KULLIYYAH OF INFORMATION AND COMMUNICATION TECHNOLOGY

CSCI 4340 MACHINE LEARNING

SEMESTER 2, 2021/2022

SECTION 1

PROJECT TITLE:

A STUDY ON IMAGE CLASSIFICATION USING Convolutional Neural Network (CNN)

PREPARED BY:

Names	Matric No.
1: Amnah salah majzob abdel maged	1824962
2: Emon Rifat Hasan	1832901
3: HIBO SULEIMAN AMEN	1825120

LECTURER: DR. AMELIA RITAHANI

DUE DATE: 24/6/2022

Table of content

Abstract

Generally,as human beings we are experts at classifying cats and dogs. Cats have perky ears while dogs generally have floppy ears. Cats have small snouts while dogs have elongated snouts. Cats have round faces while dogs are rather oval. Although the image doesn't show it, cats are generally smaller and fluffier than dogs.One might think that there is sufficient difference between cats and dogs that it should be easy to tell them apart. For us it is; for a computer, not quite so easy.In this project we will try to design and build models that would effectively classify images of cats and dogs.We will model Conventional Neural Network.In this project ,we are using a dataset containing images of cats and dogs.

1.0 INTRODUCTION

As children, we learn how to identify different kinds of animals. Our brain stores and sorts out information we receive from various sources (i.e. parents, television, books, etc.) about the distinctive features belonging to such creatures. So when we look at different animals, we are able to tell the difference between, say, a cat and a dog. That means there must be some uniquely inherent features possessed in each animal.Our goal is to create a classification algorithm that trains the computer to extract these distinc- tive features and correctly classify animals, namely cats and dogs.In Machine Learning,classification is a supervised learning concept which basically categorizes a set of data into classes.This project we will try to shed some light on how to create a system that can recognize images of cats and dogs. After analyzing the input image, the result will be predicted.To obtain reliable results, we will model Conventional Neural Network machine learning algorithms. The Dogs vs Cats dataset can be downloaded from the Kaggle website. The dataset contains a set of images of cats and dogs.

1.1 Project Objectives

- To develop a working machine learning model that can distinguish cats and dogs given images by using CNN.
- To test the accuracy of picture categorization and the performance of algorithms models.
- To predict the possibility between Cats and Dogs in each model Process.
- To evaluate the performance of CNN algorithms.

1.2 Model Objectives

- Build and tune a convolutional network with keras for image classification
- To build and train a conventional neural network for classifying images of Cats and Dogs.
- Use the keras ImageDataGenerator to augment your dataset and limit overfitting.

1.3 Expected Outcome

We will be able to categorize and show the data according to its category using the dataset we have acquired. The model's primary goal is to understand many distinguishing characteristics of cats and dogs. After the model has been trained, it will be able to distinguish between cat and dog images. The accuracy of picture categorization as well as the performance of the CNN will be evaluated.

2.0 LITERATURE REVIEW

2.1 Introduction to image classification

In this modern era,building your computer vision model is a sophisticated process that involves several steps, a high-level engineering team, and hundreds to thousands of images. Your model must be trained to identify these images through a process known as image classification (or categorization as we refer to it in the Superb AI suite), which uses an advanced algorithm to assign a label or a tag to identify each individual image. Image classification works by utilizing pre-existing datasets to train your model. Through this process, your model is studying each image at the pixel level, meaning that it is analyzing this information to determine the correct label for your image. As part of the bigger picture, image classification is used to teach your computer vision model patterns and behaviors in the real world. Through careful training, your model can achieve high levels of accuracy before being used for practical applications.Convolutional Neural Network (CNN)The goal of this project is to offer a free predictive model that will enable computers to learn whether a given image is of a cat or a dog.

2.2 Causes and implications of Image classification algorithm in machine learning

Classification between objects is a fairly easy task for us, but it has proved to be a complex one for machines and therefore image classification has been an important task within the field of computer vision.Image classification refers to the labeling of images into one of a number of predefined classes.There are potentially *n* number of classes in which a given image can be classified. Manually checking and classifying images could be a tedious task especially when they are massive in number (say 10,000) and therefore it will be very useful if we could automate this entire process using computer vision.

Although the functionality covered by this project may only be fairly basic, we think the features and functionalities may be scaled to meet the needs of picture categorization. The CNN algorithms could be used as technical support for identifying images in large-scale factory machinery. It could help with the energy saving required to avoid conducting unneeded image combustion. Additionally, we expect that through technology application of image prediction, more people may adapt sustainable consumption and production habits into facilitators for future jobs that require classifying between photos.

2.3 Machine learning used

Since our goal is to create a machine learning model that could help assist in image classification, a classification algorithm will be used as it enables us to classify given dataset into a set of categories, which in this case, the classification algorithm will identify whether a particular given image is Cat or Dog that are identifiable . To fulfill our first project objective, we decided to use advanced computer vision with a deep learning technique called CNN. CNN is a classification algorithm hence it falls under the category supervised learning algorithm.

2.3.1 Conventional Neural Network

CNN is one of the algorithms that has been created to help researchers resolve categorization difficulties. CNN is highly desired since it allows us to use pictures as input thereby getting artificial intelligence technology closer to approximating human abilities to understand their surroundings through a different sense, sight. The only inputs that its previous algorithms could accept were words and numbers. CNN operates similarly to how people utilize their brain neurons to recognize an item, but it uses artificial neurons instead of natural ones. In order for the computer to distinguish between each pixel of an image and calculate the necessary output, CNN would run an image through layers of processing and compression.

Moreover,this algorithm takes an image as input then assigns weights and biases to all the aspects of an image and thus differentiates one from the other. Neural networks can be trained by using batches of images, each of them having a label to identify the real nature of the image (cat or dog here). A batch can contain a few tenths to hundreds of images. For each and every image, the network prediction is compared with the corresponding existing label, and the distance between network prediction and the truth is evaluated for the whole batch. Then, the network parameters are modified to minimize the distance and thus the prediction capability of the network is increased. The training process continues for every batch similarly.

3.0 EXPERIMENT SETUP

3.1 Dataset Description:

The dataset that was used is Dog vs Cat (fastai) by Aprit Jain in 2019 from Kaggle. This dataset was created to train the machine to acknowledge the pet from an image, whether a cat or a dog. It consists of 25,000 in 2 files one for the cats and has 12,500 cat pictures and the second one for the dogs with 12,500 dog pictures. Manipulating the data is needed because it's in folders by using a pre-processing image function from Keras and then splitting the data into a training set and a testing set.

No.	Name	Description
1.	Image	Size 24-bit RGB (JPG File)
2.	Type	0=cats, 1=dogs

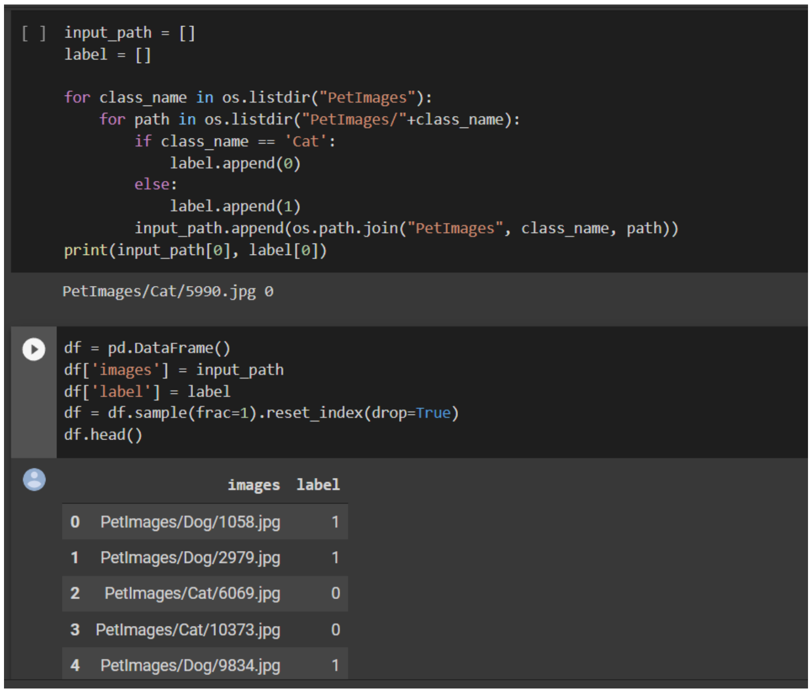
Table 1: Dataset Description

3.2 Data Pre-processing

This data set was obtained from Kaggle in the form of images in folders. A function called ImageDataGenerator, which is an image pre-processing function from Keras, was applied to prepare the data set.

3.2.1 Data Cleaning

The images are defined as 2 types only. We labeled them into 0 for cats and 1 for dogs. Then we checked the error files to delete them from the dataset.



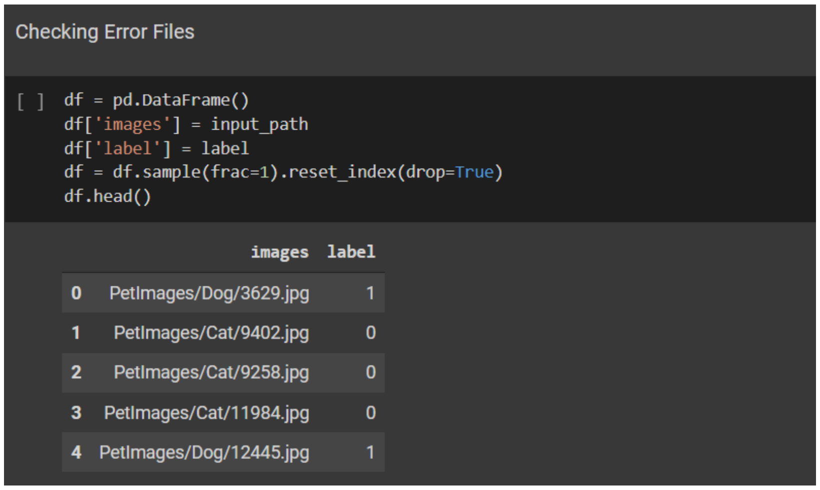
```
[ ] input_path = []
    label = []

    for class_name in os.listdir("PetImages"):
        for path in os.listdir("PetImages/"+class_name):
            if class_name == 'Cat':
                label.append(0)
            else:
                label.append(1)
            input_path.append(os.path.join("PetImages", class_name, path))
    print(input_path[0], label[0])
```

```
PetImages/Cat/5990.jpg 0
```

```
df = pd.DataFrame()
df['images'] = input_path
df['label'] = label
df = df.sample(frac=1).reset_index(drop=True)
df.head()
```

	images	label
0	PetImages/Dog/1058.jpg	1
1	PetImages/Dog/2979.jpg	1
2	PetImages/Cat/6069.jpg	0
3	PetImages/Cat/10373.jpg	0
4	PetImages/Dog/9834.jpg	1



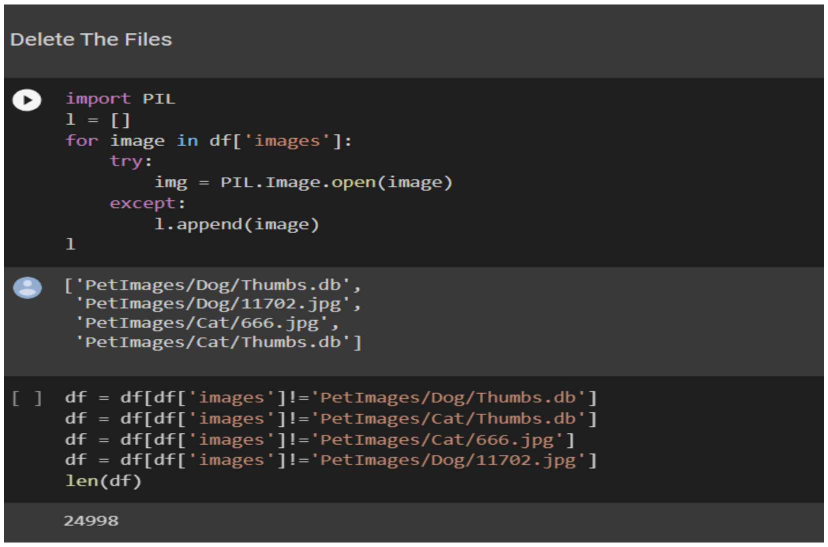
Checking Error Files

```
[ ] df = pd.DataFrame()
    df['images'] = input_path
    df['label'] = label
    df = df.sample(frac=1).reset_index(drop=True)
    df.head()
```

	images	label
0	PetImages/Dog/3629.jpg	1
1	PetImages/Cat/9402.jpg	0
2	PetImages/Cat/9258.jpg	0
3	PetImages/Cat/11984.jpg	0
4	PetImages/Dog/12445.jpg	1

3.2.2 Normalization

To conduct normalization, the values will be adjusted by eliminating the error values. It will make it easier when comparing the different types of data.



Delete The Files

```
import PIL
l = []
for image in df['images']:
    try:
        img = PIL.Image.open(image)
    except:
        l.append(image)
l
```

```
['PetImages/Dog/Thumbs.db',
 'PetImages/Dog/11702.jpg',
 'PetImages/Cat/666.jpg',
 'PetImages/Cat/Thumbs.db']
```

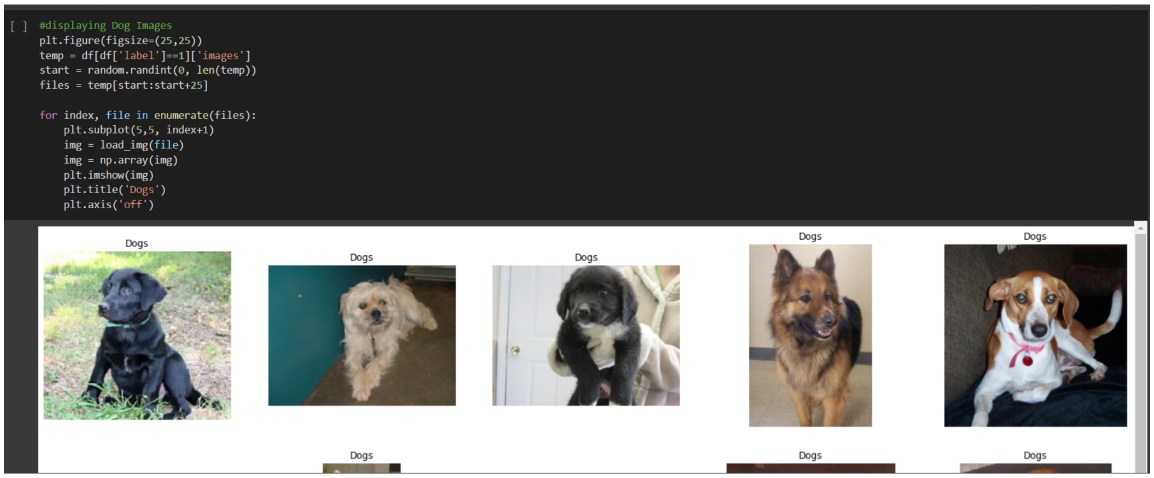
```
df = df[df['images']!='PetImages/Dog/Thumbs.db']
df = df[df['images']!='PetImages/Cat/Thumbs.db']
df = df[df['images']!='PetImages/Cat/666.jpg']
df = df[df['images']!='PetImages/Dog/11702.jpg']
len(df)
```

```
24998
```

As we mentioned in 3.1 the total number of images is 25,000 but after deleting the error it became 24,998.

3.3 Data Visualization

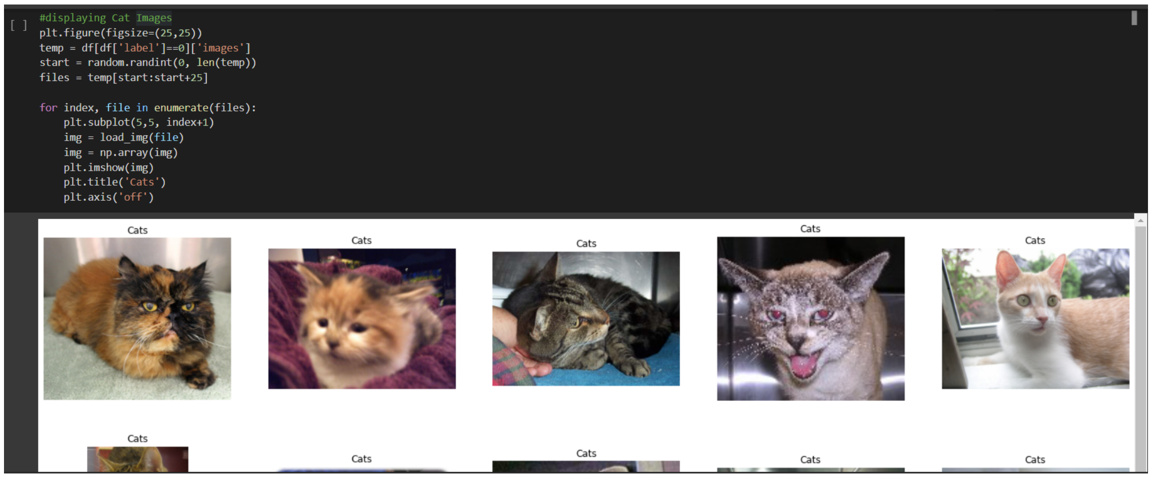
Because the datasets are in an image form it is considered large. It needs to be changed into a meaningful visual graphical representation like graphs to give better understanding of the data. In this project, there are only two types of datasets: dogs and cats. visualizing examples of both types helped to understand the differences in them for the machine. Below is the code we used to display the dogs and cats pictures with the label of them.



```
#displaying Dog Images
plt.figure(figsize=(25,25))
temp = df[df['label']==1]['images']
start = random.randint(0, len(temp))
files = temp[start:start+25]

for index, file in enumerate(files):
    plt.subplot(5,5, index+1)
    img = load_img(file)
    img = np.array(img)
    plt.imshow(img)
    plt.title('Dogs')
    plt.axis('off')
```

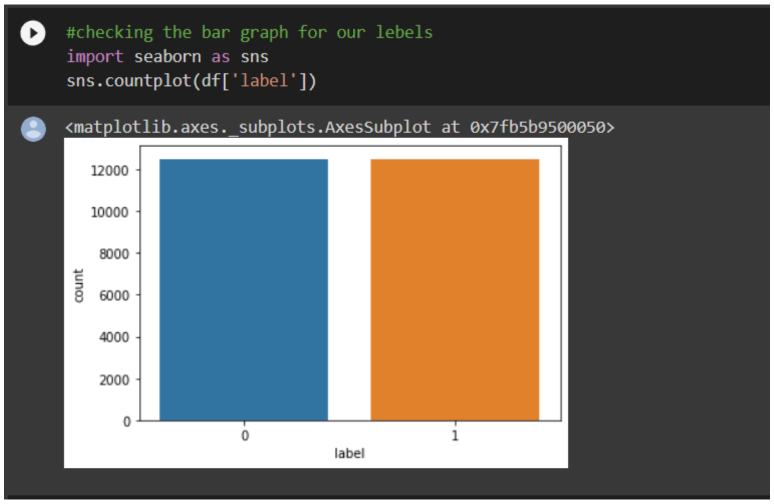
The code we used to display the dogs and cats pictures with the label of them.



```
#displaying Cat Images
plt.figure(figsize=(25,25))
temp = df[df['label']==0]['images']
start = random.randint(0, len(temp))
files = temp[start:start+25]

for index, file in enumerate(files):
    plt.subplot(5,5, index+1)
    img = load_img(file)
    img = np.array(img)
    plt.imshow(img)
    plt.title('Cats')
    plt.axis('off')
```

After changing the form of the dataset we represented it in this bar graph:

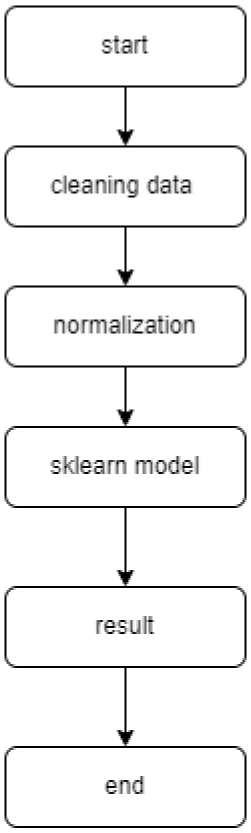


```
#checking the bar graph for our lebels
import seaborn as sns
sns.countplot(df['label'])
```

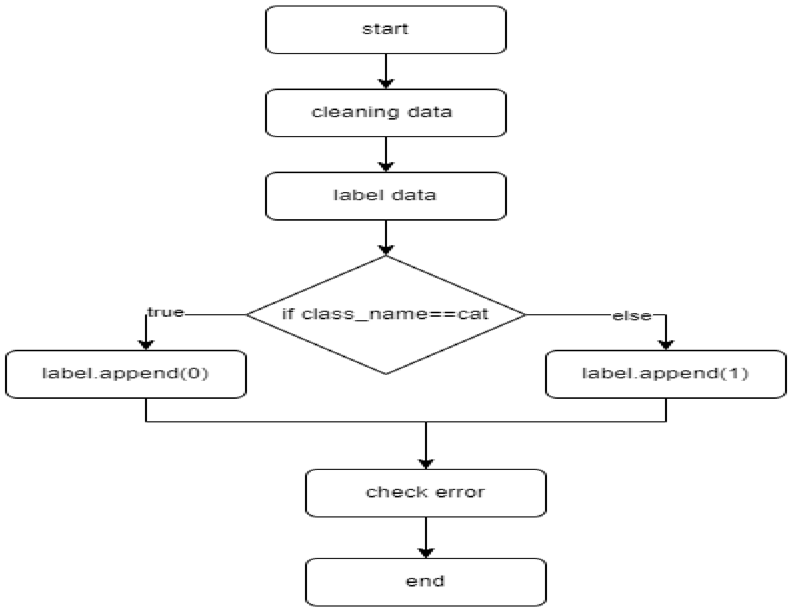
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb5b9500050>
```

This bar graph shows that both labels (1=dogs & 0=cats) have the same count number.

3.4 Model Development



Flowchart for model development

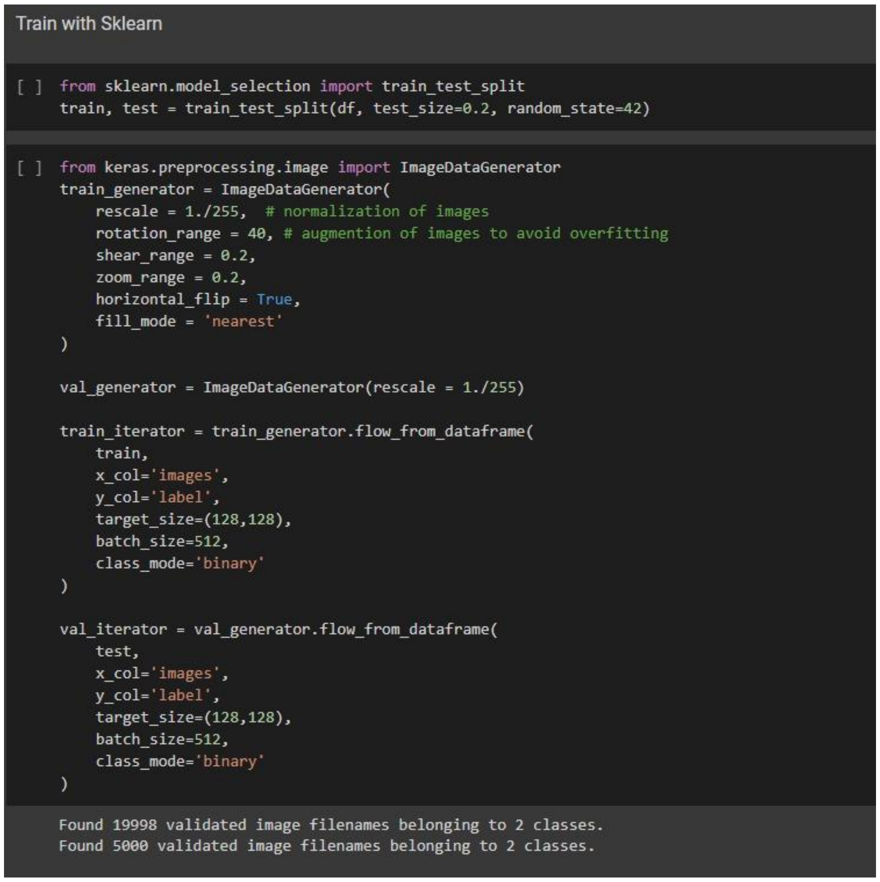


Flowchart for clearing data

4.0 MODELING

4.1 Image generator

We choose to utilize a generator object from the ImageDataGenerator package to deal with images as our input data. This step includes the normalization and scaling of several picture pre-processing tasks. Since 128 seems to be the typical length and width, we scaled each image to be 128 by 128 pixels. Three generators—one each for training, testing, and validation—are available. Each iteration in an epoch has a batch size of 40 set to 512. Given that a large batch size of input to CNN could interfere with its learning process, this is a good batch size choice. Additionally, when batch size is set to greater than 512, our virtual machine crashes every time the training code is executed.


Train with Sklearn

```
[ ]  from sklearn.model_selection import train_test_split
     train, test = train_test_split(df, test_size=0.2, random_state=42)
```

```
[ ]  from keras.preprocessing.image import ImageDataGenerator
     train_generator = ImageDataGenerator(
         rescale = 1./255,  # normalization of images
         rotation_range = 40, # augmention of images to avoid overfitting
         shear_range = 0.2,
         zoom_range = 0.2,
         horizontal_flip = True,
         fill_mode = 'nearest'
     )

     val_generator = ImageDataGenerator(rescale = 1./255)

     train_iterator = train_generator.flow_from_dataframe(
         train,
         x_col='images',
         y_col='label',
         target_size=(128,128),
         batch_size=512,
         class_mode='binary'
     )

     val_iterator = val_generator.flow_from_dataframe(
         test,
         x_col='images',
         y_col='label',
         target_size=(128,128),
         batch_size=512,
         class_mode='binary'
     )
```

```
     Found 19998 validated image filenames belonging to 2 classes.
     Found 5000 validated image filenames belonging to 2 classes.
```

Image : Full Code of To Produce Image Generator

4.2 CNN

In this section, we will be discussing the CNN overall model and its parameters. The model contains 3 convolution layers with ReLU as its activation function. The middle convolution layer has the highest number of filters which is 512 compared to the other 1 layer which is 256 filters. After every convolution layer, a pooling layer is implemented to minimize the mapping of feature extraction. Lastly, a process of flattening the 2D-matrix is done. The flattened 1D-array is connected to a fully dense neural network which is set up to perform classification of the label.

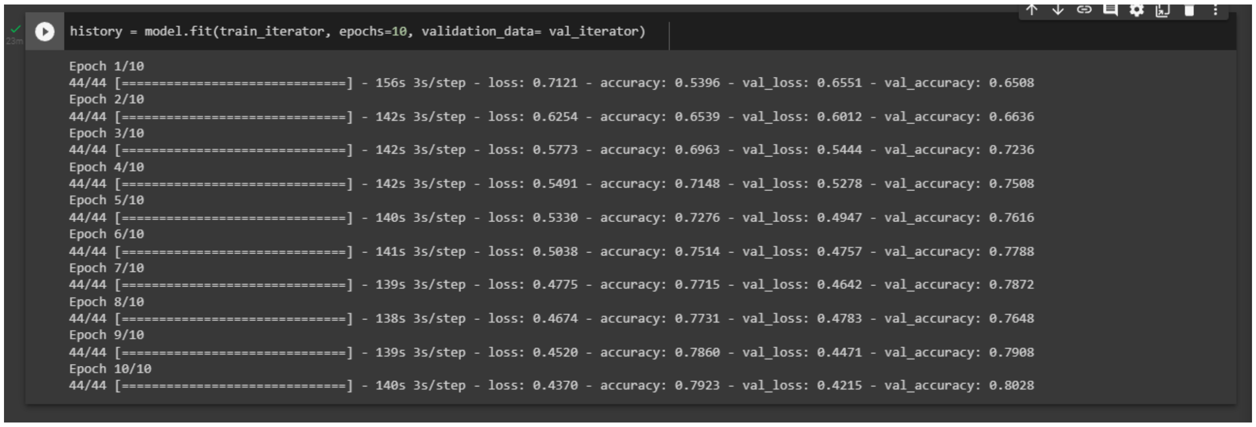
In detail, in every convolution layer, we set the Conv2D size to be a matrix of 3X3. This is due to the fact that the 2X2 Conv2D is arguably small for our input image size of 128X128 pixel image. It may not be able to acquire the needed features from images. For the activation function, we used ReLU to speed up the learning process of the model. It is a nonlinear function which converts every negative value in the kernel to zero. This is possible since a negative value indicates non-pattern in filters.

Next, pooling is a layer to minimize the computation of the neural network. It will result in a faster learning process. It will decrease the size to a defined pooling size. We decided to use the max pooling approach with a pool of size 2x2 matrix. It will stride over the Conv2D matrix by its own size. It is a suitable pool size and stride since we have the Conv2D size of 3x3 matrix. No padding is needed.

Lastly, the model is compiled with adam optimizer. Metrics of accuracy and loss of binary cross-entropy is tracked in every epoch done.

4.3 Model performance

We set the epochs at 10. Because our data is large, so it needs time to train. At the last epoch we get a maximum accuracy of 0.8028. From here we can see our model is performing very well. Its near 80% of the accuracy.



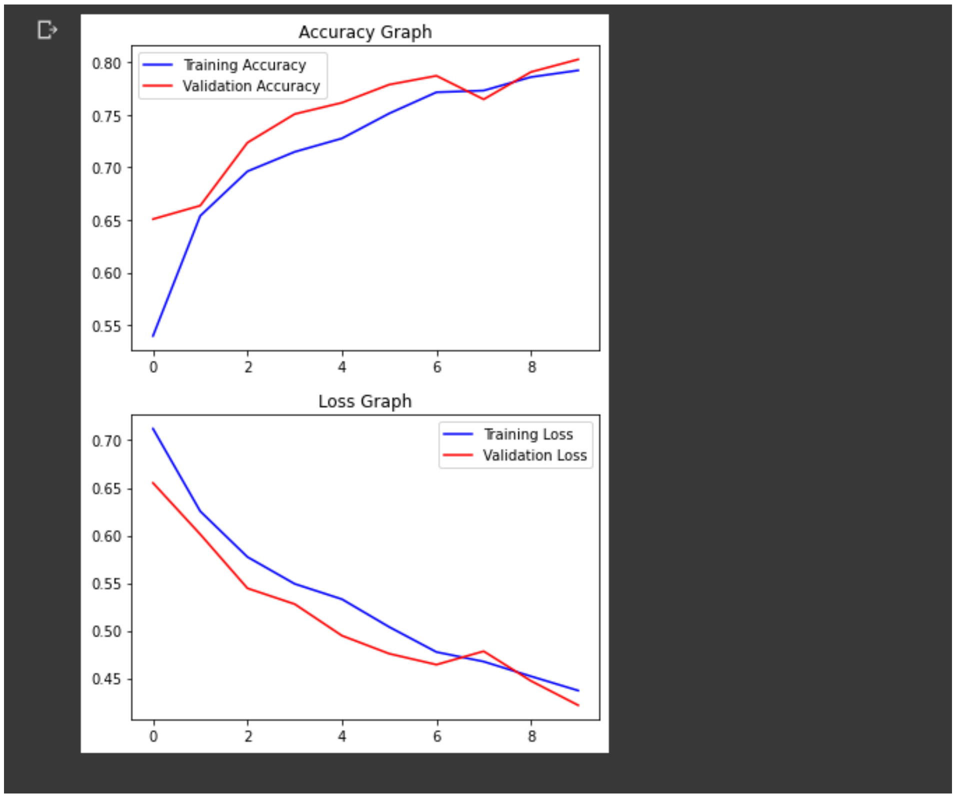
```
history = model.fit(train_iterator, epochs=10, validation_data= val_iterator)

Epoch 1/10
44/44 [==============================] - 156s 3s/step - loss: 0.7121 - accuracy: 0.5396 - val_loss: 0.6551 - val_accuracy: 0.6508
Epoch 2/10
44/44 [==============================] - 142s 3s/step - loss: 0.6254 - accuracy: 0.6539 - val_loss: 0.6012 - val_accuracy: 0.6636
Epoch 3/10
44/44 [==============================] - 142s 3s/step - loss: 0.5773 - accuracy: 0.6963 - val_loss: 0.5444 - val_accuracy: 0.7236
Epoch 4/10
44/44 [==============================] - 142s 3s/step - loss: 0.5491 - accuracy: 0.7148 - val_loss: 0.5278 - val_accuracy: 0.7508
Epoch 5/10
44/44 [==============================] - 140s 3s/step - loss: 0.5330 - accuracy: 0.7276 - val_loss: 0.4947 - val_accuracy: 0.7616
Epoch 6/10
44/44 [==============================] - 141s 3s/step - loss: 0.5038 - accuracy: 0.7514 - val_loss: 0.4757 - val_accuracy: 0.7788
Epoch 7/10
44/44 [==============================] - 139s 3s/step - loss: 0.4775 - accuracy: 0.7715 - val_loss: 0.4642 - val_accuracy: 0.7872
Epoch 8/10
44/44 [==============================] - 138s 3s/step - loss: 0.4674 - accuracy: 0.7731 - val_loss: 0.4783 - val_accuracy: 0.7648
Epoch 9/10
44/44 [==============================] - 139s 3s/step - loss: 0.4520 - accuracy: 0.7860 - val_loss: 0.4471 - val_accuracy: 0.7908
Epoch 10/10
44/44 [==============================] - 140s 3s/step - loss: 0.4370 - accuracy: 0.7923 - val_loss: 0.4215 - val_accuracy: 0.8028
```

Image : The 10 Epoch of CNN Model Training

5.0. Discussion on the Results

CNN model performance in this project was evaluated using a fitting model that will extract needed calculations like the training accuracy, validation accuracy, training loss, and validation loss. After that it will use the information gained to display it in the history graph below:



As we can see from the graph the overall accuracy for our CNN model is 79% and the chance of incorrect result is 21%. This model is trustworthy because the validation or testing accuracy is 80% which is considered a high percentage. Moreover, epoch loss is a scale of optimizing the process to minimize the training, the less percentage the better, and as we can see from the graph the training loss is 44%, while the validation loss is 42%.

6.0. Conclusion and Future Work

In conclusion, as we can see CNN algorithm is very useful in image classification using machine learning due to the high accuracy shown in the result. It will help a lot in classifying the different types of images in less time and energy. in the future, this model can be used to identify the illegal images that will help in many cases like sexualizing children, pornography, murder cases, and so on. It can also be used in schools to teach children with learning disabilities. Furthermore, it can be used in mobile phones to scan and identify the content of the picture and convert it to text or any other form.

7.0 REFERENCES

[1] Golle, P. (2008, October). Machine learning attacks against the Asirra CAPTCHA. In Proceedings of the 15th ACM conference on Computer and communications security (pp. 535-542). ACM.

[2] Tushar & Pankaj.(2019,December).Image Classification – Cat and Dog Images.Retrieved from:https://www.irjet.net/archives/V6/i12/IRJET-V6I1271.pdf

[3] Caroline,L.An Introduction to Image Classification [+ Superb AI Tutorial].Retrieved from:https://www.superb-ai.com/blog/an-introduction-to-image-classification-superb-ai-tutorial.

[4] Rostylav,D.(2019,26 April).A Brief History of Computer Vision (and Convolutional Neural Networks)Retrieved from https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3

[5] Khushi,S(2021,1 June).Beginner-friendly Project- Cat and Dog classification using CNN.Retrieved from https://www.analyticsvidhya.com/blog/2021/06/beginner-friendly-project-cat-and-dog-classification-using-cnn/

[6] Basics of Image Classification Techniques in Machine Learning.Retrieved from https://iq.opengenus.org/basics-of-machine-learning-image-classification-techniques/