

Comments from Group41

- Although they mention feature extraction and different classifiers in literature review, they did NOT implement any of these which is a big deficiency of their work. At least I wanted to see one classifier such as SVM and its performance over CNN models like we did in our project.

Thank you, we also think testing the classification task on SVM is important. However, we only have limited time and we already spent a lot of time training and comparing different CNN models. We applied data augmentation and we also test the effect of pretrained model. If we have more time, we will try this task on SVM.

- Results presentation is really vague, it is really hard to understand which part to focus on.

Thank you, we will improve and correct these things in our final report.

- What is the reasoning behind providing a number of parameters, it is really confusing.

For machine learning framework, the ability of reproducing experiment results are extremely important. We think that providing the details of our model parameters is reasonable.

- Overall, it seems very trivial to implement three CNN models as a class project.

As mentioned previously, we spent a lot of time training and comparing different CNN models. We applied data augmentation and we also test the effect of pretrained model. We also discuss our classification results based on top 1 accuracy and top 3 accuracy. We think it is not suitable to consider the effort trivial.

- Accuracy levels seem not high (all of them are less than 80%)

The dataset is not easy since the variation of the images of same class are very large. Some of them are taken from night vision camera and telescopes. Also, many species look similar to each other. For example, these two images are different species (class 4252 and class 4255).



These may be the reasons why the accuracy is not high.

- Why did you specifically select these three models? - It is a plus that they implement different data augmentation methods cause it fits their context.

ResNet, Inception and VGG are very popular CNN models. Most classification models are modified based on their architecture and concepts. Moreover, they have been trained on ImageNet and obtain good results. Choosing three models allows us to apply pretrained weight, which is very important to enhance the classification results.

Comments from Group48

We wish you would have given some more detail on your data. You mentioned how many classes and images, but did not say how many images per class. Was your data consistent? How did this affect your results?

Overall, the number of images of the 235 mammal species are not very balanced. Moreover, the variation of the images of same class is large and this makes the classification results worse than 80%.

We also really liked your comparison of images, discussing orientation, number of animals per image, distance, nighttime images, telescope images, etc. It gave us a deeper understanding of the difficulty of your project, and we recognize that you put a lot of thought into all aspects of your data. While we think all of this information is crucial for understanding your work, we wish you spent a little less time discussing the data and a little more time discussing your models. Also you mentioned that no one would take a picture of a mammal upside down. What about bats? Was that a mammal in your dataset? We also would have liked to know some examples of the mammals in your dataset.

Thanks for your compliment! We are flattered. If there are bats in our training set, some bats may rest upside down but some may fly without flipped. Our model still can learn this thing but it may take longer time and get worse results. However, this is very hard to prevent before we check all species one by one. As a result, we just decide not to vertically flip the pictures. Thank your interesting question.

We are unclear as to why you picked these 3 models, other than you saying they are the most popular. What are they most popular for? Why are they popular? At one point you mentioned models that are too deep are difficult to optimize. Then when you were describing that you chose ResNet-50 due to computational resources but would have liked to use ResNet-152, why would you want to do this if more layers may be difficult? You stated that you used sgd optimizer, but gave no explanation as to why.

Why these models and why popular: ResNet, Inception and VGG are powerful CNN models. Most classification models are modified based on their architecture and concepts. The skip connection and multi-scale features have been widely used in CNN models. Moreover, they have been trained on ImageNet and obtain good results. Choosing three models allows us to apply pretrained weight, which is very important to enhance the classification results.

Although more layers may be difficult, ResNet-152 has skip connections to avoid gradient vanishing. It may yield better results than ResNet-50.

What optimizer: we use SGD, but it's fine to use other optimization like Adam as long as the learning rate is set properly, and the results are similar. We do not find anything interesting and special so we didn't mention that.

Also why do you think that the adam optimizer gave similar results? Understanding how these optimizers behave with your data could be very interesting. Were these the only optimizers that you tried? Why did you try these specifically?

It is fine to use other optimization like Adam as long as the learning rate is set properly, and the results are similar as long as the loss converges. We do not find anything interesting and special so we didn't mention that.

We were confused with your results, especially what Prec@1 and Prec@3 means. We thought your explanation of this was unclear in the slides and in the code, as you just stated it was top 1 accuracy and top 3 accuracy. What does top 3 accuracy mean?

Sorry for being unclear. Prec@3 is the top 3 accuracy, it means our model predicts the top 3 possible species and one of them is the correct one. Prec@1 the top 1 accuracy, we predict one species and this is exactly the correct one.

Aside from that, we really liked that you have your results in a table format; it made it very easy for us to see the differences in your results. We also liked that you explained the results, specifically saying why the pre-trained models achieved better results than the non pre-trained models.

Thank you very much!

We thought that your coding demo was very good. We appreciated that you explained many lines of your code, mentioned image size, epochs, and learning rates. It helped us understand your process, from gathering data, processing, and training. We wish you would have shown us your models in the code though, such as Inception. You also mentioned that you used pretrained models from the pytorch website. Were these models pre-trained for your specific dataset? You could have discussed this a bit more in your slideshow presentation before comparing model results.

Overall, we really enjoyed your presentation and thought you did a great job.

Thank you very much! The models are pretrained on ImageNet and are not specific for our dataset.

Comments from Group84

The talk completely introduced the details of the dataset, the data augmentation method and the three basenets they used in the deep learning model. This dataset was impressive with many complicated pictures for classification, which made this project more outstanding.

Thank you for your compliment!

Some improvements/unclear:

1.Vgg16 not detail, why can help

Vgg16 is one of the powerful CNN models, and the skip connection and multi-scale features have been widely used in CNN models. Moreover, it has been trained on ImageNet and obtain good results. Choosing Vgg16 allows us to apply pretrained weight, which is very important to enhance the classification results.

2.Having only a table in the observation part is not straightforward. Graphs showing the varying trend of accuracy and loss on epochs are always important in presentation which could help us find more details about the effects of these models.

Thanks for your feedback. Although it would be more interesting to show the graphs. Having the outcome in a table format makes it clear enough to presents our results because we only have three models to present.

3.The demo part is not straightforward either. Only showing the codes and the training process are not enough. An end-to-end like demo would make the presentation more intuitive such as importing several pictures of mammals and then showing several prediction results of them.

Thanks for your feedback. Although it would make the presentation more intuitive by showing some images for the prediction results, we only have limited time and we already spent a lot of time training and comparing different CNN models. In addition, we also went through our code in details and explained image size, epochs, and learning rate etc.

4.The reference part could be a little more formal.

Thank you for your feedback!

5.Only comparing results among your own three models is not enough. The current best accuracy for solving the same problem is also important for letting us know the value of your work.

It would be a plus to mention the current best accuracy for solving the same problem. However, it is clear enough to show the value of our work by presenting the difficulty of training process, since the dataset is not simple and the variation of the images of same class are very large. And we also elaborated the importance of this problem at the beginning of our presentation.

6.I am still not clear about why the visualizing the features with tSNE can help better solve this problem.

We can project data features into 2D or 3D space for qualitative visual observation by using t-SNE. In other words, t-SNE is a good way to visualize and intuitively verify the effectiveness of a dataset features.

ECE 228 final report

Yao-Cheng Yang A53305005, Ching-Yu Chen A53298967, Chun-Yen Liou A53316963

I. INTRODUCTION

There are about 5,500 mammal species identified in our planet, and they were grouped into 1,229 genera, 153 families and 29 orders. Therefore, it is quite difficult to accurately classify different mammals without the assistance of experts. Moreover, there are some species look alike which is even more difficult to classify them(See Fig.1).



Fig. 1: Two different species look alike

Therefore, our project goal is to build a model to help general public classify different mammal species. And we aim at classifying these similar species by learning from a huge dataset collected and labeled by experts. By doing this, non-experts can predict the species in their photos using the trained model. The input to our algorithm are images of mammal. We then use a neural network to output a predicted mammal species.

II. RELATED WORK

We provide a brief review on previous related work focusing on feature extraction and choosing a desirable classifier.

In the aspect of feature extraction, Tilo and Janko [1] proposed a method for animal classification using facial features. Heydar et al. [2], developed an animal classification system using joint textural information.

After feature extraction, deciding a suitable classifier is crucial to the process of identifying different species. Matthias [3] used color features for animal classification with Support Vector Machine classifier. Deva et al. [4], also used K-way logistic regression and K-Nearest Neighbors to classify different species. Xiaoyuan et al. [5], used linear SVM classifier for the classification of animals. In their work, the multi-task joint sparse representation is an effective way to combine multiple complementary visual features and instances to improve the classification accuracy, and it performed well with SVM.

III. DATASET AND FEATURES

The dataset we used for this project is from iNaturalist Competition on 2018. And we only used the category of mammal specifically. There are 235 classes, each class represents

a specific species. And there are 20,806 images in total. We used 16644 images for training data. And For the validation data, we used 4,162 images. For data augmentation, we will go through the detail in the section of Methods.

In order to make our application closer to real life, there are several features and factors we took into consideration. For example, the scale of the animal is different because of different distance. In Fig.2, on the right hand side, we can see the animal looks much smaller than the left one because we observe it from a longer distance.



Fig. 2: Observing from different distance

Brightness and saturation should also be taken into consideration. For example, on the left hand side of Fig.2, the images may vignetting due to the use of telescope, which is quite common when observing them from a long distance. On the right hand side of Fig.3, we can also see that photos taken by night vision camera will have a totally different color than those taken in the day time.



Fig. 3: Observing from telescopes or night vision cameras

Shown in Fig.4, we can see that sometimes the camera doesn't just capture an animal, it may captures more than one animal of the same species at a time, which is also the important information for the training data.



Fig. 4: Different numbers of animal

IV. METHODS

A. Overview

We implemented three popular models, VGG16 [6], ResNet50 [7], Inception V3 [8], to train our mammal classification network. For the three models, we change the output dimension of the last fully-connected layer to the number of species in our dataset, which is 235. Note that the activation function of last layer is softmax, and the loss function is categorical cross-entropy. For better efficiency, we utilized pretrained model on ImageNet [9] at the beginning of training except for the last fully-connected layer. To prevent overfitting, data augmentation is applied. In the following paragraph, we will elaborate on each model architecture and discuss our model training strategies.

B. VGG16

VGG network [6] is simple and intuitive structure. It increases the model depth using an architecture with very small (3x3) convolution filters and down-samples the feature maps by 2x2 max-pooling. We adopt the architecture of VGG 16 without batch normalization, and the details of parameters is shown in Fig.5. After the stacks of convolutional layers, three Fully-Connected (FC) layers is applied. The first two have 4096 channels each and the third performs 235 way mammal classification and thus contains 235 channels. Although the model on have 16 layers, it contains 138 millions parameters, making this a major drawback of VGG models.

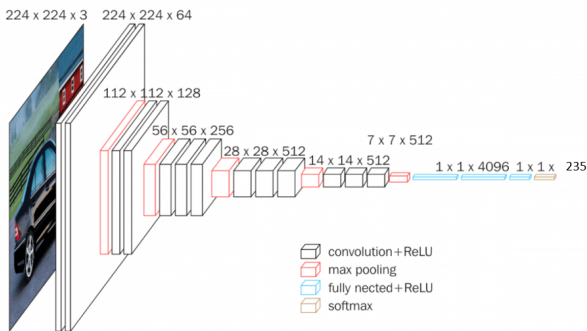


Fig. 5: Our model architecture for VGG16. Model figure reference: <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>.

C. ResNet50

In previous work, we found out that if we keep increasing the model depth using the VGG-like structure, that the ac-

curacy may start to decrease. The degradation is not caused by overfitting because training error is also high compared to the models with fewer layers. This shows that not all systems are easy to optimize. As a result, the authors of ResNet [7] design the residual block, shown in Fig.6, which utilizes the skip connections and the standard convolution layers. With this design, the layers only needs to learn the residual mapping instead of learning the direct mapping. This prevents the gradient vanishing problem, and allowing ResNet to have excellent performance on ImageNet dataset [9] with a depth up to 152 layers. In our work, we choose the ResNet50 instead of ResNet152 due to limited computing resources. Also, the number of images may not be sufficient to train the ResNet152 model. The parameters of ResNet50 is shown in Fig.7. It only contains 23.5 millions parameters, which is much fewer than VGG16.

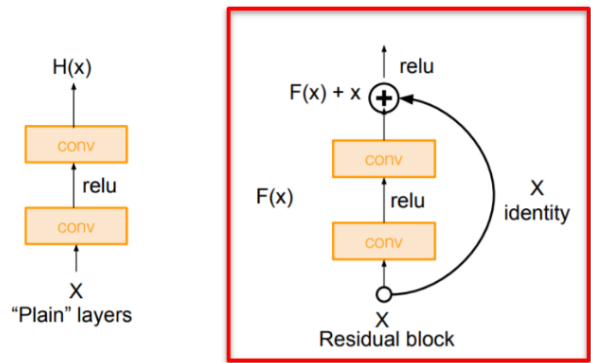


Fig. 6: Comparison between traditional convolution layers and residual layers [7].

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
		3x3 max pool, stride 2				
conv2.x	56x56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3.x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4.x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5.x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 7: Different design of ResNet [7]. We adopt ResNet 50 in our classification network.

D. Inception V3

Inception V3 [8] is an improved model of GoogLeNet [10]. The basic blocks for Inception module in Fig.8 contains filters of different sizes (1x1, 3x3). This makes the features obtained at each block contains multi-scale information. Moreover, 1x1 convolution is performed to reduce the number of dimensions, which can save more computational cost. Although the complete Inception V3 model architecture in Fig.9 is very complex, it only contains 23.8 million parameters, which is close to ResNet50 and much fewer than VGG16.

TABLE I: Mammal classification results using different model architecture. Prec@1 means top-1 accuracy and Prec@3 means top-3 accuracy.

	Pretrained	Validation acc Prec@1	Validation acc Prec@3	Training acc Prec@1	Training acc Prec@3
Inceptionv3	true	63.287	79.625	81.64	90.86
Inceptionv3	false	12.5	37.2	25.5	31.2
resnet50	true	64.344	79.409	80.28	90.5
resnet50	false	19.558	38.107	18.63	37.3
VGG16	true	47.405	67.011	57.72	76.0
VGG16	false	4.661	14.392	4.74	14.5

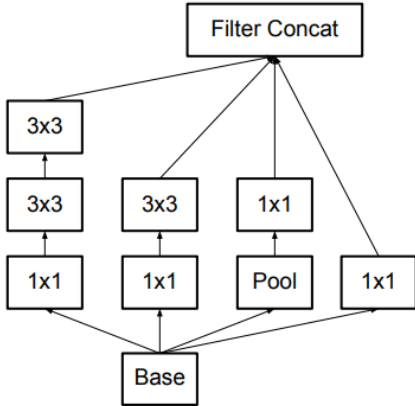


Fig. 8: Basic block of Inception V3 [8].

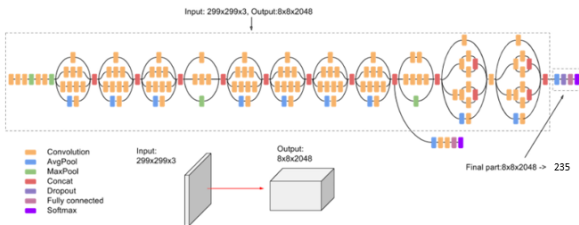


Fig. 9: Model architecture of Inception V3 [8].

E. Data augmentation

The training set is small (only 5GB), so good data augmentation is important in the training. We use three data augmentation: horizontal flip, resized flip and color jittering. I will discuss each one of them in the following. First, we use horizontal flip rather than vertical flip because the picture are never upside down and vertical flip would produce a new picture should not appear in the training set. Second, we use resized flip because the animal may appear in any part of the picture. Also, some photos are vignetting due to telescopes. As a result, resized flip would make the model more general for these cases. Third, we use color jittering to randomly change the contrast and brightness to make the model more general for image taken under different environments like some photos taken by night vision camera.

V. EXPERIMENT RESULTS

The classification results in shown in TableI. Since some species are very similar, we decide to record both top-1 accuracy and top-3 accuracy, which may be a better indicator

of the classification results on this task. First, we observed that top-3 accuracy is much better than top-1 accuracy in all cases, and is about 15 percent higher in pretrained ResNet50. This means that it is common that these models almost get the right prediction but make mistake on two or three very similar species. As a result, considering both top-1 and top-3 accuracy is a reasonable choice. Also, we observe that the training accuracy is higher than validation accuracy in all cases. This indicates that there is a certain extent of overfitting in our models although we perform data augmentation. For the comparison of the three models, the accuracy of Inception V3 model and ResNet50 is similarly high. However, VGG16 accuracy is low. We believe the reason is the difference of the number of parameter. VGG16 model has 138M parameter, but ResNet50 and Inception V3 have about 24M parameter and these are much less then that of VGG16 model. The parameter are too many in VGG16 model and the training set is too small to train the model. We may have some gradient vanish problem because there are no skip connection in VGG16 mode and the model is too deep. As a result, we think that adding skip connections (ResNet and Inception V3), extracting multi-scale features (Inception V3) is an effective strategy in mammal classification problems. Moreover, the results in TableI shows that all the three models fail to converge when not initialized with the pretrained weight on ImageNet. We come up with several possible explanations: (i) The number of training images is not sufficient to train the three classification networks from scratch. (ii) The quality of the training images are not good, as discussed in the previous section. The large variation in training images may make our models difficult to converge.

VI. CONCLUSION

To sum up, the Inception V3 model and ResNet50 are better than VGG16 model in this task. The are some possible reasons: (i) VGG16 model is huge and the number of the parameter are too many but the training data set is not big enough to train the model. (ii) ResNet50 and Inception V3 models have slip connections to prevent gradient vanish from happening. Besides, Inception V3 models extracts multi-scale features to capture more global information and this make the model more robust. Also, the pretrained model is important for the task for two reason. (i) the training data set size is too small. (ii) the training data image quality is bad. As a result, in the future, if we want to create a high accuracy model, we need to focus on collecting high quality and quantity training. Otherwise,

we can use transfer learning or pretrained model to initialize the parameter of our model. Moreover, we should put skip connection, extract multi-scale features and add dropout layer in our model.

VII. CONTRIBUTIONS

- Yao-Cheng Yang: Dataset collection. Model selection. Results analysis.
- Ching-Yu Chen: Dataset collection. Dataloader implementation. Results analysis.
- Chun-Yen Liou: Dataset collection. Related work survey. Model training. Results analysis.

VIII. REFERENCE

REFERENCES

- [1] Tilo B., Janko C., (2006), Real-time Face Detection and Tracking of Animals. IEEE 8th Seminar on Neural Network Applications in Electrical Engineering (NEUREL06), pp. 27–32.
- [2] Heydar A., Targhi H M, Eklundh A T., Pronobis J O., (2008), Joint visual vocabulary for animal classification, Pattern Recognition, ICPR 2008, 19th International Conference, pp 1– 4.
- [3] Matthias Z, Angela S. S, Christian B., (2013), Acoustic detection of elephant presence in noisy environments, In: Proceedings of the 2nd ACM international workshop on Multimedia Analysis for Ecological Data, pp 3-8.
- [4] Deva R., David A. F., Kobus B., (2006), Building models of animals from video. In: IEEE transaction on Pattern Analysis and Machine Intelligence, vol 28, pp. 1319–1334.
- [5] Xiao T Y., Xiaobai L., Shuicheng Y., (2012), Visual Classification With Multitask Joint Sparse Representation, In proceedings of the IEEE Transactions on Image Processing, vol 21, pp 4349- 4360.
- [6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
- [7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [9] Deng, J. and Dong, W. and Socher, R. and Li, L.-J. and Li, K. and Fei-Fei, L. "ImageNet: A Large-Scale Hierarchical Image Database" 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.