[[1]](#footnote-1)

Animal Silhouette Dataset Generation and Classification

Bala Siva Sai Megi Reddy, *E20CSE184*

*Abstract*—The given project involves creating black-and-white binary silhouette mask images of Animal Image Dataset from Kaggle which consists of 5400 images from 90 different animal classes and classifying them correctly. The provided RGB images are converted to silhouettes using multiple image-processing techniques from python libraries and saved in a new folder in the same format as the original dataset. Following this, we perform data augmentation on the masks for a larger training set. Then a CNN architecture created through PyTorch framework is used for the task of classification. Finally, the model is tested on the test set. We were only able to achieve an accuracy of nearly 10% on validation and test sets while above 15% on training set.

# INTRODUCTION

S

ilhouettes are the images of a creature, object or scene represented as a solid shape of single color, usually black, with its edges matching the outline of the subject. The interior of a silhouette is featureless, and is generally presented on a light background, usually white, or none at all. Classifying such featureless images is a task that could prove useful in certain situations.

## About the Problem

During our research, it was found that there is no good animal silhouette dataset available online. Also, manually collecting such images would take a long time to create it ourselves. Therefore, this task needed to be automated. The dataset then needed to be used for creating a classifier that can achieve good accuracy on the new images.

## Motivation

Analyzing and classifying silhouettes is quite difficult even for a human being, reason being that the internals of a mask are featureless. Identifying the class of silhouette is an important task as they are used in maps, traffic signs and aircraft recognition [1]. Also, photograph from a certain angle might create a silhouette due to lighting [1]. Hence, training a classifier on such images will be quite helpful.

*Fig.1 Examples of silhouettes from real life [1]*

## Background Knowledge

The reader should be familiar with basic image processing techniques such as smoothing, thresholding and morphological operations like dilation and closing. They must also be familiar with deep learning concepts such as convolutional neural networks and data augmentation. For implementation, they should be familiar with python frameworks such as OpenCV for image processing and PyTorch for data augmentation and building neural networks.

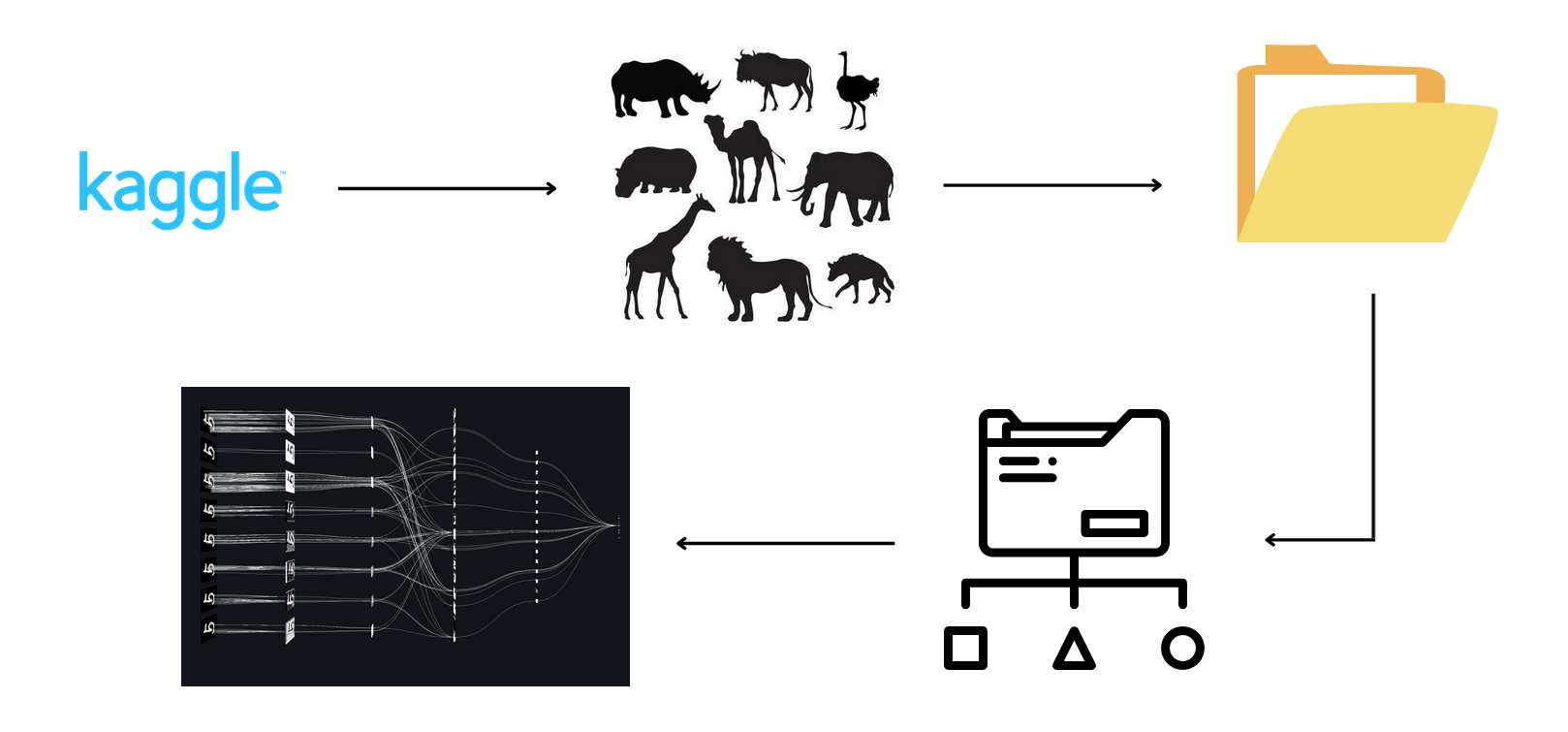
# Related work

Xiang Bai, Wenyu Liu and Z.W. Tu in their paper addressed the problem of shape classification by combining both contour and skeleton information for shape analysis and collected a shape database with 20 classes of animals each having 100 shapes. All the shapes were obtained from real images with large variation in pose, viewing angle, articulation and self-occlusion. The images of the database are basically silhouettes, with white background and the animal mask being black [2].



*Fig.2 Images of the dataset from the paper [2]*

# Pipeline



*Fig.3 Steps of implementation*

1) Import the dataset from Kaggle

2) Creating silhouettes for image dataset

3) Save these silhouettes in a folder of similar format

4) Pass the folder to the classification model

5) Modify the neural network into a Resnet architecture

# Proposed methodology

The process began by downloading the animal image dataset created by Sourav Banerjee from Kaggle [3]. The platform used for programming was Google Colab for GPU access and preinstalled machine learning libraries.

## Dataset Description

The dataset has been collected from kaggle.com. It consists of a total of 5400 images of animals, from among 90 categories containing 60 images each. It covers mammals, birds, reptiles and insects. Given images are in Joint Photographic Expert Group (jpeg) format.

## Pre-processing

Silhouette mask creation involved following steps:

1) Converting the images to grayscale form.

2) Applying a median filter of kernel size 5 on the images for removing salt and pepper noise.

3) Applying a gaussian filter of kernel size 5 and sigma 1 for smoothing the edges and reducing the noise further.

4) Applying Otsu thresholding on the images, which converts them into binary form by finding the threshold value on its own.

5) Performing morphological operation of closing (dilation + erosion) of larger kernel size 11, for filling the gaps in the objects and converting them back to original size.

6) Saving the images in a new folder named Masks with subfolders named as class of the image.

All the steps were performed using OpenCV and OS libraries of python.

*Fig 4: Creating silhouette of an animal*

The new dataset was split into training, testing and validation sets. All the images were resized to be 256x256 in dimensions and normalized to a mean and standard deviation of 0.5. The training set was augmented using PyTorch’s transforms class, by creating horizontally flipped and rotated by 15 units images.

## Feature Selection

A convolutional neural network was used for classification. It does not need manual feature selection as the convolutional layers are able to extract the features like edges from images themselves after training. Therefore, no separate feature selection method was used.

## Algorithm

A CNN architecture with 30 resnet blocks consisting of convolutional filter followed by Batch Normalization and Leaky ReLU activation function with negative slope as 0.1 was trained on the training set. The network ends with a global average pooling layer that converts the feature maps to a 1x1x512 map. It is connected to a linear layer with softmax activation function, with 90 nodes whose output will decide which class the image belongs to.

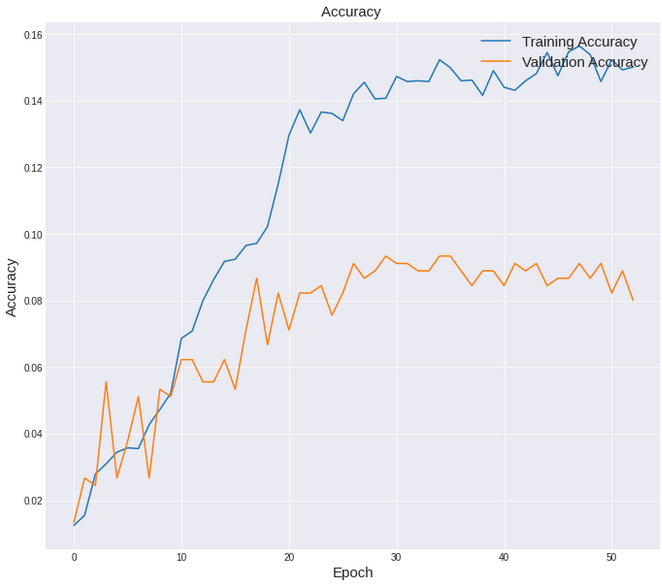
We decided to use Resnet architecture with Leaky ReLU as both of them will prevent the problem of vanishing gradient during training. Resnets use the concept of skip connections, which will pass on the information from the previous feature map to the new feature map, that could prove helpful in classifying silhouettes that hardly contain any features internally.

# Results

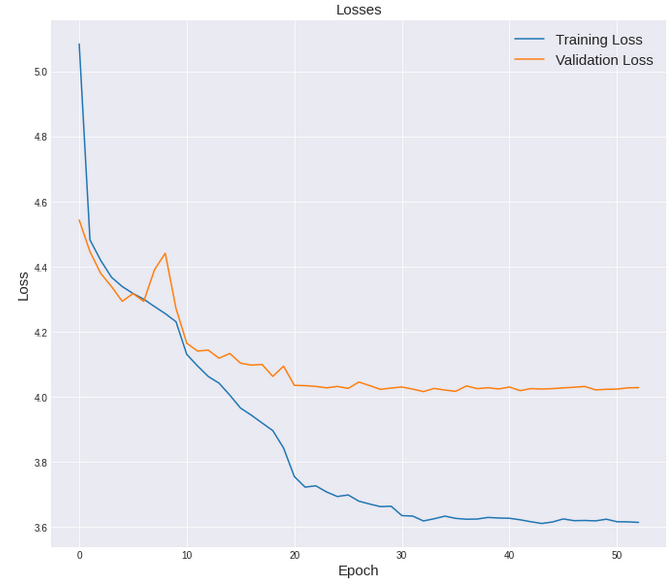
## The generated silhouette masks from some images are perfect, while in others, background objects like grass and internal features such as stripes have interfered in solid image production. This resulted in modification of shape of certain animals, making them look like others. Having such masks of bad quality made it hard for the model to train and classify the images correctly.

.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | *Fig 5: Images with their ground truth label and predicted labels* | | |
|  |
|  |
|  |
|  |



*Fig 6: Epoch vs Accuracy graph*



*Fig 7: Epoch vs Loss graph*

After training the model for 53/100 epochs using early stopping limit of 20 epochs with batch size of 64 images, the highest accuracy and least loss that we achieved on the respective subsets of data are given as follows:

|  |  |  |
| --- | --- | --- |
| Set | Train | Validation |
| Accuracy | 15.64% | 9.33% |
| Loss | 3.6203 | 4.0273 |

The accuracy on the test set was 8.33%.

It was observed that while accuracy on training set kept increasing quite slowly, the accuracy of validation set saturated really early on.

# Conclusion

## In this experiment we sought to create a silhouette mask dataset for animals and use a ResNet model for classifying them. The results were not great, as conventional image processing techniques proved unable in generating clear silhouette images. The background objects interfered in the final output, distorting the animal’s shape. This resulted in making it quite difficult even for the neural network to classify an object which was already featureless internally.

# Future scope

## The silhouette masks should be created using semantic segmentation done through CNNs instead of using traditional image processing techniques. The mask of the animals will not get affected from background objects in this case, generating clear and precise silhouettes. The model will find it much easier to classify such images after training. Changes could be made to the model itself so that it is able to capture more features from a silhouette which is featureless internally. More resnet blocks should be added and trained for a much longer period, as we observed both these acts to have improved the result slightly.

# VIII. References

1. https://en.wikipedia.org/wiki/Silhouette#Silhouette\_images
2. Bai, Xiang & Liu, Wenyu & Tu, Z.. (2009). Integrating contour and skeleton for shape classification. 360 - 367. 10.1109/ICCVW.2009.5457679.
3. https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals

1. [↑](#footnote-ref-1)