International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 04 | April - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Design and Development of Fine-Grained Image classification

Sumiran Hirpurkar, Vivek Dhore, Yash Ingole, Tulip Deshmukh, Mohammad Danish Abdul Majeed

Prof. Sanjivanee R. Kale

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

P.R.Pote College of Engineering and Management, Amravati

Abstract:

Fine-Grained Image classification is an area of machine learning in which we can perform classification between the same categories such as dog breeds, where we can determine the breed of dog. At present fine-grained image classification has developed variety of methods including multi-network learning, target part detection and visual attention. Each method is to obtain a distinguishing area in the image, which help the network to learn more effective features to complete the classification and recognition of fine-grained image. In this paper, we introduce an image dataset for fine-grained classification of dog breed: Kaggle's Dog Database. The database consists of 120 dog breeds which contains at least 60 images of each breed. It helps for the accurate prediction of images with its accuracy percentage. We tested our model and result of our model show that our proposed model can achieve the desire result.

Keyword: Classification, detection, visual attention, database, prediction, accuracy.

I. Introduction

Fine-grained image classification is also known as sub-categorization within the same category. The purpose of this classification is to make sub-classification of images such as cars, birds, flowers, dogs, etc. that belong to same category. Fine-grained image classification is more difficult than generic classification because the difference between same categories is very small. Visual difference between objects are so small, they can be influenced by the factors such as location of an object, pixels of image, pose of object. One small part of object can make huge impact on the image classification.

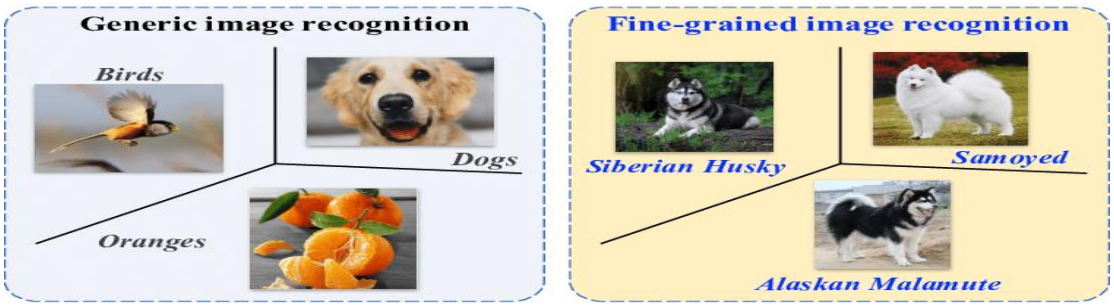


Figure 1.1: Generic image classification vs Fine-grained image classification

Fine-grained image classification is performed between two objects of same category. Different from generic image classification task such as object recognition, fine-grained image classification has more detailed class precision, and the differences between the classes are more subtle, often by small local differences [1]. The primary goal of image classification is to ensure that all photos are classified according to their respective groupings. Classification is easy for human but is has proved to be major problems for machine. It consists of unidentified patterns compared to detecting an object as it should be classified to the proper categories [2].

To face the above problems, many fine-grained classification methods have performed parts detection in order to decrease the intra-class variation [3]. The most prevalent deep learning framework is the convolutional neural network (CNN), which is a multilayer neural network. The convolution neural network (CNN) is a machine learning algorithm that performs exceptionally well in hyperspectral image classification. While primitive approaches need hand-engineering filters, ConvNet can learn these features with enough training.

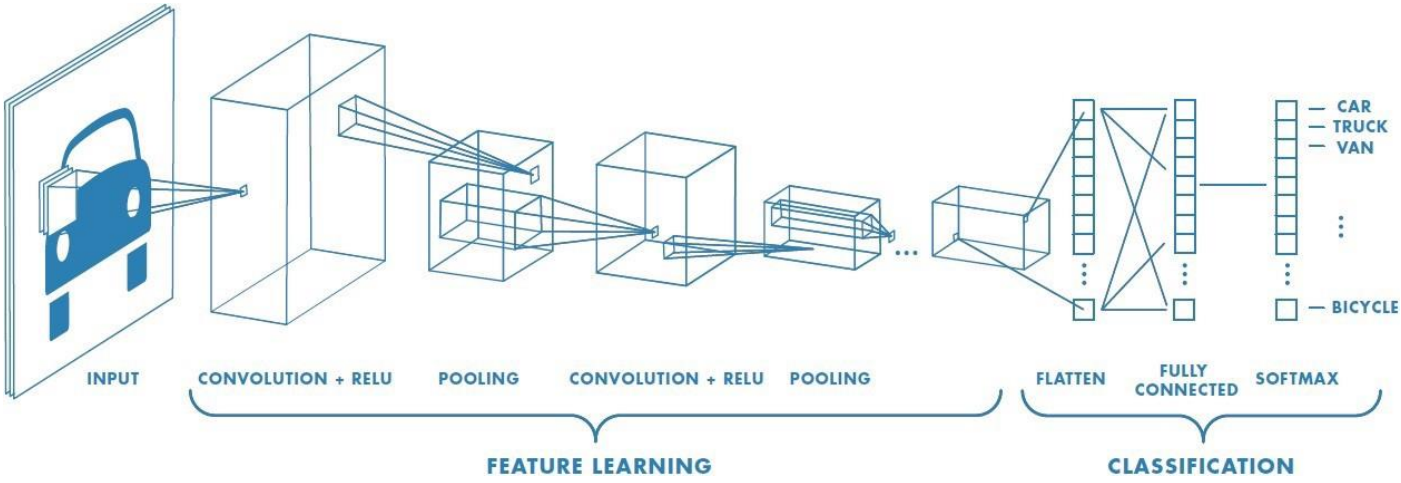


Figure 1.2: Convolutional Neural Network (CNN)

In this paper, picture classification is accomplished using a deep neural network based on TensorFlow with Python as the programming language.Hundreds of images are used as the input data in this project. The accuracy of each percentage of 'train' session will be studied and compared [2].

II. Related Work

Many scientists have spent years studying discriminative visual features or using part-based representation. The key to image classification is to extract the robust features of the object and form better feature representation. There is wide range of methods that have been developed for fine-grained object recognition. These approaches seek to distinguish subtle differences among similar classes typically by identifying and reasoning about the layout structure of object part present in fine-grained classes.

In order to make different subclasses, an intuitive approach is to explicitly take advantage of differences between corresponding object parts. Few features are extracted from object parts and fed to linear classifiers, such as SVMs. Deep learning method provide better performance, with the parts located and normalized for pose. Feature learning is a promising approach that can generate powerful appearance representation. Much work has focused on encoding low-level features such as SIFT or HOG or mining discriminative templates. The recent success of convolutional neural network on large-scale classification

and face recognition demonstrates that powerful features can be learned directly form pixels. This inspires us to adopt convolutional neural network for fine-grained recognition.

Part-based R-CNN is based on the famous target detection algorithm R-CNN. The main idea is: First, use the selective search method to generate candidate frames that may be objects or key parts of objects in fine-grained images. Then, three detection models are supervised and trained based on the label box and the label information of the key parts of the object, one of which is the object-level detection model, and the other two are the detection models of different key parts of the object. Then, the corresponding geometric position constraints are added to the predicted frames obtained by the three detection models, so as to obtain more ideal detection results of objects and key parts. However, in the Part-based R-CNN algorithm, it is not only necessary to use object position labelling boxes and key parts labelling information during training, but also to require the predicted image to provide these additional artificial labelling information during model prediction. Pose Normalized CNN is an improved version inspired by the Part-based R-CNN algorithm. Compared with the Part-based R-CNN algorithm, which simply adds the corresponding geometric position constraints to the predicted frame obtained by the detection model, Pose Normalized CNN uses a more accurate sub-image pose alignment operation.

R-CNN has been trained to detect object sections, which has been applied to fine-grained classification. We use sequential reasoning employing LSTM to search for object parts in order to classify an image, in contrast to these works, which anticipate object parts to categorise an image in one shot.Additionally, we cannot directly use these approaches since their object proposals are supported objectness, whereas we want object parts, and parts aren't objects. there's a number of search-based methods in vision.as an example, minimizing energy of graphical models has been addressed using town Markoff chain (MCMC), which successively will be viewed as an exploration. Our approach is closely associated with those methods that seek to be told the heuristic and successor functions of the search on training data, rather than using heuristics.

III. Proposed Model

In this section we describe about our model of fine-grained image classification, which contains different modules and methods to supply the require output. To use the various methods we import the module like matplotlib, seaborn, gradio, os, tensorflow keras, pandas, etc. the foremost important libraries are tensorflow and keras which contributes as follow:

Tensorflow : TensorFlow includes a special feature of image recognition and these images are stored in a very specific folder. With relatively same images, it'll be easy to implement this logic for security purposes. The dataset_image includes the related images, which require to be loaded.

Keras : Keras is employed for creating deep models which may be productized on smartphones. Keras is additionally used for distributed training of deep learning models. Keras API may be a deep learning library that has methods to load, prepare and process images.

```
import matplotlib.pyplot as plt
import seaborn as sns
import gradio as gr

import os
import gc

from sklearn.model_selection import train_test_split


import tensorflow as tf
from tqdm.autonotebook import tqdm

import numpy as np #
import pandas as pd

from keras import Sequential
from keras.callbacks import EarlyStopping


from tensorflow.keras.optimizers import Adam,SGD
from keras.callbacks import ReduceLROnPlateau
from keras.layers import Flatten,Dense,BatchNormalization,Activation,Dropout
from keras.layers import Lambda, Input, GlobalAveragePooling2D,BatchNormalization
from tensorflow.keras.utils import to_categorical
# from keras import regularizers
from tensorflow.keras.models import Model
from keras.preprocessing.image import load_img
```

Figure 3.1: Importing Modules

We load the .csv file of dog breeds into our program which contain the name and their respective id's. After loading the database in program, following figure examines the dog breed name with their maximum images available within the database.

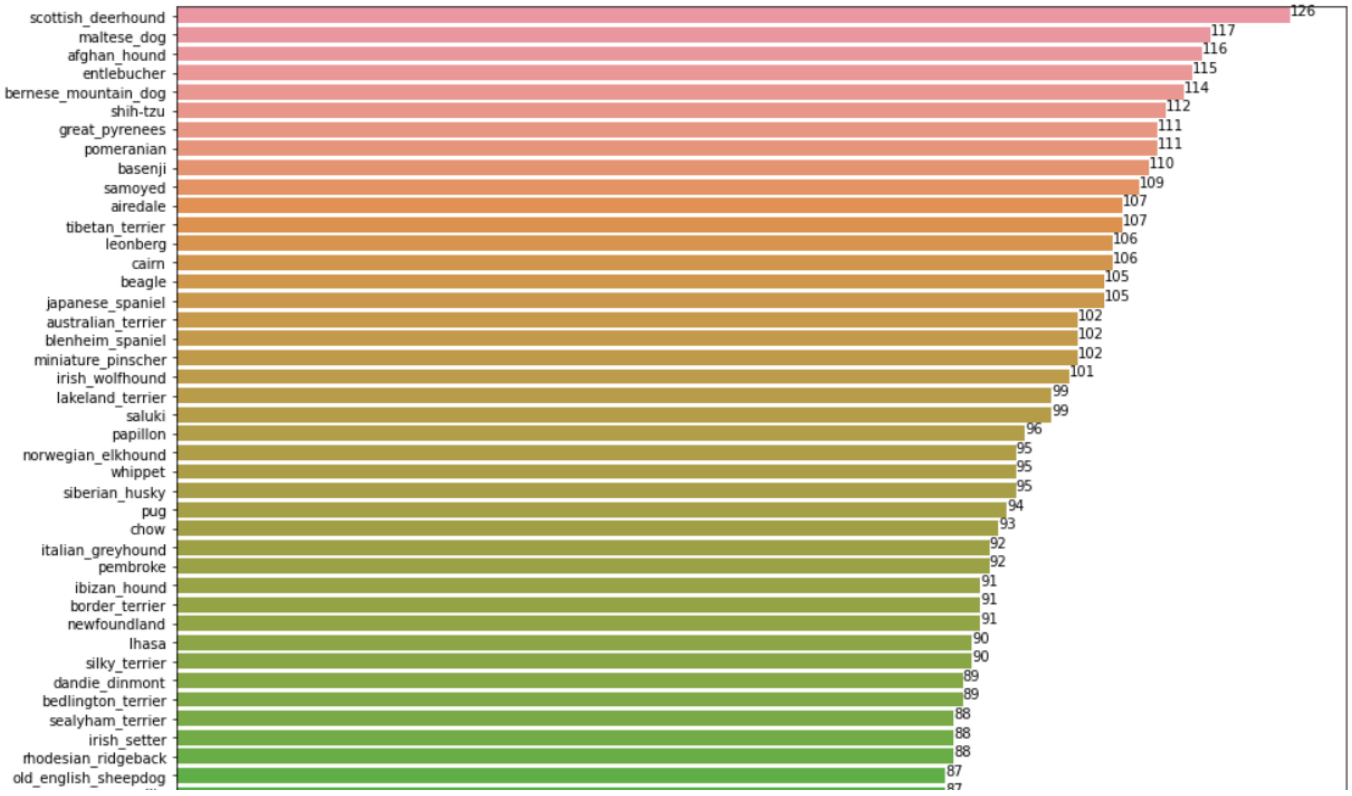


Figure 3.1: Dog breeds with their count

As the model works for the pictures of same configuration so we want to configure the pictures to specific size. To perform this task we use the plt.figure(figsize=(20,20)) method, as shown in figure 3.2.

```
# np.where(y[5]==1)[0][0]

# lets check some dogs and their breeds
n=25

# setup the figure
plt.figure(figsize=(20,20))

for i in range(n):
#     print(i)
    ax = plt.subplot(5, 5, i+1)
    plt.title(classes[np.where(y[i] ==1)[0][0]])
    plt.imshow(X[i].astype('int32')) # .astype('int32') ---> as imshow() needs integer data to read the image
```

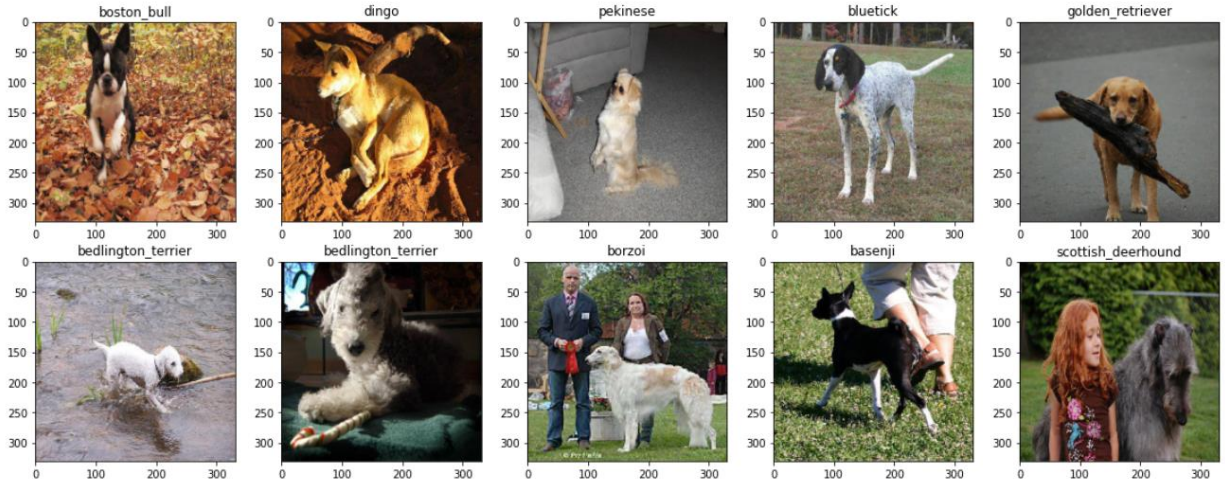


Figure 3.2: Configuring images

After configuring images to fixed size the following task is to extract the feature of images by using the pre-trained modules from the keras library.

This modules are as follow:

Inception: Inception model could be a convolutional neural network which helps in classifying the various kinds of objects on images. Also referred to as GoogLeNet. It uses ImageNet dataset for training process. within the case of Inception, images have to be 299x299x3 pixels size.

Xception: Xception could be a convolutional neural network that's 71 layers deep. you'll load a pretrained version of the network trained on quite 1,000,000 images from the ImageNet database. The pretrained network can classify images into 1000 object categories, like keyboard, mouse, pencil, and plenty of animals. As a result, the network has learned rich feature representations for a large range of images. The network incorporates a picture input size of 299-by-299.

NASNet-Large: It is a convolutional neural network that's trained on over 1,000,000 images from the ImageNet database. The network can classify images into 1000 object categories, like keyboard, mouse, pencil, and plenty of animals. As a result, the network has learned rich feature representations for a decent range of images. The network features a picture input size of 331-by-331.

InceptionResNetV2: It's possible that Inception-ResNet-v2 is a convolutional neural network trained on over 1,000,000 photos from the ImageNet collection. The network is 164 layers deep and might classify images into 1000 object categories, just like the keyboard, mouse, pencil, and much of animals. As a result, the network has learned rich feature representations for an honest range of images. The network encompasses a picture input size of 299-by-299, and so the output is also a listing of estimated class probabilities.

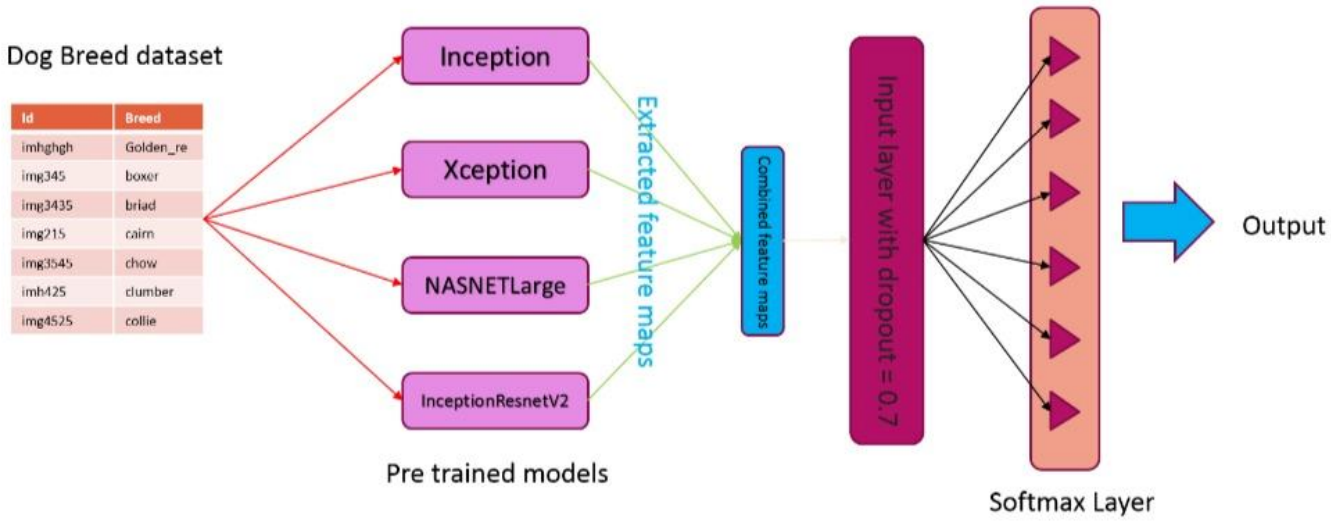
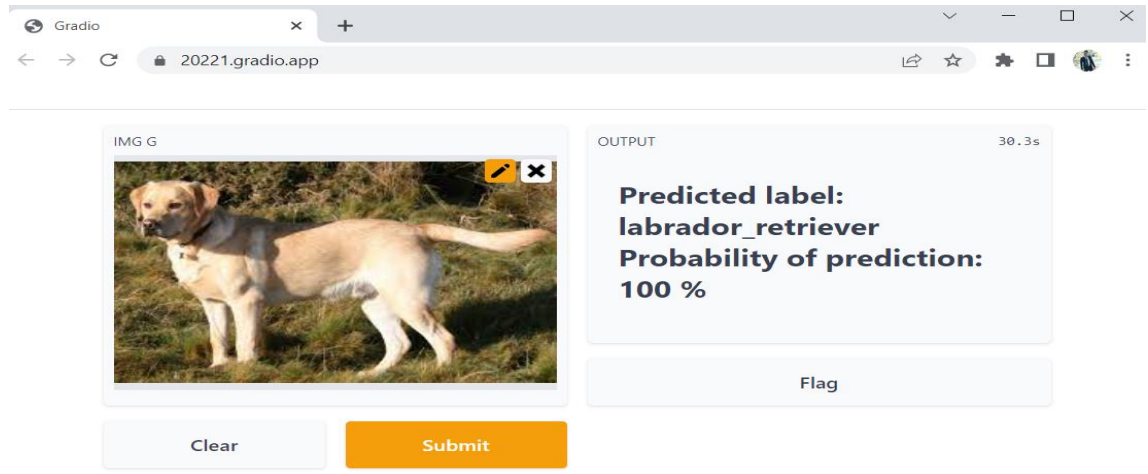
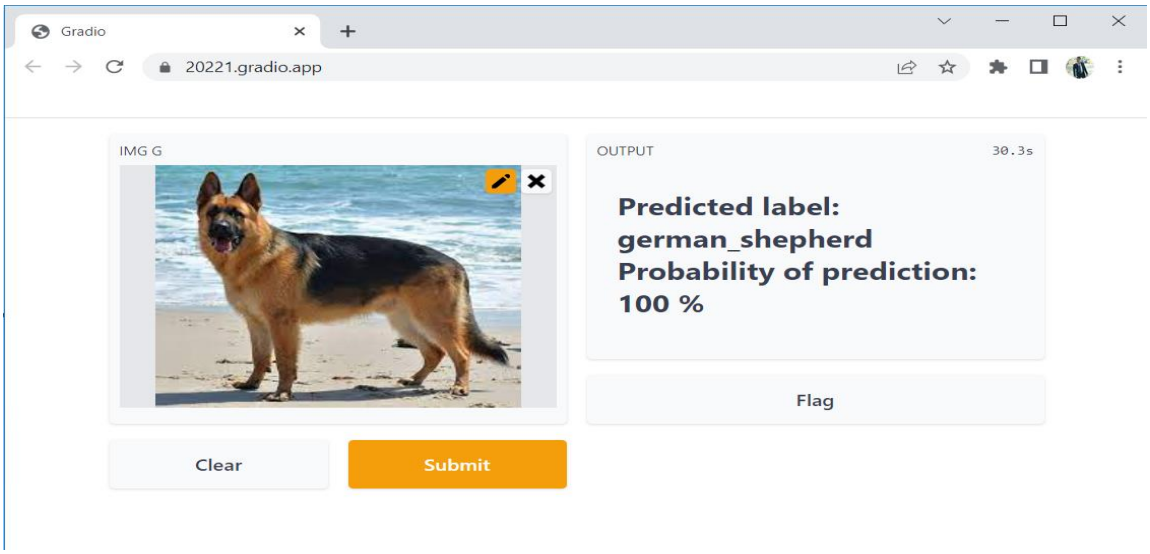
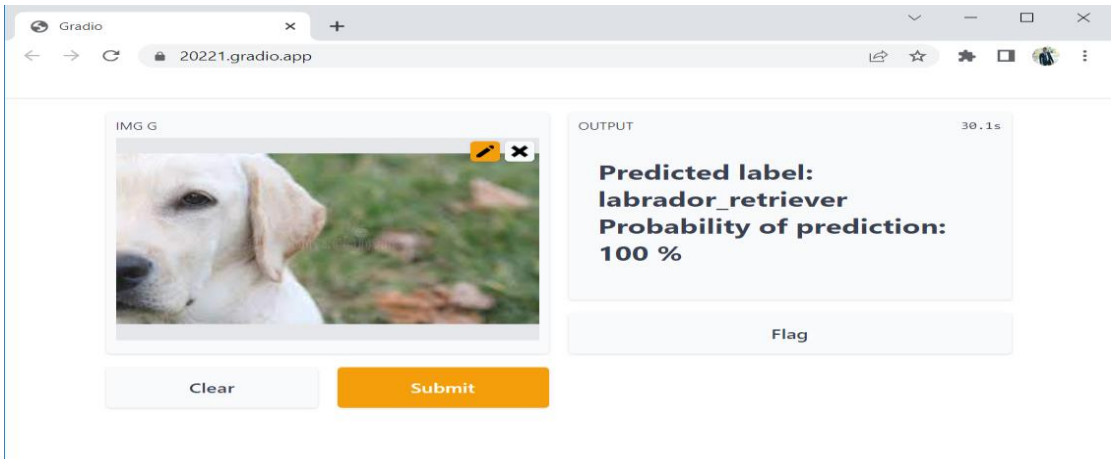


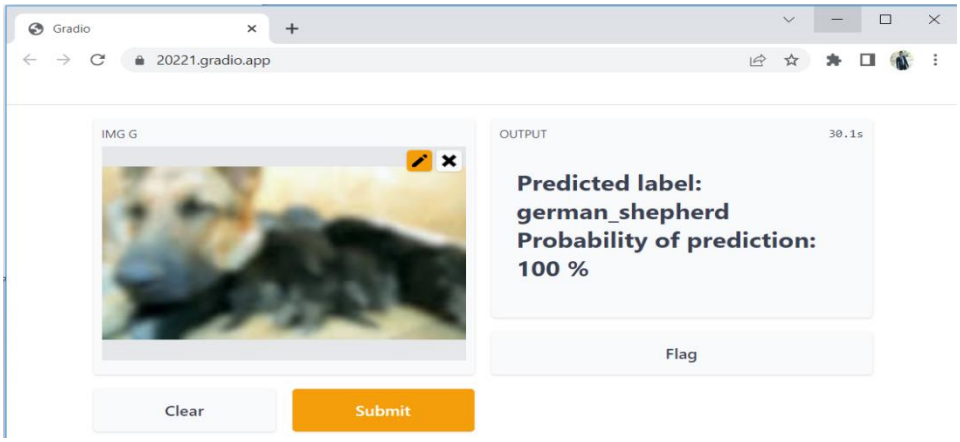
Figure 3.3: Flow of extraction features.

After giving input the to the above model it performs Inception, Xception, NASNETLarge , and InceptionResnetV2 we are ready to extract the features of the images and might perform the classification with the database images. Our model can even predict the blur and half face images moreover as predict the statue of dog breed. Following figure shows how we are going to input the image and predicted output, also few samples of images inputed to the model.









References:

[1] Chun-feng GUO, Hai-rong CUI, Kun YU and Xin-ping MO, Shandong Foreign Trade Vocational College, Qingdao University.

[2] Mohd Azlan Abu, Nurul Hazirah Indra, Abdul Halim Abd Rahman, Nor Amalia Sapiee and Izanoordina Ahmad, Universiti Kuala Lumpur British Malaysian Institute, Malaysia.

[3] ZongYuan Ge, Alex Bewley, Christopher McCool, Peter Corke, Ben Upcroft, Conrad Sanderson, Australian Centre for Robotic Vision, Brisbane, Australia Queensland University of Technology (QUT), Brisbane, Australia University of Queensland, Brisbane, Australia Data61 / NICTA, Australia.

[4] HuapengXu, GuilinQi, JingjingLi, MengWang, KangXu and HuanGao, Southeast University, Nanjing, China. University of Electronic Science and Technology of China, Xi'an Jiaotong University, Xi'an, China. Nanjing University of Posts and Telecommunications, Nanjing, China.

[5] Junming Lu, Wei Wu, Department of Information Engineering, Wuhan University of Technology, Wuhan, Hubei Province, 430000, China. Department of Information Engineering, Wuhan University of Technology, Wuhan, Hubei Province, 430000, China.