Deep Learning Classification Applied to Four Insect Genera in Order to Complement and Assist Citizen Science Monitoring of Irish Small Streams

Ciara O’Hara

*CCT College Dublin*

Ireland

sbs23015@student.cct.ie

***Abstract* - Deep learning was applied to a species classification task using CNN**

***Keywords – deep learning, CNN, machine learning, citizen science, species classification***

1. INTRODUCTION

Citizen science is the use of interested but non-professionally trained individuals who can be utilised to collect and analyse data. It may be used to bridge the gap between the resources available and the resources necessary in a geographical area, or professional field (Westphal et al., 2022). One such citizen science project is the Small Stream Characterisation System (SSCS) being put in place by the East Wicklow Rivers Trust (wicklowrivers.ie, no date), that the author has participated in. This course is designed to train interested individuals in the characterisation and monitoring of small streams by training them on hydromorphology and in the sampling and identification of indicative flora and fauna. The idea is to create a network of people across the county who regularly assess the condition of a stream or streams local to them. While the participants are trained in the classification of the river fauna, classification is complicated, especially amongst species from the same order, that nonetheless can indicate opposing river conditions. It is considered that species classification might be improved by applying deep learning techniques to images of fauna that are taken by the volunteers. Some relatively basic classification would need to be done by the volunteers, to exclude species that are not diagnostic of either polluted or pristine rivers (such as ), and to reduce the species submitted for classification down to a set of known species which the model has been trained on.

1. DEEP LEARNING APPLIED TO SPECIES IDENTIFICATION

Deep learning can assist even outside of citizen science applications by providing professionals with a rapid and efficient means of identifying species (Wäldchen and Mäder, 2018). Allowing large datasets to be rapidly processed ensures that they can be used to maximum benefit (Westphal et al., 2022), while traditional species identification can be costly as well as time consuming (Banan, Nasiri and Taheri-Garavand, 2020). A novel use of machine learning was utilised by Westphal et al. (2022) in filtering live-stream video for static images that include the species of interest, in this case, beluga whale. Their algorithm was able to distinguish images that contained beluga from images that didn’t (images with no beluga, that may have had other species, or had just water) with an accuracy of 97%. To improve retention of volunteers the method used in gathering the data must be quick and simple, and photographing organisms fits this requirement (Newcomer et al., 2019). Westphal et al. (2022) found that, using citizen scientists to classify video frames into ‘beluga’ and ‘no beluga’ reduced the level of participation of the volunteers due to the preponderance of ‘no beluga’ frames.

While in-person species diversity measurements are more accurate, it has been found that photographic analyses can accurately capture the relative abundance of species and functional groups (Newcomer et al., 2019). It is relative abundance that is key in determining the quality of small streams, so a method that accurately determines the relative abundance of relevant species could be usefully applied to photographic data gathered via citizen science. This would remove the need to train non-scientists to accurately distinguish species. Rather, the citizen scientists would only need to be trained to collect – as per the current approach - and photograph the species present in the sample collected. It is important that the process incorporates a relatively straightforward and repeatable method of gathering the image data. As the organisms are currently collected in a sample tray for identification and counting, imaging them would be relatively straightforward and repeatable. This is similar to the study by Newcomer et al, which noted that the use of settlement panels for assessing the assessing the status of marine environments are ideally suited for photographic analysis.

These images could then be uploaded and processed by the model, and species’ relative abundance determined. An added benefit would be that the images would be available for later examination by a trained professional, in the cases of outlier streams that were found to be exceptionally poor (or exceptionally good), or to spot check the citizen scientists’ work.

Rajabizadeh and Rezghi (2021) applied both traditional machine learning and deep learning (neural network) techniques to the classification of images of six Iranian snake species. Having found that feature extraction – specifically LDA – significantly improves the performance of traditional classifiers, the SVM classifier was found to give an accuracy of 84%. Of the traditional machine learning algorithms SVM (rbf kernel) performed best out of those tested, namely KNN, Logistic regression, and SVM. The authors note that the most appropriate dimension reduction algorithm to use depends on the dataset and the task, rather than the classifier, so a trial-and-error approach to determining the appropriate approach should be adopted. The researchers then turned to deep learning and achieved an accuracy of 93% using a CNN classifier.

The CNN architecture used by Rajabizadeh and Rezghi (2021) was MobileNetV2 (which can even be used on mobile devices), with 5,147,206 parameters over 150 epochs with an SGD optimizer and a learning rate equal to 0.0001 and momentum of 0.9. Images fed to the model were initially resized to 24 x 224 pixels. Their model was pre-trained using images from ImageNet to attain the initial weights and to carry out transfer learning, in order to decrease the training time of the model. Transfer learning was again used by Westphal et al. (2022) for two of their architectures, AlexNet and ResNet50. Transfer learning is recommended to not only improve training effectiveness, but also to reduce the size of the requisite training dataset and to reduce the computational load (Banan, Nasiri and Taheri-Garavand, 2022). In order to improve performance on the VGG-16 model they used attention mechanisms to focus the model on significant regions. Rajabizadeh and Rezghi had initially trained a model with no initial weights, but this model did not train effectively, indicating the importance of initial weights.

Alternative CNN architectures that could be used include ResNet (the first model to beat human accuracy in a classification task), AlexNet and VGGNet (Wäldchen and Mäder, 2018). VGG-16 was also tested by Rajabizadeh and Rezghi (2021), but MobileNetV2 provided better accuracy. Westphal et al. (2022) carried out 2-fold cross validation while training a CNN model on 3 architectures: AlexNet (8 layer architecture), VGG-16 (16 layers, but smaller filters) and ResNet50 (50 layer architecture). Banan, Nasiri and Taheri-Garavand (2022) also used VGG-16 to classify 4 species of carp with 5-fold cross validation. Westphal et al. (2022) achieved over 97% accuracy with VGG-16, and over 96% with the other architectures trained. Researchers classifying fish species using neural networks have achieved accuracies up to 93% (Banan, Nasiri and Taheri-Garavand, 2022). VGG-16 has also been successfully used in other image classification areas, with an accuracy of 96% achieved in differentiation those wearing a covid-19 prevention mask, and those without (Lavanya, 2022).

Rajabizadeh and Rezghi (2021) did not carry out feature extraction before training their deep learning model, and indeed neural networks do not require preprocessing with feature extraction as they carry out the step automatically in the convolution (Wäldchen and Mäder, 2018; Kumar, 2022). Some researchers have cropped images to focus on particular taxonomic features (James, 2017), but this approach would not be applicable to big data – for a big data processing algorithm the images gathered would need to be processed independent of any human input. The image gathering and processing approach needs to be simple and repeatable. However, R-CNN may be applicable in species identification tasks, such as identifying the head, tail, or legs of the species in question, before running the classifier algorithm (Thakur, 2020).

Rajabizadeh and Rezghi (2021) found that it was the colour pattern and shape (the dorsal patterns) that the model used to discriminate between species. The authors determined this by visualising the images after processing had been carried out by the various hidden layers. This knowledge could, in future studies, potentially be used to fine-tune a CNN model by informing on the most appropriate convolutional filter/s to use for the convolution.

The visualisation of intermediate images by Rajabizadeh and Rezghi (2021) was done using a technique developed by Selvaraju et al. (2020) who extracted visual constructs from the final convolutional (output) layers to determine the class-specific information in the image (in other words, the parts of the image that the model is assigning to each class). They did this by generating a heat map of the most highly weighted feature map activation areas.

Rajabizadeh and Rezghi (2021) used accuracy, precision and recall to evaluate the performance of the models (accuracy alone for the traditional algorithms). There were not enough samples to split the data into train and test here, so precision will not be used. Westphal et al. (2022) used accuracy with confusion matrices, and Area Under Receiver Operating Characteristics curve to evaluate performance, though the ROC-AUC did not vary significantly amongst the 3 models trained.

While CNN algorithms tend to perform better than traditional machine learning methods e.g. Rajabizadeh and Rezghi (2021) study, for image-based species classification, the opposite has been reported, notably in the identification of birds by Islam et al. (2019). It is also the case that researcher experience and expertise can make the difference between a high performing model and a poor one, for example, by applying appropriate preprocessing to the data prior to training the model, and task-specific properties, such as multi-scale images, such as small scale patches in the image that show an animal’s teeth, for example, can improve a CNN’s performance (Wäldchen and Mäder, 2018), though this would be less applicable to a citizen science application.

Rajabizadeh and Rezghi (2021) note that training a deep CNN algorithm requires a large dataset, but that images of the snakes being classified in their study - with at least 50% of the snake’s body visible in the image - were not readily available. They used a final dataset of 594 images in total. Westphal et al. (2022) used 12,678 labelled images, classified by citizen scientists. Unfortunately, there are no datasets in ImageNet for the species being examined here so this was not replicated. ImageNet, or another publicly available naturalist’s database such as iNaturalist (iNaturalist, Accessed 17/08/2023) would also have been useful in increasing the size of the dataset being used here, but images of all species being examined here were sparse. For Ireland, iNaturalist had only a handful of images of mayfly *Serratella* and *Baetis* genera in the nymph stage, with only a few more each across Europe; the Irish images were used as training data in the model. This is currently a common problem with deep learning classification of animal species, as labelled datasets are still rarely available for training (Wäldchen and Mäder, 2018).

1. PARALLEL PROCESSING OF LARGE DIGITAL IMAGE SETS USING APACHE SPARK

Apache Spark may be used to reduce processing times of image datasets, especially with high complexity algorithms. For example, [Téllez-Velázquez](https://onlinelibrary.wiley.com/authored-by/T%C3%A9llez%E2%80%90Vel%C3%A1zquez/Arturo) and [Cruz-Barbosa](https://onlinelibrary.wiley.com/authored-by/Cruz%E2%80%90Barbosa/Ra%C3%BAl) (2019) found that speeds of up to 54x were possible by extracting and storing image rasters in Spark’s RDDs and dataframes, and by treating all subsequent image processing steps as transformations on these. Similarly, Spark may be used to improve speed and reduce computational load in the processing of large images (Le Nir et al., 2020). Lavanya et al. (2022) utilised a Spark framework to fit a MobileNet model to large numbers of images gleaned from facial detection cameras.

1. DEEP LEARNING APPLIED TO SMALL STREAM SPECIES OF INTEREST

This application does not currently fulfil the characteristics of big data due to the small number of images available. It is just to test whether the approach is feasible as a deep learning application. Other citizen science ecology / biodiversity research areas could fulfil the characteristics of big data, with large volumes of varied data rapidly gathered through camera traps, for example. Nonetheless, an accurate, well-trained deep-learning model trained to classify the various species of interest for small streams would be beneficial in terms of saving time and potentially reducing costs for researchers and volunteers surveying the habitats, as well as potentially improving species classification accuracy for species image data gathered by volunteers.

Camera traps not suitable in this environment as the method of collection is kick-sampling stream beds, and using pond nets to gather the organisms that are disturbed. These are then introduced to a sample tray for identification and counting. It is the identification step that is challenging for volunteers - and where deep-learning may come in - as there are many similarities between the species of interest, especially to the untrained eye. There are 5 key macroinvertebrates of interest, mayflies, stoneflies, caddisflies, the water louse, and a varied group including snails, worms, flatworms, leeches and true flies. For the purposes of this project the final group will not be included, as they are very easily distinguished from the others, and also because it is a varied group, whose members don’t have features in common with each other.

1. METHOD

Finding images of species within the four groups was challenging, as it must be limited to individuals in the nymph stage for the 3 fly orders. 10 images of each were gathered, with some coming from the iNaturalist website. The mayfly genus used for this initial classification task was Ecdyonurus, as it is a very sensitive indicator or water quality. If it is absent – even in the presence of other mayfly genera – it indicates that there is something amiss with the quality of the stream. There are many other mayfly species that - by their presence or absence – can indicate good or poor water quality, respectively, but the aim of this part of the study was simply to distinguish mayfly nymphs from the other fly nymphs, and from the water louse.

The next part of the project was to attempt to distinguish different mayfly species from each other. Ecdyonurus was used again, along with Serratella ignita, another species whose absence can indicate a problem with water quality. However, there is a genus of mayfly – Baetis – that is very common, and that is also tolerant of pollution. The presence of this mayfly can confuse the issue, especially if mistaken for a species whose absence actually indicates a problem with the stream’s quality. Therefore another task was to attempt to train a model to distinguish mayfly species from each other, not just ‘*Baetis*’ or ‘not *Baetis*’, but to distinguish between a number of species (because information on the number of different species is, in itself, useful).

TRAINING A SIMPLE CNN MODEL

Initially a simple CNN model was trained with a

UTILISING DISTRIBUTED SYSTEMS

It was attempted to use PySpark to distribute computational load from within Jupyter Notebook, but a functioning system was not achieved until PyTorch was used (PyTorch, 2023). A number of approaches were tried. The author successfully loaded images using PySpark but could not use them in any way to either use with a CNN model (or any model), or to convert to a pandas dataframe and carry out deep learning outside PySpark. Although the file source for the images was in the hdfs and hadoop had been started, and was confirmed to be running, the error returned every time was ‘java.io. FileNotFoundException: File does not exist:’. Restarting the kernel, shutting and unshutting the hdfs, deleting and re-adding the images to the hdfs and restarting PySpark made no difference. It was also attempted to cache the image data before querying them, but that made no difference. It appeared that PySpark was renaming he images at random, and then failing to find the renamed images because the rename did not exist in the file path. For example, the error: ‘File does not exist: /user1/Species\_images/Stoneflies\_Plecoptera/stonefly%209.jpeg’ was true. The files were named stonefly1, stonefly2, ... stonefly10. It could not be found why PySpark was renaming the files then attempting to find the renamed files in the hdfs, after initially importing the images successfully (with the correct file names). Images in appendix 1 show that PySpark had initially read the images successfully. Searching for a resolution to the problem seemed to indicate that the issue was caused by the source file being located in a local folder rather than in a hdfs, but, as shown, this was not the case here (user1 is a folder on the hdfs, as shown).

Another approach was to try to use a Keras2DML wrapper to utilise parallel computing (following the Towards Data Science tutorial on the subject (Purkait, 2018)), but unresolvable errors were encountered in trying to import any mllearn algorithm (errors were ‘import \* only allowed at module level). It was also tried to use SparkDL (WRITE ABOUT THIS) but those packages failed to import with error

Finally, PyTorch was used successfully to implement an image transformation, resize and model fit with a pre-trained VGG-16 model, following a sample image processing tutorial outlined by Tandon et al. (2022). Transfer learning was used with success by various researchers working in species classification, as outlined above, and is recommended if there is a suitably trained model. VGG-16 has not, however, been trained to classify these genera, but it was tested (just for demonstration purposes) to see what it would classify the various species as.

1. RESULTS

A simple CNN model with batch size 4 (due to the small number of images, 40) gave a very high accuracy of 1, and a loss of 0.05 at the final epoch when classifying between the mayfly, caddis fly, stonefly and waterlouse groups. This is not surprising since they are not even of the same taxonomic class – water lice are class Copepoda, while the others are class Insecta – but even the 3 insect classes are different Orders (Ephemeroptera, Trichoptera and Plecoptera), so the fact that these are easily classified is, maybe, not surprising. With 25 filters applied this model ran with no issues because the images were small. A more challenging deep learning task would be classifying the various mayfly genera accurately, of which there were 3: *Ecdyonurus (dispar), Seratella (ignita) and Baetis*, and also running the model outside without distributed processing, as the image sizes were bigger, which caused the Jupyter notebook kernel to crash. It was attempted to put in place a distributed system with Apache Spark in a few different ways, but all failed. Instead, the image sizes were reduced in order to ensure that the model could run. Details of the attempt made to utilize Spark in Python are outlined in the Methods section.

Once images were reduced to 250 x 250 pixels, the deep learning model could be fit without error.

Using the pre-trained VGG-16 model the vast majority were classified as American lobsters, with a few jellyfish, one black widow, a platypus, a mouse, and a pirate (!) also returned.

REFERENCES

Anon, (n.d.). *Small Streams Characterisation System (SSCS) Course | East Wicklow Rivers Trust*. [online] Available at: https://wicklowrivers.ie/small-streams-characterisation-system-sscs-course/ [Accessed 12 Sep. 2023].

Banan, A., Nasiri, A. and Taheri-Garavand, A. (2020). Deep learning-based appearance features extraction for automated carp species identification. *Aquacultural Engineering*, 89, p.102053. doi:https://doi.org/10.1016/j.aquaeng.2020.102053.

iNaturalist. Available from https://www.inaturalist.org. Accessed 17/08/2023.

Islam, Shazzadul, et al. “Bird Species Classification from an Image Using VGG-16 Network.” *Proceedings of the 2019 7th International Conference on Computer and Communications Management*, 27 July 2019, <https://doi.org/10.1145/3348445.3348480>.‌‌

James, A. (2017) ‘Snake classification from images’, *PeerJ Preprints* [Preprint]. doi:10.7287/peerj.preprints.2867v1.

Kumar, A. (2022). *Different Types of CNN Architectures Explained: Examples*. [online] Data Analytics. Available at: <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/>.

Lavanya, K., Prakash, S., Gedam, Y., Aijaz, A. and Ramanathan, L. (2022). Real Time Digital Face Mask Detection using MobileNet-V2 and SSD with Apache Spark. *International Journal of Performability Engineering*, 18(8), p.598. doi:https://doi.org/10.23940/ijpe.22.08.p8.598604.

Purkait, N. (2018). *How to train your Neural Networks in parallel with Keras and Apache Spark*. [online] Medium. Available at: https://towardsdatascience.com/how-to-train-your-neural-networks-in-parallel-with-keras-and-apache-spark-ea8a3f48cae6 [Accessed 22 Sep. 2023].

PyTorch (2023). *PyTorch*. [online] Pytorch.org. Available at: https://pytorch.org/.

‌Rajabizadeh, M. and Rezghi, M. (2021). A comparative study on image-based snake identification using machine learning. *Scientific Reports*, [online] 11(1), p.19142. doi:https://doi.org/10.1038/s41598-021-96031-1.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, [online] 128(2), pp.336–359. doi:https://doi.org/10.1007/s11263-019-01228-7.

Tandon, A, Ryza, S, Laserson, U., Owen, S. and Wills, J. (2022). *Advanced Analytics with PySpark*. O'Reilly Media, Inc.

Téllez-Velázquez, A. and Raúl Cruz-Barbosa (2019). A Spark image processing toolkit. Concurrency and Computation: Practice and Experience, 31(17). doi:https://doi.org/10.1002/cpe.5283.

Thakur, R. (2020). Step-by-Step R-CNN Implementation From Scratch In Python. [online] Medium. Available at: <https://towardsdatascience.com/step-by-step-r-cnn-implementation-from-scratch-in-python-e97101ccde55>.

Wäldchen, J. and Mäder, P. (2018). Machine learning for image based species identification. *Methods in Ecology and Evolution*, 9(11), pp.2216–2225. doi:https://doi.org/10.1111/2041-210x.13075.

Westphal, A.M., Breiter, C.J.C., Falconer, S., Saffar, N., Ashraf, A.B., McCall, A.G., McIver, K. and Petersen, S.D., 2022. Citizen science and machine learning: Interdisciplinary approach to non-invasively monitoring a northern marine ecosystem. *Frontiers in Marine* Science, 9, p.1738.

Yannick Le Nir, Devin, F., Baldaquin, T., Pierre Mesler Lazennec, Ji Young Jung, Kim, S., Hyeyoung Kwoon, Nilsen, L., Yoo Kyung Lee and Laffly, D. (2020). Image Processing in Spark. pp.163–171. doi:https://doi.org/10.1002/9781119720553.ch5.