# Animal Classification From Animal Face Image Using Convolutional Neural Network

Sohan,Md.Tanjil Alam (ID: 18-36915-1)[1]

[1]Department of Computer Science, American International University-Bangladesh(AIUB)

**Abstract**

To classify the Animal Faces-HQ (AFHQ) [**1**] dataset we used a customized Convolutional Neural Network(CNN) model inspired from AlexNet [**3**]. We removed two Convolution layer and reduced some neurons from fully connected dance layer of Alexnet CNN model to improve the accuracy of over Animal Faces-HQ (AFHQ). We used Tensorflow to build the CNN model. Animal Faces-HQ (AFHQ) dataset contains 16,130 images with 512x512 resolution and three categories of animal face image such as cat, dog, wild. We have customized the dataset to bring it to the usable format. We have got the 99.57% percent of train accuracy. After testing the data we had got 97.6% percent of test accuracy.

keyword: Computer Vision, Deep learning, Object detection, CNN Tensorflow, OpenCV

## 1 Introduction

Animal Face classification is, classify the the animal from animal face image of Animal Faces-HQ (AFHQ) dataset. We used this datset from Kaggle dataset [**2**]. The dataset contains three categories of animals, cat, dog and wild. Wild means wild animals such as tiger, foxes, leopard which faces looks familiar with cats and dogs. The dataset has 16,130 high quality images of 512x512 pixel size. The dataset contains tarining data folder and test data folder. There are 14,630 of training images and 1500 of test images. Our objective is beat the highest accuracy reported on Animal Faces-HQ (AFHQ) dataset on kaggle.com [**4**]. Object detection is a common area in Computer Vision and Pattern Recognition. There are a lot of research and algorithm based on object detection using Convolutional Neural Network. AlexNet [**3**] CNN model proposed in 2012 to classify the ImageNet [**6**] dataset which contains 14 millions of images and more than twenty thousand categories in total. They trained that large and deep CNN model to classify the high resolution images like ImageNet dataset. Using that model they had achieved top-1 and top-5 error rates for that time being. They had used dropout layer for reducing the over-fitting which is also developed by them. On September 30, 2012, AlexNet participated in the ImageNet Large Scale Visual Recognition Challenge [**5**].The network had a achievement of top-5 error rate of 15.3 percent for that competition.

## 2 Related Works

Face recognition is also a common area in Computer Vision and Pattern Recognition as like as object detection. There are considerable amount of work done over human face where deep learning as well as Convolutional Neural Network (CNN) is used [**7**] [**8**]. But there are a few work done over animal face recognition. We collected Animal Faces-HQ dataset from kaggle.com. In there not more than 23 work had been submitted on that dataset [**9**]. Among those work contestants reached up to 99.4% of accuracy. There are a contestant who achieved 98.5% of accuracy using ResNet50 [**10**] CNN model.

## 3 Our Proposed Model

We have used a customized Colvloutional Neural Network Model which is inspired form AlexNet model. CNN Model layers description is given below that we used for the classification of Animal Faces-HQ dataset:

- **Input Layer:** First we used the input layer, with input shape = (120,120,3). As we have seen that in our dataset every images with resolution of 512x512. We resized our data which contains pixel size of 120x120, for our convenience.

- **Convolution Layer:** We used three convolutional layers serially one after another.First convolutional layer contains 96 such filters with kernel size (3x3),strides 2 and padding (valid). Second convolutional layer contains 384 such filters with, kernel size (3x3), stirdes 1 and padding (valid).In last convolutional layer we have used 256 such filters with kernerl size (3x3), strides 1 and padding(valid). We have used the same number of filters as AlexNet in our convolutional layers. They used five convolutional layers in AlexNet and we used three which had made our model different form AlexNet.

- **Activation Function:** We have used Relu activation function all over our model except output layer. In output dance layer we used SoftMax activation function. Relu becomes the default activation layer for a lot of Convloutional Neural Network model.

- **MaxPool:** We have used maxpool layer after two of our convolutional layer which contains 96 and 256 such filters. Every maxpool layer contains pool size of (2x2), strides 2 and padding (valid).

- **Flatten Layer:** We have used a flatten layer before using dance layer to keep the consistency between convolutional layers and and dance layers.

- **Dense Layers :** Four(4) dense layers has been used after flatten layer. First dense layer contains 1024 unites of filters, second dense layer contains 512 unites of filters, third dense layer contains 256 unites of filters. Fourth dense layer also called output layer contains 3 (three) unites, since we have three categories of animals in our dataset which means 3 outputs as we expected.

- **Dropout Layer:** Dropout layer is used after three of our dense layer except output layer from the model to reduce the over-fitting.
- **Batch Normalization:** Batch normalization is used after every convolutional layers and dense layer except output layer so that the network can learn independently. There are more reason for using batch normalization such as increase learning efficiency, overfitting.

## AlexNet Architecture:

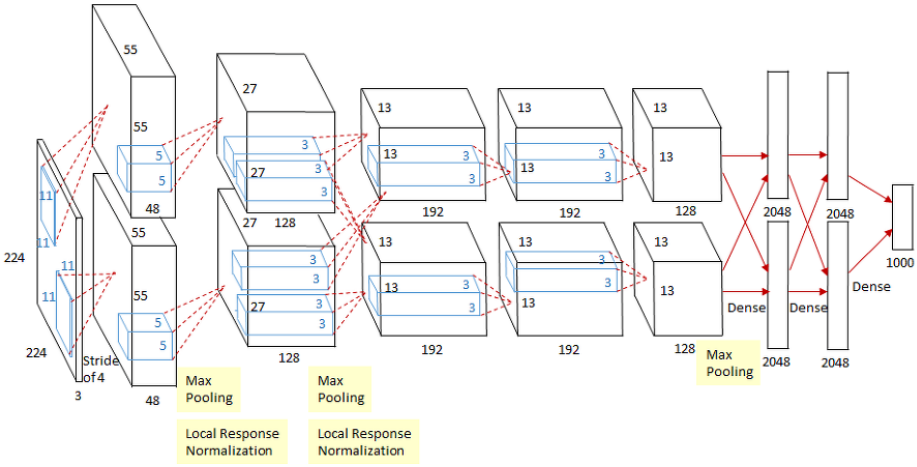


Figure 1: AlexNet Architecture.

## List of All layers in Alexnet:

| | Layer | Input Shape | No of Neurons | Kernel Size | Stride | Padding | Output Shape | Activation |
|---|---|---|---|---|---|---|---|---|
| 0 | Input | (227,227,3) | -- | -- | -- | -- | (227,227,3) | -- |
| 1 | Convolution 1 | (227,227,3) | 96 | 11 x 11 | 4 | -- | (55,55,96) | ReLU |
| | Max Pooling 1 | (55,55,96) | -- | 3 x 3 | 2 | -- | (27,27,96) | -- |
| 2 | Convolution 2 | (27,27,96) | 256 | 5 x 5 | 1 | 2 | (27,27,256) | ReLU |
| | Max Pooling 2 | (27,27,256) | -- | 3 x 3 | 2 | -- | (13,13,256) | -- |
| 3 | Convolution 3 | (13,13,256) | 384 | 3 x 3 | 1 | 1 | (13,13,384) | ReLU |
| 4 | Convolution 4 | (13,13,384) | 384 | 3 x 3 | 1 | 1 | (13,13,384) | ReLU |
| 5 | Convolution 5 | (13,13,384) | 256 | 3 x 3 | 1 | 1 | (13,13,256) | ReLU |
| | Max Pooling 3 | (13,13,256) | -- | 3 x 3 | 2 | -- | (6,6,256) | -- |
| 6 | Fully Connected 1 | 9216 | | | | | 4096 | RelU |
| 7 | Fully Connected 2 | 4096 | | | | | 4096 | ReLU |
| 8 | Fully Connected 3 | 4096 | | | | | 1000 | SoftMax |

Figure 2: Layer list.

# 4 Results and Discussions

We have used our train data from Animal Face-HQ dataset to train the model using python "fit" method with 100 epochs and batch size 64.

## Results:

After training the model with 100 epochs we have got the train accuracy of 99.72% with validaton accuracy 94.67%

**Training Accuracy and Validation Accuracy is shown below:**

```
plt.plot(h.history['accuracy'],'r',label='training accuracy')
plt.plot(h.history['val_accuracy'],label='validation accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

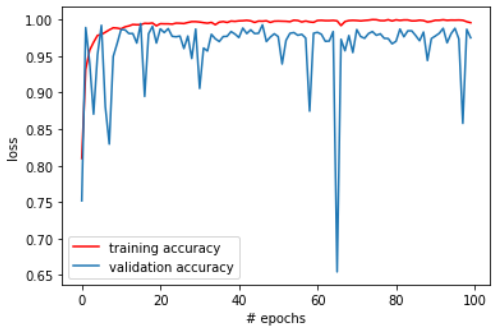


Figure 3: Train Accuracy Vs Validation Accuracy Graph

**Train Loss and Validation Loss is shown below:**

```
[ ]  plt.plot(h.history['loss'],'r',label='training loss')
     plt.plot(h.history['val_loss'],label='validation loss')
     plt.xlabel('# epochs')
     plt.ylabel('loss')
     plt.legend()
     plt.show()
```

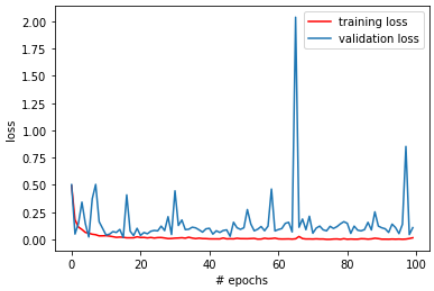


Figure 4: Train Loss Vs Validation Validation loss Graph

**We have Achieved 97.6% of Testing Accuracy using Test Data of Animal Faces-HQ Dataset:**

```
[ ]  test_loss, test_acc = model.evaluate(X_test, Y_test)
     print('\nTest Accuracy:', test_acc)
     print('\nTest Loss:', test_loss)

     47/47 [==============================] - 1s 30ms/step - loss: 0.0948 - accuracy: 0.9760

     Test Accuracy: 0.9760000109672546

     Test Loss: 0.09480511397123337
```

Figure 5: Test Accuracy:

### 4.1 Discussion:

In Animal Faces-HQ dataset we had got images with resolution of 512x512. Firstly er had read the data using openCv and then resized both train data and test data into 120x120 pixel size from 512x512 for bring all the images into same size. Then we converted the data into numPY array to bring the data into usable format. Before training the data we prepossessed the data by dividing both training data and test data by 255. We have used this particular prepossessing technique to improve the accuracy.

## 5 Conclusion

In kaggle.com the highest accuracy of 99.4% percent is reported for Animal Faces-HQ (AFHQ) dataset. We have achieved the test accuracy of 97.6%.We were so close. We have tried our best to beat it. We could have achieved more accuracy if we had used one of the models that invented earlier such as ResNet-50 e.t.c.

## References


[1] choi2020starganv2, title=StarGAN v2: Diverse Image Synthesis for Multiple Domains, author=Yunjey Choi and Youngjung Uh and Jaejun Yoo and Jung-Woo Ha, booktitle=Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, year=2020

[2] Choi, Y., Uh, Y., Yoo, J. and Ha, J., 2021. Animal Faces. [online] Kaggle.com. Available at: ¡https://www.kaggle.com/andrewmvd/animal-faces¿ [Accessed 14 December 2021].

[3] krizhevsky2012imagenet, title=Imagenet classification with deep convolutional neural networks, author=Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, journal=Advances in neural information processing systems, volume=25, pages=1097–1105, year=2012

[4] Kaggle.com. 2021. Kaggle: Your Machine Learning and Data Science Community. [online] Available at: ¡https://www.kaggle.com/¿ [Accessed 14 December 2021].

[5] image-net.org/. 2021. ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). [online] Available at: ¡https://image-net.org/challenges/LSVRC/2012/¿ [Accessed 14 December 2021].

[6] https://image-net.org/. 2021. ImageNet dataset Home page. [online] Available at: ¡https://image-net.org/¿ [Accessed 14 December 2021].

[7] abdalmageed2016face, title=Face recognition using deep multi-pose representations, author=AbdAlmageed, Wael and Wu, Yue and Rawls, Stephen and Harel, Shai and Hassner, Tal and Masi, Iacopo and Choi, Jongmoo and Lekust, Jatuporn and Kim, Jungyeon and Natarajan, Prem and others, booktitle=2016 IEEE winter conference on applications of computer vision (WACV), pages=1–9, year=2016, organization=IEEE

[8] khalajzadeh2014face, title=Face recognition using convolutional neural network and simple logistic classifier, author=Khalajzadeh, Hurieh and Mansouri, Mohammad and Teshnehlab, Mohammad, booktitle=Soft Computing in Industrial Applications, pages=197–207, year=2014, publisher=Springer

[9] Kaggle.com. 2021. Animal Faces. [online] Available at: ¡https://www.kaggle.com/andrewmvd/animal-faces/code¿ [Accessed 14 December 2021].

[10] he2016deep, title=Deep residual learning for image recognition, author=He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, booktitle=Proceedings of the IEEE conference on computer vision and pattern recognition, pages=770–778, year=2016