HDSC Winter ‘22 Capstone Project: African Wildlife Animal Conservation

A Project by Team Smote

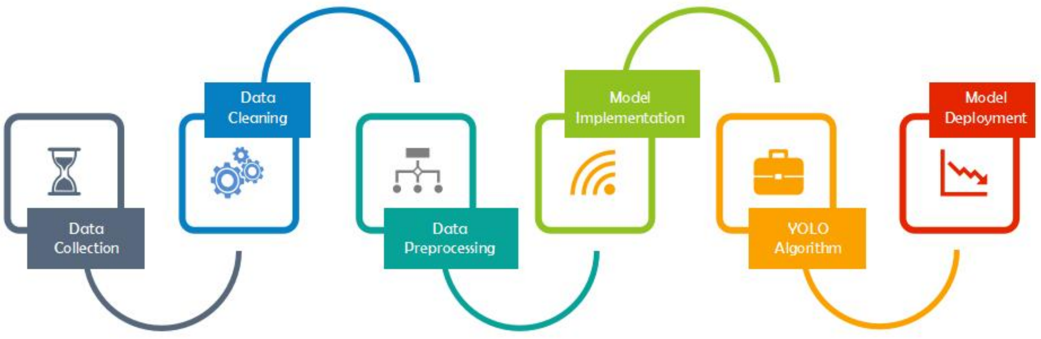
PROBLEM STATEMENT
Poaching remains one of the biggest threats to wildlife animals in Africa. Elephants are hunted for their ivory, rhinos for their horns, buffaloes for bushmeat, and zebras for their skins.

Poached Rhino (Susan Scott/STROOP via AP)

According to the African Wildlife Foundation, the population of black rhinos has decreased by 97.6 percent since 1960; up to 35,000 elephants are illegally killed each year; and only around 2,000 adult Grevy's zebras exist.

At the present rate of poaching, these famous African animals may be extinct within our lifetime.To alleviate this key threat, strict law enforcement and anti-poaching measures must be implemented.

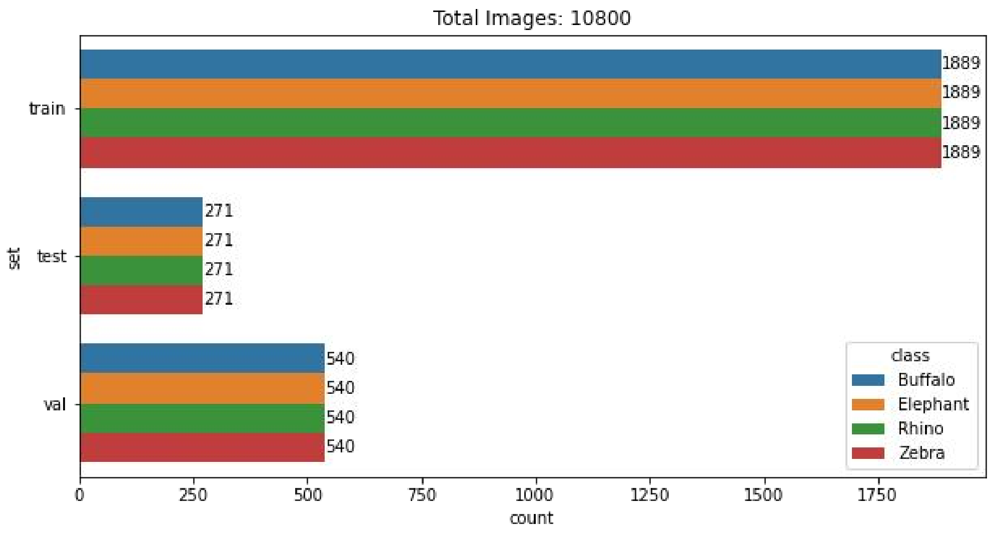
OBJECTIVE
This project seeks to develop effective computer vision models that can be deployed as part of security systems to detect poaching in wildlife reserves. This can in turn increase wildlife crime prosecution in important wildlife crime hot regions.



Flow Process

DATA COLLECTION

We scraped thousands of wildlife buffalo, elephant, rhino and zebra images from royalty-free websites using Beautiful Soup and Selenium. We then augmented these images, yielding a total of 10,800 images for the project.



MODEL BUILDING

We built an image classifier using transfer learning and an object detector with the YOLO (You only look once) algorithm.

TRANSFER LEARNING

Transfer learning is a machine learning (ML) technique that focuses on storing knowledge learned from one problem and applying it to a different but related problem. For instance, knowledge obtained when learning to recognize cars might be applied when attempting to detect trucks.

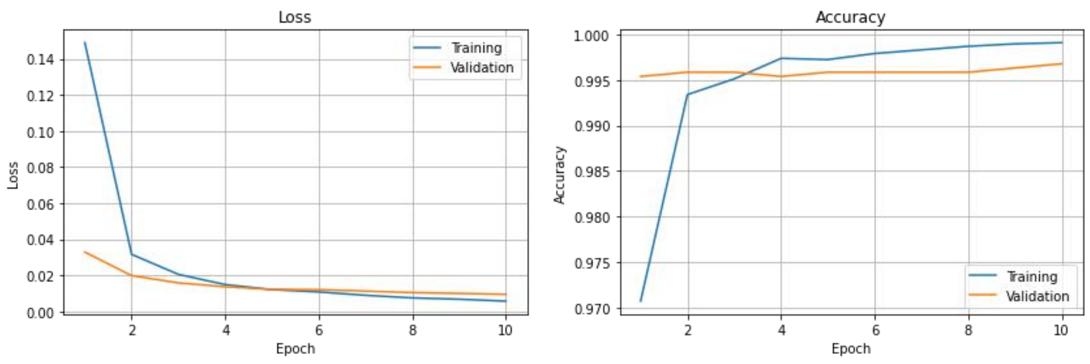
Keras provides us with a handful of pretrained models to use for our task. Keras is a high level API (application programming interface) built on top of Tensorflow. Tensorflow is an open source ML library that focuses majorly on deep learning.

These pretrained models were trained on the **Imagenet** data. Imagenet is a large collection of image data containing 1000 categories of images.

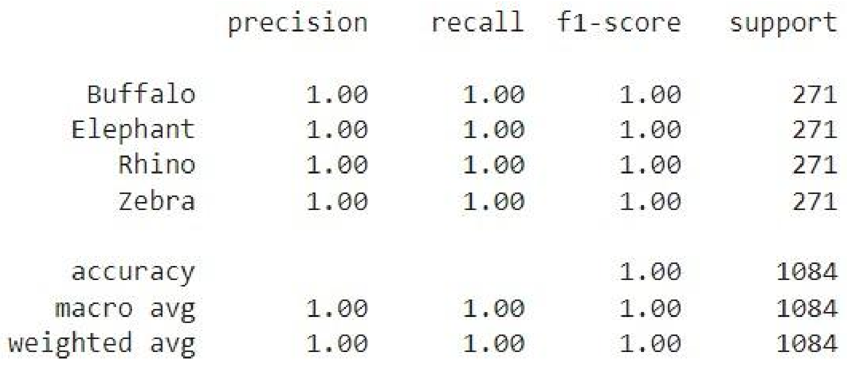
We tried several versions of the following models:

Xception
vgg
ResNet
Inception
inceptionResNet
MobileNet
DenseNet
NasNetMobile
NasNetLarge
EfficientNet

We were able to achieve encouraging results just by changing the output layer to predict only 4 classes (Buffalo, Elephant, Rhino, and Zebra). The **EfficientNetB6** outperformed the remaining models on the unseen test set.



Loss and Accuracy plots for EfficientNetB6

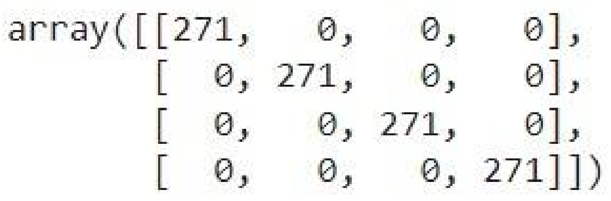


```
              precision    recall  f1-score   support

     Buffalo       1.00      1.00      1.00       271
    Elephant       1.00      1.00      1.00       271
       Rhino       1.00      1.00      1.00       271
       Zebra       1.00      1.00      1.00       271

    accuracy                           1.00      1084
   macro avg       1.00      1.00      1.00      1084
weighted avg       1.00      1.00      1.00      1084
```

Classification report for EfficientNetB6



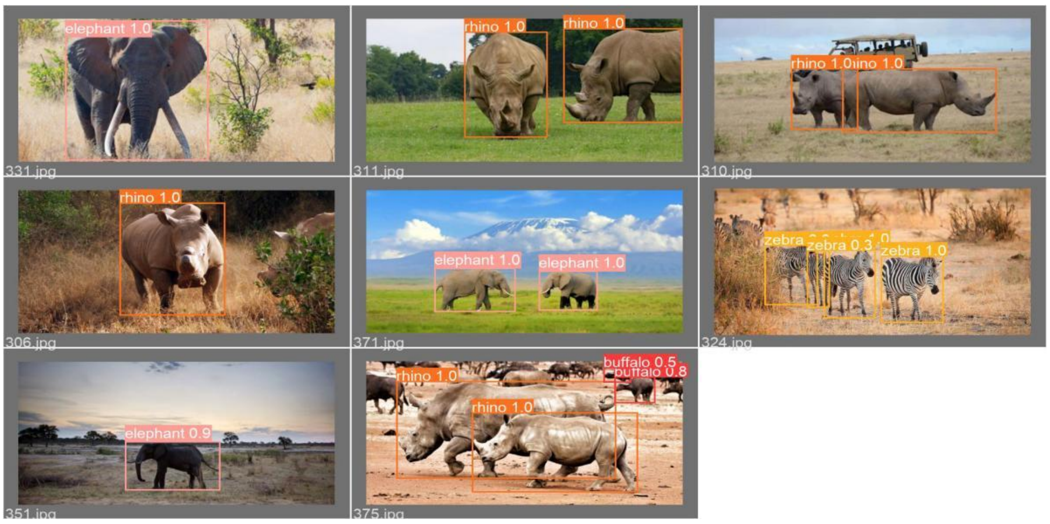
```
array([[271,   0,   0,   0],
       [  0, 271,   0,   0],
       [  0,   0, 271,   0],
       [  0,   0,   0, 271]])
```

Confusion matrix for EfficientNetB6

YOLO

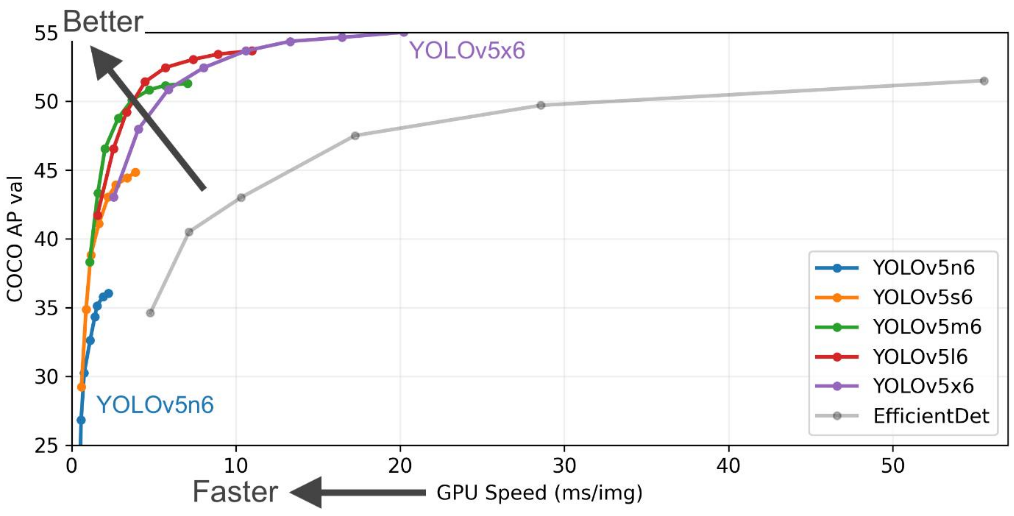
YOLO is an algorithm that detects and recognizes various objects in a picture (in real-time). This algorithm works by dividing an image into grids, each having equal dimensional regions. Each of these grids is responsible for the detection and localization of the object it contains.

These grids predict bounding box coordinates along with the object label and probability of the object being present in the cell.



Yolo in action

This performance chart guided our choice in choosing the **YOLOv5x6** for our project:

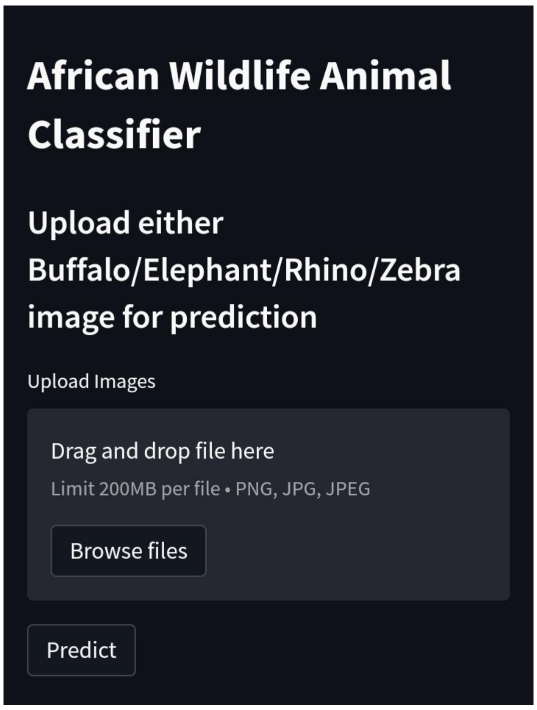


The chart represents the performance of each model on the COCO (Common Objects in Context) dataset. This dataset is is widely used to benchmark the performance of computer vision models.

The y-axis represents the average precision value (AP val) of a model on each class while the x-axis represents the time it takes a model to make an inference on an image. We chose Yolov5x6 because it has the best AP value among the models.

MODEL DEPLOYMENT

We created web applications for both image classification and object detection. These applications were deployed using streamlit. Streamlit is an open-source app framework does the heavy lifting for us. It enables the data app to run with just few lines of code. We do not need to worry about building a full-fledged website to show how our model works in production



African Wildlife Animal Classifier

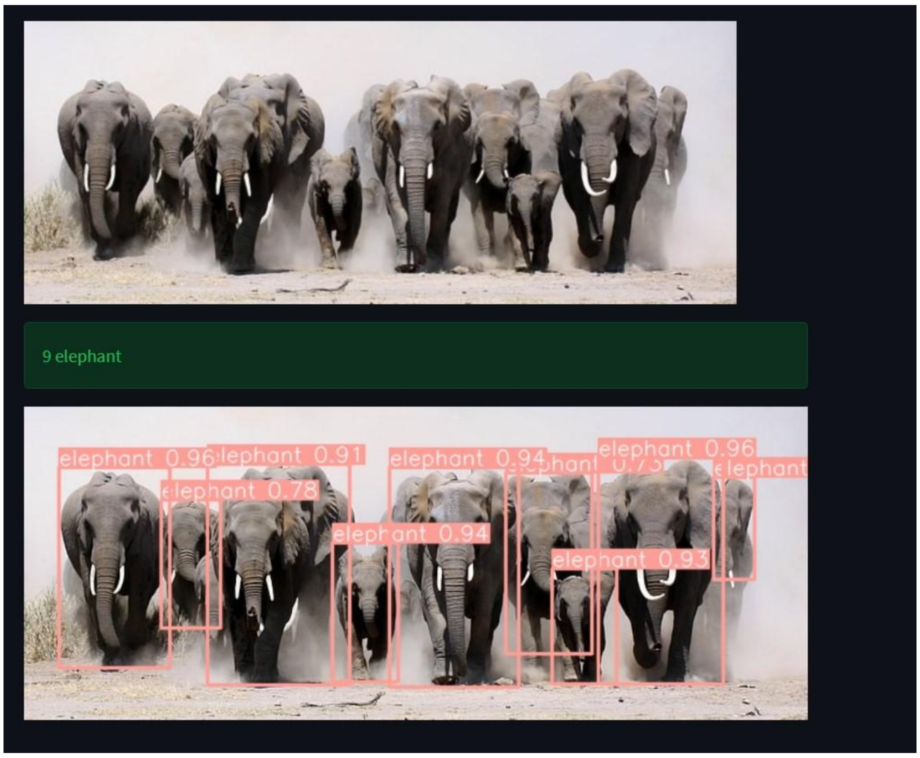
Upload either Buffalo/Elephant/Rhino/Zebra image for prediction

Upload Images

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

Image classifier web app



9 elephant

Object detection web app

CONCLUSION AND RECOMMENDATIONS

Threat detection is an integral part of threat prevention. When ineffective detection mechanisms are put in place, prevention attempts can be foiled.

Satellite cameras should be put in strategic locations to take real-time images and transmit them to the model. The model then generates a prediction and triggers the security system to inform the appropriate authorities of an invasion. These strategic locations could include paths that lead to and from the reserves.

In order to reduce the number of false alarms, the results of the YOLO algorithm can be fed to a real time monitor for the necessary authorities to confirm if there truly is an invasion. Footage of these events can also be analyzed to discover poaching patterns. Some interesting questions to ask could be:

- What animals are more likely to be hunted?
- What time of the year are these animals most vulnerable to poaching?
- In what manner(s) are they poached?

Link to project: https://github.com/heyakshayhere/Hamoye_capstone_project_smote
Link to Web apps: https://african-wildlife.herokuapp.com