Animal Recognition System Based On Convolutional Neural Network

Rajan Pandey [a], Dilip Paudel [b], Sailesh Sapkota [c], Diwash Bhandari [d], Raju Shrestha [e]

[a, b ,c ,d, e] *Department of Electronics and Computer Engineering, National College of Engineering, Lalitpur, Nepal*

[a] 12pnrj@gmail.com, [b] pdldilip00@gmail.com, [c] sapkotasailes@gmail.com,[d] diwashbhandhari99@gmail.com [e] raju@nce.edu.np

***Abstract*—Efficient and accurate object recognition has been an important topic in the field of computer vision systems. Object recognition is concerns with determining the identity of an object being observed in the image from a set of known labels. With the advent of deep learning with neural network techniques, the accuracy for object detection has increased drastically. Animal recognition system is implemented using Convolution Neural Network (CNN) for classification. CNN model of the system is based on our own architecture. Kaggle animal-10 and monkey species dataset with 10 animals as cat, chicken , cow , dog, elephant, horse , monkey ,sheep, spider, squirrel are used in this system. The system achieved 77.2 % accuracy during testing.**

Keywords—Computer Vision, deep learning, convolutional neural network, classification

I. INTRODUCTION

Animal recognition is one of the major and important fields in computer vision. Animal recognition is concerns with determining the identity of an animal being observed in the image from a set of known labels. Generally, the animal recognition algorithms implement a binary pattern classification task[1]. That means, given an input image is divided in blocks and each block is transformed into a feature. Features from the animal that belongs to a certain class are used to train a certain classifier. Then, when given a new input image, the classifier will be able to decide if the sample belongs to certain class or not.

This system uses Deep Learning techniques with convolutional neural network [2], and large labeled datasets to address the problem of automated animal identification in the images. A convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. They have wide applications in image and video recognition, recommendation systems and natural language processing. It take in an input image, assign learnable weights and biases to various aspects/objects in the image and will be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms in terms of features sizes. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

II RELATED WORK

One of the earliest works on automatic animal detection [4] uses Haar-like features and a low-level feature tracker to detect a lion's face and extract information to predict its activity like still, walking or trotting. The system works in real time and is able to detect faces at multiple scales although only with slight pose variations.

Zhang et al. [5] detect heads of animals like tiger, cat, dog, cheetah, etc. by using shape and texture features to improve image retrieval. The approach relies on prominent 'pointed' ear shapes in frontal poses which makes it sensitive to head-pose variations. These approaches rely on identifying different parts of the animal to detect and track an individual but are likely to fail in case of occlusion or significant pose change.

Hung Nguyen et al. (DSAA, 2017) [6] attempted to automate the wild-life monitoring by using CNN, but on a much smaller scale and with only 3 classes. They used only 55 000 image for training and achieve a 90.4% accuracy.

Gyanendra K. Verma and Pragya Gupta [7]: In this paper, wildlife monitoring and analysis through animal detection. The images obtained from camera-trap consist of highly cluttered images that hinder the detection of animal resulting in low-detection rates. In this detection model using DCNN features provides accuracy of 91.4% on standard camera-trap dataset that contains 20 species of animals with around 1000 image sequence for each species.

III. ANIMAL RECOGNITION SYSTEM

Animal Recognition algorithm (image classifier) is deep learning model which takes images (or a batch of the image)

from user, process it and gives numeric values (probability). At last, it gives output what the image contains using features that it have learned during training phase. In other words, the output is a class label (cat , cow , horse etc.).

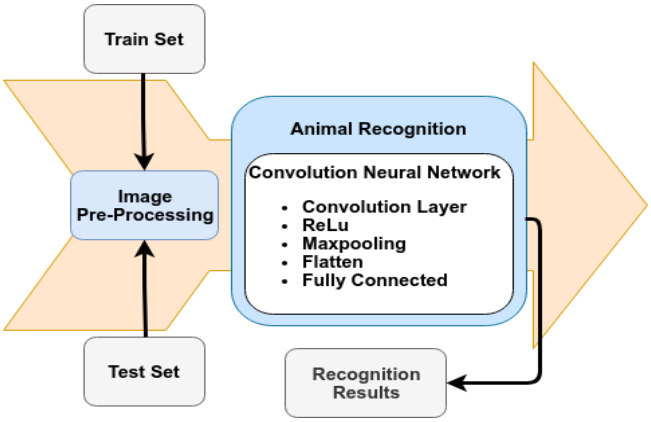

Fig 1 : The animal recognition system.

In fig 1, training images and test images are passed to pre-processing module. Pre-processed matrix is then passed in convolution Neural Network architecture where operations such as convolution, maxpooling, flatten happens. After flattening, all training data are passed in fully connected layer where classification and optimization was done. At last, all weights and biases were saved into a .h5 files to predict animals later. But in testing phase, test images were used to test the system with same architecture.

A) Dataset Selection

Since Convolution Neural network is Supervised Machine Learning technique, a large amount of labeled data (images) are needed for the purpose of training. The datasets were obtained from Kaggle and WSID-100[5] repository. From kaggle repository, Animal-10 dataset[3] and Monkey species[4] were used and rest of the data were taken from WSID-100 having similar feature as Kaggle dataset. All together 2000 images were taken for each class. For testing, 200 images were taken. Thus, the dataset classes consist of elegant, chicken, cow, sheep, dog, cat, horse, monkey, squirrel and spider with 2000 images per class and a total of 20000 images were taken.

B) Pre-processing Modules

Image pre-processing is the method to enhance image or extract some useful information. For pre-processing images, pre-processing module was used. In pre-processing module, images were read from directory, 3D images were converted into grayscale by using weighted or luminosity method/ algorithm as it is easy to compute and system learn from geometry, images were reshaped into 200 * 200 sizes using LINEAR Interpolation and each image was normalized by divided with maximum channel values i.e. 255 to convert values into float32.

C) Convolution Neural Network Layer

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep learning, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. CNN compares any image piece by piece and the pieces that it looks for in an image while detection is called as features [2].

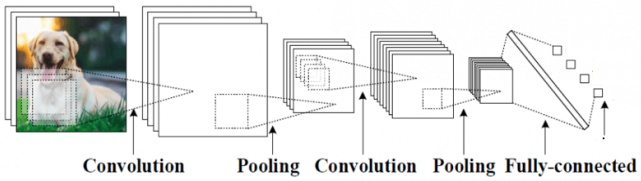

Fig 2: Examples of Convolution Neural Network

There are five main operations in the CNN:

a) Convolution.

b) ReLU.

c) Pooling or Sub-Sampling.

d) Flatten

e) Fully Connected Layer (classification)

a) Convolution layer

The primary purpose of Convolution in case of a CNN is to extract features from the input image. Each convolution layer takes image as a batch input of four dimension N x Color-Channel x width x height. Kernels or filters are also four dimensional (Number of feature maps in, number of feature maps out, filter width and filter height) which are set of learnable parameters (weights and biases). In each convolution layer, four dimensional convolution is calculate between image batch and feature maps by dot product between them. After convolution only parameter that changes are image width and height.

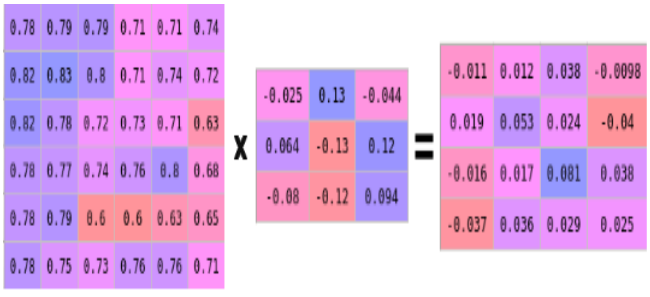

Fig 3: Convolution applied to input image

b) Rectified Linear Unit

An additional operation called ReLU has been used after every Convolution operation. A Rectified Linear Unit (ReLU) is a cell of a neural network which uses the following activation function to calculate its output given x:

$$R(x) = Max(0,x) \quad (1)$$

where x is pixel value.

Using these cells is more efficient than sigmoid and still forwards more information compared to binary units. It simply removes all negative values that comes from convolution layers by comparing with its function and replace negative with zeros.

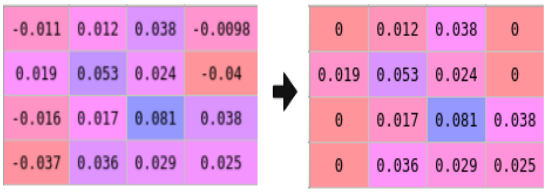

Fig 4: Rectification applied to Feature Maps

c) Pooling Layer

In this layer, the dimensionality of the feature map reduces to get shrink maps that would reduce the parameters and computations. Pooling can be Max, Average or Sum. Number of filters in convolution layer is same as the number of output maps from pooling. Pooling takes input from rectified feature maps and then downsized it according to algorithm.

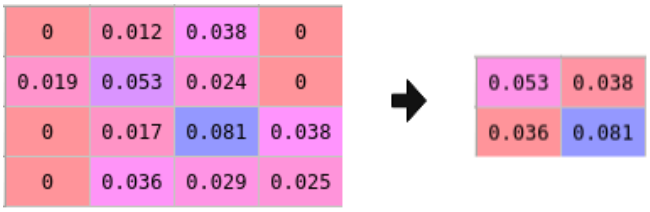

Fig 5: Max Pooling applied to Rectified Feature Maps

d) Flatten layer

Once the pooled featured map is obtained, entire pooled feature map matrix is transform into a single column which is then fed to the neural network for classification.

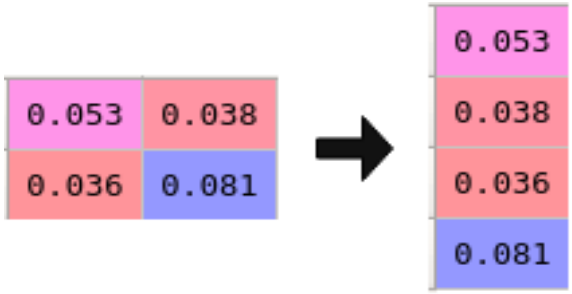

Fig 6 : Flattening applied to pooled Feature Maps

e) Fully Connected Layer

This is the final layer where the actual classification occurs where this layer takes downsized or shrink feature maps obtained after convolution, ReLU and pooling layer and flatten. It is a traditional Multi-layer perceptron which uses a softmax activation function. Convolution and pooling layers generate high-level features. The purpose of the fully connected layer is to use these features to classify into various classes based on labels.

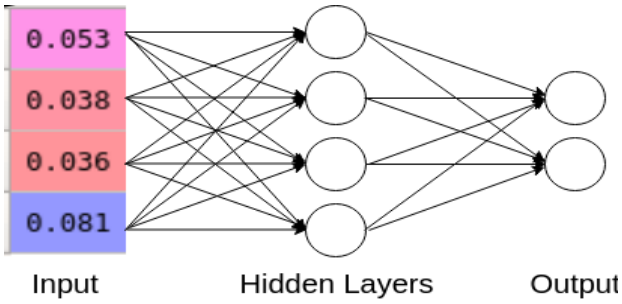

Fig 7: Fully Connected layer

D) Training of CNN using Back-Propagation

All filters and parameters are initialized with random values. The input images are taken for training and are propagated through the forward propagation and output probability is obtained for each class using softmax function. Then total error is calculated by categorical cross entropy [7] formula.

$$CE:=-\sum (y'i\log(yi)+(1-y'i)\log(1-yi))/n \quad \ldots\ldots.(2)$$

where CE is cost errors, y is actual output, y` is predicted output in terms of probability, n is total numbers of inputs.

The gradient of the error is calculated with respect to the weights and then RMSprop algorithm is used to update the filter values and parameters to minimize the output error. The RMSprop optimizer formula are:

$$V_{dw} = \beta * V_{dw} + (1- \beta) * dw^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

$$V_{bw} = \beta * V_{bw} + (1- \beta) * db^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

$$W = W - \alpha * dw / (\sqrt{} V_{dw}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

$$b = b - \alpha * db / (\sqrt{} V_{bw}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

where,

V_{dw} is small change in weight,

V_{bw} is small change in biases,

β is momentum (0.9),

α is learning rate(0.0005),

dw is gradients error of weight obtained by partial derivatives with total error,

db is gradients error of biases obtained by partial derivatives with total error,

W is value of weight,

b is value of bias

Repeat above steps for all images in the training set.

E) CNN Architecture

Thus convolutional neural network was developed using Keras and was trained with datasets. The pre-trained model for initializing the network was not used.

In the proposed CNN (see Fig. 8), input image contains 200 *200 *1 and was passed to convolution layers where training image batch size was taken as 150, while filter map was of size 64x3x3 for each convolution layer. Then convolutional layer was followed by Pooling Layer. The pooling operation was applied separately to each feature map. In general, the more convolutional steps we have the more complex features (such as edges) is possible to recognize using proposed network. The whole process is repeated in successive layers until the system can reliably recognize objects. For example, in image classification a CNN may learn to detect edges from raw pixels in the first layer and then use the edges to detect simple shapes in the second layer. Then use these shapes to determine higher-level features, such as body shapes in higher layers. The architecture of the Convolutional Neural Network (CNN) is shown in Fig 8.

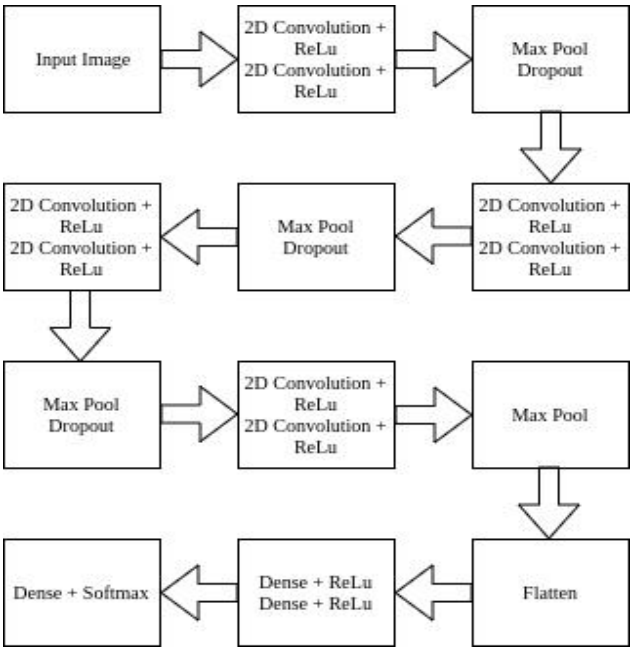

Fig 8 : Block diagram of the proposed CNN.

IV. RESULT

For each iteration, batch of dataset was passed through the neural network. The training process updated the weights of feature maps and hidden layers based on hyper-parameters such as learning rate, momentum, regularization and decay. In this system batch-wise learning rate was used as 0.0005, momentum as 0.99. The accuracy gradually increased according to the iteration giving 94.45% final training accuracy and test accuracy of 77.67 %.

The comparisons of validation cost, validation accuracy, training cost and training accuracy while training are shown in Figure.

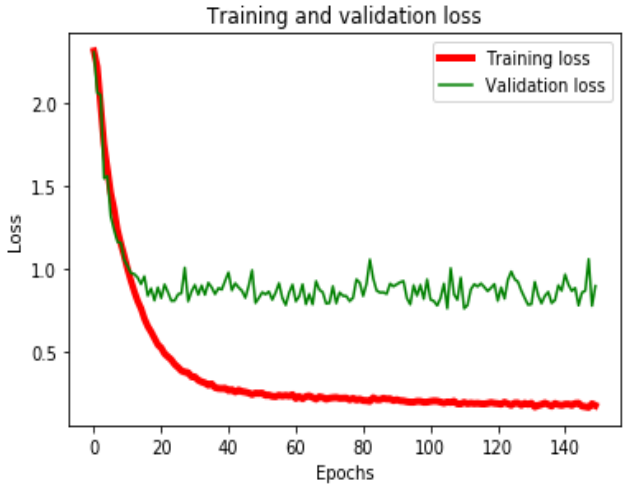

Fig 9: Graph between Training and Validation Accuracy

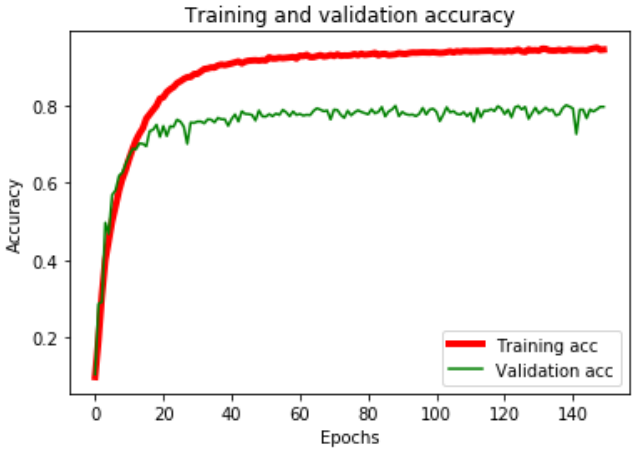

Fig 10: Graph between Training and Validation loss

During testing, the random image of the animal that belongs to one of the class of dataset was used to test with animal recognition system. We have found that small misalignment in the image, caused by the many challenges of these images (occlusions, lighting, blur, etc) can have a

noticeable impact on the result. The confusion matrix for ten animal's classes is shown below:

TABLE I. CONFUSION MATRIX FOR ANIMALS RECOGNITION

Index	cat	dog	monkey	cow	elephant	horse	squirrel	chicken	spider	sheep	classified prob.
cat	126	37	7	0	1	0	18	7	4	0	0.63
dog	18	133	5	10	6	12	9	7	4	3	0.643
monkey	2	6	173	0	7	0	10	0	0	2	0.865
cow	2	4	5	158	6	4	2	1	2	4	0.84
elephant	1	4	3	2	186	0	2	0	0	2	0.93
horse	2	4	1	5	9	164	7	2	3	1	0.828
squirrel	2	4	12	0	0	0	85	2	1	3	0.78
chicken	4	2	3	6	0	0	18	159	8	0	0.795
spider	6	2	0	0	1	0	7	2	180	2	0.9
sheep	4	6	4	12	5	3	7	5	1	152	0.764

V. CONCLUSION

In this system, architecture based on seven layer convolution neural network is implemented to classify animals recognition such as cat, chicken, cow, dog, elephant, horse, monkey, sheep, spider and squirrel. Our model has training loss 0.1788, training accuracy 0.9445, and validation loss 0.8973 and validation accuracy 0.7767 after trained for 150 iterations. Model performance seems weaker across negative images (out of classes) on average.

In the future work, this system plan to perform experiments and also test on more complex architecture such as AlexNet[10] or VGGNet [11] for animals recognition. The model can be extended to color images. Future works can also include experiments with this method on other animal classes.

ACKNOWLEDGMENT

The authors would like to acknowledge the support provided by National College of Engineering. They would also like to extend appreciation to the reviewers for their valuable positive suggestions.

REFERENCES

[1] XIE, Z., A. SINGH, J. UANG, K. S. NARAYAN and P. ABBEEL. Multimodal blending for high-accuracy instance recognition. In: 2013 IEEE/RSJ International Conference on Intel-ligent Robots and Systems. Tokyo:IEEE,2013, pp. 2214–2221. ISBN 978-1-4673-6356-3.DOI: 10.1109/IROS.2013.6696666.

[2] "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.

[3] https://www.kaggle.com/alessiocorrado99/animals10

[4] https://www.kaggle.com/slothkong/10-monkey-species

[5] drive.google.com/drive/folders0B7dS7AFpUzt1b0UzeDVhWVg0Y2s

[6] EITEL, A., J. T. SPRINGENBERG, L. D. SPINELLO, M. RIEDMILLER and W. BURGARD. Multimodal Deep Learning for Robust RGB-D Object Recognition. In: 2015

[7] IEEE/RSJ, International Conference on Intelligent Robots and Systems (IROS). Hamburg:IEEE, 2015, pp. 681–687. ISBN978-1-4799-9994-1.DOI:10.1109/IROS.2015.7353446

[8] Verma, Gyanendra & Gupta, Pragya. (2018). Wild Animal Detection Using Deep Convolutional Neural Network. 10.1007/978-981-10-7898-9_27.

[9] Kamdem Teto, Joel and Xie, Ying, "AUTOMATIC IDENTIFICATION OFANIMALS IN THE WILD: A COMPARATIVE STUDY BETWEEN C-CAPSULENETWORKS AND (2018).Master of DEEP Science CONVOLUTIONAL in Computer

[10] https://digitalcommons.kennesaw.edu/cs_etd/2027NEURALScience NETWORKS."

[11] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. Computing Research Repository, abs/1311.2901, 2013.

[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[13] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv: 1409.1556(2014).