Snake Species identification using Machine Learning

Mohd Aamir, Emerson Cardoso, Sameer Hans
EURECOM
MALIS project progress update report

Abstract: *Snakebite poisoning is one of the tropical diseases responsible for many deaths globally and particularly in developing countries. In most cases, identifying the species of the snake following the bite is the life-saving factor for snakebite poisoning patients. However, snake species identification is not a trivial task, it can be misleading and dangerous. That is why this study proposes the use of Machine Learning approaches to automate snake species identification.*

1. Introduction

Snakes are vertebrate animals that belong to the group of reptiles, among which approximately 700 species are described as poisonous. Snakes are animals that share physical characteristics between different species, have different characteristics than snakes of the same species in different geographic regions, the characteristics of species vary at different ages and in opposite sexes.

Traditionally the snake identification process is carried out by making a visual analysis. This approach proves to be inefficient, as it can be misleading due to the need for in-depth knowledge about these animals. For example, to identify correctly, one needs to be able to distinguish different patterns that can be found in a snake which may vary in age, location, sex, the climate in the region, among other features. Also, visual analysis can be dangerous as it needs certain physical contact or approximation between the snake and the specialist

Snakebite poisoning is a potentially fatal disease, causing about 81000 - 138000 deaths annually, and about 421000 and 1.2 million poisonings per year (Bolon et al., 2020). Moreover, the world health organization (WHO) has declared snakebite to be a neglected tropical disease (NTD) that causes a humanitarian crisis (de Castañeda et al., 2019; *Snakebite*, n.d.).

2. Motivation

According to WHO the snakebit causes over 100,000 human deaths and 400,000 victims of disability and disfigurement globally every year. It affects poor and rural communities in developing countries, which host the highest venomous snake diversity and the highest burden of snakebite due to limited medical expertise and access to antivenoms. Death or harmful effects of snakebites can be reversed or prevented by administering antidotes. However, administering an antidote is only efficient when done quickly and correctly -with a positive snake species classification.

3. Methods

In this study, we will make a comparison between Convolutional Neural Network Xception and one method of machine learning(K-NN).

3.1.Models

a) **Xception:** (Chollet, 2017) which means "eXtreme Inception", is a linear architecture network of 36 layers organized in 14 modules that have a residual connection in each module except the first and last module. This architecture uses depth-wise separable convolutions instead of Convolution Layer. The Xception used a modified depth wise separable convolution where the first layer performs a pointwise convolution followed by depthwise convolution.

b) **K-NN:** The k-nearest neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. K-NN - algorithm assumes that similar things exist in proximity. In other words, similar things are near to each other.

3.2.Dataset

In this project we will be using part of the dataset released in the challenge proposed by the world health organization (WHO) for snack identification, more concrete in this study we will work with 6 species of venomous snake (table 1) from France and Europe in general. From the WHO dataset, we were able to collect 1222 Images in total.

ID	Snake Species	Images	Train	Test
000	Vipera aspis	237	190	47

001	Vipera berus	280	224	56
002	Coronella girondica	196	157	39
003	Vipera seoanei	93	74	19
004	Vipera ammodytes	119	95	24
005	Malpolon monspessulanus	239	191	48
Total		**1164**	**931**	**233**

Table 1: Species of Venomous Snake

From the images collected the species 000(Vipera aspis) have more than 237 images in total and the class 005 (Vipera seoanei) is the species with less image 93. To split our dataset into train and test we use the ratio 80:20, where 20% of the images per class was attributed to the test set and we use k-fold cross validation (k=3) on our training set. Before splitting the dataset in training and test we did some data normalization such as resizing all the images to the same size 256x256.

3.2.1. Data Augmentation

The ratio between the class with the lowest sample and the one with the higher sample is disproportionate, for that reason, we decide to use data augmentation on the train set to provide a well-balanced dataset. We apply data augmentation for classes with few samples so that we can have the same number of samples as the class with more samples. The following techniques of data augmentation were used randomly: Random rotate, Random flip, saturation, crop, dilation, erosion, brightness, Histogram and gaussian blur.

3.2.2. Key Characteristics

The dataset has some interesting characteristics that make the classification of these species a challenge, some of these characteristics are described below:

a) The dataset contains images of snakes in their habitat (bushes, rocks, etc.), or other objects that we call noise here.
b) The species 003 presents two different colour patterns depending on where the pictures were taken.
c) Snakes are very diverse animals which make some species patterns (colour, shape etc) very different.
d) The same species can present different characteristics and patterns if the age, location and sex are different.

3.3. Training and testing

The models were trained in the Kaggle kernel using Tensor Processing Unit (TPU) accelerator v3-8 that supports 4 dual-core TPU chips for a total of 8 TPU cores, with 16GB of RAM. The training and testing phase followed the k-fold cross-validation approach, with k = 3. To evaluate the model performance the following metrics were calculated, accuracy, precision, f1 score, and recall. The metrics described were applied using the scikit-learn library

a) **Precision** is a metric that allows us to determine if the model is correct when the prediction of a snake species is positive when the predicted species is equal to the real species.
b) **Accuracy** is the most widely used measure for classifying models; it is the ratio of the classes correctly predicted to the existing class total.
c) **The recall** metric is a metric that allows you to determine how often the model correctly determines species in the test data set.
d) **F1score** determines the weighted average (harmonic) between Precision and Recall, in this metric the closer to 1, the better the model's performance.

4. Preliminary Experiments and Results

Our first experiment shows us that we need more data and some good features extraction for our models, the partial result is detailed below per model:

4.1. Xception

We implement Xception Architecture from scratch based on the description provided by the author in his article. We also add a Batch Normalization layer after each Convolution operation to add some noise and bust our training. The xception was implemented with the following hyperparameters:

Learning rate	0.01
Loss function	Stochastic Gradient Descent
Optimizer	Sparse Categorical cross-entropy
Batch size	32
Epoch	100

We train the Xception model using the k-fold cross-validation method with k=xx and where k estimation points were generated, and the weighted average was calculated for each of the metrics used. The use of cross-validation allows the model to be trained and validated throughout the data set (Yadav & Shukla, 2016).

	Mean	**Std.**
Accuracy	0.21806853582554517	0.01588479599250089
f1_score	0.21489046569440176	0.015468902801510457
Precision	0.23068421875807987	0.021468165392170165
Recall	0.221525342270023	0.012792121322788706

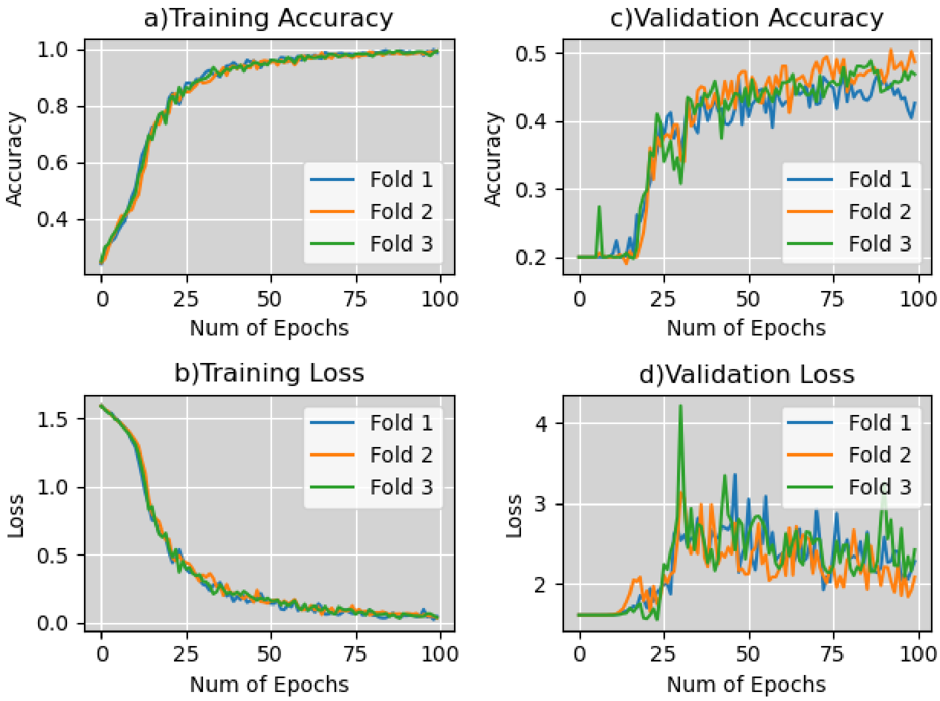

Figure 1: Xception Performance

The model xception performs well during the training phase achieving an accuracy of about 90% but we still have low validation accuracy. There is a clear sign that our model is overfitting as we look at our losses for the training and validation because our training loss decreases but the validation increases which is a clear indication of overfitting. It's learning the training data well but fails to generalize the knowledge to the test data and we get a very low f1 score which is not good.

4.2.KNN

We implement knn using the euclidean distance and for the features extraction, we compute the histogram of the image. The knn algorithm had very low accuracy using histogram colour as features to train the model, we believe that some key dataset characteristics influence this behaviour on the knn model.

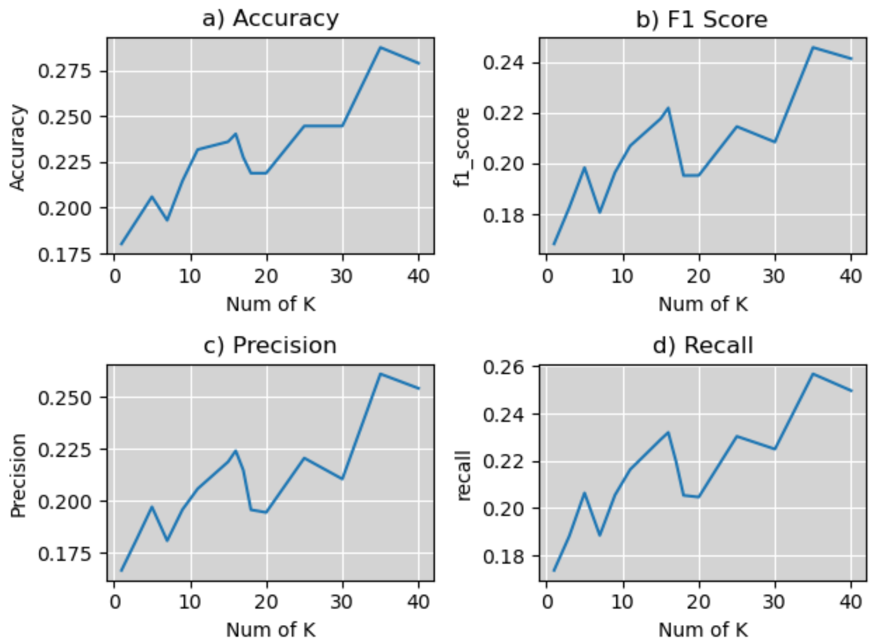

Figure 2: performance of different k in knn

As we can see that the training accuracy itself is very low for our KNN model for different values of k. One of the reasons we think this might be happening is because of the object which we are trying to classify i.e the snake. Snakes tend to merge around with their environment like trees, mud, leaves which is what according to us, making it hard for our model to learn just based on the histogram of the images.

5. Contributions & Next Steps

Our first approach to the problem of identification of snakes shows us that we have a long way to go for achieving good performance, the table below shows in brief what we have done and what we still have to do in the next weeks.

	Task performed	**Future task**
Emerson Cardoso	Dataset cleaning, preparation, data augmentation.	Funding more data, normalising data, try different techniques for data augmentation.
Sameer Hans	Knn implementation and training, tuning.	Research and implement a new strategy for features extraction.
Mohd Aamir	Xception implementation and tuning.	Improve Xception accuracy and avoid overfitting.

Addition things to do when time allow us:

- Try another algorithm learned in class, Neural Network and SVM to find out if there is something else which works better for our dataset.
- Try Convolution for features extraction to train KNN in order to give our model a better chance to learn from our dataset.
- Try different hyperparameters for our Xception model in order to find the one which works the best for our dataset.

6. References

Abdurrazaq, I. S., Suyanto, S., & Utama, D. Q. (2019). Image-Based Classification of Snake Species Using Convolutional Neural Network. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 97–102. https://doi.org/10.1109/ISRITI48646.2019.9034633

Abeysinghe, C., Welivita, A., & Perera, I. (2019). Snake image classification using siamese networks. *ACM International Conference Proceeding Series*, *June*, 8–12. https://doi.org/10.1145/3338472.3338476

Amir, A., Zahri, N. A. H., Yaakob, N., & Ahmad, R. B. (2017). *Image Classification for Snake Species Using Machine Learning Techniques*. https://doi.org/10.1007/978-3-319-48517-1

Bolon, I., Durso, A. M., Mesa, S. B., Ray, N., Alcoba, G., Chappuis, F., & De Castañeda, R. R. (2020). Identifying the snake: First scoping review on practices of communities and healthcare providers confronted with snakebite across the world. *PLoS ONE*, *15*(3), 1–24. https://doi.org/10.1371/journal.pone.0229989

Chollet, F. (2017). Xception: Deep Learning with Depth Wise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

de Castañeda, R. R., Durso, A. M., Ray, N., Fernández, J. L., Williams, D. J., Alcoba, G., Chappuis, F., Salathé, M., & Bolon, I. (2019). Snakebite and snake identification: empowering neglected communities and health-care providers with AI. *The Lancet Digital Health*, *1*(5), e202–e203. https://doi.org/10.1016/S2589-7500(19)30086-X

Snakebite. (n.d.). Retrieved February 26, 2020, from https://www.who.int/health-topics/snakebite#tab=tab_3

Yadav, S., & Shukla, S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *Proceedings - 6th International Advanced Computing Conference, IACC 2016*, *Cv*, 78–83. https://doi.org/10.1109/IACC.2016.25