Assignment 1 - Computer Vision Concepts Implementation (8890 CVIA PG)

The context of this assignment is the common computer vision task object detection and recognition. Specifically, the task is to implement and evaluate classifiers to automatically recognise different species of birds in images.

Dataset:

We will use the Caltech-UCSD-Birds_200_2011 dataset. This dataset contains 11,788 images of 200 bird species (in the range of 40-60 images per bird species). Here are some examples:

Figure 1. Example images: Parakeet Auklet (left), Green Kingfisher (centre), and Horned Puffin (right)

The dataset is available on the Nutanix SciTech Student Virtual Desktop with GPU (in folder U:\Faculty of SciTech\Units\un8890 – Computer Vision and Image Analysis PG\CUB_200_2011) or you can download a ZIP file from OneDrive. The size of the ZIP file is about 1.1GB.

The images can be found in the `images` folder, with a subfolder for each of the 200 classes. The `images.txt` file contains a list of the individual image file names including the subfolder path. The `classes.txt` file contains the class names (bird species). The `image_class_labels.txt` file contains the ground truth class labels (bird species label) for each image. The dataset is 'benign' in the sense that each image contains only one bird. The `bounding_boxes.txt` file contains the (x, y, width, height) parameters of a single bounding box for each image, describing the image area that contains the bird. More information about the structure of each of these files can be found in the `README.txt` file.

We use a 60:20:20 split for training, validation, and test data. That is, for each class, the first 60% of images are to be used for training, the next 20% of images for validation, and the remaining 20% for testing.

Task:

Your task is to implement and evaluate classifiers for the automatic classification of bird images from this dataset. Specifically, you are asked to **implement** (both!):

1. A handcrafted image feature method with a classic machine learning approach (e.g. SIFT features with a Support Vector Machine classifier).
2. A deep learning-based approach, this is both features and classifier are learned together (e.g. a Convolutional Neural Network).

The choice of specific method/approach is up to you, as are the specific parameters (for example, will you resize the images first? How many layers does your DL network have?) but must be clearly documented in the Matlab code and described in the performance evaluation report (see below).

Both types of approaches need to be evaluated in two scenarios (both!):

1. The entire images are used as input.
2. Only the image area marked by the bounding box information is used as input.

Consequently, you will need to perform four experiments to **evaluate the performance**:

	Classic handcrafted feature + classic ML classifier	Deep learning feature and classifier
Whole image as input	Experiment 1	Experiment 2
Bounding box image area as input	Experiment 3	Experiment 4

Performance evaluation here means (a) the class-weighted average accuracy and (b) the confusion matrices with individual class correct and incorrect recognition rates for the test partition of the data.

You will then need to run a fifth experiment that performs fivefold cross-validation on Experiment 4. Assume we partition the data in each class into five parts of 20% each. We can then run the experiment five times with different parts used for training, validation and test, while maintaining the 60:20:20 split. For example:

- Run 1: Parts 1-3 for training, Part 4 for validation, Part 5 for test
- Run 2: Parts 2-4 for training, Part 5 for validation, Part 1 for test
- Run 3: Parts 3-5 for training, Part 1 for validation, Part 2 for test
- Run 4: Parts 4, 5, and 1 for training, Part 2 for validation, Part 3 for test
- Run 5: Parts 5, 1, and 2 for training, Part 3 for validation, Part 4 for test

Report on the average overall accuracy and the average class correct and incorrect recognition rates by averaging across the five runs. This is what we mean by cross-validation.

Finally, you are to **write a short report** (max two A4 pages) that describes:

- Briefly describe the methods/approaches you chose to implement, including any specific parameter choices. **[8 marks per method (classic vs deep learning) = 16 marks]**
- Present and discuss the results of the performance evaluation of the first four experiments. **[4 marks for each of the four experiments = 16 marks]**
- Present and discuss the results for the fivefold cross-validation experiment **[8 marks]**
- Describe the lessons learnt: What could be done differently to improve the results? **[5 marks]**

A further **five marks** will be awarded for using good programming principles such as clear program structures, good use of comments, use of functions (where appropriate) etc.

Total: 50 marks