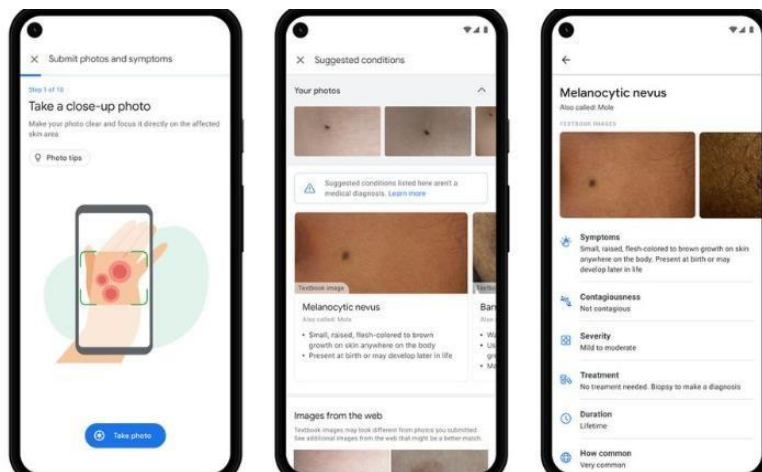


# AI in Project Management

---

## Assignment 7: Skin Disease Detection Model development

Masters in Artificial Intelligence 2024-25



Submitted by

**Alejandro Esperón Couceiro**

**Ayesha Munir**

23<sup>rd</sup> March 2025

## **Index of Contents**

1. Introduction
2. Xception model
3. Interface development
4. Steps to run the interface

## 1. Introduction

During the sprint, we were tasked with developing a simple interface that allows users to upload images, receive predictions, and display an alert when the analysis is complete. This report outlines the development of the interface and discusses the improvements made, including the implementation of a deep learning model to enhance prediction performance.

## 2. Xception model

To improve the model's prediction accuracy, we applied deep learning methods, particularly the Xception model, which is an efficient architecture for image classification tasks. Initially, we encountered challenges related to memory limitations when working with large images, resulting in poor model performance. This issue was addressed by leveraging the Xception layer, which helps manage memory usage more effectively.

We preprocessed the data, slightly resized the images to match the input size of Xception (71x71x3) and retrained the model. The performance of the new model was significantly better, achieving an accuracy of 48% on the test data. While this is a marked improvement, further enhancement is possible by increasing the image size. However, memory limitations remain a challenge that we are still working to overcome.

## 3. Interface development

The interface was developed within a Docker container, which runs Uvicorn to serve the application, allowing us to easily deploy and manage the HTML interface. The backend is built using FastAPI, which facilitates efficient image preprocessing and prediction generation.

Upon image upload and prediction, the system provides feedback via JavaScript, displaying a message to indicate whether the prediction process was successful or not.

Key Components of the Interface:

- **Frontend:** A simple HTML interface where users can upload an image and trigger predictions with the click of a button.
- **Backend:** FastAPI to handle image preprocessing, model inference, and serve the prediction results.
- **Alert System:** JavaScript code that provides a status message to the user after the analysis is complete.

## 4. Steps for run the interface

- **Navigate to the Project Folder:** Ensure you're in the root folder of the project.
- **Create the Docker Image:** Build the Docker image using the following command:

- `docker build -t counter-app .`
- **Run the Docker Application:** Launch the Docker container to run the app:
  - `docker run -p 8000:8000 counter-app`
- **Access the interface:** Open html file in the navigator
- **Upload images:** Load image from computer and press the predict button