**Assignment 02**

**Part A**

1. The two image datasets used for the class were a corgi plushie and a novel. I took 50 images for each object. I decided to use these objects because I could take a variety of interesting pictures with them (e.g., showing different pages of the book.)

Here is an image of the corgi plushie: Here is an image of the book:

A stuffed dog on a wood floor

Description automatically generated A book on a wood surface

Description automatically generated

1. For this assignment, we are asked to create an image detection AI model to put objects into their specified category.
2. These are the steps used to train the model:
   1. First, I created an account on Edge Impulse
   2. I added a device that can access the training models.
   3. I uploaded the image folders of each object, plus a third folder named Others with 50 pictures of random objects, all folders had the option of dividing their content for training and testing (i.e., 80% of images of each folder were used for training and 20% were used for testing), I also added labelling on each object.
   4. After uploading the images, I created an impulse for the model.
   5. After creating the impulse, I ensured the raw images were on RGB and then generated their features.
   6. Once their features were generated, I selected neural network settings and a model and then started training the object detection model.
   7. After transferring images into the model, I enabled EON Tuner and selected the most accurate architecture that does not exceed the performance limits of the model.
   8. After selecting the architecture, I retrained the model.
   9. Once retrained, I did a live classification by loading several samples from each image category.
   10. Once I tested the image samples, I did the model testing to verify the model's accuracy.
3. The model had very good results in terms of accuracy after the model testing (i.e., 90% accuracy). However, due to its lack of clustering, the model is not precise. Regarding its recall, the model can distinguish the images and put them in the right category.
4. Generated feature clustering:

A screen shot of a graph

Description automatically generated

Initial trained model:

A screenshot of a computer

Description automatically generated

Live classification:

A screenshot of a computer

Description automatically generated

Model testing results:

A screenshot of a computer

Description automatically generated

The accuracy of the model increased while testing the unknown images. The clustering was very bad, regardless of the model training stage. Most of the data is spread out. The only step in the process where the clustering is a bit more defined in the initial model training.

1. I think better is about accuracy, precision, and relatively tight clusters. It is when the model can distinguish the images and associate them with its specified label. I think this model could improve its clustering because the data was always spread out in the graphs. I think choosing another model basis and a more accurate architecture could help to the improvement of the model’s precision.

**Part B**

1. I used Pat's image dataset for the second part of the assignment. This dataset contained three object folders: fork pictures, pin pictures and miscellaneous items. Each folder contained 50 images. Pat had the generosity of sharing their images with our group, so I used their dataset for the assignment. Because I have no idea of what the images look like, I decided to use the fork and miscellaneous items folders from Pat’s dataset, so I am sure I am not biasing any results. I combined them with my corgi image folder because the colour tones between the fork and the plushie are similar (or at least in the same range of colour.)

Here’s an image of the fork: Here’s an image of the corgi plushie:

A wooden fork on a white surface

Description automatically generated A stuffed dog on a wood floor

Description automatically generated

1. For this part of the assignment, the goal was to train a model by using someone else image dataset and verify if there are any issues within both datasets while generating with unknown images. It allows us to verify any biases involved within the previously trained model and make sure that the images from both datasets provide accurate or reliable results.
2. Overall, the model is highly accurate (95% for the initial trained model and 90% for the model testing results). It is a bit hard to define if the model is precise or not because at certain stages of the training, some results shown very tight clusters and others spread out data, but the model has still some imperfections regarding its precision. Regarding its recall, the model can distinguish the images and put them in the right category.
3. Generated feature clustering:

A screenshot of a graph

Description automatically generated

Initial trained model:

A screenshot of a data analysis

Description automatically generated

Live classification:

A screenshot of a computer

Description automatically generated

Model testing results:

A screenshot of a data analysis

Description automatically generated

From the results of the graphs, the model's accuracy decreased when the model was used on the testing images. The clustering is very tight in the live classification. There are a lot of overlaps between the data. However, the data is spread out on the initial trained model; no clustering occurs. The generated features are the only well-balanced clustering in this trained model.

1. This model performed better in precision and accuracy. The results in the model training and model testing were highly accurate, which was not the case in the model in Part A. Some tight clustering happened at certain stages of the training, which was not the case in the Part A model.

**Part C**

I think this type of object recognition model can be used to collect data about the fauna of a forest and see if there’s a risk of endangered animals within the environment. For example, if we do not detect many beings of a certain species, it can be an indicator of extinction.

And provide a storyboard describing the scenario above as well.

Once the folders were uploaded, I selected the position of the object in each image and labelled them with their specific category