Analyzing the Species, Geography, and Price Dynamics of Online Wildlife Trafficking

HANSAEM PARK, VANESSA SINAM, and ZHAO GUAN, New York University Tandon School of Engineering, USA

1 INTRODUCTION

Wildlife trafficking poses a severe threat to global biodiversity, endangering species, disrupting ecosystems, and potentially facilitating diseases that affect human health. Projects aimed at combating this illicit trade are of paramount importance, as they not only contribute to the preservation of endangered species but also uphold environmental sustainability. Such initiatives are crucial in maintaining ecological balance, supporting economies reliant on ecotourism, and ensuring ethical and cultural conservation. One of the fundamental notions is that the initial goal is to "understand the picture," which leads to a constant increase in the requirement to grasp the high-level meaning of things when it comes to object recognition and image identification[1]. Furthermore, these efforts aid in enforcing international legal frameworks like cities, fostering necessary global collaboration to address this transnational challenge. In essence, tackling wildlife trafficking is integral to protecting our planet's natural heritage and promoting a balanced coexistence between human activities and wildlife conservation.

The goal of this project is to analyze the data crawled from multiple websites that might be involved in wildlife trafficking. We will use the data collected from multiple platforms to study wildlife trafficking patterns. While analyzing the large data set of ads, we recognized the need to clean the dataset for public use or further studies. Although the data we are working with is already preprocessed, there are some data quality issues, such as wrong labels and missing attributes. In the first part of our project, we dealt with cleaning the dataset and handling missing values. We then tried to answer questions such as the species traded, geographical distribution of ads posted, and the price range of the animals trafficked.

2 RELATED WORK

2.1 Image identification and classification

In our pursuit to classify animal species within a large, unorganized dataset, especially in the context of wildlife trafficking, we recognize the importance of methodologies capable of handling unlabeled data. While our dataset, primarily composed of crawled ads' URLs with images, titles, and descriptions, lacks pre-defined labels, we draw inspiration from the work 'Animal image identification and classification using deep neural networks techniques' by Battu and Reddy Lakshmi. This study, focused on animal species classification from camera-trap photos, employs advanced deep learning techniques, particularly suited for environments with high levels of noise and human activity. Object identification is the name given to this procedure, and it has become a popular study topic in the area due to the focus on understanding what a picture represents[1]. We also look at the fully and semi-automated means of assembling and structuring scientific data through NLP and text mining[4]. The specific type and structure of the information to be extracted depend on the need of the particular application[3]. Similarity is also used in information extraction. Given two sets of objects, similarity join aims to find all similar pairs from the two collections[2].

Although their method relies on labeled data, its innovative use of k-means clustering for data segmentation and dual network architectures to handle varying data characteristics offers valuable insights. Adapting these principles, we aim to implement a semi-supervised or unsupervised learning approach, possibly leveraging clustering techniques to discover inherent patterns and groupings within our unlabeled dataset. The methodology of utilizing deep neural networks, as demonstrated in the study, could be crucial in developing a model that effectively categorizes animal species in our context, despite the absence of explicit labels.

2.2 Species identification using TaxoNERD

TaxoNERD (Taxonomic Named Entity Recognition and Disambiguation) is a specialized tool designed for the precise identification and disambiguation of taxonomic entities within textual data. It provides deep neural network (DNN) models to recognise taxon mentions in ecological documents. This tool serves a critical role in biodiversity informatics and bioinformatics, where accurate recognition and categorization of taxonomic names, such as species and genera, are essential. TaxoNERD operates through two primary functions: Named Entity Recognition (NER) and Disambiguation. In the NER phase, it identifies mentions of

Authors' address: Hansaem Park, hp2229@nyu.edu; Vanessa Sinam, vs2476@nyu.edu; Zhao Guan, zg2501@nyu.edu, New York University Tandon School of Engineering, New York City, New York, USA, 11201.

taxonomic entities in unstructured text, including scientific articles or biodiversity databases. The subsequent Disambiguation process involves resolving potential ambiguities in taxonomic names, linking them to standardized taxonomic databases. This tool is particularly valuable in the understanding of species distributions, ecological interactions, and evolutionary relationships. TaxoNERD can be integrated into Natural Language Processing (NLP) pipelines, enhancing text analysis and information extraction. The tool addresses challenges related to variability in taxonomic naming conventions, including synonyms and multiple names for a given species.

3 PROBLEM FORMULATION

Our dataset consists of 20 columns, most of which contain data that has been roughly crawled and not subjected to any cleaning process. To effectively identify species, we need to analyze both animal images and text data, including product descriptions, names, and titles. The text data in its current state is disorganized, posing significant challenges in accurately extracting species information. In addition to utilizing text data for species identification, we also implement image classification and identification techniques. These methods will allow us to extract species information from the image data, providing a more comprehensive approach to understanding and analyzing our dataset.

It is also important to understand the global impact of wildlife trafficking. Exploring the extent of involvement in this trade by different countries is a key step. In our project, we utilize visualization techniques allows us to observe the distribution of wildlife trade across different countries. One problem is the lack of unified and data driven approach to estimate the market price of animal trading. This issue is particularly challenging due to the various currencies used in most of the trading advertisements. What's more is the unevenly distribution of the prices animals are traded in, makes it difficult to visualize it in a way that best displays the features of the wildlife animal trading prices. To effectively address these issues, we need to identify the species of animals that are being trafficked.

4 METHODS

Our codebase is available here - https://github.com/sammitako/CS-GY-6513-Big-Data.

To make the data more useful, we are focusing on identifying species in the dataset by extracting keywords from text-based data and classifying animals from image data. This approach involves both a detailed analysis of the text data to identify relevant keywords and an examination of the image data to classify the animal species accurately.

4.1 Data Cleaning and Pre-Processing

We excluded columns containing irrelevant data and retained only these values in each column:

- `Label_product`: 'an animal body part' and 'a real animal'.
- `Category`: 'None' or a null value.

4.1.1 Animal Species Identification. The 'label_product' column contains seven categories, and we cannot guarantee that these categories are free from potential trafficking. Therefore, we will retain only those records where 'label_product' is categorized as 'an animal body part' or 'a real animal'.

Originally, the dataset contained 375,726 rows. However, after excluding specific 'label_product' categories, the total number of rows has been reduced to 95,543.

4.1.2 Processing NULL Value Data. The columns 'description', 'country', 'image', 'location', 'loc_name', 'lat', 'lon', and 'price' contain NULL values. We discovered that the columns 'title', 'product', 'description', and 'currency' represent NULL values as 'None' in the dataframe, resulting in an incorrect count of the number of rows. Therefore, we need to examine these columns as well. We also removed the 'ships to' column because it contains no data. There are missing values in the columns. We can find these rows with missing (null or empty) values. If the columns 'name', 'title', 'product', and 'description', which we use to identify 'species', all have NULL values, we can decide to drop them.

4.1.3 Geographic Information Processing. The dataset also has the country attribute missing from most of its entries. Other attributes which could have been used to infer the country such as seller, latitude, and longitude also have missing values in the dataset. We used various techniques to determine the country from where the ads were posted. The first approach is to use currency as a way to map the ads to their originating country. This approach has its limitations in that several countries share the same currency and therefore it is difficult to point out the exact country where the ad originates from. The next approach is using the domain attribute to infer the country. For domains such as 'ebay.uk', a simple Regex is enough to extract the country name.

Since we have location data, we will check for unique values. We can see that the Domain column has no null values. We can use the domain names to infer the country (using the country code top-level domain). The 'country', 'loc_name', 'lat', and 'lon' columns each originally have 66 data entries. The goal is to ensure that the 'country' field is populated. Since the 'location' value is available when other geography-related columns are NULL, we will use the 'location' data to populate the 'country' records. Manually populating the 'country' field with 'location' data (2212) is not possible because some records only include the city name, not the country name. For example, 'Paisley, Renfrewshire' is a town in Scotland (United Kingdom), and 'Eastleigh, Hampshire' is a town in England (United Kingdom). By observing the capital letters in the 'location' field, we can parse and populate the 'country' field. For instance, 'Guangzhou, CN' indicates China. However, there are some issues with the order of country codes in the 'location' data. For example, in some records, the country code appears at the beginning, while in others, it is at the end. There is no consistent ordering. To comprehensively address these cases, we will use an API to obtain the country name. Since we have a list of 57 countries with names in different languages, manual translation might be possible. We can use a library containing translations of country names in different languages to successfully populate the country data.

For the domain, we can use ccTLDs (Country Code Top-Level Domains) to populate the 'country' columns, which still remain NaN after being pre-processed with 'location' data. Before handling gTLDs (Generic Top-Level Domains), we need to view the unique domain names where the 'country' columns has NaN values. Then we Extract Country Codes for country code top-level domains (ccTLDs) and handle generic top-level domains (gTLDs) like ".com",".org", or ".net". We will use other geography-related columns. In the end, only gTLDs are left, which can be further used for DNS whois lookups with the assumption that the trading happens at the location where the website is registered or created.

4.1.4 Price range calculation. For the price range calculation, we considered the attributes price and currency. What comes after is that we convert the various currencies to US dollar using currency rate API. Then calculate the price range of the animals traded online. We have set 10 dollars as price slot and calculated for more than 150,000 ads so far.

We normalize prices using an API. To minimize redundant API calls, we store the conversion rates in a dictionary. By using 'USD' as the base currency, we will calculate the normalized prices.

4.1.5 Text Standardization. We will use Fingerprinting method that is the least likely to produce false positives, which is why it is the default method for text standardization. We generate the key from a string value by removing leading and trailing whitespace, changing all characters to their lowercase representation, removing all punctuation, and non-alphabetic characters, and control characters, normalizing extended western characters to their ASCII representation (for example "gödel" → "godel"), splitting the string into whitespace-separated tokens, sorting the tokens and remove duplicates and joining the tokens back together.

4.2 Keyword Extraction for Species Detection

4.2.1 Data pre-processing for keyword extraction. Upon analyzing the data, we discovered that the 'name', 'title', and 'product' columns contain identical data in the field. Therefore, we chose the 'product' column as it has the longest average text length. Then we dealt with duplicate entries in our dataset. We observed that the same advertisements are posted on different URLs. Upon examination, we noted that the 'name', 'title', and 'product' are the same; however, the descriptions vary. To better process these rows to extract species keywords, we removed duplicate rows where 'name', 'title', 'description', and 'product' are the same. Then we remove duplicate rows where the description is the same.

4.2.2 Dropping rows that are not related to wildlife trafficking. We removed rows where the text in the 'product' column indicates that it might not be related to wildlife trading. From the dataset, we remove rows that have terms such as 'faux' and 'replica' in the product column which implies that the product is not a real animal or an animal body part. More examples: 'peluche' translates to a stuffed toy, so we removed those entries too.

In contrast to the previous approach, we can also assume that an ad listing is related to wildlife trafficking if the product column contains certain keywords. We load the pre-defined most common keywords from the 'common_kws.txt' file and retain those ad listings that have any matching keywords from the file. When considering rows that contain any of the common keywords, the data is further reduced.

4.2.3 Keyword extraction. To extract the species from each row, we tokenize the product column in an attempt to find relevant keywords. However, the column contains a lot of text, most of which is irrelevant to the species name. By removing these words, we remove the low-level information from the product description to give more focus to the species name.

To do this, we need to remove stopwords, numerals, and other irrelevant words. Stopwords are the most common words in any language (ex. articles, prepositions, pronouns, conjunctions, etc) and do not add much information to the text. Examples of a few stop words in English are “the”, “a”, “an”, “so”, and “what”. We also removed numerals, verbs, and other words that are not relevant to the species name. The removal of such words will not have any negative consequences for our task. It reduces the number of tokens and leads to a more accurate species extraction. We then populated a new column ’product_cleaned’ that contains the pre-processed text for species extraction.

4.2.4 Using TaxoNERD for Species detection. TaxoNERD is a domain-specific tool for recognizing taxon mentions in the biodiversity literature. We used the ’en_core_eco_biobert’ model to extract species names from the text in the pre-processed ’product_cleaned’ column.

However, Taxonerd does not perform well with long strings. So we removed some irrelevant words to make species extraction faster and more accurate. We then used Taxonerd on the cleaned text to extract species names.

We analyzed the result of using TaxoNERD for species extraction and concluded that most of the rows contain ’None’ values. Taxonerd also returned multiple species names for some of the rows. This might be an indication that Taxonerd is not the most appropriate tool for our task. Therefore, we move on to a different approach for species extraction, focusing more on string matching.

4.2.5 Using Fuzzy String Matching for Species detection. We used rapidfuzzy, a python library for comparing text based on the similarity of strings. The library uses Levenshtein distance to calculate the difference between two strings. It is an alternative to the fuzzy-wuzzy library and provides fast and efficient string matching and fuzzy string matching functions.

The extractOne function returns a tuple containing the best matching string and its similarity score. For our project, the similarity score is computed using the ’partial ratio’.

Partial Ratio: It finds the ratio similarity measure between the shorter string and every substring of the longer string, and returns the maximum of those similarity measures. It searches for the optimal alignment of the shorter string in the longer string and returns the fuzz.ratio for this alignment.

We have the ’final_animal_list.csv’ file that contains 8130 entries of endangered animal species names. This file has been computed using the animals list from the ’CITES’ website. It contains normalized species names which we used for string matching. We checked if our dataset contains any of the animals listed here. Using fuzzy string matching, we compared the long text in the ’product_cleaned’ column and found if any part of that text matches with anything in the animal list.

The ’product_cleaned’ might have more than one potential match, but we will consider the best match out of all the potential matches. We set our fuzz.ratio to 95 as we are trying to find the exact species name.

From the results of fuzzy string matching, we were able to identify the unique species of endangered animals traded online. We were also able to analyze the top ten species traded online and their geographical and price distribution.

4.3 Image identification and classification

In our methodology, we have chosen various deep learning models like ResNet50, VGG16, EfficientNetV2M and InceptionV3 for the classification of animal species in our dataset. This approach is advantageous in handling the unstructured and unlabeled nature of our data, enabling us to extract meaningful insights and accurate classifications without the need for prelabeled training data.

Before applying the deep learning models, we need an ’image’ column to classify species, which is the image URL provided by the page’s metadata. We found out that the image is not directly relevant to the product advertised. We also discovered that even though the domains differ, the image URL is duplicated. I assume the advertisement is posted by a bot on multiple websites. We will remove the duplicate image URLs; however, at this stage, we will not address the irrelevant images. We will assume that the image URL represents the "most relevant image. We will eliminate duplicates based on the 'image' field, retaining the first occurrence associated with the longest text. This approach is grounded in the assumption that longer text yields more information for species identification. To implement this, we will assess text length in the following order: description, product, title, and name. After that, we will download the images and identify the specific images that failed to download and attempt to download only those images again, without re-downloading the images that were downloaded successfully.

4.3.1 Comparison of Deep Learning Models. To extract animal species, we use four deep learning models to predict the content of the images downloaded from the 'image' column, focusing solely on the top-1 prediction. The ImageNet dataset is one of the most widely used datasets for training deep learning models in image recognition tasks, providing a diverse range of images.

Each of the models — ResNet50, VGG16, EfficientNetV2M, and InceptionV3 — were typically trained on the ImageNet dataset.

- `ResNet50`: Trained on the ImageNet database, which contains over 14 million images categorized into more than 20,000 categories. It was part of the model's success in the ILSVRC (ImageNet Large Scale Visual Recognition Challenge).
- `VGG16`: Also trained on the ImageNet dataset. The VGG16 model was one of the top performers in the 2014 ILSVRC competition, mainly due to its deep architecture trained on this extensive and varied dataset.
- `EfficientNetV2M`: The EfficientNet models, including the V2 series, were trained on ImageNet. The EfficientNetV2 models introduced additional improvements and optimizations, but they continued to leverage the ImageNet dataset for training, which provides a broad and challenging range of image recognition tasks.
- `InceptionV3`: Like the others, InceptionV3 was trained on the ImageNet dataset. Its unique architecture was particularly well-suited to the diverse and complex images found in this dataset, allowing it to achieve high accuracy in classification tasks.

We encountered '404 Not Found' errors or inaccessible pages at some image URLs, necessitating the removal of null entries before proceeding with the preprocessing of extracted terms.

When we examine the unique items in the column, we notice that when a term consists of more than two words, these words are connected with an underscore '_'.

Since our goal is to identify animal species, we will extract the last word in multi-word terms like 'great_white_shark' (in this case, 'shark') and replace the underscores with spaces.

The new column 'combined_keyword' contains the most overlapped second element (keyword) from each row of the ResNet50_, VGG16_, EfficientNetV2M_, and InceptionV3_ columns. If there's a clear majority, only the most common keyword is selected; otherwise, all unique keywords are listed.

- For rows where one keyword is dominant (like 'shark' or 'bear'), only that keyword is listed. For rows with diverse keywords (like different animal names), all unique keywords are combined.
- For rows with diverse keywords (like different animal names), all unique keywords are combined.

The analysis revealed interesting insights about the top-1 predictions made by these four deep learning models.

- **One Common Keyword (11,594 rows):** This represents approximately 80.8% of the cases. In these instances, all four models either agreed on a single prediction or had no more than one unique prediction after processing the second element of each prediction. This high level of agreement suggests that for the majority of the data, the models are consistent in their predictions, indicating reliability and certainty in these particular instances.
- **More Than Two Keywords (2,825 rows):** This accounts for approximately 19.7% of the cases.

4.3.2 Extract Keywords Relevant to Animal Species. We utilized four deep learning models to predict the content of the images; however, we were unable to determine whether the content was related to animals. Therefore, it is necessary to define a list of keywords considered relevant to animal species. This list will be used to filter and extract the relevant keywords from each entry in the 'combined_keyword' column. The list, provided by CITES (Convention on International Trade in Endangered Species of Wild Fauna and Flora), enabled us to identify specific names of endangered animals and the countries where they reside.

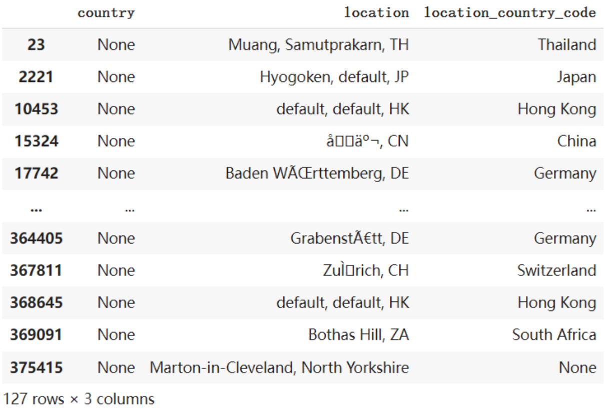
The list is first filtered by the Kingdom 'Animalia', as we will only consider animals, not plants, and we make a species name list from "EnglishNames". The list obtained from CITES contained some duplicate names, so we also performed normalization on them. We also utilized this list to identify general species by extracting the last word; for instance, by transforming 'Abaco Island Boa' into 'Boa', we were able to gain insights into broader animal species categories. We can conclude that, according to CITES (the Convention on International Trade in Endangered Species of Wild Fauna and Flora), there are 906 general animal species classified as endangered.

Here we use fuzzy matching (String Similarity). Fuzzy matching algorithms, particularly those used in libraries like rapidfuzz or fuzzywuzzy, are generally designed to handle variations in case (uppercase vs. lowercase) and can effectively manage minor typos or variations in the spelling of words. This capability is part of what makes fuzzy matching useful for comparing strings where exact matches are not expected due to inconsistencies or human error in the data. One of the most common metrics used in fuzzy matching is the Levenshtein Distance. It measures the minimum number of single-character edits required to change one word into another. The lower the number, the more similar the two strings are. By comparing each row of the list with our predicted keywords from the four deep learning models, we were able to identify the best matches.

As a result of the image classification process and the subsequent comparison with the animal list using the fuzzy string matching method, we were able to gain insights into the general and specific species in the images. Additionally, this approach enabled us to understand the distribution of these species across different countries.

5 RESULTS

5.1 Data cleaning



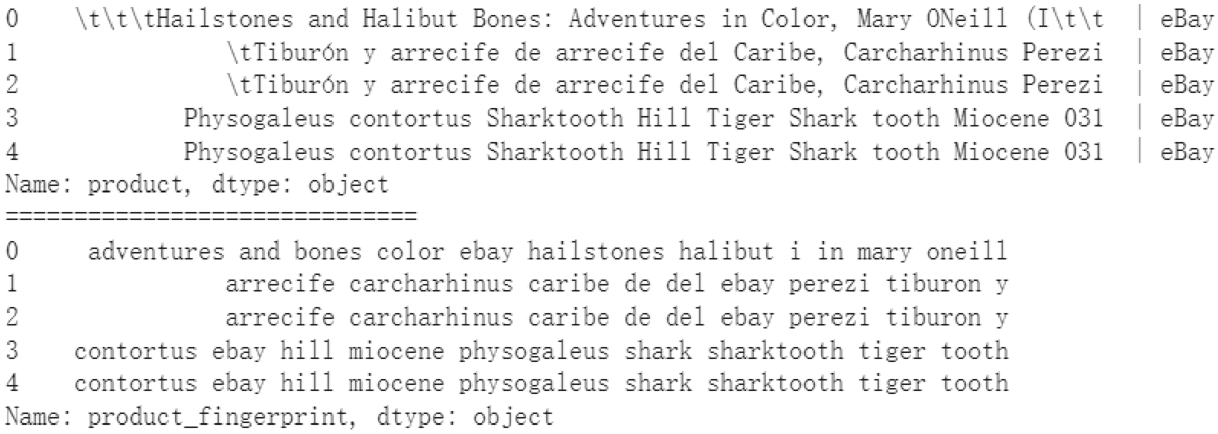
	country	location	location_country_code
23	None	Muang, Samutprakarn, TH	Thailand
2221	None	Hyogoken, default, JP	Japan
10453	None	default, default, HK	Hong Kong
15324	None	å□□ä°¬, CN	China
17742	None	Baden WÃŒrttemberg, DE	Germany
...	...	...	...
364405	None	GrabenstÃ€tt, DE	Germany
367811	None	ZuÌ□rich, CH	Switzerland
368645	None	default, default, HK	Hong Kong
369091	None	Bothas Hill, ZA	South Africa
375415	None	Marton-in-Cleveland, North Yorkshire	None

127 rows × 3 columns

Fig. 1. Processed Country Name

We can see from the Figure 1 that locations in different languages have been correctly set to correspond to the country name.

5.2 Text standardization

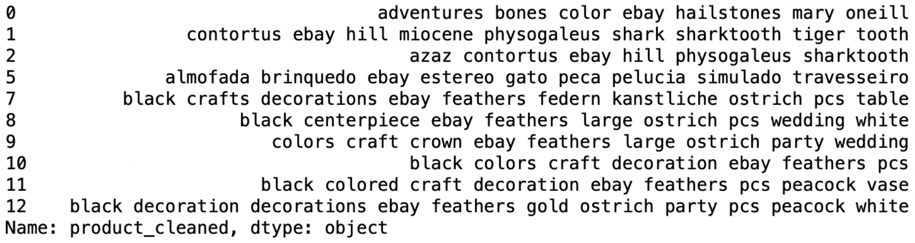


```
0    \t\t\tHailstones and Halibut Bones: Adventures in Color, Mary ONeill (I\t\t  | eBay
1              \tTiburón y arrecife de arrecife del Caribe, Carcharhinus Perezi  | eBay
2              \tTiburón y arrecife de arrecife del Caribe, Carcharhinus Perezi  | eBay
3          Physogaleus contortus Sharktooth Hill Tiger Shark tooth Miocene 031  | eBay
4          Physogaleus contortus Sharktooth Hill Tiger Shark tooth Miocene 031  | eBay
Name: product, dtype: object
==============================
0     adventures and bones color ebay hailstones halibut i in mary oneill
1               arrecife carcharhinus caribe de del ebay perezi tiburon y
2               arrecife carcharhinus caribe de del ebay perezi tiburon y
3    contortus ebay hill miocene physogaleus shark sharktooth tiger tooth
4    contortus ebay hill miocene physogaleus shark sharktooth tiger tooth
Name: product_fingerprint, dtype: object
```

Fig. 2. Text Standardization

We can see that leading or trailing whitespaces have been removed and the information have been converted to lowercase. Punctuation and control characters have also been removed. The result after text standardization is more clear and easy to read.

5.3 Data pre-processing for keyword extraction

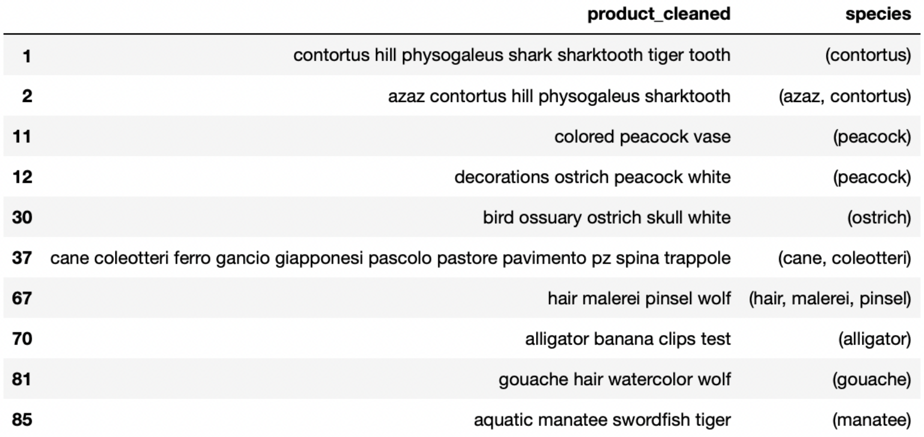


```
0                                   adventures bones color ebay hailstones mary oneill
1                  contortus ebay hill miocene physogaleus shark sharktooth tiger tooth
2                                      azaz contortus ebay hill physogaleus sharktooth
5                almofada brinquedo ebay estereo gato peca pelucia simulado travesseiro
7           black crafts decorations ebay feathers federn kanstliche ostrich pcs table
8                       black centerpiece ebay feathers large ostrich pcs wedding white
9                          colors craft crown ebay feathers large ostrich party wedding
10                                     black colors craft decoration ebay feathers pcs
11                         black colored craft decoration ebay feathers pcs peacock vase
12    black decoration decorations ebay feathers gold ostrich party pcs peacock white
Name: product_cleaned, dtype: object
```

Fig. 3. 'product' column after retaining just Nouns, Proper Nouns and Adjectives

We can see that by removing stop words, numerals and other irrelevant words, the text in the 'product' column have become simpler and contain more compact information of species.

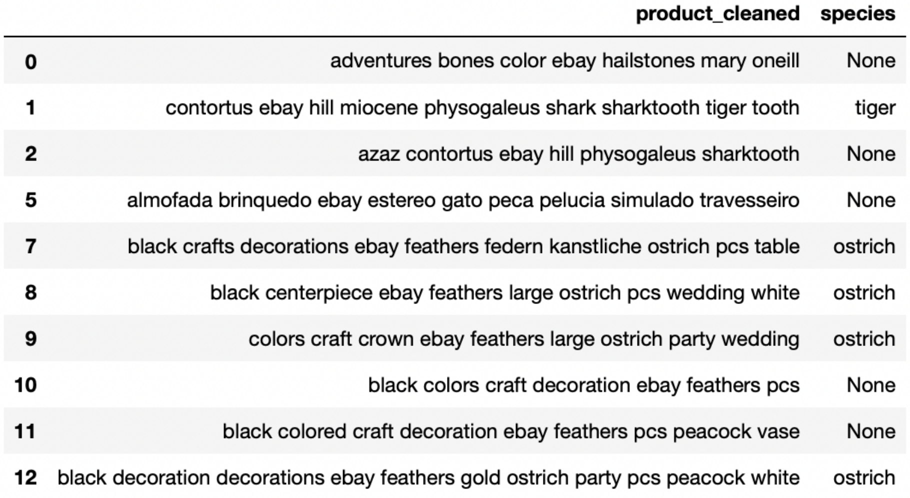
5.4 Species Extraction using NLP



	product_cleaned	species
1	contortus hill physogaleus shark sharktooth tiger tooth	(contortus)
2	azaz contortus hill physogaleus sharktooth	(azaz, contortus)
11	colored peacock vase	(peacock)
12	decorations ostrich peacock white	(peacock)
30	bird ossuary ostrich skull white	(ostrich)
37	cane coleotteri ferro gancio giapponesi pascolo pastore pavimento pz spina trappole	(cane, coleotteri)
67	hair malerei pinsel wolf	(hair, malerei, pinsel)
70	alligator banana clips test	(alligator)
81	gouache hair watercolor wolf	(gouache)
85	aquatic manatee swordfish tiger	(manatee)

Fig. 4. Species extraction using TaxoNERD

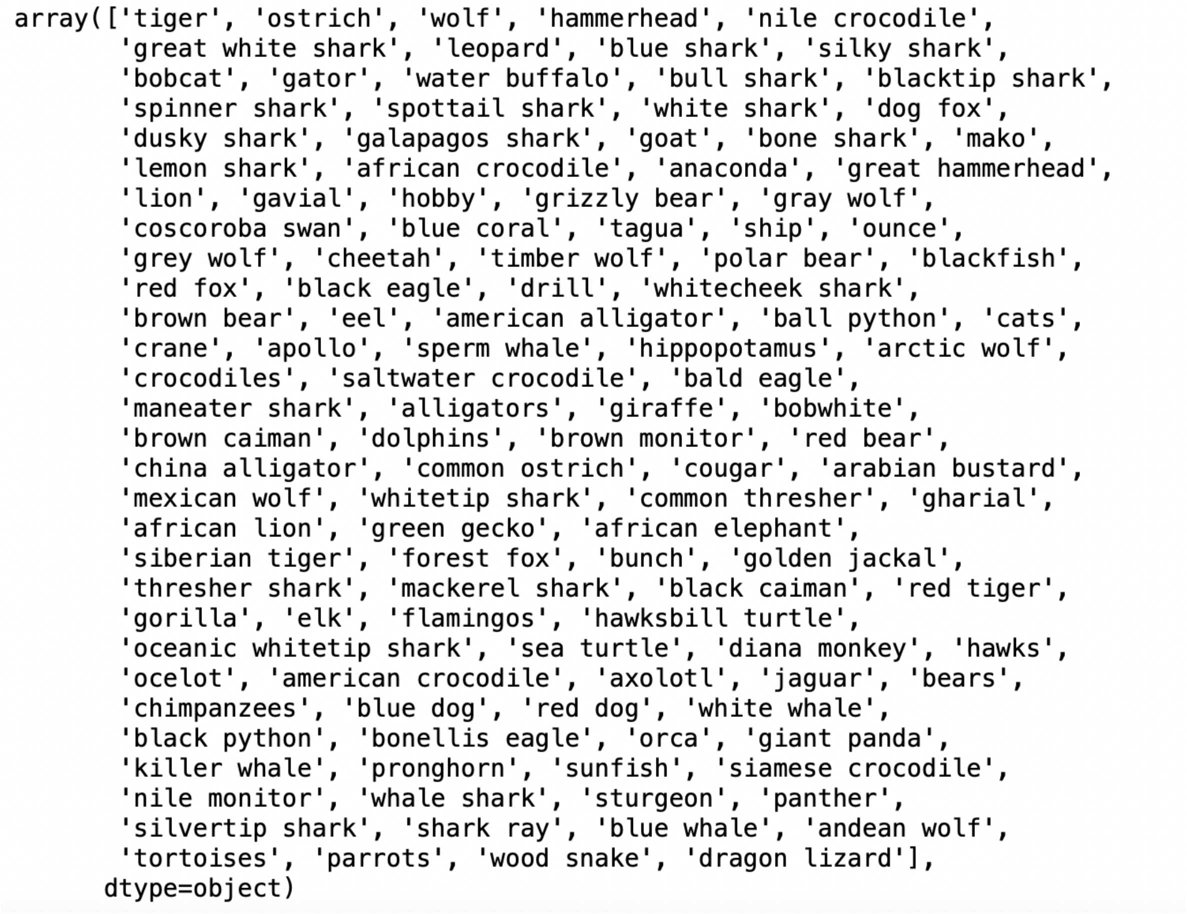
TaxoNERD was able to extract the species name for about half of the dataset. However, from the result, we can see that it is sometimes not accurate.



	product_cleaned	species
0	adventures bones color ebay hailstones mary oneill	None
1	contortus ebay hill miocene physogaleus shark sharktooth tiger tooth	tiger
2	azaz contortus ebay hill physogaleus sharktooth	None
5	almofada brinquedo ebay estereo gato peca pelucia simulado travesseiro	None
7	black crafts decorations ebay feathers federn kanstliche ostrich pcs table	ostrich
8	black centerpiece ebay feathers large ostrich pcs wedding white	ostrich
9	colors craft crown ebay feathers large ostrich party wedding	ostrich
10	black colors craft decoration ebay feathers pcs	None
11	black colored craft decoration ebay feathers pcs peacock vase	None
12	black decoration decorations ebay feathers gold ostrich party pcs peacock white	ostrich

Fig. 5. Species extraction using Rapid fuzz String matching

Fuzzy String Matching gave more accurate results and it was able to extract the species of animals that are illegal to trade and are in danger of extinction.



```
array(['tiger', 'ostrich', 'wolf', 'hammerhead', 'nile crocodile',
       'great white shark', 'leopard', 'blue shark', 'silky shark',
       'bobcat', 'gator', 'water buffalo', 'bull shark', 'blacktip shark',
       'spinner shark', 'spottail shark', 'white shark', 'dog fox',
       'dusky shark', 'galapagos shark', 'goat', 'bone shark', 'mako',
       'lemon shark', 'african crocodile', 'anaconda', 'great hammerhead',
       'lion', 'gavial', 'hobby', 'grizzly bear', 'gray wolf',
       'coscoroba swan', 'blue coral', 'tagua', 'ship', 'ounce',
       'grey wolf', 'cheetah', 'timber wolf', 'polar bear', 'blackfish',
       'red fox', 'black eagle', 'drill', 'whitecheek shark',
       'brown bear', 'eel', 'american alligator', 'ball python', 'cats',
       'crane', 'apollo', 'sperm whale', 'hippopotamus', 'arctic wolf',
       'crocodiles', 'saltwater crocodile', 'bald eagle',
       'maneater shark', 'alligators', 'giraffe', 'bobwhite',
       'brown caiman', 'dolphins', 'brown monitor', 'red bear',
       'china alligator', 'common ostrich', 'cougar', 'arabian bustard',
       'mexican wolf', 'whitetip shark', 'common thresher', 'gharial',
       'african lion', 'green gecko', 'african elephant',
       'siberian tiger', 'forest fox', 'bunch', 'golden jackal',
       'thresher shark', 'mackerel shark', 'black caiman', 'red tiger',
       'gorilla', 'elk', 'flamingos', 'hawksbill turtle',
       'oceanic whitetip shark', 'sea turtle', 'diana monkey', 'hawks',
       'ocelot', 'american crocodile', 'axolotl', 'jaguar', 'bears',
       'chimpanzees', 'blue dog', 'red dog', 'white whale',
       'black python', 'bonellis eagle', 'orca', 'giant panda',
       'killer whale', 'pronghorn', 'sunfish', 'siamese crocodile',
       'nile monitor', 'whale shark', 'sturgeon', 'panther',
       'silvertip shark', 'shark ray', 'blue whale', 'andean wolf',
       'tortoises', 'parrots', 'wood snake', 'dragon lizard'],
      dtype=object)
```

Fig. 6. Unique species traded online

We found a total of 123 unique species that are traded online. These are part of the endangered species from the CITES list of animals threatened by over-exploitation through international trade.

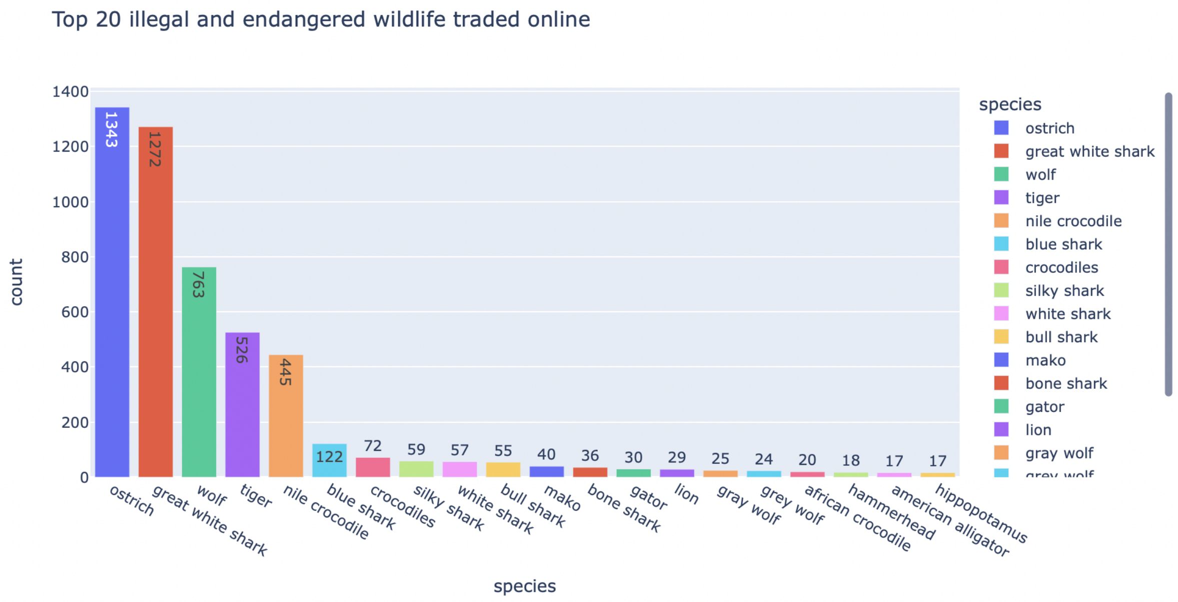


Fig. 7. Top 20 animals traded online

From the result, we can conclude that ostrich and great white shark have the highest count. This is followed by wolf, tiger, and nile crocodile. The other species are fewer in comparison.

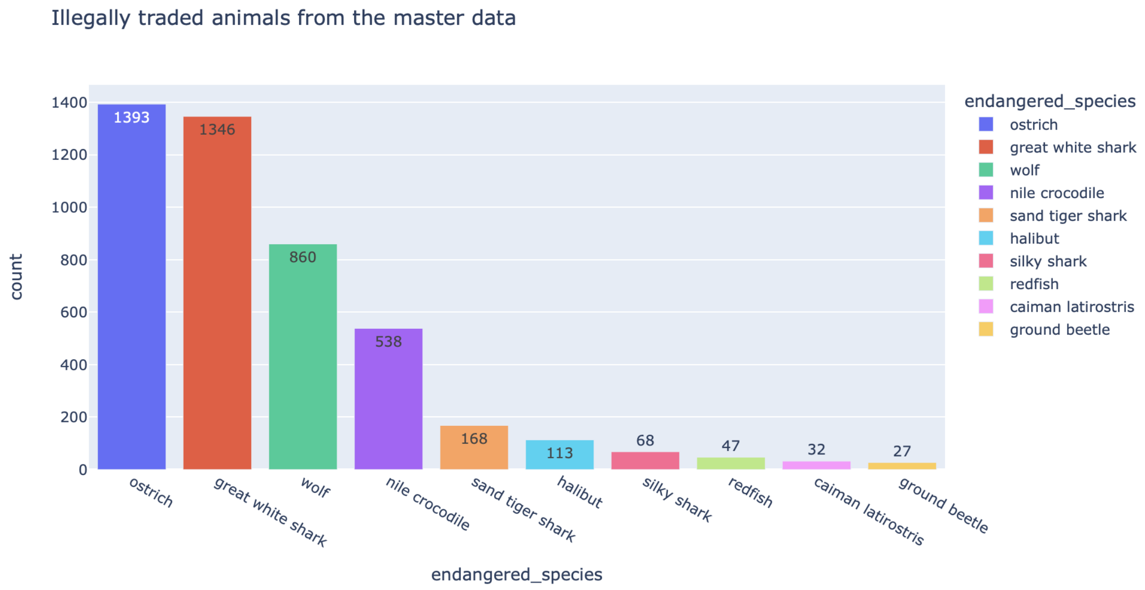


Fig. 8. Top 10 animals from the master data

The count of endangered animals from the master data is almost similar to the result from fuzzy string matching.

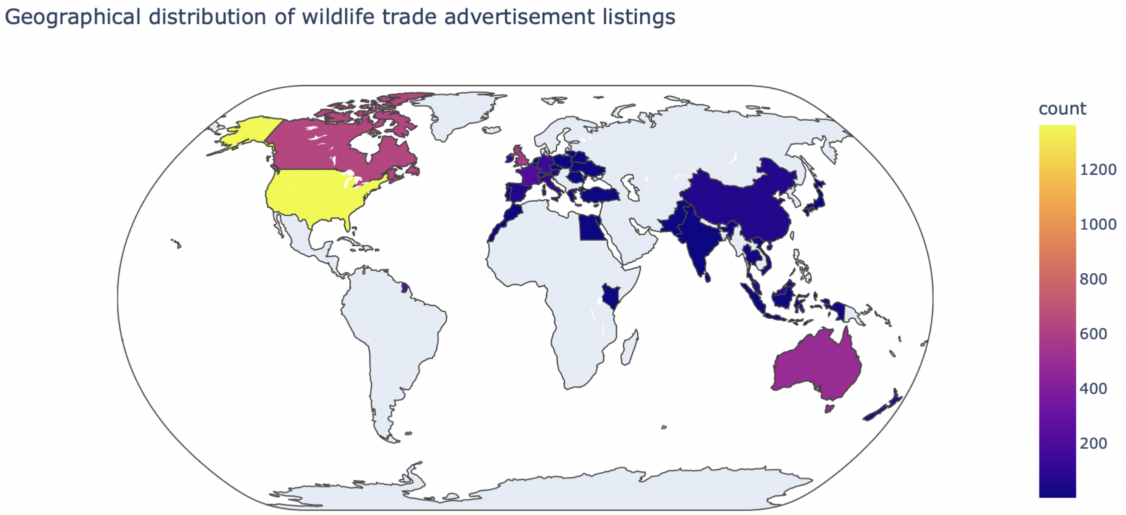

Geographical distribution of wildlife trade advertisement listings
count
1200
1000
800
600
400
200

Fig. 9. Geographical distribution of the illegal wildlife trade

Most of the wildlife trading is from the United States and Canada, followed by the United Kingdom and Australia.

5.5 Species Extraction using Image Classification

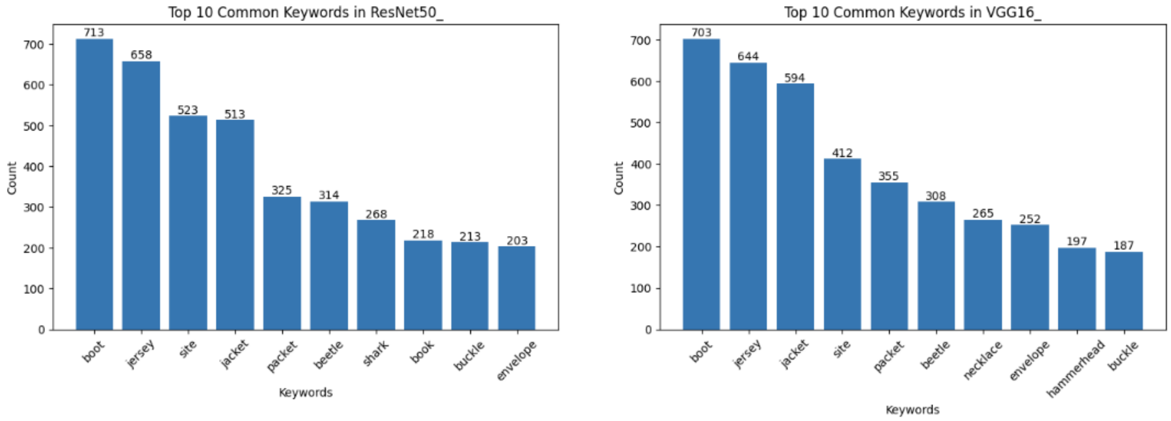
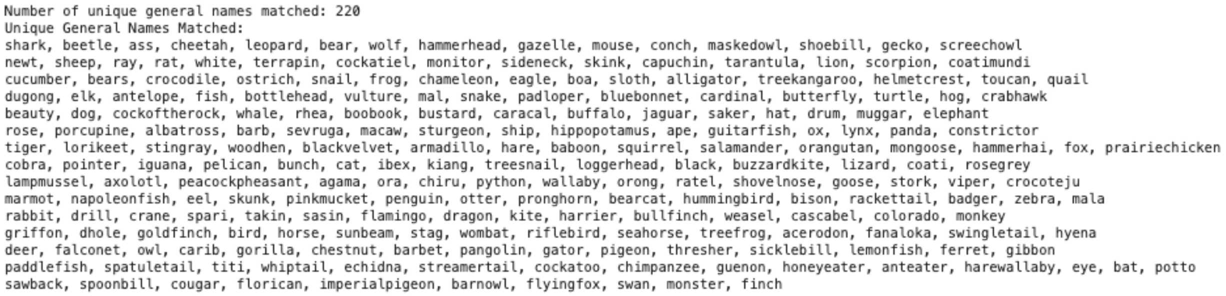


Fig. 10. Mix of Irrelevant and Relevant Animal Names

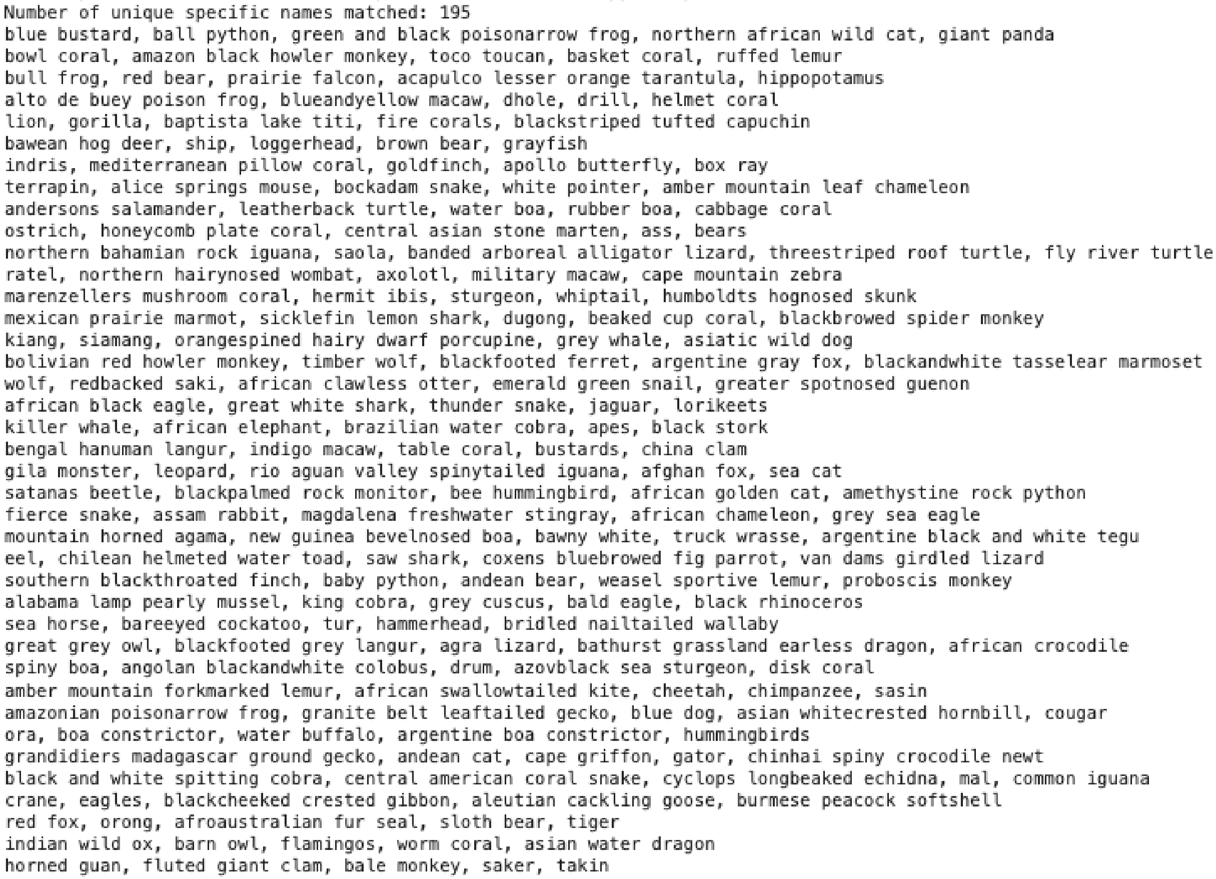
We discovered that non-relevant animal keywords were predicted solely by the deep learning models before merging with the animal list. This proves that multiple images were identified as non-relevant animals, even after we limited labels to ’animal body part’ and ’a real animal’ during data preprocessing.

After merging the animal list with keywords predicted by the four deep learning models from image data, we were able to identify both general and specific species.



```
Number of unique general names matched: 220
Unique General Names Matched:
shark, beetle, ass, cheetah, leopard, bear, wolf, hammerhead, gazelle, mouse, conch, maskedowl, shoebill, gecko, screechowl
newt, sheep, ray, rat, white, terrapin, cockatiel, monitor, sideneck, skink, capuchin, tarantula, lion, scorpion, coatimundi
cucumber, bears, crocodile, ostrich, snail, frog, chameleon, eagle, boa, sloth, alligator, treekangaroo, helmetcrest, toucan, quail
dugong, elk, antelope, fish, bottlehead, vulture, mal, snake, padloper, bluebonnet, cardinal, butterfly, turtle, hog, crabhawk
beauty, dog, cockoftherock, whale, rhea, boobook, bustard, caracal, buffalo, jaguar, saker, hat, drum, muggar, elephant
rose, porcupine, albatross, barb, sevruga, macaw, sturgeon, ship, hippopotamus, ape, guitarfish, ox, lynx, panda, constrictor
tiger, lorikeet, stingray, woodhen, blackvelvet, armadillo, hare, baboon, squirrel, salamander, orangutan, mongoose, hammerhai, fox, prairiechicken
cobra, pointer, iguana, pelican, bunch, cat, ibex, kiang, treesnail, loggerhead, black, buzzardkite, lizard, coati, rosegrey
lampmussel, axolotl, peacockpheasant, agama, ora, chiru, python, wallaby, orong, ratel, shovelnose, goose, stork, viper, crocoteju
marmot, napoleonfish, eel, skunk, pinkmucket, penguin, otter, pronghorn, bearcat, hummingbird, bison, rackettail, badger, zebra, mala
rabbit, drill, crane, spari, takin, sasin, flamingo, dragon, kite, harrier, bullfinch, weasel, cascabel, colorado, monkey
griffon, dhole, goldfinch, bird, horse, sunbeam, stag, wombat, riflebird, seahorse, treefrog, acerodon, fanaloka, swingletail, hyena
deer, falconet, owl, carib, gorilla, chestnut, barbet, pangolin, gator, pigeon, thresher, sicklebill, lemonfish, ferret, gibbon
paddlefish, spatuletail, titi, whiptail, echidna, streamertail, cockatoo, chimpanzee, guenon, honeyeater, anteater, harewallaby, eye, bat, potto
sawback, spoonbill, cougar, florican, imperialpigeon, barnowl, flyingfox, swan, monster, finch
```

Fig. 11. Counts and Names of Unique General Species



```
Number of unique specific names matched: 195
blue bustard, ball python, green and black poisonarrow frog, northern african wild cat, giant panda
bowl coral, amazon black howler monkey, toco toucan, basket coral, ruffed lemur
bull frog, red bear, prairie falcon, acapulco lesser orange tarantula, hippopotamus
alto de buey poison frog, blueandyellow macaw, dhole, drill, helmet coral
lion, gorilla, baptista lake titi, fire corals, blackstriped tufted capuchin
bawean hog deer, ship, loggerhead, brown bear, grayfish
indris, mediterranean pillow coral, goldfinch, apollo butterfly, box ray
terrapin, alice springs mouse, bockadam snake, white pointer, amber mountain leaf chameleon
andersons salamander, leatherback turtle, water boa, rubber boa, cabbage coral
ostrich, honeycomb plate coral, central asian stone marten, ass, bears
northern bahamian rock iguana, saola, banded arboreal alligator lizard, threestriped roof turtle, fly river turtle
ratel, northern hairynosed wombat, axolotl, military macaw, cape mountain zebra
marenzellers mushroom coral, hermit ibis, sturgeon, whiptail, humboldts hognosed skunk
mexican prairie marmot, sicklefin lemon shark, dugong, beaked cup coral, blackbrowed spider monkey
kiang, siamang, orangespined hairy dwarf porcupine, grey whale, asiatic wild dog
bolivian red howler monkey, timber wolf, blackfooted ferret, argentine gray fox, blackandwhite tasselear marmoset
wolf, redbacked saki, african clawless otter, emerald green snail, greater spotnosed guenon
african black eagle, great white shark, thunder snake, jaguar, lorikeets
killer whale, african elephant, brazilian water cobra, apes, black stork
bengal hanuman langur, indigo macaw, table coral, bustards, china clam
gila monster, leopard, rio aguan valley spinytailed iguana, afghan fox, sea cat
satanas beetle, blackpalmed rock monitor, bee hummingbird, african golden cat, amethystine rock python
fierce snake, assam rabbit, magdalena freshwater stingray, african chameleon, grey sea eagle
mountain horned agama, new guinea bevelnosed boa, bawny white, truck wrasse, argentine black and white tegu
eel, chilean helmeted water toad, saw shark, coxens bluebrowed fig parrot, van dams girdled lizard
southern blackthroated finch, baby python, andean bear, weasel sportive lemur, proboscis monkey
alabama lamp pearly mussel, king cobra, grey cuscus, bald eagle, black rhinoceros
sea horse, bareeyed cockatoo, tur, hammerhead, bridled nailtailed wallaby
great grey owl, blackfooted grey langur, agra lizard, bathurst grassland earless dragon, african crocodile
spiny boa, angolan blackandwhite colobus, drum, azovblack sea sturgeon, disk coral
amber mountain forkmarked lemur, african swallowtailed kite, cheetah, chimpanzee, sasin
amazonian poisonarrow frog, granite belt leaftailed gecko, blue dog, asian whitecrested hornbill, cougar
ora, boa constrictor, water buffalo, argentine boa constrictor, hummingbirds
grandidiers madagascar ground gecko, andean cat, cape griffon, gator, chinhai spiny crocodile newt
black and white spitting cobra, central american coral snake, cyclops longbeaked echidna, mal, common iguana
crane, eagles, blackcheeked crested gibbon, aleutian cackling goose, burmese peacock softshell
red fox, orong, afroaustralian fur seal, sloth bear, tiger
indian wild ox, barn owl, flamingos, worm coral, asian water dragon
horned guan, fluted giant clam, bale monkey, saker, takin
```

Fig. 12. Counts and Names of Unique Specific Species

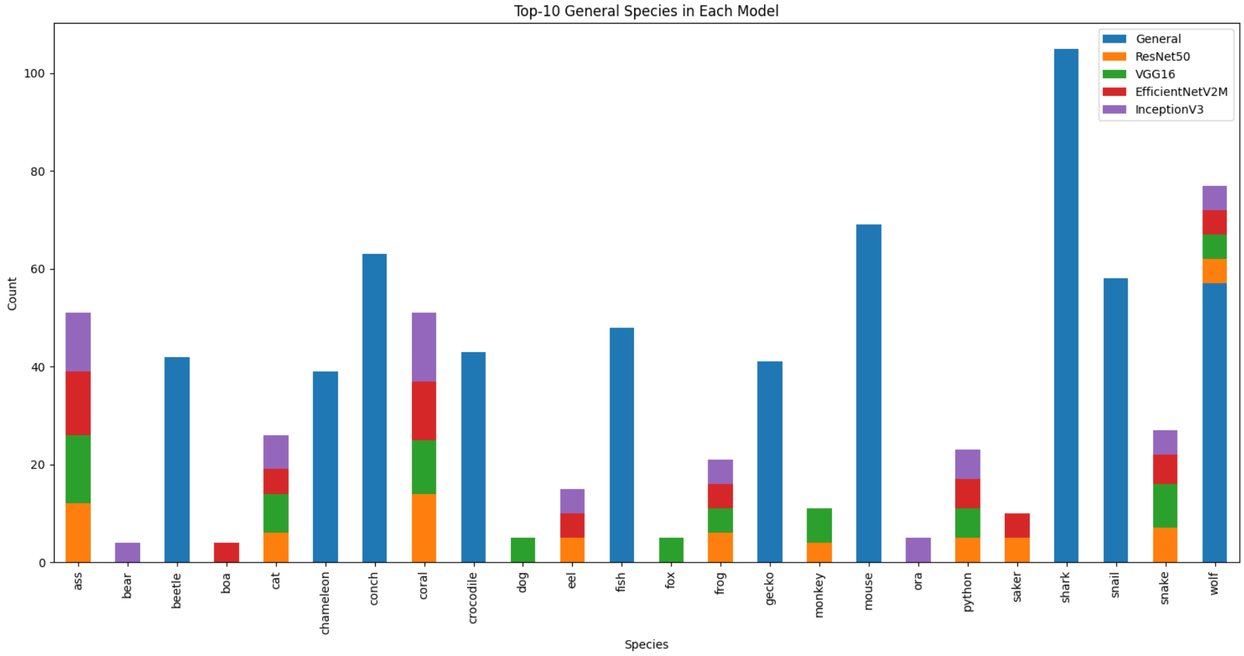


Fig. 13. Top 10 General Species in Each Model

From the results of the general species, we can see that shark takes up the most count of over 100. This is followed by mouse, conch and snail, each takes up about 60 counts. Beetle, chameleon, crocodile, fish and gecko each takes up about 40 counts. Species like bear, dog, fox have the least counts.

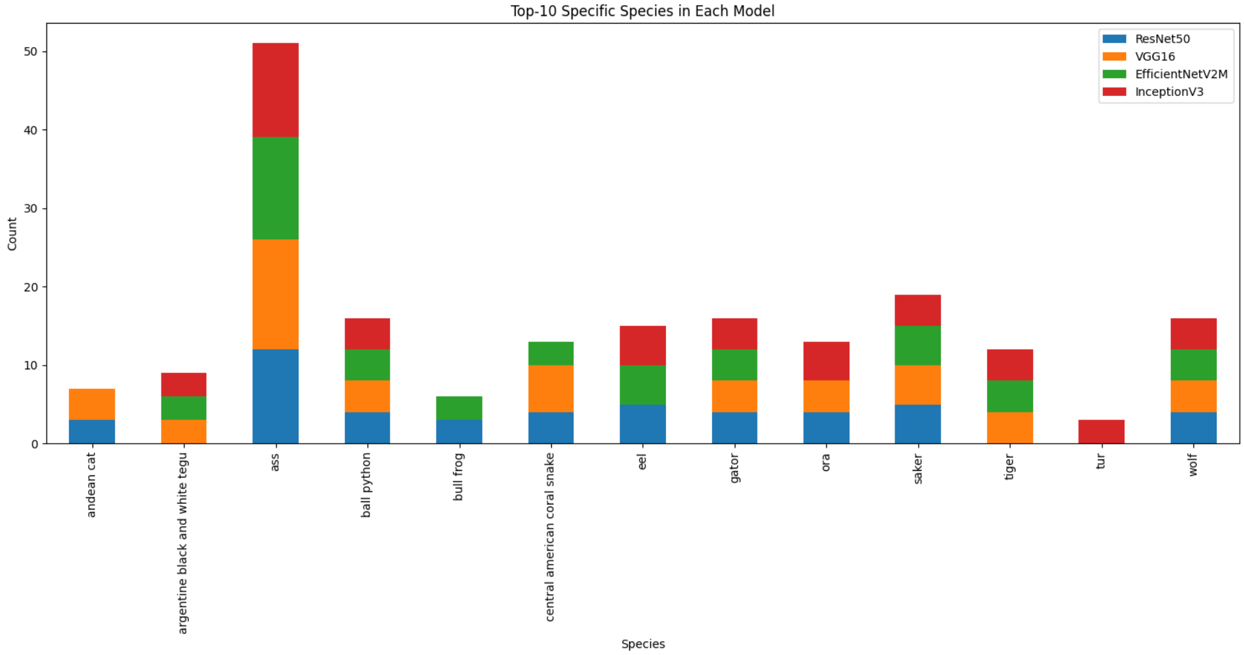


Fig. 14. Top 10 Specific Species in Each Model

For specific species, ass takes up the most counts of over 50, which is far more than other species. Ball python, eel, wolf, American coral snake each takes up about 10 to 20 counts. Species like bull frog and andean cat have the least counts of below 10.

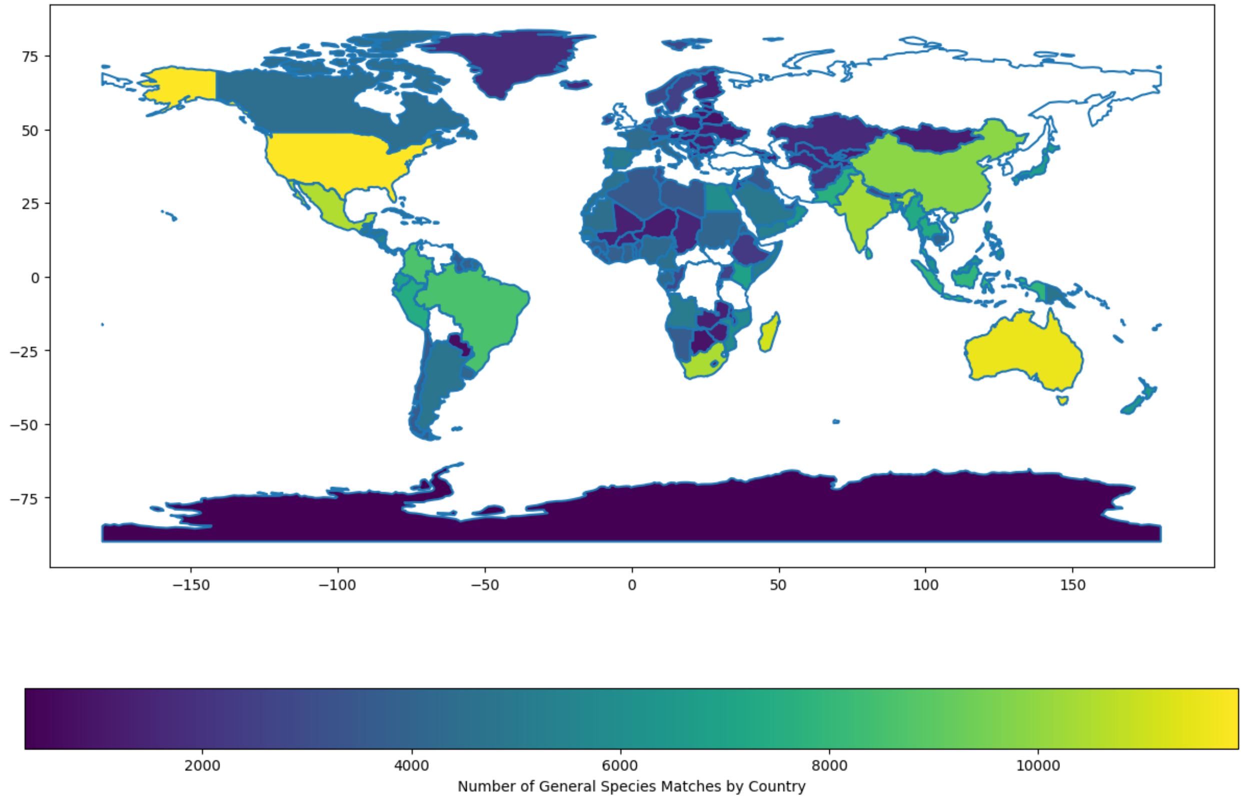


Fig. 15. Number of General Species Matched by Country

From the graph, we can see that United States and Australia have the most number of general species, followed by China, India, Madagascar and South Africa of more than 10000 number of general species. These countries might be biodiversity hot-spots. Greenland, Mongolia and Antarctica have the least number of below 2000 general species, which might indicate less biodiversity, less data available, or possibly a lower efficiency in species documentation. Other countries would have moderate levels of species matches.

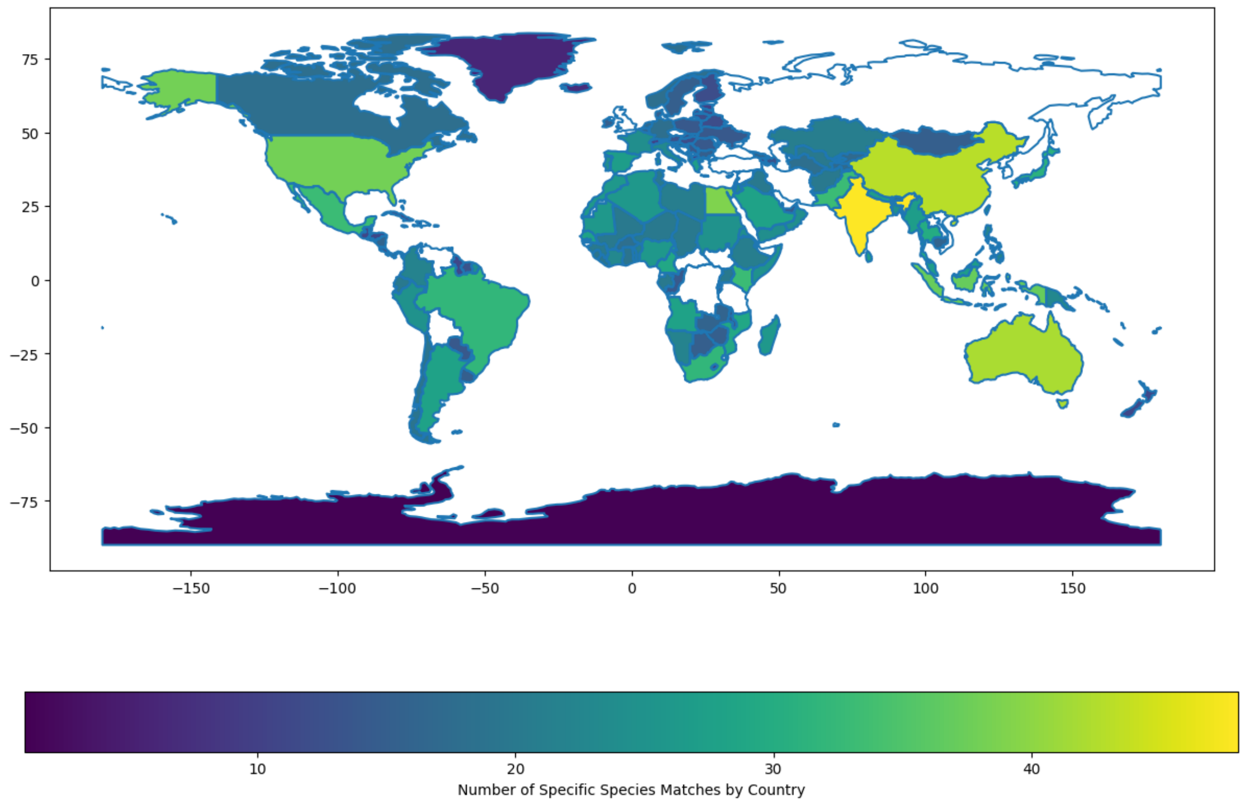


Fig. 16. Number of Specific Species Matched by Country

From this graph, we can see that India, China, Australia have the most number of specific species of over 40. This is followed by United States, Egypt, Brazil, Argentina an South Africa ranging from 30 to 40. Greenland, Mongolia and Antarctica have the least number of below 10. This could indicate less biodiversity within the same species due to extreme temperature.

5.6 Geographical distribution of the ads posted

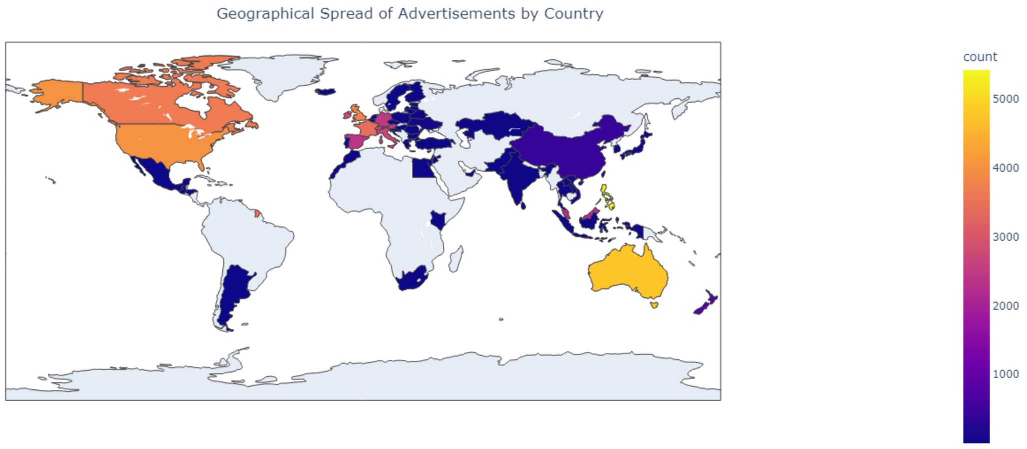


Fig. 17. Geographical distribution of the ads posted

From Figure 17, we can see that there are clear regional differences in the volume of advertisements. Southeast Asia has the most advertisements of over 5000, followed by Oceania with advertisements of nearly 5000. North America and West European and have volume of ads ranging from 2500 to 3500. Asia, East European, south region of South America and parts of Africa have volume of ads ranging from 0 to 500. This could suggest that Southeast Asia, North America, along with West European and Oceania have a more active market or a higher tendency to post advertisements in animal trading.

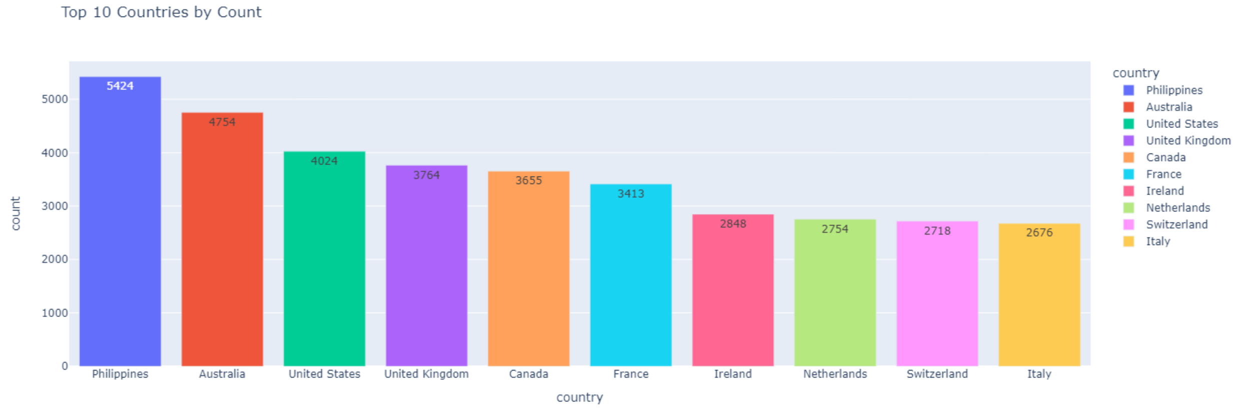


Fig. 18. Top 10 Countries by count

From this figure, we can see that Philippines and Australia have the highest count among the top ten countries, each contributing over 4500 counts. United States, United Kingdom ,Canada and France each contributes 3500 to 4000 counts. The other four countries have relatively smaller counts, with the Italy contributing the least among the top ten.

5.7 Price range of wildlife sold online

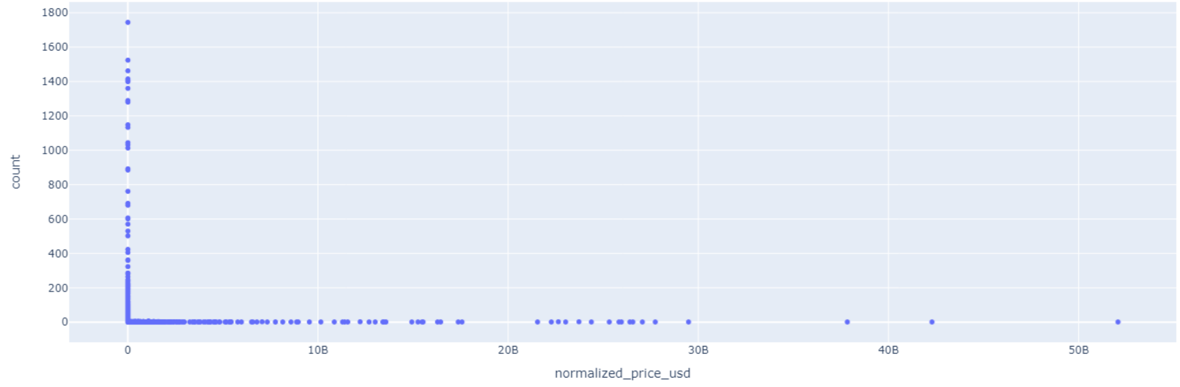


Fig. 19. Normalized Price in USD in All Price Ranges

From Figure 19, we can see that trading price focuses mainly below 17 billion, especially below 5 billion.

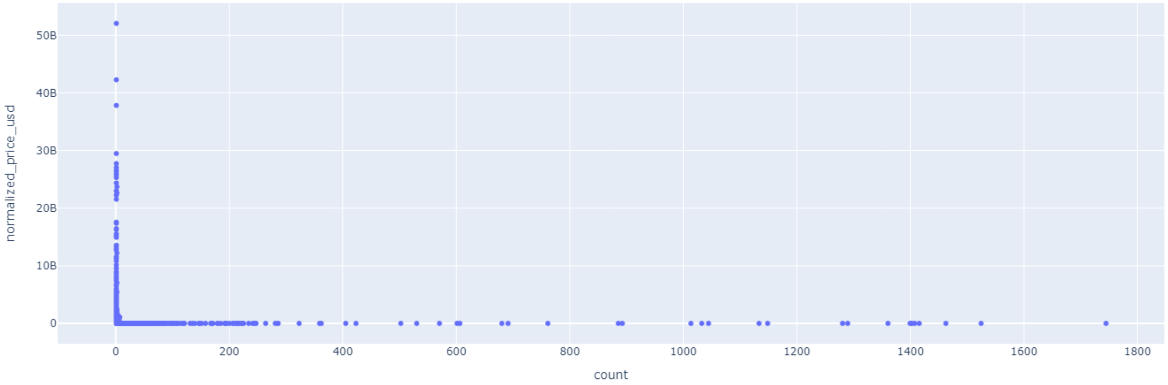


Fig. 20. Normalized Price in USD in All Count Ranges

From Figure 20, we can see that for each trading price, the trading count falls mostly below 300.

Considering the above situation, we focus on the data where the normalized price in usd is less than $130,000.

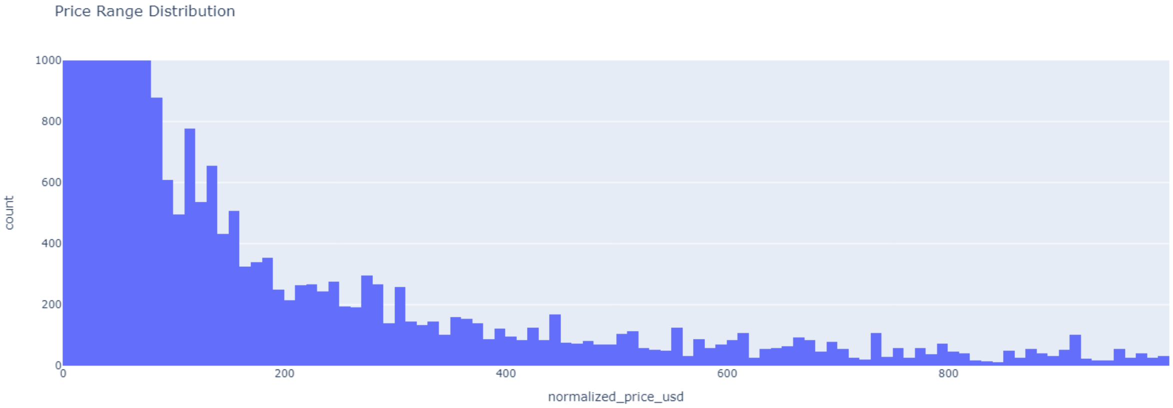


Fig. 21. Price Range Distribution below $130,000

Judging by the distribution of the price range, the trading price falls mostly in the range of 0 to 100 dollars and falls drastically between 100 to 200 dollars, the slowly declines after 200 dollars.

Since we observe a tight zone in the price range, we experimented by analyzing the data after excluding outliers to observe the distribution.

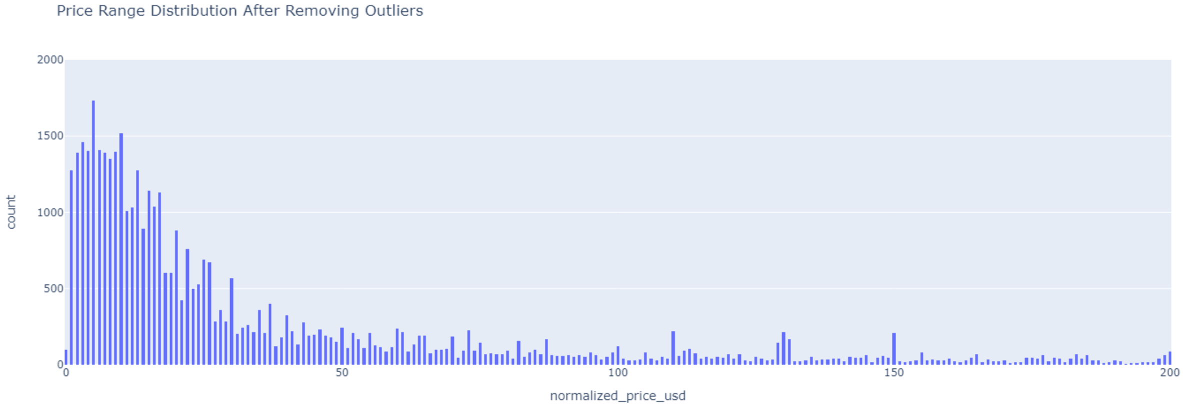


Fig. 23. Price Range Distribution below $130,000 after Removing Outliers

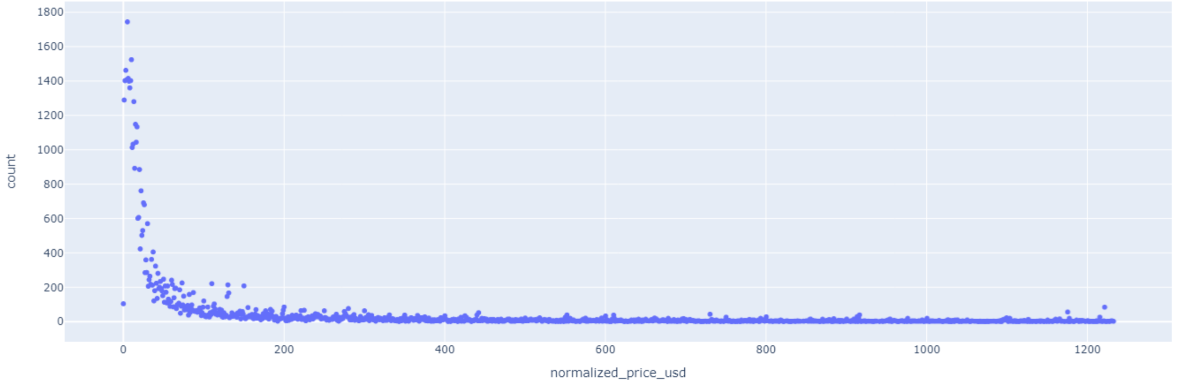


Fig. 22. Price Range after Removing Outliers

We can have a clearer view of the price range of the advertisements after removing the outliers. The majority of the advertisements falls in the range 0 to 100 dollars. After this range, the count of the advertisements falls below 50.

We can see a more accurate and fine-grained result of the trading price range after removing outliers. The result clearly shows a tight cluster of advertisements in 0 to 100 dollars, especially price under 50 dollars. We can also see a drastic decline of the advertisement counts when the price goes from 0 to 50, and a relatively slower decline from price 50 to 100.

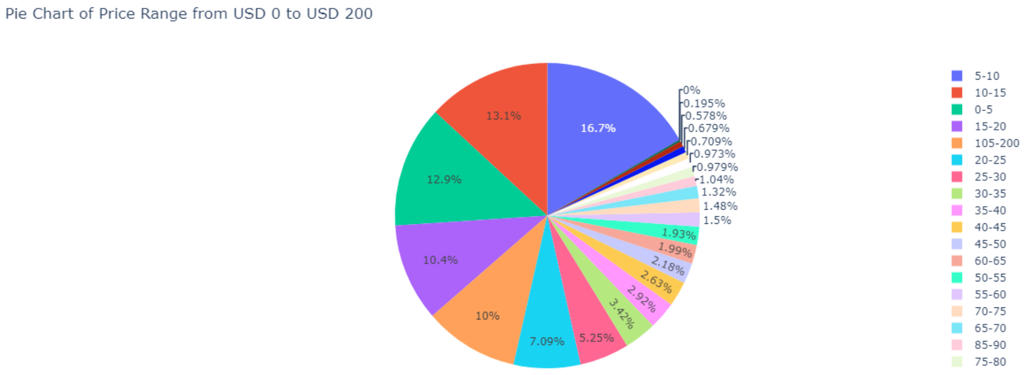


Fig. 24. Price Range from 0 to 200

Given the price distribution, we can focus on the advertisements with price in the range from 0 to 200 dollars. From Figure 24, we can see that the majority of the data falls within the 5 to 10 USD price range. Smaller percentages are distributed among higher price ranges, with some as low as below 1 percent, indicating these price ranges are less common in the dataset. Notably, the "105-200" USD range has a noticeable slice, suggesting a significant portion of the data is in this higher price range, though it's not as common as the lower ranges. The smallest slices, many under 1 percent, represent higher and more specific price ranges like "100-105", "200-300", and various increments between "75-300" USD. This effectively shows that lower price ranges are more common in the dataset, with fewer instances of higher-priced advertisements.

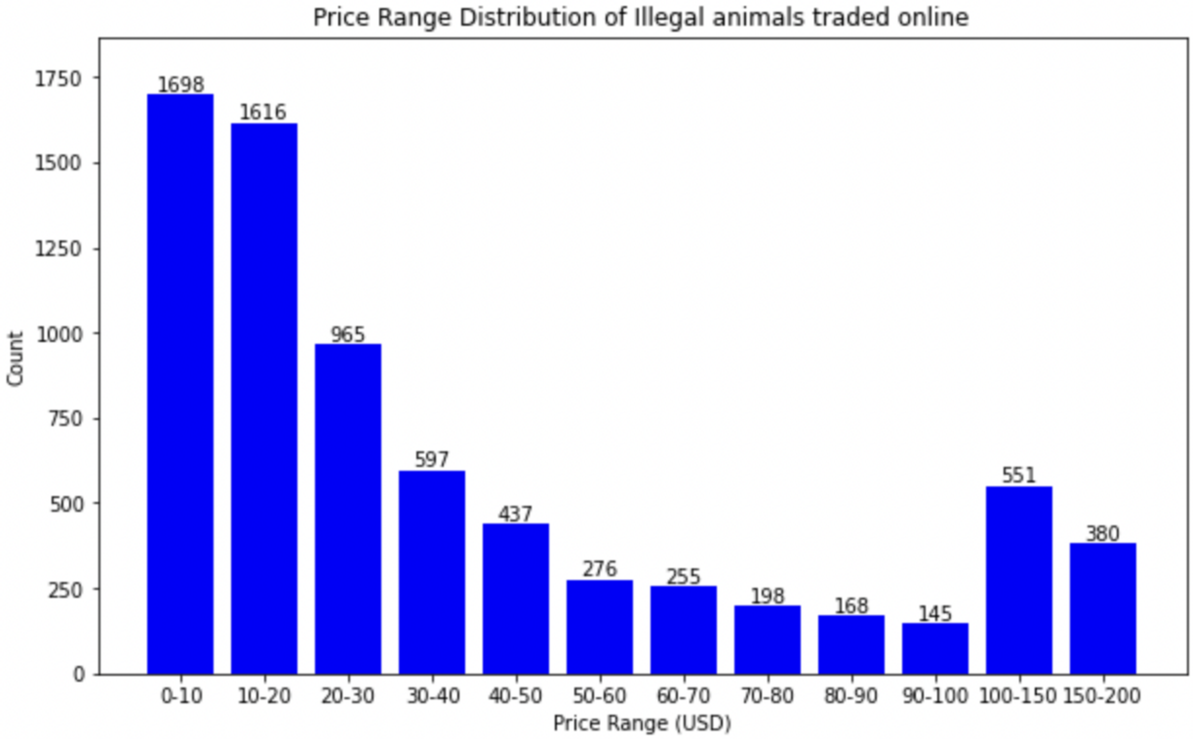


Fig. 25. Price Range of Illegally traded wildlife

Considering the animals that are illegal to trade and are in danger of extinction, we find that most of the listings for these trades fall in the price range of 0 to 20 USD.

6 FUTURE WORK

We have observed that images in the dataset sometimes do not accurately represent the advertised item but are instead associated with the user profile of the person who updated the page. Furthermore, while these pages often contain multiple images, our current approach only allows for the storage of a single image per ad. To address this limitation and rectify instances where animals are incorrectly represented, we are considering the use of a NoSQL database. This would enable us to include multiple images per ad, thereby significantly enhancing the accuracy of our species identification process. By leveraging these additional images and the URLs for improved image retrieval, our goal is to address the prevalent issue of misleading or irrelevant images, thereby ensuring the reliability and validity of our data for comprehensive analysis in the field of wildlife trafficking. We also verified the nearly 99 percent accuracy of ChatGPT 4 in extracting animal species from text and image data. This model can additionally assess whether the data is relevant to wildlife trafficking, based on provided text-based or image advertisements. Future work will involve using the OpenAI API to aid in the prevention of wildlife trafficking.

REFERENCES

[1] Thirupathi Battu and D. Sreenivasa Reddy Lakshmi. Animal image identification and classification using deep neural networks techniques. *Measurement: Sensors*, 25:100611, 2023. ISSN 2665-9174. doi: https://doi.org/10.1016/j.measen.2022.100611. URL https://www.sciencedirect.com/science/article/pii/S2665917422002458.

[2] Patrick A. V. Hall and Geoff R. Dowling. Approximate string matching. *ACM Comput. Surv.*, 12(4):381–402, dec 1980. ISSN 0360-0300. doi: 10.1145/356827.356830. URL https://doi.org/10.1145/356827.356830.

[3] Jing Jiang. *Information Extraction from Text*, pages 11–41. Springer US, Boston, MA, 2012. ISBN 978-1-4614-3223-4. doi: 10.1007/978-1-4614-3223-4_2. URL https://doi.org/10.1007/978-1-4614-3223-4_2.

[4] Elsa A. Olivetti, Jacqueline M. Cole, Edward Kim, Olga Kononova, Gerbrand Ceder, Thomas Yong-Jin Han, and Anna M. Hiszpanski. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, 7(4):041317, 12 2020. ISSN 1931-9401. doi: 10.1063/5.0021106. URL https://doi.org/10.1063/5.0021106.

[1–4]