet pretraining)

TS-ViT Architecture Components

1. **Shared Transformer Encoder**

- Load ViT-B/16 from torchvision with weights=None
- Remove original classification head
- Extract CLS token (768-dim) and 196 patch embeddings

2. Global Classification Stream

- Input: CLS token (768 dimensions)
- Two-layer feedforward network: 768 → 384 → 3 dimensions
- Dropout: 0.3
- Purpose: Overall lesion risk assessment

3. Spatial Attention Stream

- Input: 196 patch embeddings
- Lightweight attention scoring network
- Softmax normalization for attention weights
- Weighted aggregation of patches
- Output: Attention heatmap + risk logits

4. Uncertainty Quantification Stream

- Input: CLS token
- Monte Carlo Dropout: 5 forward passes
- Fully connected network → sigmoid activation
- Output: Confidence estimate and prediction variance

7 Comparison

The developed TS-ViT model was evaluated against the following baseline architectures:

1. ResNet-50
- Convolutional neural network with residual connections
- Parameters: 25.6 million
- Training: From scratch (no ImageNet pretraining)

2. ViT-B/16 (Vanilla)
- Single-stream Vision Transformer
- Parameters: 86.6 million
- Training: From scratch (no ImageNet pretraining)

3. MobileNetV3-Large
- Efficient CNN with depth-wise separable convolutions
- Parameters: 5.5 million

- Training: From scratch (no ImageNet pretraining)

Model	Accuracy	Precision	Recall	Macro F1	Brier Score
TS-ViT (Ours)	**89.3%**	**0.799**	**0.853**	**0.847**	**0.074**
ViT-B/16	67.2%	0.612	0.638	0.625	0.167
ResNet-50	63.8%	0.666	0.567	0.591	0.236
MobileNetV3	59.4%	0.521	0.565	0.543	0.284

Table 1. Model comparison

References

Google. (n.d.). Google Colaboratory. Retrieved December 7, 2024, from https://colab.research.google.com/

PyTorch developers. (n.d.). PyTorch: An open source machine learning framework. Retrieved December 7, 2024, from https://pytorch.org/

Scikit-learn developers. (n.d.). Scikit-learn: Machine learning in Python. Retrieved December 7, 2024, from https://scikit-learn.org/

Hunter, J. D., Droettboom, M., & Caswell, T. A. (n.d.). Matplotlib: Visualization with Python. Retrieved December 7, 2024, from https://matplotlib.org/

Yilmaz, A., et al. (2024). DERM12345: A Comprehensive Dermatoscopy Dataset. Harvard Dataverse. Retrieved December 7, 2024, from https://dataverse.harvard.edu

Python Software Foundation. (n.d.). PyPI: The Python Package Index. Retrieved December 7, 2024, from https://pypi.org/