

# CT5133/CT5145 2020/2021 Assignment 2: Transfer Learning, Convolutions, and Object Localisation in Keras

James McDermott

- **Weight:** 20% of the module
  - **Deadline:** Midnight Sunday 2nd May, end of Week 11.
  - **Teams:** You may work solo, or in a group of 2. The assignment is the same either way. If working in a group of 2, you may not work with a student with whom you have worked in any previous assignment in this module or any other module.
1. Create a labelled image classification dataset, OR choose an existing one. It should include **exactly 2 classes**. The classification problem should be suitable for deep convolutional networks, e.g. not trivially solved with colour thresholding. If you like, you can choose just two classes from an existing dataset. If using an existing dataset, do not use MNIST, Fashion MNIST, CIFAR, ImageNet, or any subset of them. **Update:** do not use a dataset created for any previous module, such as the SUV versus Sedan car dataset. Write a short paragraph to describe the dataset: background, size/shape, classes, method of collection. Include a typical image from each class.
  2. Choose any pretrained image classification model, e.g. one of those found at <https://keras.io/api/applications/>. Import it, excluding the dense classification head. Make sure to use the appropriate preprocessing for input images. We will call this the base model. Add layers to the base model and train them to produce classifications on your dataset. Report your choice of base model; state and explain the shape of the base model's output; explain the layers you added and training carried out; report classification performance on unseen data.
  3. Now investigate the base model as follows. Extract a scalar output from each neuron in the final layer of the base model. You can use a `GlobalMaxPooling2D` layer or `GlobalAveragePooling2D`, or another method if you prefer. Add a `Softmax` layer after. Using this, **identify one or more neurons whose scalar output is strongly correlated with the class label of your dataset**. Such neurons can be seen as (weak) class detectors. Specify your method, state results, and give brief interpretation of the results.
  4. Now build a simple approach to object localisation, that is deciding **where** in the input image an object is present, as follows. For each neuron identified in (3), write code to **visualise its 2D output**. Use this to write a brief comment: **does this 2D output allow localisation of objects in the input image?** Include several example images and the corresponding 2D outputs to support your comment. Note: if we fail to make step (3) work, step (4) can still be attempted by choosing some neurons manually. By the way, this approach is related to but not very similar to state of the art systems for object localisation, such as YOLO. We are not required to investigate such systems in this assignment.
- **Submission:** Submit **either** a `.ipynb` file, or a `.pdf` and a single `.py` file. Include student name(s) and ID(s) at the top of every file submitted. The report and code should be in four sections corresponding to points 1-4 above. Code should be well structured and commented. Include a citation and/or URL for any sources used, such as code, papers, and blogs. Do not submit the dataset, but include a link, source, or download instructions if available. Marks will be deducted for incorrect submission formats.
  - **Plagiarism:** NUI Galway guidelines on plagiarism apply. You may discuss the assignment with other

students/groups but may not view any part of their solution or allow them to view yours.

## Grading Rubric

### 1. **Dataset** (10%)

- A. Choice of a non-trivial dataset, with a full description of relevant details and representative images.
- B. A non-trivial dataset, but lacking some description.
- C. A more trivial dataset, OR lacking important details of description.
- D. A trivial dataset AND lacking important details of description.
- E. No real attempt.
- F. No attempt.

### 2. **Transfer learning** (30%)

- A. Correct use of a pre-trained model for the new task, with appropriate choice of classification head and re-training. Results reported and interpreted.
- B. Complete use of a pre-trained model, but with some inappropriate/unmotivated choices. Complete results with some flaws.
- C. Incomplete or incorrect use of a pre-trained model. Incomplete results or lack of understanding.
- D. An attempt at implementation with significant omissions; no real results.
- E. No real attempt.
- F. No attempt.

### 3. **Inspecting convolutional features** (30%)

- A. A complete implementation of the requirements, in clean, idiomatic Keras code with good comments, leading to a clear result with understanding.
- B. Some flaws in execution and code quality, but basically complete, with a result.
- C. Larger flaws which cast doubt on correctness of the result. Or a good attempt which is somewhat incomplete. Lacking interpretation.
- D. Significant flaws and omissions, and no real result.
- E. No real attempt.
- F. No attempt.

### 4. **Object localisation** (30%)

- A. Complete implementation in clean code, leading to multiple visualisations and an interpretation/comment on the possibility of localisation.
- B. A complete but flawed implementation with good interpretation, OR a complete implementation lacking interpretation of the results.
- C. Some flaws in implementation/quality, and little interpretation/understanding.
- D. Significant flaws and no interpretation/understanding.
- E. No real attempt.
- F. No attempt.