Image Classification using SVM

Submitted by

Abu Nahid Anik (160104147)	**nahid420anik@gmail.com**
Durjoy Roy (150204027)	**durjoyroyaust@gmail.com**
Abdul Hadi (160104145)	**topwin34@gmail.com**

Supervised by

Mir Tafseer Nayeem

Assistant Professor

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

1. Introduction:

1.1 Definition: Image classification is one of the most classical problems of concern in image processing. There are various approaches for solving the problem of image classification. In this project, we used SVM (Support Vector Machine) to classify images. For data we took the images of cats and dogs and shuffled the data then applied SVM model to classify whether it's a cat or a dog and also calculated accuracy of the prediction. we splitted the data into train and test data and tried various experiments to improve our performance on the test dataset, and finally got the best accuracy of 84.00%. Image classification has huge amount of usage in fields like medical analysis, computer vision, image processing and many more real-life applications.

1.2 Motivation: The motivation for choosing this project is to learn and have real experience of applying image classification. There are also many competitions like Dogs vs Cats competition from Kaggle which also encouraged us to choose this topic as our project. The competition of Dogs vs Cats relies on the problem of distinguishing images of dogs and cats.

1.3 Challenges: Distinguishing images of dogs and cats is easy for humans, but evidence suggests that cats and dogs are particularly difficult to tell apart automatically by computers. As we are working with images there can be many complications like a big fluffy cat can be groomed just like a dog and that will be very difficult for computer to determine whether it's a cat or dog. Here is a pic of a cat and a dog that looks quite similar.

Fig: Similar looking cat and dog

Moreover, cats and dogs can have clothes, masks etc. Images can have noise, distortion. And image can also have subjects other than cats and dogs. These things will also cause problem while distinguishing between cats and dogs

2. Related Works:

Classification between objects from image 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. There are statistical machine learning classifiers like Decision Tree and deep learning architectures like Convolutional Neural Networks.

Decision Trees: It is also a supervised machine learning algorithm, which at its core is the tree data structure only, using a couple of if/else statements on the features selected. Decision trees are based on a hierarchical rule-based method and permits the acceptance and rejection of class labels at each intermediary stage/level.

K Nearest Neighbor: The k-nearest neighbor is a very simple machine learning algorithm. This algorithm simply relies on the distance between feature vectors and classifies unknown data points by finding the most common class among the k-closest examples.

Artificial Neural Networks: Artificial Neural Networks are statistical learning algorithms and are used for a variety of tasks, from relatively simple classification tasks and it is Inspired by the properties of biological neural networks. ANNs are implemented as a system of interconnected processing elements, called nodes, which are functionally analogous to biological neurons. The connections between different nodes have numerical values, called weights, and by altering these values in a systematic way, the network is eventually able to approximate the desired function.

Convolutional Neural Networks: Convolutional neural networks are comprised of two very simple elements, namely convolutional layers and pooling layers.

So, there are many image classification methods but SVM provides good classification as it implements a technique called kernel trick. Kernel trick transforms the data and then based on these transformations it finds an optimal boundary between the possible outputs. It generally transforms the data in higher dimension and tries to find optimal hyperplane. As it transforms the data the training time is much longer as it's much more computationally intensive but provides good results

3. Project Objective:

3.1 Task of the System:

3.1.1 Subtasks:

- **Data processing**
  - **Data Transformation**: we transformed the data or images from colored images to greyscale images
  - **Data resize**: we used images as our data. To work without images first we converted all our images to the same size.
  - **Data scan:** we checked if any of the data are corrupted or damaged. All corrupted data were excluded
  - **Data storage**: we used pickle to store all the valid and resized images and data.

- **Data normalization:** Normalization was done by shuffling all the data.
- **Data splitting:** we splitted the data for training and testing purpose**.** we used 75% data for training and 25% data for testing.
- **Model training:** we imported SVC (support vector classifier) and trained the model using train data.
- **Model Storing:** after training the model we saved and stored the model using pickle.
- **Model testing:** we predicted the nature of test data and calculated accuracy of our prediction.

3.1.2 Flow Chart:

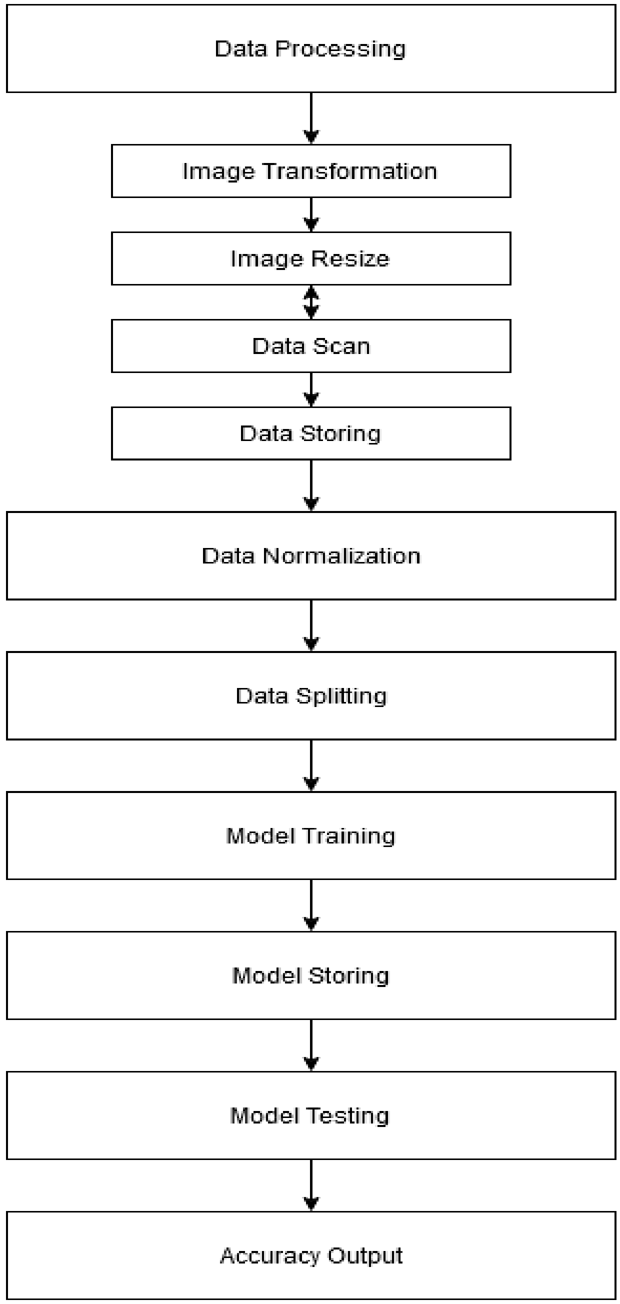


Fig: Flow Chart

3.2 Dummy Input Output:

Input: Here is an example input of data image.

Fig: Input Image

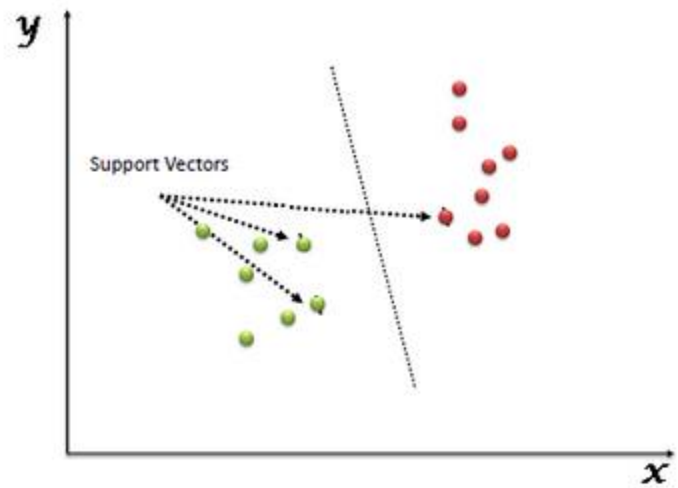
Output: Here is the output image for the input image.


```
Accuracy:  0.848
Prediction is :  Dog
[Finished in 577.7s]
```

Fig: Output Image and Accuracy

4. Methodologies:

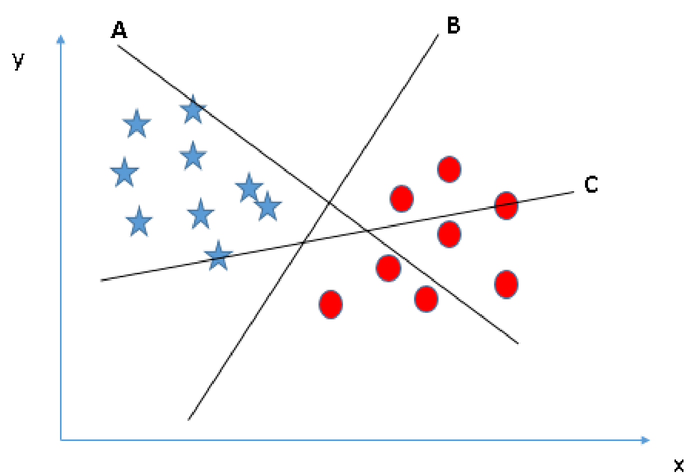
"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well



Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

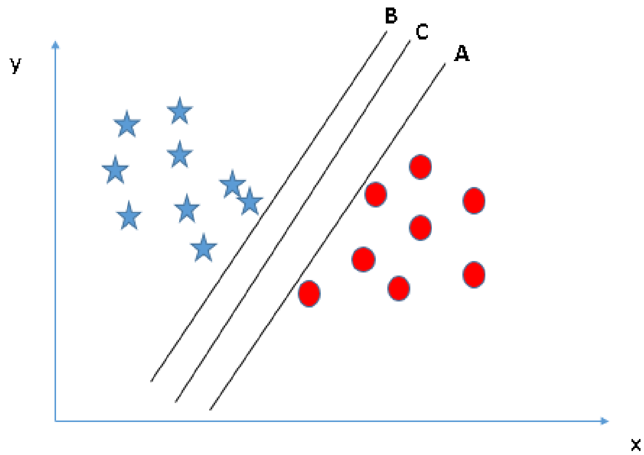
How does SVM work-

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C)

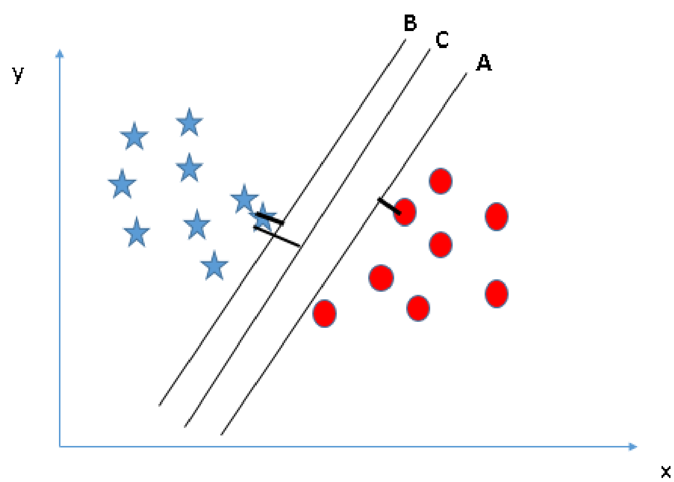


Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.

- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well.

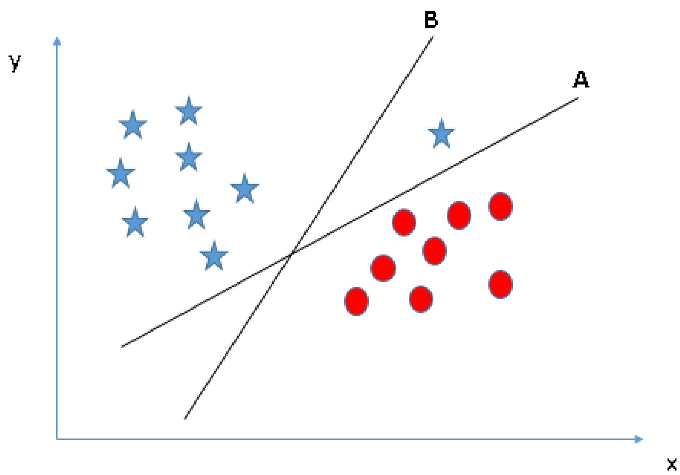


Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin.



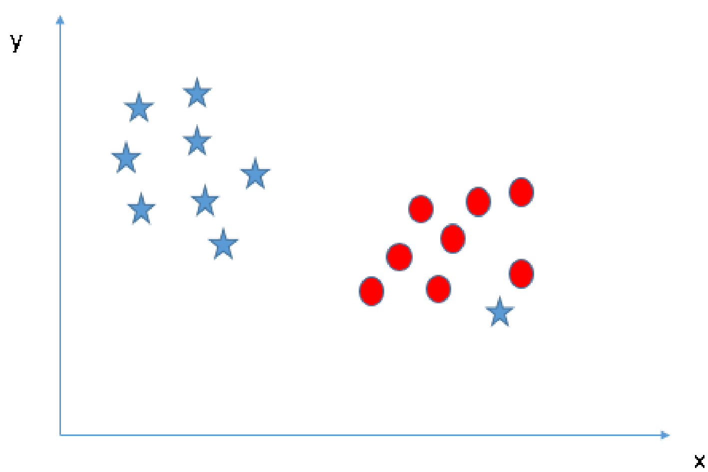
Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- **Identify the right hyper-plane (Scenario-3):** Hint: Use the rules as discussed in previous section to identify the right hyper-plane

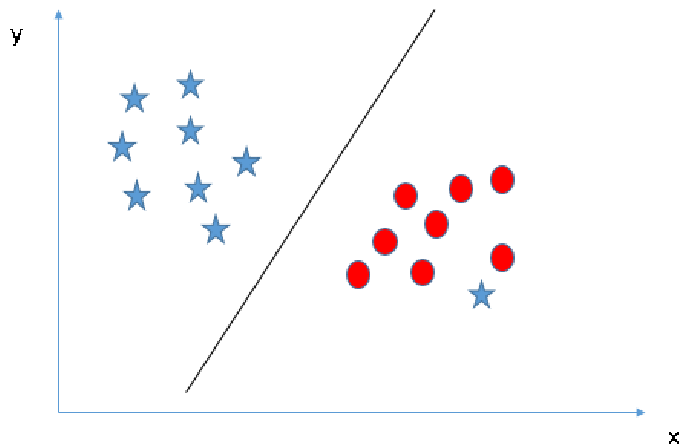


Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

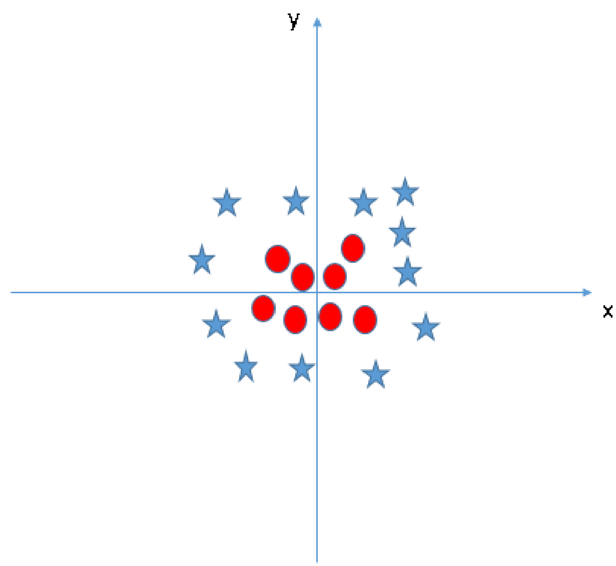
- **(Scenario-4):** Below, it's impossible to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



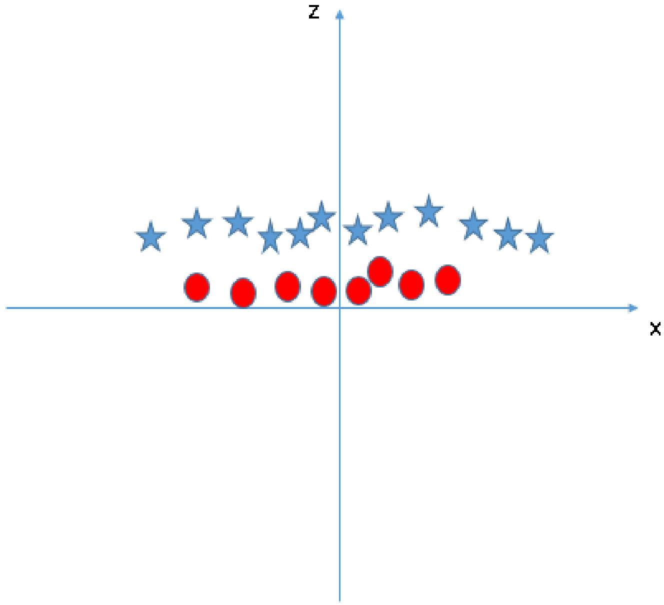
One star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



- **(Scenario-5):** In the scenario below we cannot have a hyperplane that separates two classes



SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:



In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

5. Experiments:

5.1 Dataset:

- The most important role of data is classifying the datasets into various categories. It helps to recognize and classify similar objects. So the selection of dataset is very important. We searched in the internet for a good dataset, and found many datasets related to our project. After that we selected one of them.
- In our dataset, we have images as our data. In total there are 25000 images in our dataset. We have used two classes in it. The names of the classes are “Cat” and “Dog”.
- In the “cat” class there are 12500 data and in the “dog” class, there are also 12500 data.

Image for the class “Cat “:

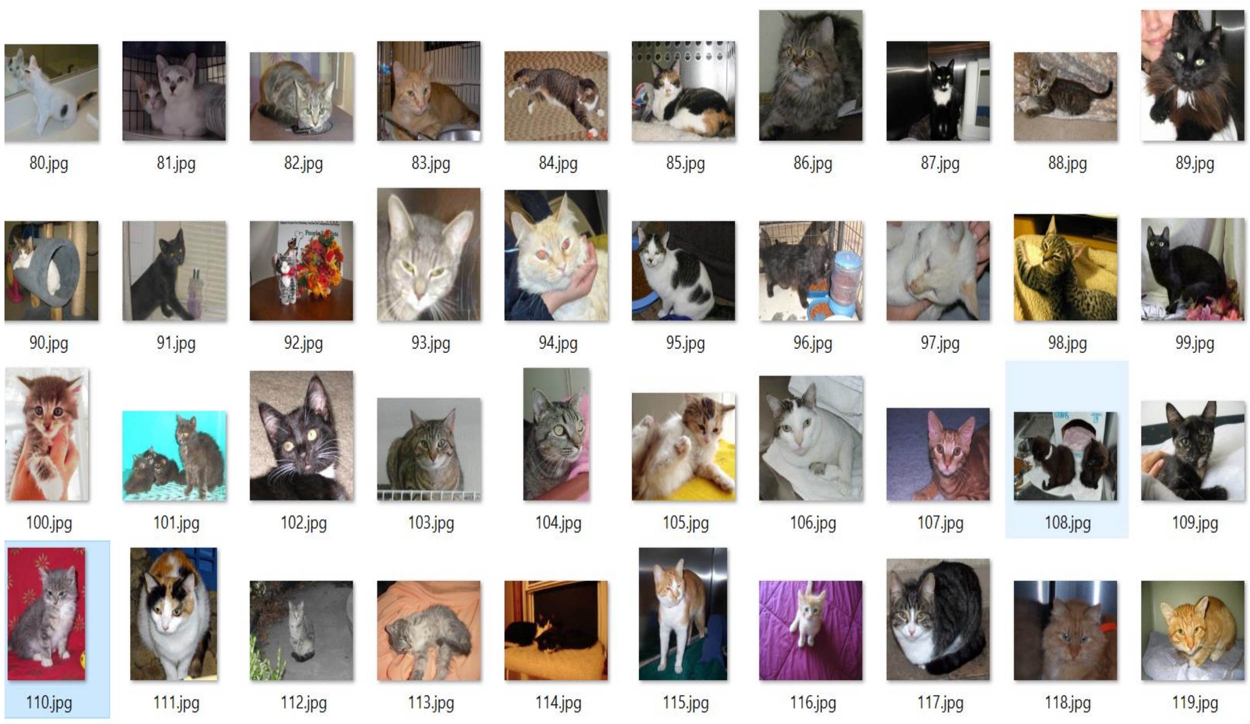
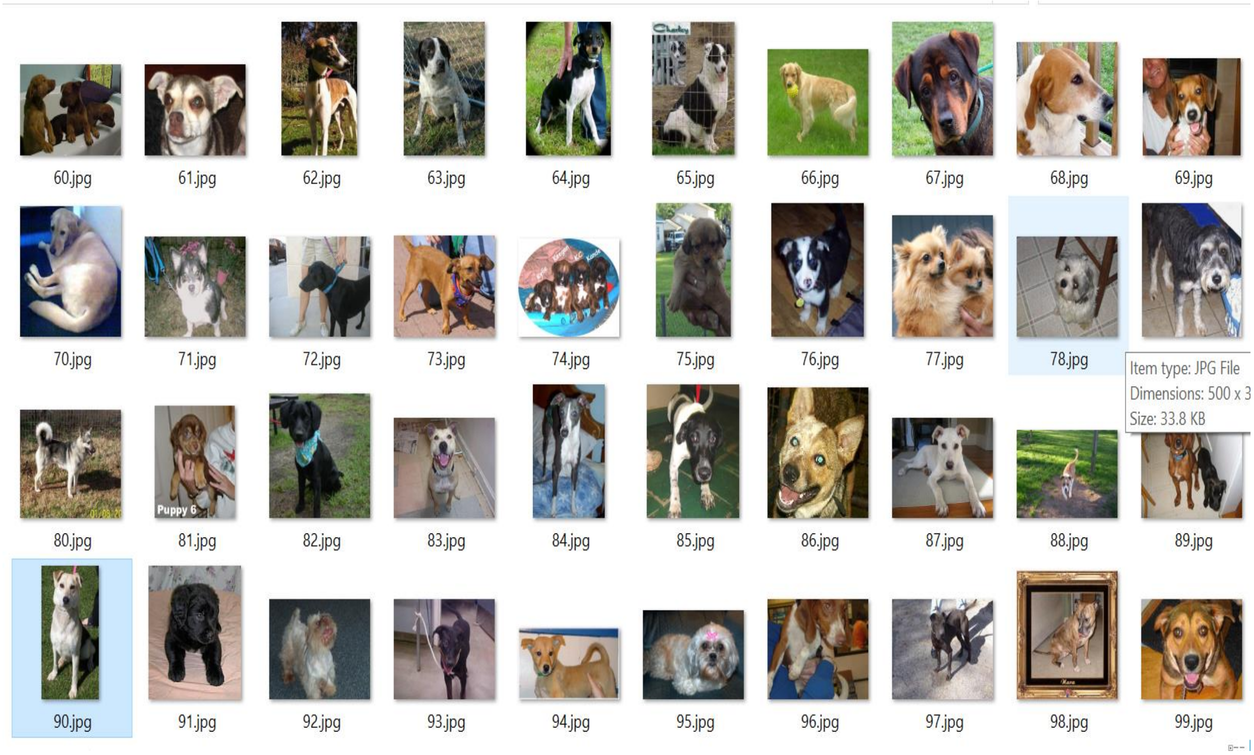


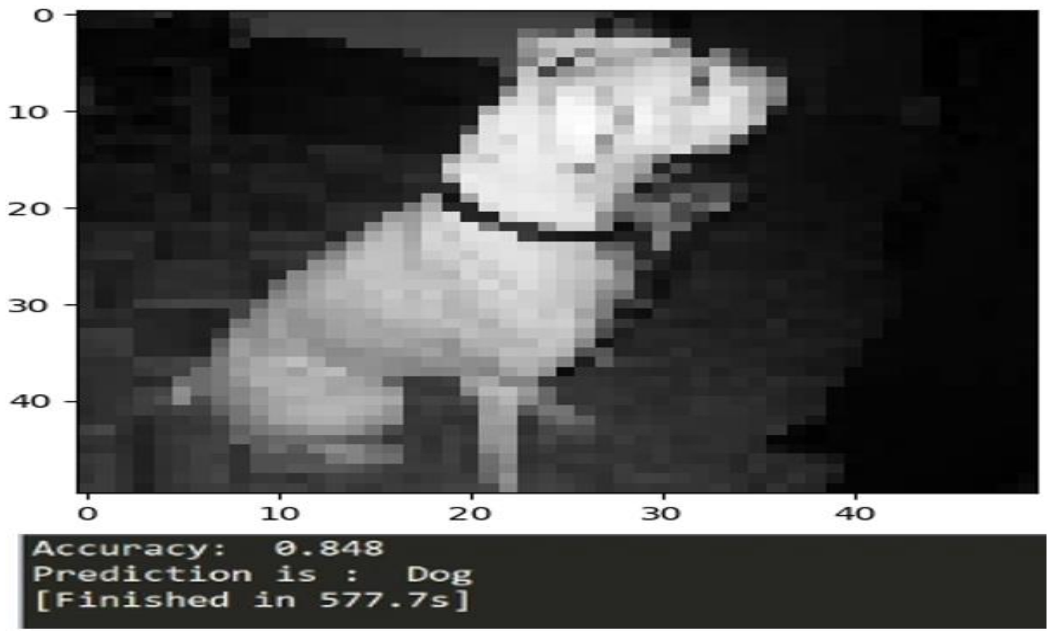
Image for the class "Dog":



5.2 Results:

Input: Here is an example input of data image.

Output: Here is the output image for the input image.



6. Conclusion:

For human it is very easy to differentiate between cats and dogs. But for computers it is not that easy. We have tried our best to make our model as much efficient as possible. As we can see above, our model learned to successfully identify both dogs and cats. Its predictions are very confident in the majority of the cases and they are lower only for anything that’s unusual like pose, fur color etc. It’s definitely expected behavior because our model has trained on less amount of such uncommon data. Possibilities of improvement in this field are boundless. So in the near future, we will be trying to build a much more balanced system.