AI-BHARATA EMERGING TECHNOLOGIES PVT LTD

INTERN STARTER PROJECT LEVEL-1(a) REPORT

TITLE: CAT AND DOG CLASSIFIER USING PYTORCH

By-

M.Surya

AI/ML Developer Intern

BE 8th Semester

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SL NO** | **TITLE** | **PAGE NO** |
| 1) | ABSTRACT | 01 |
| 2) | INTRODUCTION | 02 |
| 3) | METHOD | 03 |
| 4) | MODEL ARCHITECTURE | 15 |
| 5) | GRAD-CAM | 18 |
| 6) | RESULTS AND ANALYSIS | 23 |
| 7) | CONCLUSION | 29 |
| 8) | REFERENCES | 30 |
| 9) | CODE REPOSITORY | 31 |

**ABSTRACT**

The ever-increasing growth of data as well as the need of gaining insights from those data, has prompted us to develop sophisticated tools and technologies which help in increasing the accuracy of insights gained from the data.

Image classification is the process of predicting the class of an image based on the features provided in the training of the model. It is a type of supervised learning model.

The purpose of this project is to develop a state of the art deep learning model using Py-torch which aims to classify an input image into one of the two classes either cat or dog with the sole aim of achieving high amount of accuracy.

A convolutional neural network has been used for initial training of the model, later I have used VGG16.

Key Words: Image classification, convolutional neural network, VGG16, Deep learning

**INTRODUCTION**

Image classifications forms an integral part of everyday life, wherein we need to distinguish between different things to take better decisions. Although it seems easy to human eyes to distinguish, but when the classification is automated, it becomes difficult to conclude when the process is automated.

Although there exist various machine learning methods for solving this problem, by using deep learning models, we can successfully increase the accuracy by many folds.

In this project, I have used convolutional neural network as my base model and VGG16 as a pre-trained model, to solve the problem and achieve greater performance as well as accuracy.

The dataset used for this project is kagglecatsanddogs5340.zip from Microsoft which consists of 25000 images in total.

Link to the Dataset:

https://www.microsoft.com/en-us/download/details.aspx?id=54765

**METHOD**

1)**Convolutional Neural Network**

A Convolutional Neural Network (CNN) is a type of Deep learning architecture commonly used for image classification and recognition tasks. Convolution is a special type of linear operation. CNN consists of many layers namely:

* Input layer
* Convolutional Layer
* Activation Function Layer
* Pool layer
* Fully Connected Layer

In convolutional networks, we represent an image using a cuboid which has a defined length, width, and height as dimensions. Images generally have red, green and blue channels.

Convolutional layers consist of a set of learnable filters, every filter has small width and height, and depth is approximately equal to input volume. In forward pass, we slide each filter across the whole input volume in strides and compute the dot product.

**2) DATA COLLECTION**

The data is collected and downloaded using the below given link.

<https://www.microsoft.com/en-us/download/details.aspx?id=54765>

The images are stored in a tree like structure wherein the root directory is /content/PetImages and its subdirectories are Cat and Dog which consists of 12500 images each of cats and dogs respectively.

**Fig: Representation in Local File System**

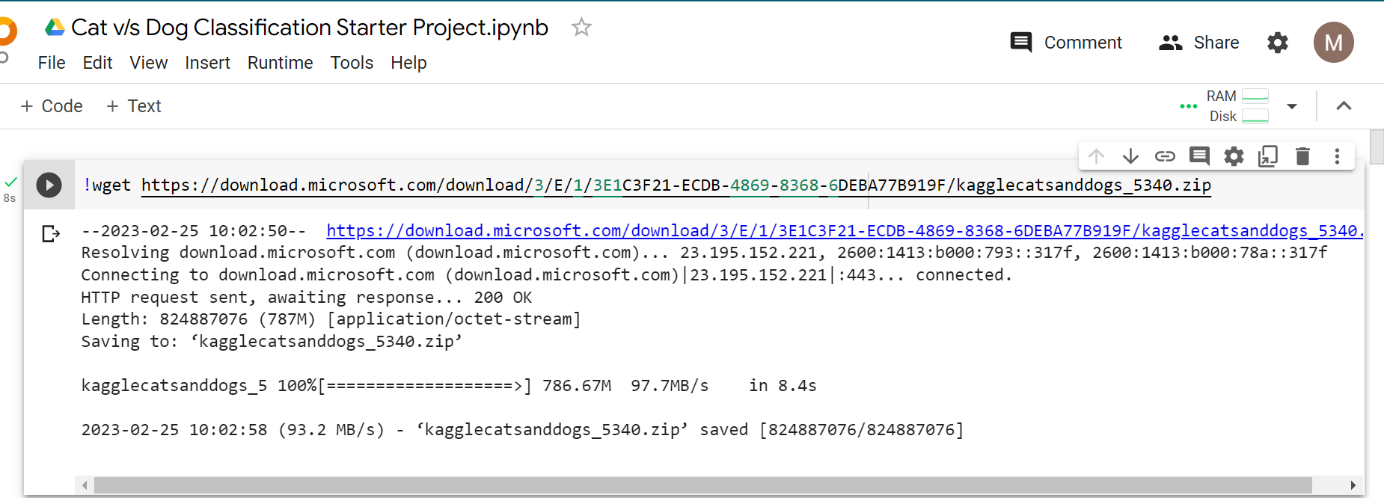


Graphical user interface, text, application

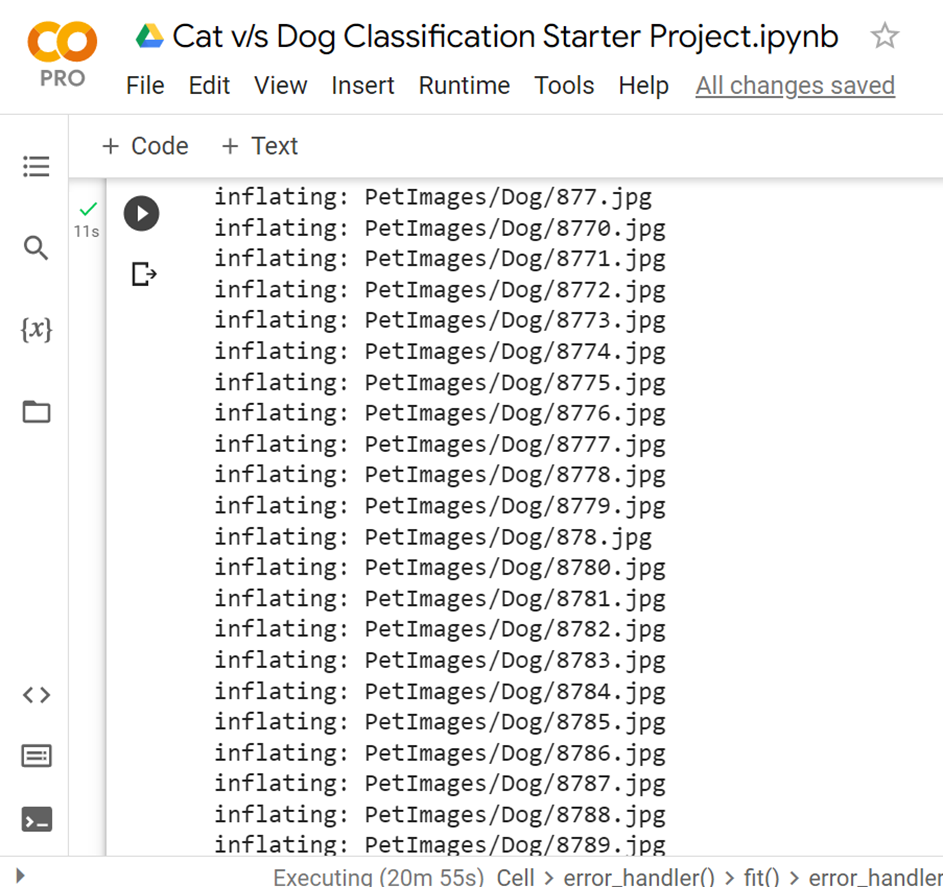
Description automatically generated

Graphical user interface, text, application

Description automatically generated



**Fig: Downloading of Dataset**



**Fig: Unzip Operation**

**3) ORGANIZATION OF DATA:**

After the data collection stage is complete, we need to store and organize the data into an appropriate form so that it can be easier to process the data.

Here, we have used a tensor to convert the input and output paths of an image for easier processing.

A Tensor in Pytorch is same as a numpy array, it is a multidimension matrix containing elements of a single data type.

Each class of images viz cats and dogs are labelled as 0 and 1 respectively for distinguishing between the two classes of animals.

**4)DATA PREPROCESSING:**

Once the dataset is built, there arises a need to pre-process the data to extract useful features from our deep learning models.

Data pre-processing is quintessential to gain an understanding regarding how it is organized and to be familiar with the structure of data.

Data pre-processing helps in increasing the accuracy of the model by removing unnecessary images which are not helpful to our analysis.

Three steps of pre-processing have been applied in this project:

* Identifying files which are not images.
* Creating a list of non-image type files and corrupted images.
* Deleting the corrupted files and non-image files from the

dataset.

Text

Description automatically generated

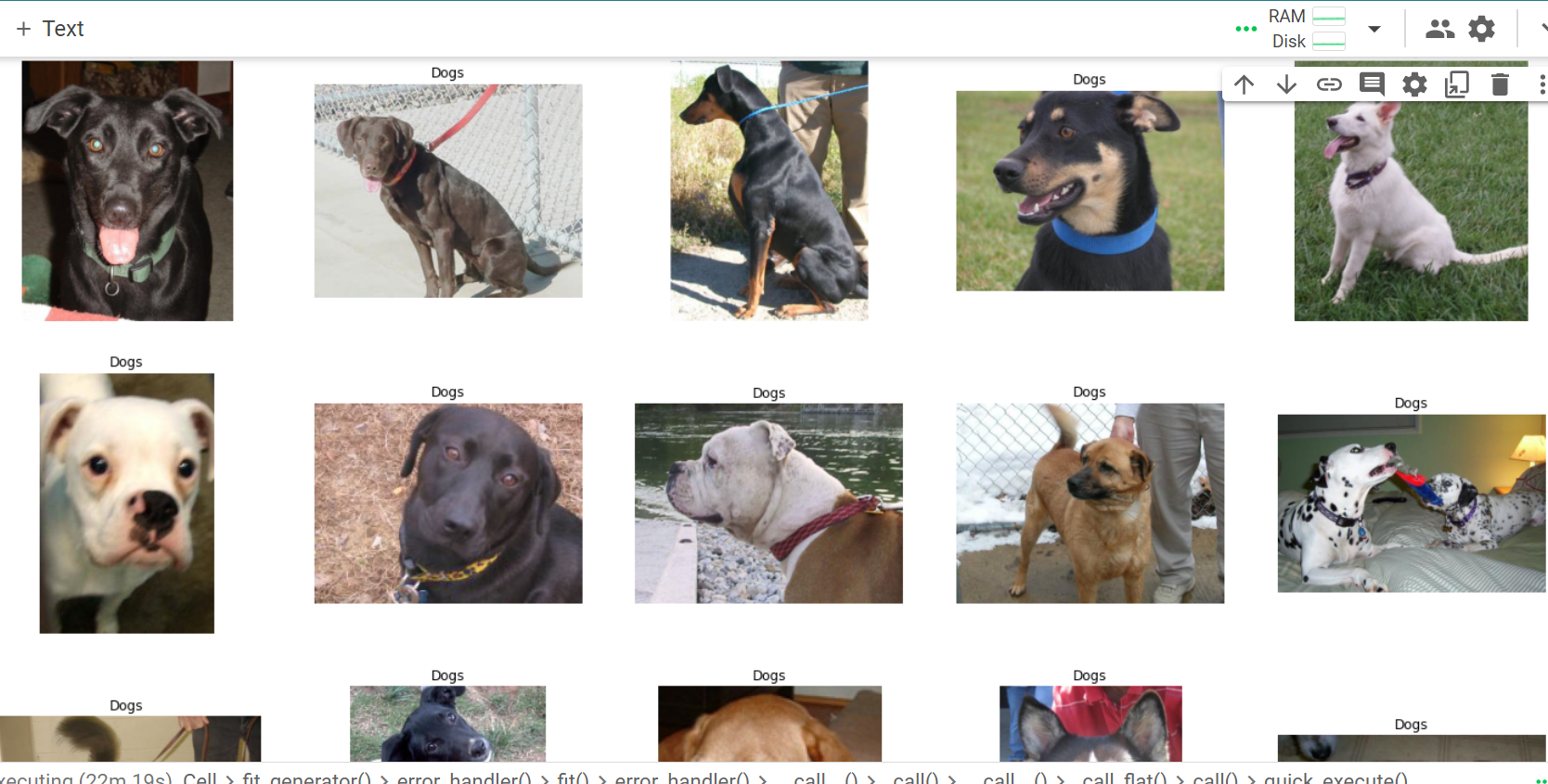
**Fig:Code Snippet representing Preprocessing of image**

**5)EXPLORATORY DATA ANALYSIS (EDA):**

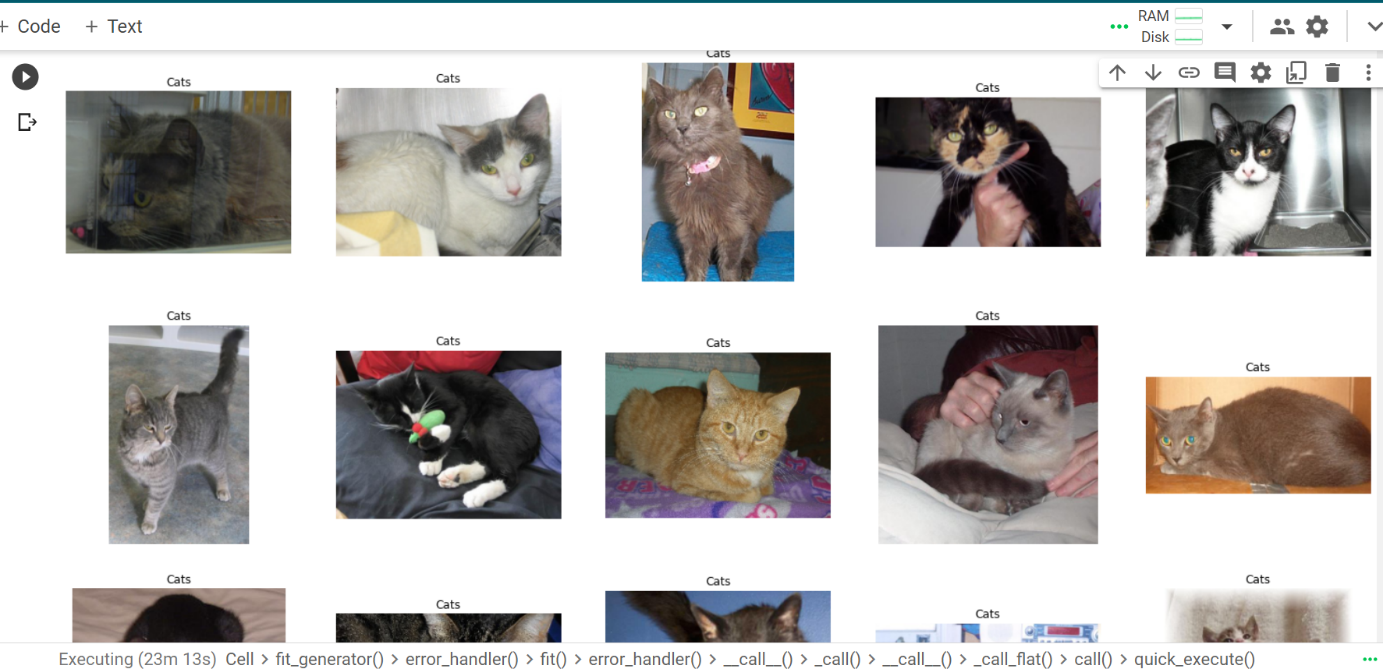
The pre-processing step is helpful to extract images which are not having any sort of defect from the dataset.

Exploratory data analysis (EDA) is an approach to analyse the data using visual techniques. This approach is applied to discover the trends, underlying patterns, or to check assumptions with the help of statistical summary and graphical representations.

Here, we display the images of cats and dogs using a 5 x 5 grid by using the label assigned to the images. Each image represented in the grid can have different saturation and qualities.



**Fig: Grid representation of dogs**



**Fig: Grid representation of Cats**

**6)AUGMENTATION OF DATASET AND SPLITTING INTO TRAIN AND TEST SET:**

Here we apply various augmentation techniques to the images, to increase the size of the dataset. Once the dataset size is increased, we apply ‘torch.utils.data.random\_split’ method to split the entire dataset into train\_set and test\_set.

Once that is done, we can find out the length of train\_set as well as test\_set.

Graphical user interface, text, application, email

Description automatically generated

**Fig: Splitting the datasets into train and test and Augmentation**

**Table

Description automatically generated with medium confidence**

**Fig: Training length and test length**

**8)MODEL ARCHITECTURE:**

* Classifier is given the name Model. The first layer is called MaxPool2D layer.
* The next layer is Dropout layer, which is used to get maximum pixel value to the next layer. Then there are five Conv2D layers with different channel sizes and widths. Followed by five linear layers.
* Maximum pooling or max pooling is a pooling operation that calculates the maximum or largest value in each patch of a feature map. The results are down sampled or pooled feature maps that highlight the most present feature in the patch.

Text

Description automatically generated

**Fig: Model Architecture**

**9)MODEL COMPILATION AND TRAINING:**

In the model compilation, the optimizer algorithm, loss function and metrics are the parameters are passed.

In optim.Adam method, we have certain parameters such as learning rate, weight\_decay.

* **Adam:** Optimization Algorithm
* **Cross Entropy:** It is the loss function for binary outputs.
* **net.parameters:** Model parameters defined in CNN architecture.

**Graphical user interface, text, application

Description automatically generated**

**Fig: Training of CNN Model**

**Text

Description automatically generated**

**Fig:Training Accuracy and Loss Output**

**Text

Description automatically generated**

**Fig: Printing Test Accuracy and Loss along with saving the model**

**Graphical user interface, application

Description automatically generated**

**Fig: Making Predictions on a test image**

****

**Fig: Test Image used for CNN**

**10) USING PRETRAINED MODEL:VGG-16**

After using CNN model on the dataset, it is time to test the classification by transfer learning using the pretrained model. Here I have used VGG-16 architecture as a pretrained model.

VGG-16 architecture mainly has three parts:

* **Convolutional Layer**: Filters are applied to extract the features of the image. The most important parameters are size of the kernel and stride.
* **Pooling Layer:** This function is used to reduce the spatial size to reduce the number of parameters and computation in the network.
* **Fully Connected:** These are fully connected connections to the previous layers as in a simple neural network.

****

**Fig: Initializing parameters to VGG16**

**Text

Description automatically generated**

**Fig:Training the model using VGG16**

**Text

Description automatically generated**

**Fig: Test Accuracy and Loss using VGG16**

**Graphical user interface, text, application

Description automatically generated**

**Fig:Prediction using VGG16**

**GRAD-CAM**

* Deep learning methods are considered to have a major issue of model interpretability, in other words, humans cannot interpret what is happening once the model is being trained, so they are called “black box methods”.

There are a few important factors which needs to be considered for model interpretability:

* Where the network is “focussing” in the input image.
* Which series of neurons activated in the forward-pass during inference/prediction.
* How the network arrived at the final Output.

**Definition of GRAD-CAM:**

GRAD-CAM uses the gradient of any target concept (for example dog in a classification network flowing into the final convolutional layer to produce a localization map highlighting the important regions of an image for predicting the concept.

Grad-CAM is applicable for various CNN model families:

* CNN with fully connected layers.
* CNN used for structured outputs.
* CNN used for multimodal inputs.

In this project, I have obtained Grad-CAM heatmap representation of images which are trained by CNN architecture.

Here, I have used Albumentations library which is an image augmentation library it is very useful to apply augmentation techniques to enhance the quality of images. It is actually an image augmentation package.

Augmentation techniques are applied to images wherein the existing images are applied suitable enhancement techniques so that the resulting images are of different shapes.

**Dataset used:** Fruits dataset.

**Dataset Contents:** Images belongingto three classes which are cucumber, eggplant, mushroom.

**Library Used- Albumentations:** It is a python library which is used for fast and flexible image augmentations.

**Steps used to apply Grad-CAM on images:**

1)Cloning the repository, importing the necessary library, updating OpenCV:

Table

Description automatically generated

**Fig: Cloning and importing necessary repositories and libraries**

In this step, **!git clone** command is used to clone the github repository into google colab, and make the folders available to access as it will be available in file explorer of google colab

**!pip install:** this command is used to install a particular library, here albumentations library for applying Grad-CAM on our model.

This command can also be used as a prerequisite for upgrading a particular library in python.

2) Importing necessary libraries

Text

Description automatically generated

**Fig: Importing necessary libraries**

****

**Fig: Train-Test Split along with path to CSV file**

**3) Augmentation Techniques**

The techniques which are used for augmentation include:

* Rotate
* Horizontal Flip
* Vertical Flip
* Normalize

These techniques are applied to the train as well as validation datasets.

**4)Preparing Data Loaders for train and validation datasets:**

Once we split our data into training and validation sets, our next step is to load our train and validation datasets into a dataloader.

Dataset is responsible to store images as well as its labels, whereas DataLoader is responsible to wrap an iterable around a dataset to enable easy access to the samples.

Graphical user interface, text, application, email

Description automatically generated

**Fig: No of batches in train and ValidLoader**

In the above image we are determining the no of batches in train and validloader which are 10 and 3 respectively.

The code in the second cell is responsible to determine the image shape in one batch and batch label shape.

* The first value represents, the batch size = 16
* Second value represents = No of channels(3) = RGB
* Third and fourth values represents image height and width respectively.
* One batch label shape represents no of labels in one batch, which are 16 respectively.

**5)Building a CNN Model:**

Text, letter

Description automatically generated

**Fig: CNN Model for using Grad CAM**

In the above image, we have name of the CNN model as ImageModel.

* **Feature\_extractor** name is Sequential.

There are 4 Conv2d layers, 4 ReLU layers, 3 MaxPool2d layers.

* **MaxPool2d layer**
* Flatten() function is used to flatten the outputs.

**6) Defining train and evaluation function**

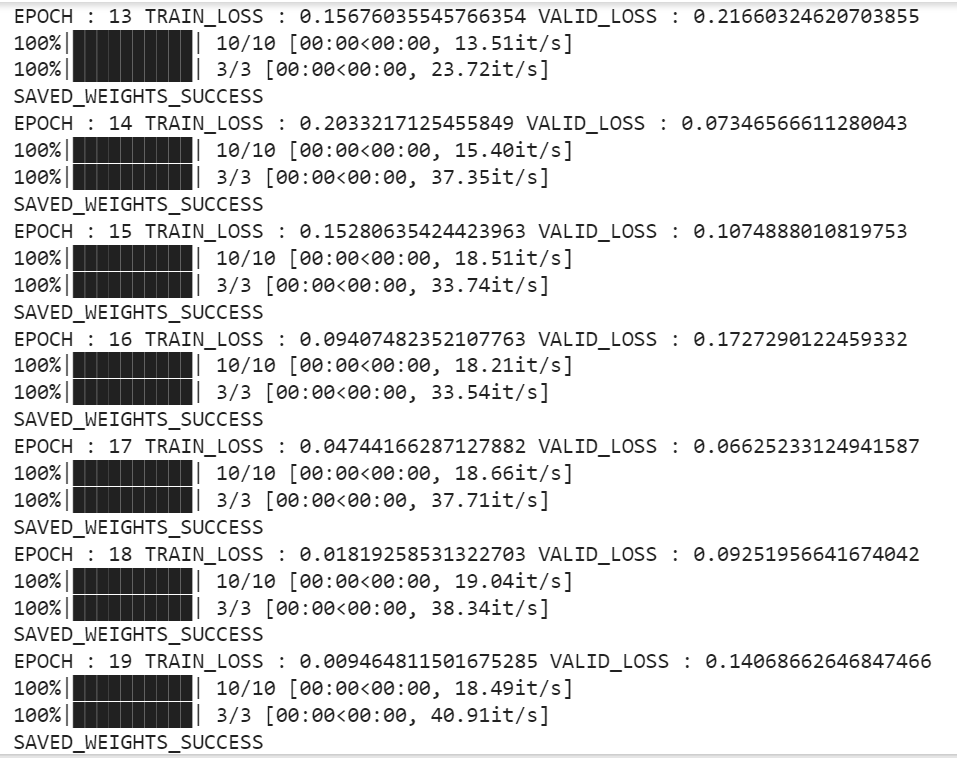
**7)Specifying Optimizer and Loss Function**

**8)Training the Model**

**Text

Description automatically generatedText

Description automatically generated**

****

**Fig: Training CNN Model**

**9) Defining grad\_cam method:**

The ‘get\_gradcam’ method takes in (model,image,label,size) of an image as input and produces a heatmap as output.

The heatmap is further responsible to highlight the specific regions of an image in which our desired class is present.

Heatmap mainly shows the region in which our desired class is present.

We can actually verify, where in the image the CNN is looking.

**10) Displaying the image along with its predicted class using heatmap:**

After defining the Grad-CAM function, we later move on to choosing the image in the valid set, so we iterate the valid set to find the desired image by using the index given to an image.

After obtaining the image, we move on to converting the image into numpy array for easier processing.

Later we apply the CNN model on that image, to extract the important features in that image.

Once the features are extracted, later we proceed towards plotting the heatmap and predicting the desired class.

**A picture containing graphical user interface

Description automatically generated**

**Fig:Image displaying GradCAM applied to image**

**RESULTS AND ANALYSIS**

The CNN model was trained for 10 epochs. The observations which I came across while training the model are:

* Training accuracy increases every iteration.
* Training Loss decreases for every iteration.

**Text

Description automatically generated**

**Fig:Training of CNN Model**

**For VGG16 ModelA screenshot of a computer

Description automatically generated with medium confidence**

* Training accuracy tends to increase as the iteration progresses.
* Validation accuracy tends to decrease as the iteration progresses.

**CNN and VGG16 MODELS**

|  |  |
| --- | --- |
| **CNN** | **VGG16** |
| Training accuracy increases | Training accuracy increases |
| Loss decreases | Loss decreases |
| As the parameters change,simulataneously accuracy changes | As parameters change, simultaneously accuracy and loss changes. |
| Accuracy is high as compared to VGG16. | Accuracy is slightly less as compared to CNN. |

**By Applying Grad CAM**

By applying Grad CAM, and while training the model, what we try to observe is training loss tends to decrease and validation loss tends to increase as the iteration progresses.

Here I have used 20 epochs for training.

Text

Description automatically generated

**Fig:Grad CAM Training**

**CONCLUSION**

Hence the given dataset of cats and dogs are carefully processed, analysed and trained using CNN and VGG16 models and their accuracies are determined and compared.

The accuracy is more for CNN architecture.

Later for enhancing model Interpretability, I have used Grad-CAM, to highlight important regions in an image.

**CODE REPOSITORY**

I have hereby uploaded my project onto my Github repository.

Link to the drive:

<https://colab.research.google.com/drive/1WKFZ-Ud_qZlPFEbh18Dmp4ASa25zrUBW>

Link to the repository:

<https://github.com/Surya0907/Ai-Bharata-Internship/blob/main/Cat_v_s_Dog_Classification_using_PyTorch.ipynb>

It contains the report as well as code file which is uploaded from google colaboratory and the problem statement file also being uploaded.