Macromedia University of Applied Sciences

Course Title: Basics of Machine Learning
Name of Examiner: jeniffer Callou

To be completed by students:

B - 304415	B - UBr DT ATI 6e 24W
Student ID number	Matriculation
ANTONY	NIXON
Last name	First name

The student work will be submitted as:
(Please fill in the letter X in the appropriate box)

[x] Individual work

[ ] Group work

Does only apply to group work: (Complete only if it is a group work)

If you submit a group assignment, please list the first and last names of all group members. By entering their names, the students confirm that they agree to the assignment being submitted in its current form. The contribution of each group member must be indicated in the assignment (e.g. in the outline or chapter headings). Furthermore, by entering their name, the student declares that the entire project work, and in particular the part created by each group member, has been produced independently and without outside help. No aids other than those listed in the attached list of sources and AI tools have been used. All passages taken verbatim or paraphrased from publications are identified as such. Content generated using AI has been marked at the relevant point. Furthermore, it is confirmed that the use of AI tools and AI-supported aids is listed in full in the attached AI directory. It is also assured that all AI-generated content has been checked to the best of our knowledge and belief and in accordance with the general principles of good scientific practice. The work has not yet been submitted to any examination authority in the same or a similar form. By submitting the work, the group members agree that all assessments and comments made by the examiners will be stored in the uploaded work. The group member who uploaded the work must make the correction notes available to the other group members.

1) .... 5) ....
2) .... 6) ....
3) .... 7) ....
4) .... 8) ....

Assessment of group work:
(Please fill in the letter X in the appropriate box)

[ ] I apply for an <u>individual</u> evaluation (i.e. each member of the group will receive an individual mark)

[ ] I apply for a <u>group</u> evaluation (i.e. each member of the group receives an identical grade)

Berlin, 25.06.2025	NIXON ANTONY
Place/Date	Complete First Name and Last Name

Evaluation (according to grading scale), result of initial inspection: total points: _______

Date: **Name, first name First Examiner (to be filled in digitally)**

To be completed by the examiner: (Text area for the second examiner)

Vegetable Classification Using Convolutional Neural Network (CNN)

Project Overview

This project is about building a computer program that can recognize vegetables in images. We used a machine learning technique called Convolutional Neural Networks (CNN), which is especially good at understanding pictures. The aim was to create a simple, working model that can tell which vegetable is in a photo. This work is submitted as part of an academic course and shows how artificial intelligence can help with image-related tasks.

Objective

- To create a model that identifies 15 types of vegetables from images.
- To use simple preprocessing steps to improve training results.
- To test the model using new images to see how well it works.
- To understand and apply deep learning concepts in a practical way.

Dataset Information

- The dataset has 15 folders, one for each vegetable category.
- Each class includes about 1,000 training images.
- There are separate folders for validation and test images with 200 images per class.
- The pictures have different lighting and backgrounds to mimic real-life conditions.

Steps for Data Preparation

- All images were resized to 150 by 150 pixels.
- The pixel values were scaled between 0 and 1 for better learning.
- We used Keras ImageDataGenerator to load and process images in batches.
- The training data was shuffled to help the model generalize.
- No image augmentation was used in this version, but it can be added later.

Model Structure

The model is made using basic CNN layers:

- Input layer accepts 150x150 size RGB images.
- Three convolutional layers with 32, 64, and 128 filters, each followed by max pooling layers.
- Flatten layer converts data to a 1D format.
- Dense layer with 128 neurons and a dropout layer to avoid overfitting.
- Output layer with softmax activation to predict one out of 15 vegetable classes.

Model Training

- We trained the model for 10 epochs using batch size 32.
- RMSprop optimizer and categorical crossentropy were used.
- The validation accuracy reached over 90%, which is good for a basic CNN.

Making Predictions

After training, we saved the model. To test it:

1. Load the saved model file (.keras).
2. Load the image you want to test.
3. Resize the image to 150x150.
4. Convert it to an array and normalize it.
5. Use the model to predict the vegetable.
6. Print the name of the predicted class.

User Interface (Optional)

A basic web interface was created using Streamlit. This allows users to upload a photo and get the prediction directly in a browser. It is optional and just shows how the model could be used in a simple app.

Files Included in the Project

- vegetable_classifier_model.keras – The trained model file.
- vegetable_classifier_notebook.ipynb – Notebook with training and testing code.
- prediction_script.py – Script for testing the model with a new image.
- streamlit_app.py – Simple Streamlit-based web UI (optional).
- requirements.txt – List of required Python libraries.
- project_documentation.docx – Full project write-up.
- README.txt – Short instructions on how to use the files.
- image2.jpeg – Sample image for testing.
- screenshots/ – Folder with screenshots of training results and predictions.
- dataset_sample/ – Few example images from the dataset.

How to Run the Project

1. Make sure Python is installed and activate your virtual environment.
2. Open a terminal and go to the project folder.
3. Run the following command to install the needed libraries:

pip install -r requirements.txt

4. To make a prediction using a test image, run:

python prediction_script.py

5. To use the web interface, run:

    streamlit run streamlit_app.py

6. Make sure the .keras model file is in the same folder as the script or notebook when testing.

Conclusion

This project was a great way to understand how deep learning can be used in real-world tasks. We learned how to prepare data, train a CNN model, and test it on new images. The results were good and show that even a simple CNN can work well when trained properly. More improvements can be made by adding more data and using advanced techniques, but this version already shows the main concepts clearly.