COMP - 6721 Applied Artificial Intelligence
Project Progress Report By Group - O

A. Introduction and Problem Statement

Snakebites envenoming [5] is one of the life threatening diseases caused by snakebites, especially in developing countries. Snakebites are treated using antivenom which are selected based on the snake species and snake identification is one of the crucial step in the process. Correctly identifying snake species is very difficult since there are more than 3700 different species of snakes, also it's not very common to have knowledge on snakes in general. Many researchers tried solving this problem by automating the process of identifying snake species using deep learning techniques.

With the help of this application,we will study the three CNN architectures ResNet18, ResNet34, and MobileNetV2 on three different datasets and then compare their performance against a dataset under same training setup and hyperparameters. Additionally, we are going to perform transfer learning on two models with lower accuracy and compare their outcomes. Further,hyperparameter tuning will also be performed on one of the poorly performing model and will try to fine tune that model.To evaluate the performance of models,five different measures are used.

The chosen datasets vary from one another in terms of the classes, samples per class, and image quality. We have used a variety of data prepossessing techniques because the datasets are imbalanced and contain many noisy objects. Due to high complexity of images and higher depth of chosen CNN architecture we require larger training cycles and high computational power.

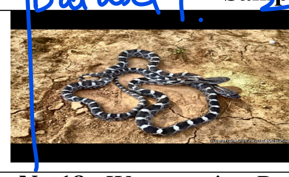
The main goal of our project is to study comparative analysis of all the models and to find the model which can identify the species more efficiently and give better accuracy among all other models. Also to learn deep learning concepts like transfer learning, hyper parameter tuning and perform ablation study.

B. Proposed Methodologies:

Dataset 1 [4] This dataset has total of 17,380 images that are divided into 5 different snake species. This dataset has already undergone some pre-processing so when we train our models we are anticipating better results as compared to the other two datasets.

Dataset 2 [6]There are 3,588 total images in this collection, which are split up into 15 different classes. Images in this dataset are relatively complex and the samples are represented in various terrains and the camouflage nature of the snakes makes it even more challenging to pinpoint the specific subject. Most of the pictures in this dataset are surrounded by noisy subjects and the number of images per sample is also relatively low so we are expecting some underfitting issues.

Dataset 3 [2] This dataset has total of 13,150 images which are divided among 35 classes. This dataset was created using web scrapping and some classes had some irrelevant pictures so we had to manually cleanup dataset before doing any further pre-processing.

DataSet Exploration	
Image Compression Format	JPG
Image Type	RGB
Dataset 1. **Sample Images** 2 3.	
	

ResNet18: We are using ResNet18 model definition from pytorch library as it comes with pretrained weights which will be useful for transfer learning later in our analysis. This model has 11.4 million trainable parameters. With residual blocks this model addresses degradation problem which is encountered with traditional models such as VGG.

ResNet34: We initially proposed ZF Net as part of project proposal but since pytorch didn't had the support for ZF Net and we couldn't find relevant papers on ZF Net after consulting with respective stakeholders we switched to ResNet34 model. This model is an extended variation of ResNet18 with more depth. It has total of 21.5 million trainable parameters.

MobileNetV2 [10]: We initially proposed to use VGG16 as part of project proposal but during training we observed that because of it's 138 million trainable parameters it was taking too long to complete and it was simply not scalable for the dataset and computing resources we had, so after consulting with respective stakeholders we changed our model from VGG16 to MobileNetV2. With just 3.5million trainable parameters this model is designed to perform well with low computational cost.

C. Attempts at Solving the Problem

Dataset Preparation The datasets are divided into 3 parts before training the model. 70% of the total images from each dataset are considered as training images, 20% are considered as validation images and 10% are considered as testing images. But for Dataset 2 there are already test data available in the data source so we have divided remaining images into 0.9:0.1 ratio.

Furthermore, we applied normalisation as part of data preprocessing using the mean and standard deviation computed from our dataset.Along with that we also used transforma-

tions like resize, randomHorizontalFlip and randomRotation which will help model to generalize better.
So far we have made progress on training and evaluating our Resnet18 and ResNet34 framework with all 3 datsets. In future we will do the comparative analysis with MobileNetV2 as well. Additionally, in order to optimise our models, we looked at both the Adam and SGD optimizers. Despite the fact that Adam converges well, we ultimately chose to use SGD with momentum because it has higher generalisation than Adam and eliminates the need for bias rates in our models [13].We also evaluated various learning rate scheduler and decided to proceed with Cosine Annealing by stepping at each epoch. [3]

Model Parameters	
Epochs	100
Input Size	224 x 224
Batch Size	32
Loss Function	Cross Entropy

After training for 100 epochs we obtained below testing accuracy for Resnet18 and ResNet34 models.

Model	**ResNet18**	**ResNet34**
Dataset 1	74.54	63.53
Dataset 2	39.16	37.78
Dataset 3	40.92	45.68

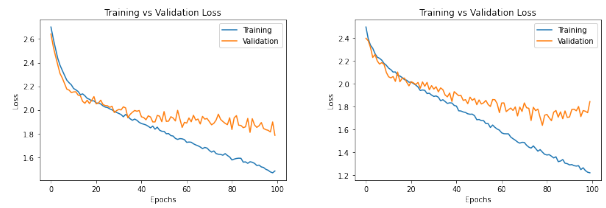
We choose dataset 2 with ResNet18 for transfer learning. We used the pre-trained weights from pytorch which was trained using ImageNet 1k dataset. We used this weights and preformed fine tuning and trained our model with dataset 2 and we got the training accuracy of 94% just in 40 epochs which is a dramatic change when comparing with the model trained from scratch. Below are the results from transfer learning.

Model	**Without TL**	**With TL**
Testing accuracy	39.16	62.78

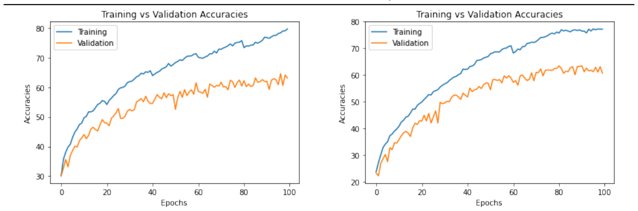
We evaluated these models on the basis of 100 epochos which means longer training time and severe resource crunch . We tried utilizing Concordia's lab GPU's for training but with limited resource we were not able to train our larger model. We used Google collab free resources for training but for dataset 1 [4] which had around 17,380 images we simply ran out of resource mid way resulting in progress loss. To resolve this for this large dataset issue we trained our models in batches of 20 epochs, saved weights and then loaded last batch weights as input and then trained next batch. This was done with same batch for all 3 CNN framework for dataset1 to keep the comparison fair. For dataset 2 and datset 3 we trained in google collab pro.

DataSet 2 Loss Plot

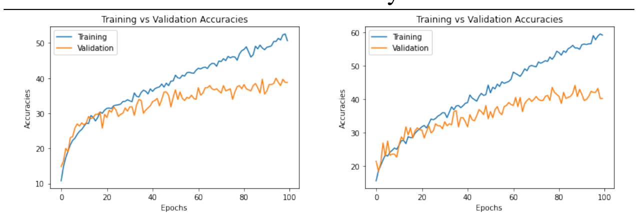
ResNet18 **ResNet34**



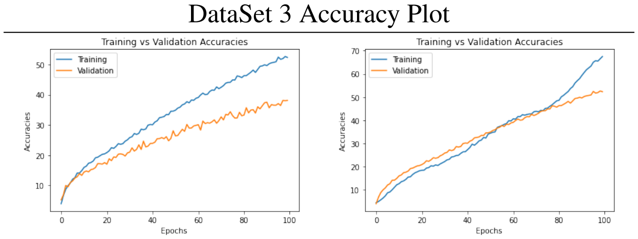
DataSet 1 Accuracy Plot



DataSet 2 Accuracy Plot



DataSet 3 Accuracy Plot



Please refer the appendix section on the last page for evaluation matrix obtained after training for 7 models so far.

D. Future Improvements

Once the training of all models is completed we will compare the models based on the accuracy and will perform hyperparamter tuning on the worst performing model. Based on the findings and computation capabilities we currently have, we decide to optimize batch size and learning rate using random search. [1] While training our models we observed that for dataset 2 [6] is facing overfitting issue, even after using few data augmentation technique. We are going to try few more techniques such as random crop, pad, random invert, color jitter and will compare how it impacts our overfitting issue. Further we are going to explore dataset 2 [2] with MobileNetV2 model for transfer learning. We will be using sklearn TSNE to visualize our models. We are going to visualize how data is separated for different models, how it varies with dataset size and how transfer learning will impact it.

Appendix

	Dataset1		Dataset 2			Dataset3	
Metrics	ResNet18	ResNet34	ResNet18	ResNet34	ResNet18 with TL	ResNet18	ResNet34
Precision Macro	0.72	0.71	0.40	0.39	0.61	0.37	0.48
Precision Micro	0.73	0.71	0.39	0.38	0.60	0.36	0.46
Recall Macro	0.72	0.71	0.38	0.37	0.59	0.36	0.46
Recall Micro	0.73	0.71	0.39	0.38	0.60	0.36	0.46
F1-score Macro	0.72	0.71	0.39	0.38	0.60	0.36	0.46
F1-score Micro	0.73	0.71	0.39	0.38	0.60	0.36	0.46

References

[1] Grid search vs random search vs bayesian optimization. https://towardsdatascience.com/grid-search-vs-random-search-vs-bayesian-optimization-2e68f57c3c46. 2

[2] Identifying different breeds of snakes. https://www.kaggle.com/datasets/duttadebadri/identifying-different-breeds-of-snakes?select=dataset. 1, 2, 5

[3] Learning rate scheduler test. https://github.com/ozwin/Snake_Species_Recognition/blob/main/Output/Practice%20%26%20failures/learning_rate_calculation_for_spliting_epochs_for_cosine.ipynb. 2, 5

[4] Pre-processed snake images. https://www.kaggle.com/datasets/sameeharahman/preprocessed-snake-images?datasetId=578272. 1, 2, 5

[5] Snakebite envenoming optimization. https://www.who.int/news-room/fact-sheets/detail/snakebite-envenoming. 1, 5

[6] Snakes species. https://drive.google.com/file/d/1rYOgC6qPHs1Z6520DPNnS2HAHtHLdpDK/view?usp=sharing. 1, 2, 5

[7] Sharada P. Mohanty Isabelle Bolon Marcel Salathé Andrew M. Durso, Gokula Krishnan Moorthy and Rafael Ruiz de Castañeda2. Supervised learning computer vision benchmark for snake species identification from photographs: Implications for herpetology and global health. In *https://www.frontiersin.org/articles/10.3389/frai.2021.582110/full*, 2021. 5

[8] Kamron Bhavnagri. Modern algorithms: Choosing a model. In *https://www.kamwithk.com/modern-algorithms-choosing-a-model*, 2020. 5

[9] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint*, 2014. 5

[10] Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen Mark Sandler, Andrew Howard. Mobilenetv2: Inverted residuals and linear bottlenecks. In *https://proceedings.neurips.cc/paper/2020/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf*, 2018. 1, 5

[11] Rob Fergus Matthew D. Zeiler. Visualizing and understanding convolutional networks. In *paper*, 2013. 5

[12] Mohammad Shahadat Hossain Raihan Ul Islam Nagifa Ilma Progga, Noortaz Rezoana and Karl Andersson. A cnn based model for venomous and non-venomous snake classification. In *Communications in Computer and Information Science book, volume 1435*, 2021. 5

[13] Chao Ma Caiming Xiong Steven Chu Hong Hoi Weinan E Pan Zhou, Jiashi Feng. Towards theoretically understanding why sgd generalizes better than adam in deep learning. In *https://proceedings.neurips.cc/paper/2020/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf*, 2020. 2, 5

[14] Abhishek Verma. Zfnet: An explanation of paper with code. In *https://towardsdatascience.com/zfnet-an-explanation-of-paper-with-code-f1bd6752121d*, 2020. 5

[15] Richard O. Sinnott Zihan Yang. Snake detection and classification using deep learning. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021. 5