Laguna State Polytechnic University
Republic of the Philippines
Province of Laguna
Los Baños

Cat and Dog Image Classification

A Final Project Presented to the Faculty
of Computer Studies
of Laguna State Polytechnic University

In partial fulfillment of the requirements in
Advance Machine Learning
Bachelor of Science in Computer Science-III A

By:
Gupo, Jhon Vincent A.

July 2022

INTRODUCTION

As we live in an era where not just numbers but also pictures and videos can be seen as an increasing data through the records of CCTV and IP Camera, as well as taking photos that can be shared online, especially on social media, we cannot deny that the field of machine learning can also be used here which talks about especially computer vision.

Boesch, G. (2022) stated that the primary use cases in computer vision include object recognition, segmentation, classification, and localization of images. Among these, picture categorization might be thought of as the core issue. It serves as the foundation for more computer vision issues. Applications for image classification are employed in a variety of fields, including medical imaging, satellite image object identification, traffic control systems, brake light detection, machine vision, and more. So, for our final project in Advance Machine Learning, we used cat and dog images that can be found in Kaggle to talk about Convolutional Neural networks which is a special type of Neural Network, particularly for image classification.

The dataset contains around 25,000 images of cat and dog. Sample batch of the image can be seen below. 0 corresponds to cat while 1 corresponds to dog.

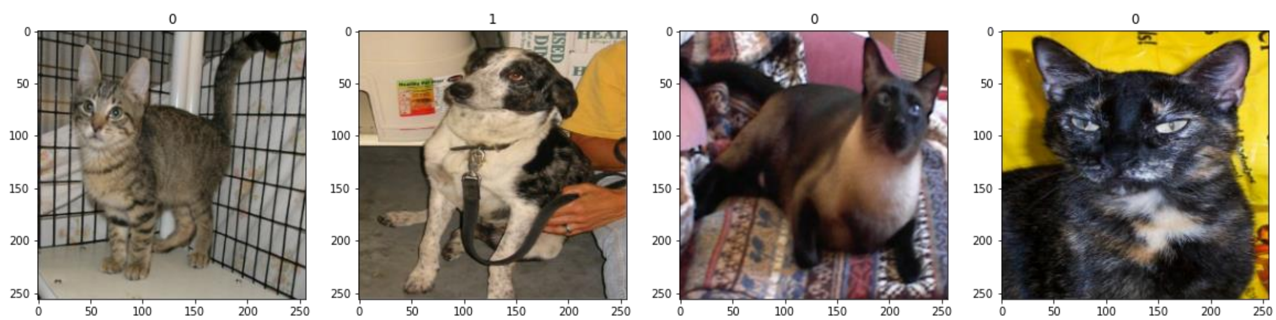


Figure 1. Dataset batch of images

METHODOLOGY

The project pipeline was divided into four components: Data Processing, Data Splitting, Model Training, and Model Evaluation.

First, the data are preprocessed, such as removing images that are not in .jpeg or .jpg since file types other than that cannot be fed into the model. The images were then resized into 256 width x 256 height since the model requires data with the same data shape training, and then each was converted into a NumPy array since only numbers can be fed into the model. After that, the numbers are normalized by dividing each number into 255, which is the range of RGB values in an image. The second stage is splitting the data with 80:20 conversion for training and testing data. Three different Convolutional Neural Network configuration was used to classify the image, which is the plain Convolutional Neural Network for the 1st model. Image Augmentation + Convolutional Neural Network for the 2nd model and lastly, the Mobile Net V2, which is a pre-trained Convolutional Neural Network trained to classify thousands of images, such as a mouse, pencil, keyboard, etc. which is built by Mark Sandler et al. and is available to use at TensorFlow hub. You can learn more about the Mobile Net V2 architecture here. Lastly, the models are evaluated with the Accuracy compared to the Validation Accuracy per epoch and the model's Loss compared to their validation loss. You can refer to figure 2 for the project pipeline visualization.

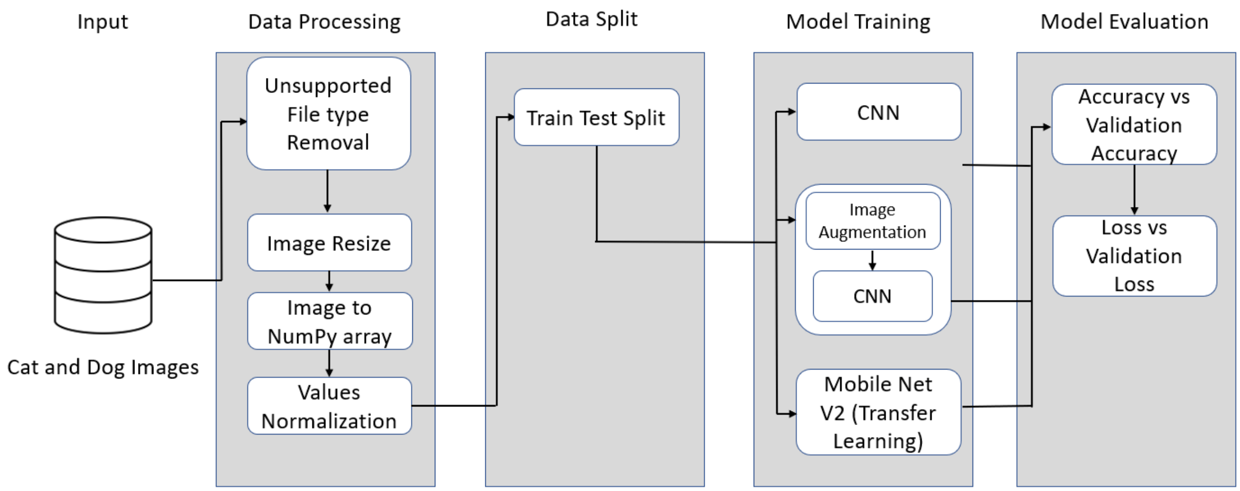


Figure 2. Project Pipeline

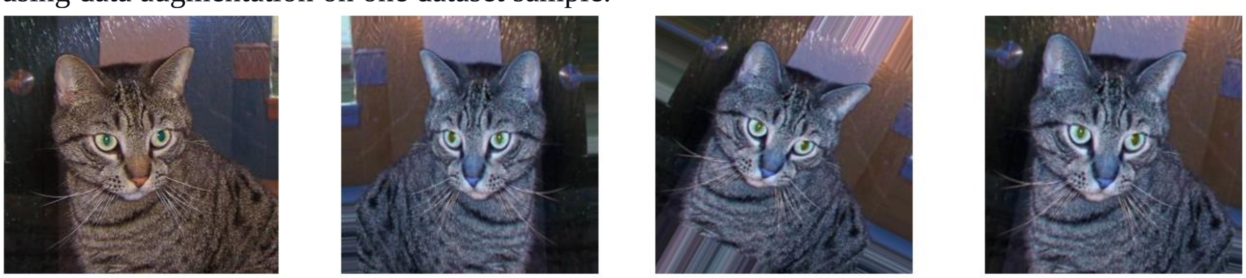
The configuration of the Convolutional Neural Network (model 1) consists of 3 conv2 layers with two max-pooling 2d in between them to get the maximum element from a particular region of the feature mapped by the max pooling. The first conv2 layer consists of 32 filters, while the 2nd and 3rd conv2 layers consist of 64. After the final convolutional layer, flatten was used to convert all the resultant 2D arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image, which consists of 64 neurons that takes the features that were obtained and, lastly, are fed to the last dense layer with one neuron that classifies less than 0.5 as a cat while more than 0.5 as a dog. All conv2d layers has a rectified linear activation unit for the activation function and SoftMax for the last dense layer. You can refer at figure 3 at the next page for model 1 summary.

The compiler configuration has Adam for the optimizer while binary cross-entropy for the loss function and is the same all throughout three models.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 32)      896

 max_pooling2d (MaxPooling2D  (None, 127, 127, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 125, 125, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 62, 62, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 64)        36928

 flatten (Flatten)           (None, 230400)            0

 dense (Dense)               (None, 64)                14745664

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 14,802,049
Trainable params: 14,802,049
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. Model 1 summary

The configuration of model 2 is the same as model 1, with only data augmentation as a difference in the input layer. Image augmentation is the process of artificially expanding the available dataset for training a deep learning model to address overfitting. Techniques such as image rotation, image zoom, image flipping, image noising, and image blurring can be applied to make a synthetic image from the original data. Image flip, image rotation, and image zoom were used in model 2 to expand the data artificially. Figure 4 illustrates the synthetic images generated using data augmentation on one dataset sample.



Original | Image Flipped | Image Rotation | Image Zoom

Figure 4. Image Augmentation Example

For model 3, Mobile Net V2 was used, a pre-trained model that is retrained to classify cat and dog images with the technique called transfer learning was used. Transfer learning is a method that uses a pre-trained model as the new base layer for a new task. In simple terms, it is reusing an already trained model on a large amount of data into a new problem, such as a model that can classify a car by repurposing it into another problem, such as classifying a truck. Figure 5 shows the visualization of how transfer learning works theoretically.

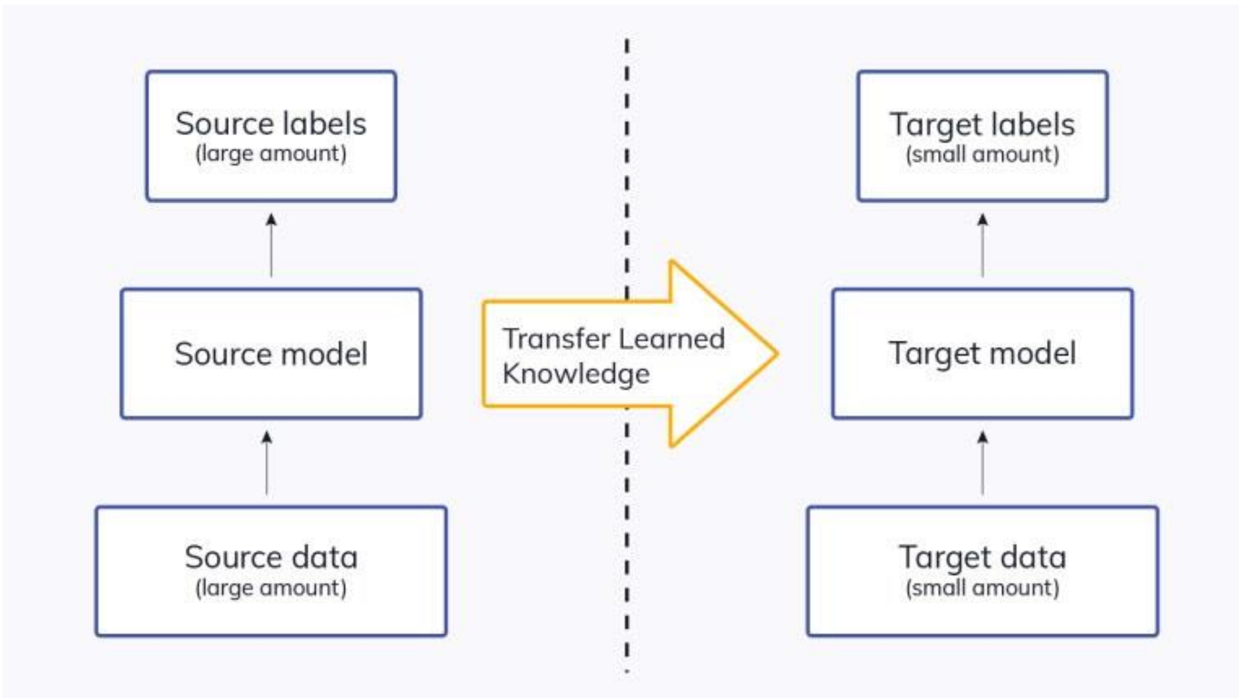


Figure 5. Transfer Learning
Image source: www.v7labs.com

Model 3 just use the pre train model with a dense layer at the end with 1 neuron for predicting cat or dog. You can refer at figure 6 at the next page for model 3 summary.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer_1 (KerasLayer)  (None, 1280)              2257984

 dense_5 (Dense)             (None, 1)                 1281

=================================================================
Total params: 2,259,265
Trainable params: 1,281
Non-trainable params: 2,257,984
_________________________________________________________________
```

Figure 6. Model 3 summary

RESULTS AND DISCUSSION

Training time varies on the epoch inputted in the model, which can be seen in table 1. The higher the epoch, the longer the training time it takes for the model.

Models	Epoch	Training Time
Model 1	10	30 mins
Model 2	15	45 mins
Model 3	5	15 mins

Table 1. Models Epoch per training time

Models' accuracy and results will be discussed as follows.

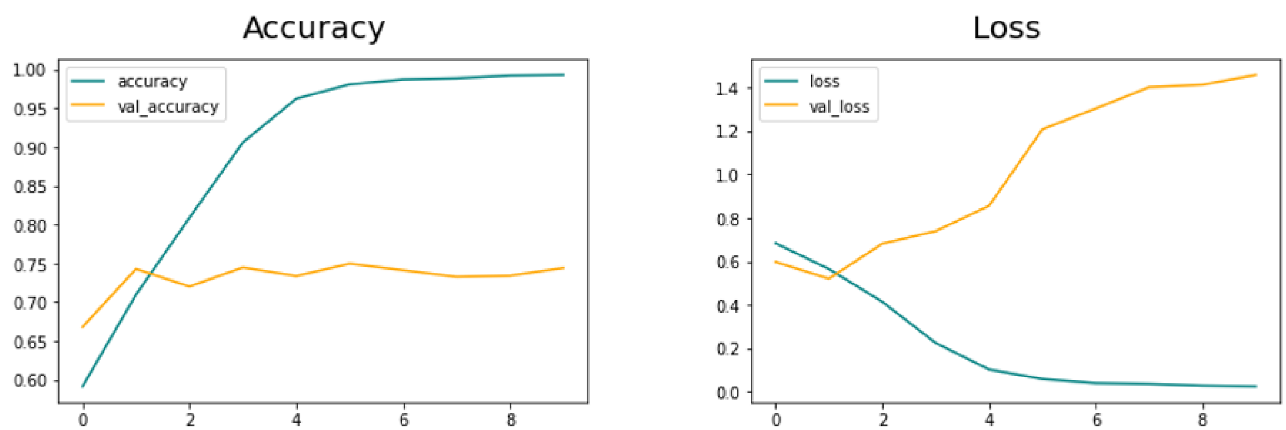


Figure 7. Model 1 accuracy and loss

Even though the accuracy is almost perfect, there is a huge gap between the validation of both loss and accuracy. This is a great sign of overfitting

There are three ways to fix overfitting which are:

- Data Augmentation - Increase the amount of the data, collect new data or derive new data from existing data / data augmentation.
- Dropout - Reduce the complex of the model. For example, reduce the hidden layers or number of units in the hidden layer.
- Regularization - Put constraints in the coefficients/parameters. For example, L1 or L2 regularization.

Data augmentation is the technique that will be used for the next model.

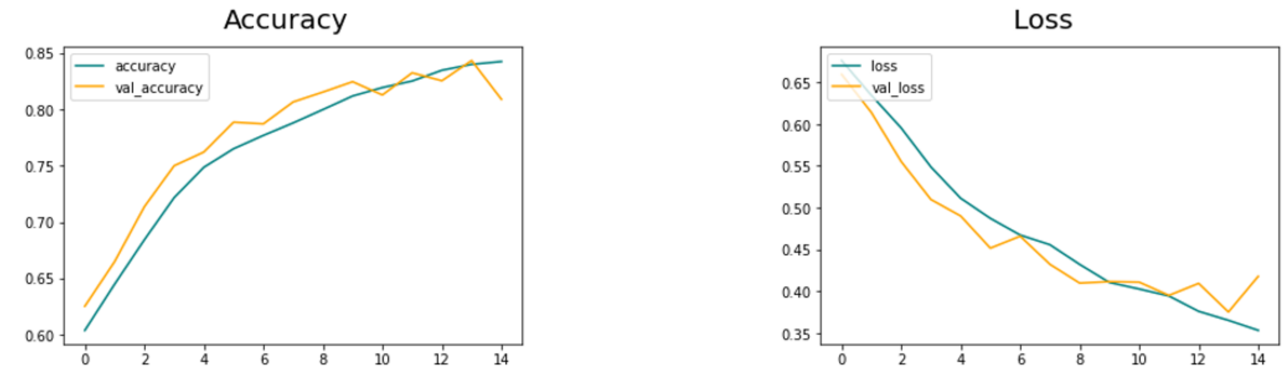


Figure 8. Model 2 accuracy and loss

Image Augmentation solves the overfitting problem from Model 1 but the accuracy is still not that high (which is around 83%). The next model that we will try is Mobile Net V2 which utilize the technique of transfer learning.

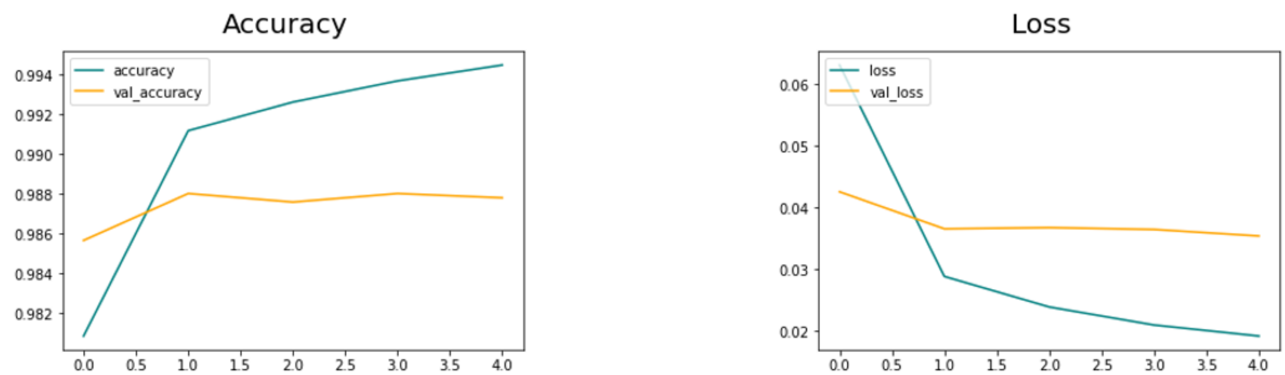


Figure 9. Model 3 accuracy and loss

The 3rd model has almost perfect accuracy with lesser training time compared to the two previous model. It also doesn't overfit compared to model 1.

For the next test, we will test all model's predictions as well as their confidence in their prediction on these ten images downloaded from the internet. Correct predictions are marked as red.

Images	**Prediction and Score Confidence**		
	Model 1 (CNN)	**Model 2 (Image Augmentation + CNN)**	**Model 3 (Mobile Net V2 – Transfer Learning)**
	Cat (0.98)	Dog (0.57)	Dog (0.99)
	Dog (0.99)	Dog (0.79)	Cat (0.99)
	Dog (0.98)	Dog (0.85)	Dog (0.99)

<table>
<tr>
<td>
</td>
<td>Cat
(0.99)</td>
<td>Dog
(0.55)</td>
<td>Dog
(0.99)</td>
</tr>
<tr>
<td>
</td>
<td>Cat
(0.79)</td>
<td>Cat
(0.92)</td>
<td>Dog
(0.99)</td>
</tr>
<tr>
<td></td>
<td>Cat
(0.92)</td>
<td>Cat
(0.71)</td>
<td>Cat
(0.99)</td>
</tr>
</table>

	Dog (0.99)	Dog (0.60)	Cat (0.98)
	Cat (0.99)	Cat (0.74)	Cat (0.99)
	Dog (0.99)	Dog (0.95)	Dog (0.99)

	Dog (1)	Dog (0.94)	Cat (0.63)

Table 2. Test images model prediction

Model 1 has predicted 4/10 of the test data while model 2 predicted 6/10 and model 3 predicted 10/10 of the test data.

CONCLUSION

Based on this study, the use of transfer learning (model 3) is the most effective method in dealing with image classification problems both with accuracy and with the time spent training the model. Model 3 is also seen to be good at predicting unseen data and other types of cat and dog images (such as the animated cat and dog in examples 9 and 10).

RECOMMENDATION

It is recommended to have the same research be done in the future that checks the following:

Since model 3 is already excellent at predicting cat and dog images, no tweaking is suggested to fine-tune the model. You may try other instances of data and check model 3 predictions such as:

- Images with multiple dogs and a cat in a picture (or vice versa). An example is three dogs and one cat in the same picture or three cats and one dog in the same picture.
- A picture that has noise/ blurred or has very low resolution.

It is also time to step up the task of classifying the images. You can try testing models 2 and 3 in predicting the breed of cat and dog by retraining them with images of cat and dog breeds.

REFERENCE

Boesch, G. (2022). A Complete Guide to Image Classification in 2022 available at https://viso.ai/computer-vision/image-classification/#:~:text=Image%20Classification%20is%20the%20Basis%20of%20Computer%20Vision,-The%20field%20of&text=It%20forms%20the%20basis%20for,%2C%20machine%20vision%2C%20and%20more.

Bandyopadhyay, H. (2022, June 20). Image Classification Explained available at https://www.v7labs.com/blog/image-classification-guide

Sandler, M. et al. (2018, January 13). MobileNetV2: Inverted Residuals and Linear Bottlenecks available at https://arxiv.org/pdf/1801.04381.pdf

Google Developers (n.d.). Convolutional Neural Networks through TensorFlow available at https://developers.google.com/codelabs/tensorflow-3-convolutions#0

Saxena, S. (2021, March 10). Image Augmentation Techniques for Training Deep Learning Models available at https://www.analyticsvidhya.com/blog/2021/03/image-augmentation-techniques-for-training-deep-learning-models/

Baheti, P. (2022, June 22). A Newbie-Friendly Guide to Transfer Learning available at https://www.v7labs.com/blog/transfer-learning-guide