FINAL PROJECT ANALYSIS REPORT

Presented to Mehrdad Tirandazian

COMP 263-002 Deep Learning - Centennial College

Adrian Posadas

Centennial College

aposadas@my.centennialcollege.caAiju Kinoshita

Centennial College

akinoshi@my.centennialcollege.caJestoni Vasquez

Centennial College

jvasqu16@my.centennialcollege.caOscar Quispe Guanoluisa

Centennial College

oquispe@my.centennialcollege.ca

# Description and Scope of the Problem

Using digital images of dog facial expressions, the problem of dog emotion in neural networks is a classification task aimed at predicting a dog's emotional state. There are 4000 images of dogs used in this task, each labeled with one of four emotions: happy, sad, angry, relaxed.

In this problem, we will experiment with different deep learning techniques, such as supervised and unsupervised learning, and state-of-the-art models, to predict the emotions of dogs.

In supervised learning, data is divided into training, validation, and testing sets, and a Convolutional Neural Network (CNN) architecture is designed and trained on the training set. In the validation set, hyperparameters will be tuned, and in the testing set, their performance will be evaluated. This model will be evaluated for accuracy, precision, recall, and F1 score.

Unsupervised learning involves learning the features of the images without labels using GAN data augmentation model trained on the dataset. The encoder part of the model extracts the features of the images, which are then used to train a clustering algorithm to group similar images. In order to measure the accuracy of the clustering algorithm, we will separate the images according to their emotional states.

A state-of-the-art model will be trained using pre-trained models such as ResNet, and Inception and VGG, which will extract features from the images, and these features will be used to train a classifier for predicting dogs' emotional states. A comparison will be made between the accuracy of these models and the results obtained from supervised and unsupervised learning.

A successful solution to this problem can have a wide range of applications, such as animal behavior research, pet care, and veterinary science.

# Dataset

Dataset for training have retrieved from Kaggle dataset: <https://www.kaggle.com/datasets/danielshanbalico/dog-emotion>

### Dog-Emotion dataset:

This dataset consists of images of dogs with labels indicating their emotional state: angry, happy, relaxed, or sad. The images were collected from various online sources.

Number of Images: The dataset includes 4000 images of dogs in total, with 1000 images for each emotion category (angry, happy, relaxed, sad).

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

License: [CC0: Public Domain](https://creativecommons.org/publicdomain/zero/1.0/)

A black dog with its mouth open

Description automatically generated with medium confidenceHere is the sample image from the dataset:

Shape of the image: (384, 384, 3)

Pixel value range: 0-255

# Feature Engineering

Since resolution of 384x384 would lead our model training so time consuming, we compress the image accordingly to the use cases:

For Supervised Image Classification model: (192, 192, 3)

For Unsupervised Image Augmentation model: (32, 32, 3)

For State-of-Art Image Classification model: (192, 192, 3

For our label, we currently have four class values.

We would like to have numerical classes, so we one-hot encoded the label.

Practically, scaling the pixel values into the range of 0-1 increases model quality. Thus, we scale the images’ pixel value into 0-1 by dividing all pixel values with 255.

We also stratified split the dataset into training and validation dataset to evaluate the model in each epoch.

The ratio of Training:Validation is 17:3. We set random state to 42 to retrieve static output for developing.

# Methodologies

We use Convolutional Neural Network architecture since we are dealing with images to classify what the dog in the image feels for supervised classification model,

# Supervised Image Classification Model - CNN

## Parameters

As a parameter we will use 25 epochs for training supervised model.The model is built using TensorFlow keras library.

## Modelling

The first layer is constructed with Convolutional 2D layer with 64 units, 3x3 window sizes, ReLU activation function, and “same” padding. Note that input\_shape parameter uses size of image as we specified in prior section (192, 192, 3). The second layer is constructed with MaxPooling2D layer with 2x2 pool size to pool the output from our prior Convolutional 2D layer. For the third layer, drop out the output units of the second layer for 20% to avoid overfitting. For the fourth layer, we constructed with second Convolutional 2D layer with 32 units, 3x3 window size, and ReLU activation function. Then for the fifth layer, we pool the output in another MaxPooling 2D layer with 2x2 pool size. Again, drop the trained units for 20% in sixth layer. Before we pass to the seventh layer, we flatten the output from the fourth layer because the fifth layer is Dense layer. The Dense layer for the seventh layer is constructed with 64 units and ReLU activation function. For the eighth layer, we drop the trained units at a rate of 50% to counterplan overfitting. The final layer would have 4 units and Softmax activation function to retrieve the predicted output. The model is compiled with Adam optimizer with learning rate of 0.001 and Categorical Cross-entropy loss function and model would evaluate with accuracy metrics.

## Training

In training, pass the training dataset to the model, and evaluate with validation dataset. We train the model with 25 epochs as we discussed. Save the model to make it reusable.

# Unsupervised Learning – GAN data augmentation

To improve the performance of deep learning models for dog emotions, we used GAN data augmentation to generate similar new data. A GAN works by training a generator and a discriminator neural network. The generator creates fake data by using random noise, and the discriminator determines whether it is real or fake. Using the original dataset of dog images as a training set, the GAN can generate images of dogs with different emotions. With the new images generated, the original dataset can be expanded and used to train a deep learning model, ultimately increasing the size of the training dataset and improving its performance.

## Parameters

As parameter we will use batch size of 256, buffer size with 3000, and 50 epochs for training the model.

The model is built using TensorFlow keras library.

### Modeling

For the generator model we define the following architecture:

The first layer is a dense layer with 8x8x256=16,384 neurons, with no bias term and a specified input shape of 100. This layer acts as a mapping from the random noise input to a higher dimensional space. The next layer is a batch normalization layer that normalizes the outputs from the previous layer, reducing the internal covariate shift and stabilizing the learning process. The following layer is a leaky ReLU activation layer, which introduces a small negative slope for negative inputs, preventing the dying ReLU problem and allowing for a more expressive model. The next layer is a reshape layer that transforms the output of the previous layer into a 4D tensor of shape (batch\_size, 8, 8, 256). Next, there are three transposed convolutional layers, each with a batch normalization layer and a leaky ReLU activation layer. These layers learn to upsample the low-dimensional feature representations learned by the earlier layers into higher resolution image representations. The first transposed convolutional layer has 128 filters, a kernel size of 5x5, and a stride of 1x1. The second transposed convolutional layer has 64 filters, a kernel size of 5x5, and a stride of 2x2. The third transposed convolutional layer has 3 filters, a kernel size of 5x5, a stride of 2x2, and uses a hyperbolic tangent activation function to produce a pixel intensity value between -1 and 1.

On the other hand, for the discriminator model we have the following layers:

The first layer is a convolutional layer with 64 filters, a kernel size of 5x5, a stride of 2x2, and padding to ensure the output size is the same as the input size. The input shape is specified to match the size of the images that the discriminator will classify. The next layer is a leaky ReLU activation layer, which introduces a small negative slope for negative inputs, preventing the dying ReLU problem and allowing for a more expressive model. The following layer is a dropout layer, which randomly drops out 30% of the outputs from the previous layer during training, helping to prevent overfitting. The next layer is another convolutional layer with 128 filters, a kernel size of 5x5, a stride of 2x2, and padding. This layer learns to extract higher level features from the lower level feature maps learned by the earlier layers. The next layer is again a leaky ReLU activation layer, followed by another dropout layer with a 30% dropout rate. The following layer is a flatten layer that flattens the 2D feature maps from the previous layer into a 1D vector. The final layer is a dense layer with a single output neuron. This layer computes a scalar output that represents the probability that the input image is real (as opposed to generated by the generator network). The output does not have an activation function applied to it, so the output range is not constrained.

## Training

The training function is a loop that trains a GAN model by generating fake images from random noise, feeding both real and fake images to the discriminator, computing the loss for both the generator and discriminator, and updating the trainable variables of each network using the respective optimizer. This process is repeated for a specified number of epochs, after which the model is expected to be able to generate realistic images from random noise.

The training happens taking a batch of real images as input and performing the following steps:

* Generate random noise vectors which are used as input for the generator network.
* Use the generator network to generate fake images from the random noise.
* Pass both the real and fake images through the discriminator network to get the discriminator's prediction for each image.
* Calculate the loss for both the generator and discriminator. The generator loss is calculated based on how well the discriminator was fooled by the generated images, while the discriminator loss is calculated based on how well it was able to distinguish between real and fake images.
* Calculate the gradients of the loss with respect to the trainable variables of each network.
* Use the calculated gradients to update the trainable variables of each network using the respective optimizer.

# State-of-the-Art Model

According to Kurama (2021), InceptionV3 is the most computationally efficient model to be trained with an image

classifier. Since we lack our powerful resources, the amount of computation we have to do was our primary concern. With addition of research by ourselves, we decided to use InceptionV3 with same number of batch size and number of epochs as our prior classification model.

## Parameters

As parameter we will use same batch size and number of epochs with Supervised Classifier model for training the model: 32 and 25 accordingly. The model is built using TensorFlow keras library.

### Modeling

The first layer is constructed with InceptionV3 pre-trained model with its output. The weights parameter for InceptionV3, use “imagenet” since it is image classification tasks and disable include\_top parameter. The second layer comes with Global Average Pooling 2D layer to pool the output from the InceptionV3. For the third layer, add Dense layer with 1024 units and ReLU activation function. The final layer is added to get the prediction of four classes using Dense layer with 4 units and SoftMax activation function. Freeze the weights for InceptionV3 layers. Lastly finalize the model with keras Model function with inputs as input of InceptionV3, and outputs as last prediction layer. The model is compiled with Adam optimizer with learning rate of 0.001 and Categorical Cross-entropy as a loss function, and record with “accuracy” metrics.

### Training

In training, pass the training dataset to the model, and evaluate with validation dataset. We train the model with 25 epochs with batch size of 32 as we discussed. Save the model to make it reusable.

## Evaluation

## CNN

Calendar

Description automatically generated

Chart, treemap chart

Description automatically generated Chart, line chart

Description automatically generated

From the above scores and matrix heatmap, the model prediction has high reliability, but computation required huge resources.

## GAN

A collage of dogs and people

Description automatically generated with low confidenceA picture containing window

Description automatically generated

The left images plot shows the sample images used to train the model and the right images plot shows the generated image after 50 epochs. As shown, the GAN are not matured to generate the image with 50 epochs, and we lacked our resources to train further more.

InceptionV3

A picture containing text, calculator, black

Description automatically generated

Chart, treemap chart

Description automatically generated Chart, line chart

Description automatically generated

From the above scores and matrix heatmap, the model prediction has reached to it best state in early epochs compared to prior classification model.

## Summary

Our supervised model has learned well about the training dataset, and the model quality is satisfactory, but the computation time is long. On the other hand, State-of-Art model have balanced with its quality and computation resources. However, our State-of-Art model fell into overfitting problem. Overall, our supervised model would be picked compared to the later model.

As of Unsupervised model, we have lacked resources to train fully capable image generator model, so it is required for larger resources to accomplish the goal

### Contributions and Credentials

Aiju Kinoshtia: Overall code management, technical document writing, and presentation preparation.

Adrian Posadas: Supervised model and InceptionV3 model development, and technical document writing

Jestoni Vasquez: Overall code management, technical document writing, and presentation preparation.

Oscar Quispe Guanoluisa: Unsupervised model development and technical document writing.

References

Balico, D. S. (2023, February 9). Dog Emotion - Dog Emotion Image Classification. Retrieved from kaggle: https://www.kaggle.com/datasets/danielshanbalico/dog-emotion

Kurama, V. (2021, April 9). A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet. Retrieved from Paperspace: https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/