Configuration Manual

MSc Research Project
MSC ARTIFICIAL INTELLIGENCE

SAYALI BHAWALKAR
Student ID: X23321750

School of Computing
National College of Ireland

Supervisor: PROF. MOHIT GARG

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: SAYALI VISHWAS BHAWALKAR …….…………………………………………………………………………………………………

Student ID: X23321750 ……………………………………………………………………………………………………..……

Programme: MS IN ARTIFICIAL INTELLIGENCE ………………………………………………………………… **Year:** 2024-2025 …………………………..

Module: RESEARCH PRACTICUM ……………………………………………………………………………………………………….………

Lecturer: PROF. MOHIT GARG ……………………………………………………………………………………………………….………

Submission Due Date: 11/08/2025 ……………………………………………………………………………………………………….………

Project Title: CNN BASED SYSTEM FOR DIABETIC RETINOPATHY DETECTION ……………………………………………………………………………………………………….………

Word Count: 836 ……………………………………… **Page Count:** 3…………………………………….…….………

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: SAYALI VISHWAS BHAWALKAR ………

Date: 11/08/2025 ………

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

SAYALI BHAWALKAR
Student ID: X23321750

1 System Requirements

The project needs a certain hardware as well as software setup that should support its project implementation. On the hardware side, a computer with a minimum of Intel i5 or equivalent processor is required with minimum 8GB of RAM (16GB recommended to us train and evaluate newer models quicker). It is highly recommended that it be powered by GPU acceleration with an NVIDIA CUDA, or similar, powered card like the GTX 1660 or higher and with 6GB of VRAM or more. It is suggested to have at least 20GB of free space to store the datasets and trained models and their outputs. Software-wise, the project can be used with Windows 10/11, Ubuntu 20.04 or newer, and macOS, systems that support GPU acceleration. The Python 3.8 to 3 instructors However, students will also need a few necessary libraries such as TensorFlow/Keras, OpenCV, NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, and Streamlit. Grad-CAM functionality may be realized either with a personal process or with the tf-keras-vis package. Experimentation should be carried out in Jupyter Notebook, and version control should be done using Git.

2 Installation & Setup

The user will be required to clone the project repository in GitHub under the github and followed by typing the relevant project directory by use of git clone command. Then, a Python virtual environment must be generated and activated--python -m venv and the corresponding system-specific activation commands--after which all requirements as specified in requirements.txt must be installed using pip install. A separate download of the dataset of interest will be required, e.g., APTOS or EyePACS and saved in the data/raw folder. It is possible then to apply preprocessing using the preprocess.py script which cleans the images, resizes, and then augments the images. Training parameters such as resolution of the images (224224), model (ResNet50, VGG16, or EfficientNetB0) and training parameters (number of epochs, number of units in the batch, learning rate) are set in config.yaml.

3 Dataset Preparation

The datasets supported in the project are pictures in APTOS and EyePACS, which can be downloaded on Kaggle. Upon download, the datasets are to be saved at the data/raw folder preserving the original folders (by class, and with labelled data). Preprocessing has to be undertaken so that data is assured of consistency before training. It requires executing the preprocess.py script that, in its turn, resizes, normalizes and augments the images with rotation, flipping, zooming, and brightness changes. Data/processed saves the processed images and may be used to train and test.

4 Configuration File Details

The param definition files determine the basic parameters of training and evaluation. These are the target resolution of the images to be constructed (e.g. 224224), their selection of CNN-based neural network (e.g. resnet50, vgg16, or efficientnetb0), and various parameters of importance, and most are the number of epochs, the number of batches, and the learning rate. It also defines paths to save model weights, and evaluation logs, and also- Grad-CAM outputs. The optimization and experimentation on the model can be performed by adjusting these parameters.

5 Running the Project

The process of training launched with the help of the train.py script where the required model architecture and settings of the training are sent as an argument. After training the trained model weights are saved into the models/ directory. One can evaluate the models through the evaluate.py script, which reports accuracy, precision, recall, F1-scores, AUC as well as visualizations that represent learning curves and confusion matrices. The system can subsequently be executed as a web application, incorporating use of the Streamlit script (app.py), availed through http://localhost:8501, where uploading of images and real-time Grad-CAM visualizations can be done.

6 Model Checkpoints and Logs

Model parameters are periodically saved as checkpoints during training, in case training or the model wishes to resume. These checkpoints are saved to the models folder and they are named in such a way that they contain the name of the epoch, as well as the validation accuracy so you can recognize them easily. Logs are kept in the logs directory and training and evaluation logs can be used to monitor performance and optimise hyperparameters.

7 Deployment and User Interface

The deployment phase resorts to the graphical user interface built with Streamlit, which helps to expose the model to ordinary users. After the interface is launched, the user can amend fundus retinal images; they then get filtered through the trained CNN to achieve classification. The level of severity of diabetic retinopathy according to the prediction is presented, together with a Grad-CAM heatmap that visually demonstrates how the retina areas affect its decision-making most of all. This form of prediction and explanation facilitates the clinical interpretation.

8 Troubleshooting

Mismatching of CUDA version, an unset dataset or directory and memory overflow during training are common issues. In case some CUDA-related errors arise, be sure the installed TensorFlow version corresponds to the installed CUDA and cuDNN versions. In the errors of datasets, ensure that the dataset has been put in the appropriate data/raw folder with the appropriate folder structure.

References

[1] kaggle.com. (n.d.). *APTOS 2019 Blindness Detection*. [online] Available at: https://www.kaggle.com/competitions/aptos2019-blindness-detection/data.

[2] kaggle.com. (n.d.). *Diabetic Retinopathy Detection*. [online] Available at: https://www.kaggle.com/competitions/diabetic-retinopathy-detection.