Detecting Illegal Wildlife Trade on Twitter

Author

Gabriel Deml

Faculty Advisor

Kyumin Lee

Graduate Advisor

Guanyi Mou

An Interactive Qualifying Project
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirement for the
Degree of Bachelor of Science in Computer Science

Abstract

The Illegal Ivory Trade (IIT) is becoming an emerging concern in the world today. It is estimated that over 30,000 African elephants are killed by illegal trade each year. To help stop IIT, we investigated their existence and activities on Twitter. There have been attempts to do this, but they did not apply a strong machine learning model. Inspired by the belief that if poachers have no clients, they have no incentive to participate in IIT, we attempted to build frameworks that potentially remove the offending tweets. Insufficient data and crude algorithms limited prior works, thus not thoroughly exploring the possibility of an automatic deep learning framework for IIT detection. We, however, collected and annotated a sizable multimodal dataset (text, account profile, and image) of IIT tweets in this work. We then built a BERT-based deep learning model to identify these IIT tweets. Our model achieved an overall average accuracy of 94% and an average macro F1 score of 93% in a 10-fold cross-validation experiment setting on the dataset. We further provided some insights into the annotated data to shed more light on future works.

Acknowledgments

Firstly, I would like to thank Guanyi Mou, a graduate student at WPI, for his help in the development of this project. Secondly, I would also like to thank Kyumin Lee, PhD for advising me on the development of this project. Without either of them, this project would not have been possible.

Table of Contents

List of Figures

List of Tables

Introduction

Illegal wildlife trafficking (IIT) is an emerging problem causing global concern. In this research, we primarily focus on the ivory-related IIT (mainly ivory from elephants, sometimes the rhinos[3] and walruses). To fight against these trafficking behaviors, countries worldwide have established various policies and laws for preventing IIT. For example, the U.S. Fish and Wildlife Service (FWS) prohibits the "import and export of African elephant ivory with limited exceptions for musical instruments, items that are part of a traveling exhibition, and items that are part of a household move or inheritance when specific criteria are met; and ivory for law enforcement or genuine scientific purposes [1]."Despite these bans in these countries, people are still observing an average of 30,000 African elephants poached each year with a continent-wide population of only 400,000[1]. To help mitigate the problem, we searched for methods that automatically detect IIT-related postings on online social platforms (more specifically, Twitter). However, there are two major obstacles in such a task: 1) prior works lack enough data with annotated labels for training an effective model, and 2) prior work leveraged rather crude and low accuracy methods for effectively and efficiently detecting IIT postings [4].

Project Goal

There were two goals for this project:

- The first goal is to build a large enough dataset of IIT texts that such platforms can use as a seed dataset. One of our biggest challenges was generating our initial dataset to seed the model. We believe that this dataset will be large enough to be used by researchers in the future.
- The second goal is to demonstrate that it is possible to classify user-generated content automatically. We believe that this model could potentially be applied to a dynamic and high-speed platform. This paper focuses on Twitter, but the same principles should also apply to other platforms. We hope that it will demonstrate to media platforms that it is possible and practical to implement an algorithm that automatically removes posts promoting the sale of IIT. We also hope that this paper would be a good starting point for any platform implementing such an algorithm.

With these challenges in mind, in this work, we aim to make the following contributions[2]:

- We collect, annotate and analyze a novel multimodal dataset of IIT Tweets at scale.
- We created the different variations of BERT-based models for identifying IIT postings. Our best model obtained an average accuracy of 94% and an average macro F1 score of 93% under a 10-fold cross-validation setting.

[1] Is elephant poaching really declining?
[2] Details of our dataset and experiments are shown in https://github.com/GabrielDeml/IWT

Background

There is a monetary incentive for poaching elephants. Each ivory on a single elephant is worth about $18,000 or about $1,800 per kg. Even though that is just the wholesale value on the street, it is worth about $6,000 per kg of carved ivory or about $60,000 per elephant[3]. African ivory is more valuable than Asian ivory since Asian ivory yellows over time[4]. Ivory poaching is much more professional than it used to be [6]. Poachers are now well-equipped to catch the elephants, using everything between rocket launchers and helicopters [6]. Most recruits come from rural areas, and they are often hired by organized crime syndicates [6]. Frequently, even the military is involved in poaching elephants [6].

It can be hard for law enforcement to catch the poachers [6]. This is for a couple of reasons. The first is that once ivory enters the supply chain, it is hard to know if it is legal or not [2]. Elephants are outside of reserves and are constantly crossing borders [6].

The ivory on the black market is being sold in two different forms, raw and carved ivory. It also goes through several people before being sold to the end consumer [5]. The poachers are the first people in the chain. They take the ivory and sell it to carvers [5]. In this stage, the ivory is in a raw form [5]. Carvers are the second people in the chain. They are the artisans who take the ivory and carve it into works of art [5]. The ivory was often turned into items such as pendants, rings, bangles, guru beads, and simple figurines [7]. From there, a trader is the third person in the chain. In this stage, the ivory is carved. The traders are the people who sell the ivory to the end consumer [5].

According to Gao and Susan [7], consumers want ivory for a couple of reasons. The first is that ivory is typically used in traditional Chinese medicine. They believe that the ivory will remove toxins from the body. The second reason is decorative purposes, either jewelry or ornamental. The third is as an investment. Some think that the price of ivory will continue to rise, thus making it a good investment. Lastly, it has religious significance. The ivory is believed to bring good fortune, making it of intelligence. There are three different states of legality that ivory is being sold in. White, black, and gray markets. The first is the white or legal markets. These are usually factories or retail outlets and are well established. The white markets are regulated by the government. All white markets must hold licenses for the ivory being sold and operate within the law. The second, Black markets, are split up into two types online and unauthorized outlets. These markets are not regulated and unauthorized. The seller typically directs the client/customer to contact them directly. They then sell and transfer the ivory to the buyer. For the third, Gray markets, the ivory is frequently sold in live auctions. The legality of these is questionable. They

[3] Elephant Slaughter Escalates as Illegal Ivory Market Thrives | Animal Welfare Institute
[4] Detailed Discussion of Elephants and the Ivory Trade | Animal Legal & Historical Center

may be completely legit, but there is a good chance that they are not. The questionability comes from items claiming that they predate the ban on ivory harvesting.

It is hard to quantify the punishment for poaching elephants or trading ivory since many countries are involved. Also, some countries are corrupt, so the laws are not enforced. The punishment for poaching elephants ranges from a couple of thousand dollars to a lifetime in prison. Generally, the punishment is a couple of thousand dollars on the side of the spectrum[5].

Ivory trade happens over the internet [8]. Sometimes sellers of ivory use code words to mask what they are selling [9]. It is known that there are ivory sellers on eBay UK, France, Italy, and Spain [9]. There is also evidence that the ivory trade happens on Twitter [4].

[5] Detailed Discussion of Elephants and the Ivory Trade | Animal Legal & Historical Center

Related work

Xu et al. [4] did some work about automatically detecting illegal wildlife trade on Twitter and Facebook. They used a bi-term clustering algorithm to identify the illegal wildlife trade on Twitter [4]. They specifically were focused on elephant ivory and pangolin. To start, they did manual searches using keywords. The keywords were both known keywords and common code words for the targeted items. They used the Twitter API to collect 138,357 filtered based on the keywords for data collection. They used a bi-term model to cluster the tweets for data processing automatically. Once they had the tweets clustered, they could then label the tweets as illegal wildlife trade. They found 53 actual ivory trade tweets.

There has been a lot of work in NLP text classification, specifically BERT which we use in this paper[10]. BERT is a transformer based model that is used for various NLP tasks such as paraphrasing, question answering, and classification.

Sara and David [9] have looked into the usage of code words for ivory trading. Even across different countries speaking different languages, it was found that they were using the same code words. They found 19 code words used across linguistically different places.

Methodology

In this section, we will introduce our definitions and criteria for IIT-related postings. We then describe our methods for collecting, processing, and annotating our IIT-related posting data. Finally, we mention how we leveraged the collected IIT-related data through various deep learning frameworks.

Definition and Criteria

We define IIT-related postings through the following criteria: the black circles are criteria, and the white circles are typical examples that correspond to the specific criteria. A posting can be deemed IIT-related if it meets at least one of the criteria.

- Selling: An IIT-related product item could potentially be bought
    - The user says to go to their website to buy it
    - The user says to contact them for details
    - The user is a business that the item can be bought at based on the user description, E.g., auctioneer, antique dealer
- Buying: The person is trying to or has already bought IIT-related products
    - The user is asking to buy ivory
    - The user said that they bought a piece of ivory

Dataset

In this section, we will first discuss the creation of the dataset in three steps: seed tweet collection, candidate tweet collection, preprocessing, and tweet annotation.

Seed tweet collection: We were grateful to receive the userid of 10 seed users identified by prior work [9]. Furthermore, we were also provided with 9 IIT-related tweets posted by 9 out of the 10 users. Upon further investigation, we realized that 4 of them did not meet our criteria to be considered IIT-related. Thus we eventually identified 5 seed tweets from prior work.

Below is a detailed analysis of the 4 tweets that did not meet our criteria.

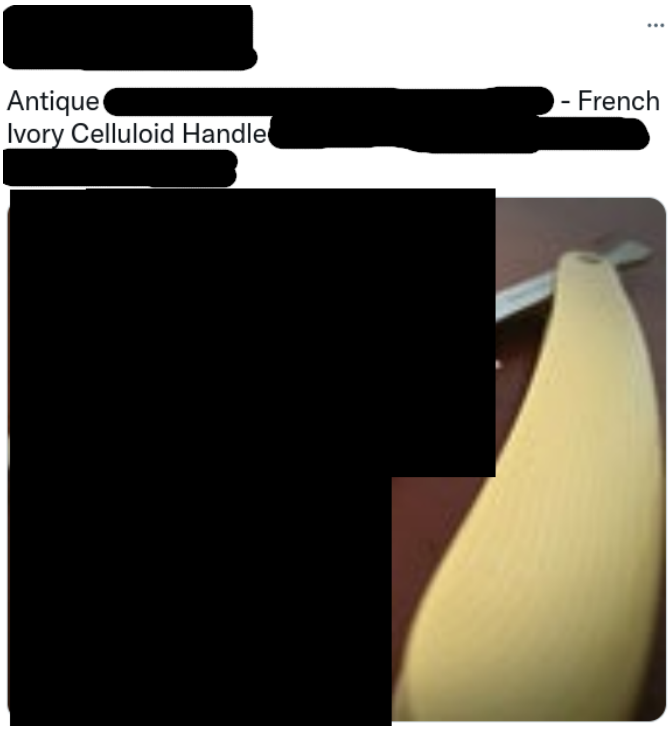

Figure 1. Seed Tweet Example 1.

Tweet #1: In Figure 1, the tweet clearly states that the item is made of “french ivory”. “French ivory’’ is a term for plastic made to resemble ivory. Furthermore, the image shows no Schreger lines typical for ivory products. We thus believe the current information is not sufficient to tell whether the product is IIT-related.

Tweet #2: We realized that the tweet text and image in Figure 1 was identical to a second tweet in the seed tweets. The tweet in Figure 1, along with its duplicate tweet, accounted for two of the incorrectly labeled seed tweets. The user that posted the tweets were different accounts, along with the tweet IDs being different. This makes us believe that the same person might control the two accounts.

Figure 2. Seed Tweet Example 2

Tweet #3: In Figure 2, we find no clear evidence that the tweet contains ivory. The text does not state that the item is made of ivory. The image also does not show any signs of ivory. We believe that the original labelers might have gotten confused by the white clock face. Upon further inspection, though, it is more likely that it is made of wood for the following reasons:

- The clock face shows no signs of Schreger lines.
- The clock face is too large to be made of ivory.

We believe the tweet in Figure 2 is not an IIT tweet for those reasons.

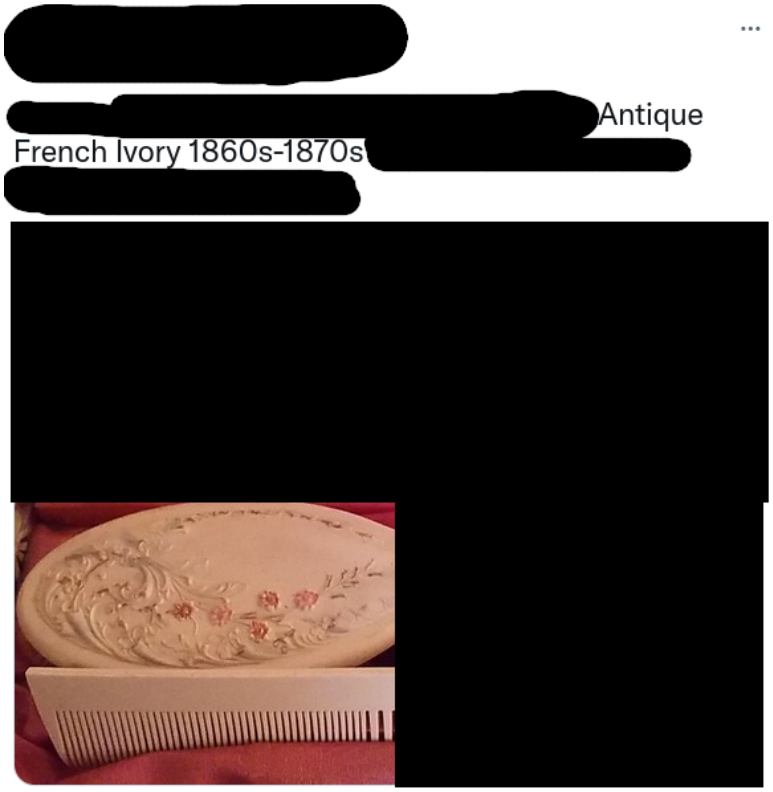

Figure 3. Seed Tweet Example 3

Tweet #4: Figure 3 clearly states that the item is made of French ivory. For the same reasons as in Figure 1, we believe that the tweet in Figure 3 is not an IIT tweet.

Candidate tweet collection: We then recovered the timeline (i.e., sequence of posted tweets) of the 10 seed users and their 13,377 followers, treating them as possible IIT-related tweets. Such a method is based on the assumption that a possible seller of IIT-related products may post more than once and that the followers of an IIT-related product seller have a higher chance of being other sellers or buyers.

With each timeline containing at most 3,200 tweets, the overall number of possible IIT-related tweets can be huge. Manually labeling all these tweets can be labor-intensive and time-consuming. We thus adopted a keyword checklist[6] method for filtering these tweets. We only keep those tweets containing at least one of the keywords in the checklist. In addition, we also add 240 random tweets from Twitter, with each tweet from a unique random Twitter user

[6] The keyword list is shared in our github repo.

account. For each tweet, we recovered the 1) **tweetid** 2) **tweet text content** 3) **images (if available) posted with the tweet**, and **4) the tweet owner's profile description in the text.** A processed dataset is shared in our github repo.

Preprocessing: Before we introduce the data to the labeling phase, we need to clean the text to respect the user's privacy (i.e., make it infeasible for people to backtrack and find the original tweets based on the given material). We also presume that our text preprocessing method could ease the model's learning procedure.

All text in the dataset was cleaned using the following steps sequentially:

- Replace all URLs with a {{URL}} token.
- Replace all @mentions with a {{MENTION}} token.
- Replace all email addresses with a {{EMAIL}} token.
- Removing all tweets that were not in English.
- Removing all tweets that have duplicate tweet IDs.
- Remove all & characters using "beautifulsoup"[7]. These were fragments of HTML that were not removed.
- Remove all Tweets with duplicate text. Some duplicate tweets were not removed before labeling because they had different tweet IDs, which we were checking for.
- Replaced all tweet ids with fake ones so that the labelers would have a more challenging time finding the original tweet.
    - We created a mapping between the original and fake tweet ids.

Labeling / Tweet annotation: The final size of the dataset contains 492 tweets awaiting further careful labeling procedures. With the filtered 492 tweets, we are ready for further data labeling with human annotators.

We had three volunteers to label our dataset. Each annotator is provided with three files:
1) a file with our definition and criteria for IIT-related tweet, together with some typical examples for better illustrating our criteria[8].

2) a CSV file with the processed dataset, containing text content of the tweet and the user description of the tweet. The CSV contains in the following entries:

- Fake Tweet ID: The fake tweet id was created to make it harder for the labelers to find the original tweet.
- Tweet Text: The text of the tweet.
- User description: The description of the user from their Twitter profile.

[7] Beautiful Soup: We called him Tortoise because he taught us.
[8] For example tweets see appendix part A.

3) A zipped folder contains all images for the tweets, mapping each tweet with a consistent Fake Tweet ID. Note that each tweet may be associated with no image, one image, or even more than one image.

We asked each labeler to provide labels for each and every tweet, based on the given information. The labels are supposed to be binary: 1 for tweet being IIT-related, and 0 for tweet being not IIT-related.

The labelers agreed on 99.6% of the tweets. There were only two tweets that the labelers disagreed on. The final labels for these tweets were decided by majority voting. The final dataset had 315 tweets without IIT and 177 tweets with IIT, totaling 492 tweets. All label information is shared in our github repo.

Deep Learning Model

We decided to embrace BERT [10] as a state-of-the-art text encoder model for classifying IIT-related tweets, where it consumes text as input and outputs a prediction probability for classification tasks. We wanted to try training the model on three variations, each leveraging different levels of dataset information.

1. **The first variation** only leveraged the text of the tweets. The token length was set to 95 since that was the max token length in the dataset.
2. **The second variation** leveraged both the text of the tweets and the user description. In between the text and user description, the token [sep] was added. If there was no user description (not all Twitter users have descriptions), we put [nodes], which we added as a custom token to the model. The token length was set to 137 since that was the max token length in the dataset.
3. **The third variation** leveraged the text of the tweets, the user description, and optical character recognition (OCR) of the images. In between the text and user description, we added the token [sep]. If there was no user description (not all Twitter users have descriptions), we put [nodes], which we added as a custom token to the model. In between the user description and the OCR, we added the token [sep]. If there was no OCR, we put [noocr], which we added as a custom token to the model.

   To obtain the OCR of the images, we used the following steps:
   - Run "Tesseract"[9] on each image.

[9] Tesseract Open Source OCR Engine (main repository)

- If the tweet has more than one image, combine the text from all the images using spaces.
- Replace all URLs with a {{URL}} token.
- Replace all @mentions with a {{MENTION}} token.
- Replace all email addresses with a {{EMAIL}} token.
- Clean using beautifulsoup.

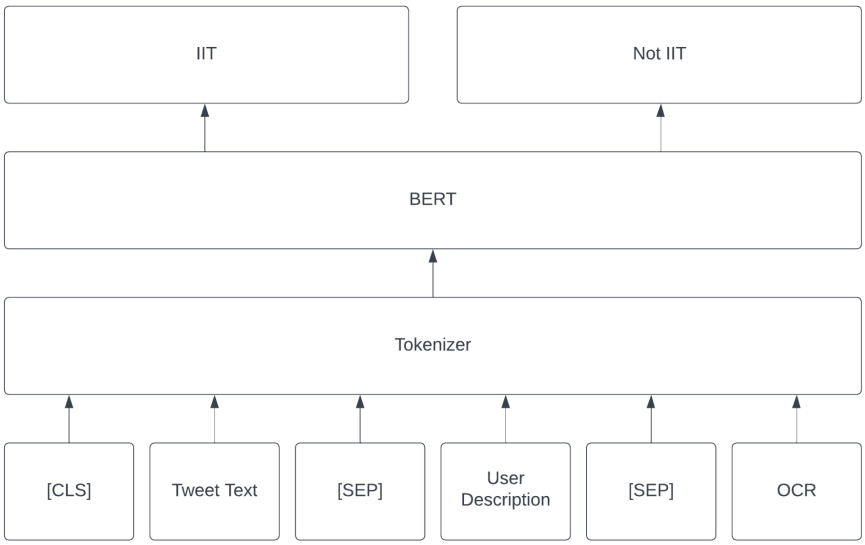

Figure 4. Model Architecture

In Figure 4 the overall architecture of the model can be seen. The input starts with [CLS] to tell the model to do classification. The tweet text is then added. If the model is doing the second variation, the [SEP] and user description is added. If the third variation is selected, another [SEP] is added along with the OCR. Next, the input is tokenized for the model. The tokenized input is then given to BERT. BERT predicts the class that the tweet should be part of.

Experiments

Experiment setting

Train-Test-Split: We experimented with the three variations under a 10-fold cross-validation scenario. For each fold, the data was split using a stratified split with respect to the labels.

BERT Model: The BERT model that we used was the pretrained BERT-uncased from Hugging Face. We fine-tuned the model using the Hugging Face[10] glue_run example code. We also added the aforementioned custom tokens to the model: [nodes], [noocr], {{URL}}, {{MENTION}}, and {{EMAIL}}.

Hyperparameters: We used the following hyperparameters:

	Data	Max Sequence Length	Batch Size	Train Epochs
Model 1	Text	95	16	5
Model 2	Text + User Description	137	16	5
Model 3	Text + User Description + OCR 313 Model	313	16	5
Model 4	Text + User Description + OCR 476 Model	476	16	5

Table 1. Model Hyperparameters

Table 1 contains all the hyperparameters used for each of the four models. For data sources Model 1 uses the tweet text, Model 2 uses the tweet text and user description, Model 3 uses the tweet text, user description and OCR, and Model 4 also uses the tweet text, user description and OCR. The difference between model 3 and 4 is the max sequence length. Model 3 uses 476 tokens and model 4 uses 313 tokens.

[10] transformers/run_glue.py at main · huggingface/transformers · GitHub

Experiment Results

In this section, we will show the results of all four models. Firstly, we will show model variations' performance. Secondly, we will discuss why we believe the models performed with the given accuracy.

The average accuracy and the standard deviation are shown in Table 1[11]. The average results for the four models can be seen in Figure 5 and Table 2. Each model's classification report for all 10 folds can be seen in the appendix.

For OCR, we used a token length of 476 since that was the max token length in the dataset. We also tried a token length of 313. The reason for this is that there was only one tweet with a token length of 476. All other tweets were at or below 313. 476 is 163 or 34% larger than the next longest token length. We thought this was an outlier, so we decided to use a token length of 313.

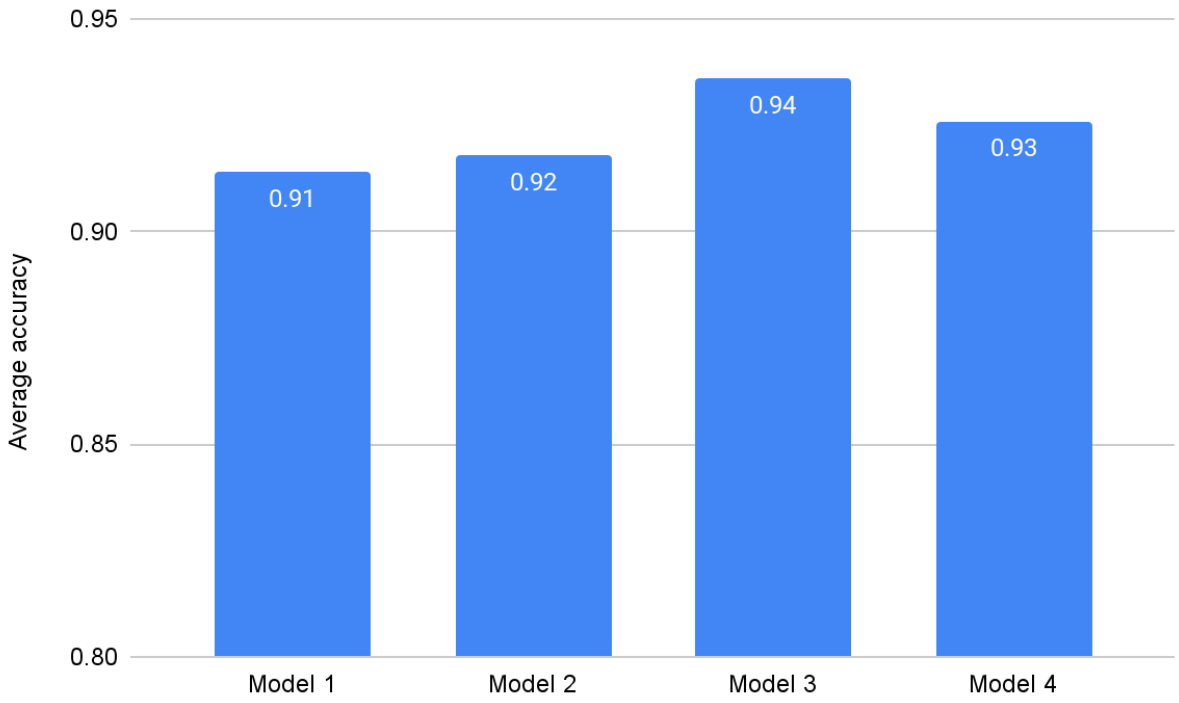

Figure 5. Model Average Accuracies

In Figure 5 it can be seen that model 3 performs the best overall compared to the other three models with an average accuracy of 0.94. Model 4 came in second with an average accuracy of 0.93. Model 2 performs the next best with an average accuracy of 0.92. Model 1 performs the worst with an average accuracy of 0.91.

[11] For all runs see appendix part B

	Model 1	Model 2	Model 3	Model 4
Average	0.91	0.92	0.94	0.93
Standard Deviation	0.04	0.03	0.03	0.03

Table 2. Average Accuracy for Each Dataset

In Table 1 it can be seen that model 3 performs the best. Model 4 is second best. Model 2 is third best. Model 1 is the worst. Standard deviations are shown for each model. Every standard deviation is 0.03 except for model 1 which is 0.03.

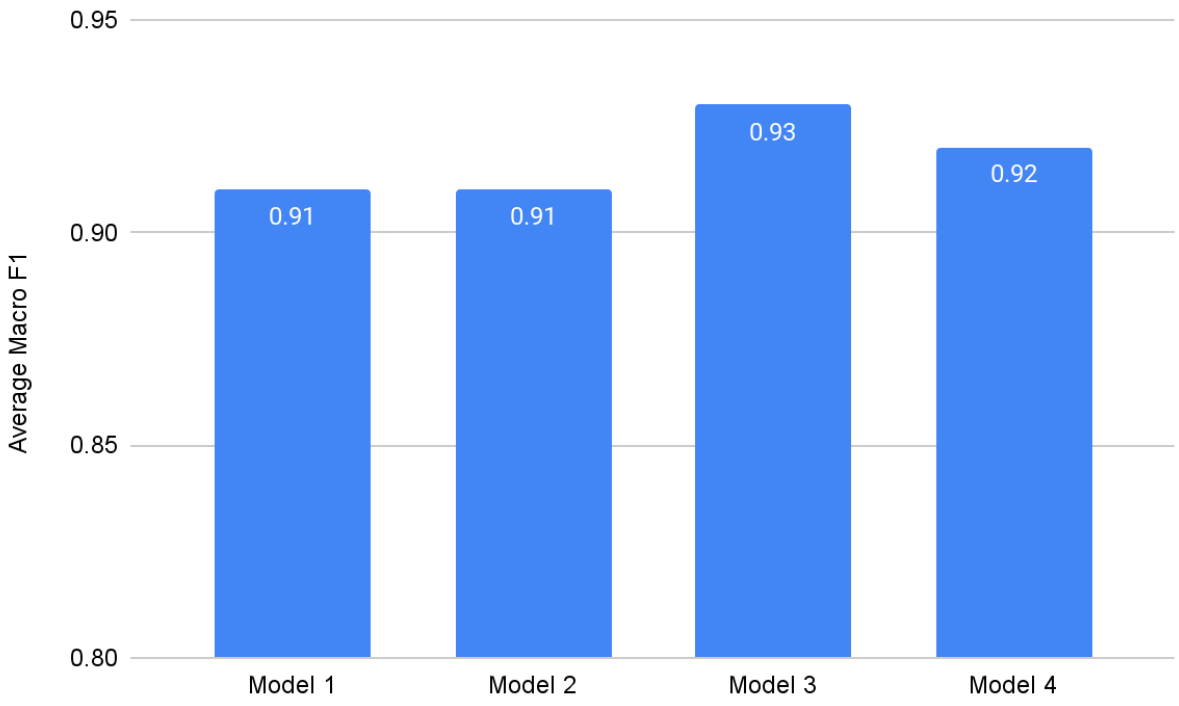

Figure 6. Model Macro F1

In Figure 6 it can be seen that model 3 performs the best compared to the other three models with a macro f1 of 0.93. Model 4 performs the second best with a macro f1 of 0.92. Model 1 and 2 perform the same with a macro f1 of 0.91.

	Model 1	Model 2	Model 3	Model 4
Macro F1	0.91	0.91	0.93	0.92
Standard Deviation	0.05	0.03	0.03	0.03

Table 3. Average Macro F1 for Each Dataset

In Table 3 it can be seen that model 3 performs the best. Model 4 performs the second best. Models 1 and 2 perform the worst with the same macro f1. The standard deviation is the same for all models with a value of 0.03 except for model 1 with a standard deviation of 0.05.

Discussion of the results

Model 1:

This model performing the worst is most likely because it has the least amount of data. It only had the text of the tweets. Even being a human looking at just the text without the user description, it can be challenging to make a prediction. This is from our criteria for labeling the data. Our criteria are on buying or selling IIT. We classify tweets that are talking about an ivory item and are a user that is an auctioneer or antique seller as IIT. We assume that they are attempting to sell or buy the item. Often the only way to tell that they are an auctioneer or antique seller is to look at their user description. Model 1 is at a disadvantage because it only has the text of the tweets.

Model 2:

It makes sense that this model performs better than model 1. The user description is important because, even as a human, it is often necessary to look at the user description to make a prediction. This is for the same reason as described in model 1. It is interesting that it only performs 1% better than model 1. We initially assumed it would perform better than model 1 by a more significant margin. We think that the model might be smart enough to tell whether the user is an auctioneer or an antique seller based on the tweet text.

Model 3:

This model performs the best. It has the most amount of data to work with. The model has the tweets text, user description, and OCR. When looking at the images in the tweets, the user is often advertising IIT in the text in the image. Often there is even a URL linking to the website to buy the IIT item. Being a human, it is often hard to parse the output from the OCR. The text is generally half broken with a bunch of random characters. The model must be able to parse the OCR and use it to help make a prediction. This model also does not have a large amount of padding for most of the tweets. This probably helps the model not get confused about the end of the tweet.

Model 4:

It makes sense that this model performs the second best. The large amount of padding at the end of most tweets confuses the model instead of helping it.

Case Study

In this section we will give an example where two of the models disagree. We will then give analysis as to why we think they disagree.

Tweet text	User description
magnifying glass ivory 10 cm c.1890 {{url}} via {{mention}}	… london silver company … we are a specialist in magnifying glasses …

Table 3. Case IIT Tweet from Dataset

Model 1 (text only) classified the tweet in Table 3 as not IIT, while model 2 (text and user description) classified it as IIT. We believe that the tweet is IIT because it is made of ivory and trying to be sold by a seller. This means that model 2 is correct while model 1 is incorrect. This is most likely because model 2 has more data to work with, which we think is critical to properly classifying the tweet. Based on our criteria, just looking at the tweet's text, we would classify the tweet as not IIT. It is made out of ivory, but there is no indication that it is being bought or sold. Once we look at the user description, we see that the user is a seller. They are a company that sells magnifying glasses. Knowing that the user is a seller, we would classify the tweet as IIT. Since model 1 does not have the user description, it is correct to classify the tweet based on its knowledge. However, we know that it is incorrectly classifying the tweet.

Future Work

Four main future works would be worth looking into: Creating a larger dataset, having input from wildlife experts, incorporating images into the prediction, and unskewing the data towards the keyword "ivory".

Even though the dataset is the biggest dataset we know of, it is still not a very large dataset. More tweets need to be collected and labeled to make the dataset larger. This should make the model more accurate.

It would be good to get input from wildlife experts. There may be tweets that are mislabeled. It might also be possible that there are some obvious ways to improve the dataset, but since we are not wildlife experts, we do not see it.

Incorporating images into the prediction will most likely significantly improve the model's accuracy. The model now is not looking at the image, putting it at a considerable disadvantage. Even being a human, it is sometimes hard to tell, if not impossible, without looking at the image.

Right now, the dataset is greatly skewed towards the keyword ivory. This is because we were filtering for the keyword ivory when creating the dataset. We did the filtering to reduce the number of tweets that we manually needed to label. This does cause the model to be biased. If an IIT tweet does not contain "ivory," then there is a good chance that the model will not be able to label it as an IIT tweet.

It would be good to run the same experiment on another platform that is not Twitter. Twitter might not be where most IITs are happening on the internet. We could be missing many trades by limiting ourselves to twitter.

Conclusions

IIT is a problem that is currently being faced by many countries. The goal of this project was to help fight against IIT. To do so, we created the largest dataset of IIT, which we are aware of. Hopefully, it is large enough to be used by researchers in the future. We also wanted to demonstrate that it would be possible to detect IIT-related posts automatically. We believe that a similar model to the one we created could be applied to any dynamic platform and high-speed social media platform. We hope that this paper will be a good starting point for any platform implementing such an algorithm. We believe that a similar model could potentially be applied to any dynamic and high-speed platform. This paper focuses on Twitter, but the same principles should also apply to other platforms. We hope that it will demonstrate to media platforms that it is possible and practical to implement an algorithm that automatically removes posts promoting the sale of IIT. We also hope this paper would be a good starting point for any platform implementing such an algorithm.

Bibliography

1. US Fish and Wildlife Service. "Endangered and Threatened Wildlife and Plants; Revision of the Section 4 (d) Rule for the African Elephant (Loxodonta africana)." *Federal Register. https://www.federalregister.gov/documents/2016/06/06/2016-13173/endangered-and-threatened-wildlife-and-plants-revision-of-the-section-4d-rule-for-the-african* (2015).
2. Bennett, Elizabeth L. "Legal ivory trade in a corrupt world and its impact on African elephant populations." *Conservation Biology* 29.1 (2015): 54-60.
3. Eikelboom, Jasper AJ, et al. "Will legal international rhino horn trade save wild rhino populations?." *Global ecology and conservation* 23 (2020): e01145.
4. Xu, Qing, et al. "Use of machine learning to detect wildlife product promotion and sales on Twitter." *Frontiers in big Data* (2019): 28.
5. Martin, Esmond, and Daniel Stiles. *The ivory markets of Africa*. Nairobi: Save the Elephants, 2000.
6. Walker, John Frederick. "Rethinking ivory: Why trade in tusks won't go away." *World Policy Journal* 30.2 (2013): 91-100.
7. Gao, Yufang, and Susan G. Clark. "Elephant ivory trade in China: Trends and drivers." *Biological conservation* 180 (2014): 23-30.
8. Matsumoto, Tomomi. "A review of online ivory trade in Japan (Briefing Paper: PDF, 1.4 MB." (2015).
9. Alfino, Sara, and David L. Roberts. "Code word usage in the online ivory trade across four European Union member states." *Oryx* 54.4 (2020): 494-498.
10. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
11. Q. Xu, M. Cai, and T. K. Mackey, “The illegal wildlife digital market: an analysis of chinese wildlife marketing and sale on facebook,” Environmental Conservation, vol. 47, no. 3, pp. 206–212, 2020.

Appendix

A: Labeling Criteria

Negative examples:

- Ivory tweet, but nothing being sold:
    - Tweet Text: Mammoth Ivory Carved
    - User Description: El Cid Gallery is an art gallery … antiques and modern design
- Talking about Ivory, but no items being sold:
    - Tweet Text: … sign this petition … Ivory laws, … devastating … laws go through.
    - User Description: … Auction House …

Positive examples:

- Is ivory and is Auctioneer:
    - Tweet Text: A beautiful Hispano-Philippine carved ivory head … . The sale of items … history … #ivory
    - User Description: I love to ride … Auctioneer & Valuer
- Item is ivory and on sale:
    - Tweet Text: asian art … ivory shibayama fruit … sale
    - User Description: No user description

B: 10-Fold Results

Model 1

	precision	recall	f1-score	support
0	0.96	0.81	0.88	32
1	0.74	0.94	0.83	18
accuracy			0.86	50
macro avg	0.85	0.88	0.86	50
weighted avg	0.88	0.86	0.86	50

Table 4. Model 1 Fold 1

	precision	recall	f1-score	support
0	0.88	0.91	0.89	32
1	0.82	0.78	0.80	18
accuracy			0.86	50
macro avg	0.85	0.84	0.85	50
weighted avg	0.86	0.86	0.86	50

Table 5. Model 1 Fold 2

	precision	recall	f1-score	support
0	0.91	0.97	0.94	31
1	0.94	0.83	0.88	18
accuracy			0.92	49
macro avg	0.92	0.90	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 6. Model 1 Fold 3

	precision	recall	f1-score	support
0	1.00	0.94	0.97	31
1	0.90	1.00	0.95	18
accuracy			0.96	49
macro avg	0.95	0.97	0.96	49
weighted avg	0.96	0.96	0.96	49

Table 7. Model 1 Fold 4

	precision	recall	f1-score	support
0	0.91	0.94	0.92	31
1	0.88	0.83	0.86	18
accuracy			0.90	49
macro avg	0.89	0.88	0.89	49
weighted avg	0.90	0.90	0.90	49

Table 8. Model 1 Fold 5

	precision	recall	f1-score	support
0	0.96	0.84	0.90	32
1	0.76	0.94	0.84	17
accuracy			0.88	49
macro avg	0.86	0.89	0.87	49
weighted avg	0.89	0.88	0.88	49

Table 9. Model 1 Fold 6

	precision	recall	f1-score	support
0	1.00	0.97	0.98	32
1	0.94	1.00	0.97	17
accuracy			0.98	49
macro avg	0.97	0.98	0.98	49
weighted avg	0.98	0.98	0.98	49

Table 10. Model 1 Fold 7

	precision	recall	f1-score	support
0	1.00	0.94	0.97	32
1	0.89	1.00	0.94	17
accuracy			0.96	49
macro avg	0.95	0.97	0.96	49
weighted avg	0.96	0.96	0.96	49

Table 11. Model 1 Fold 8

	precision	recall	f1-score	support
0	0.93	0.88	0.90	32
1	0.79	0.88	0.83	17
accuracy			0.88	49
macro avg	0.86	0.88	0.87	49
weighted avg	0.88	0.88	0.88	49

Table 12. Model 1 Fold 9

	precision	recall	f1-score	support
0	0.97	0.94	0.95	32
1	0.89	0.94	0.91	17
accuracy			0.94	49
macro avg	0.93	0.94	0.93	49
weighted avg	0.94	0.94	0.94	49

Table 13. Model 1 Fold 10

Model 2

	precision	recall	f1-score	support
0	0.94	0.91	0.92	32
1	0.84	0.89	0.86	18
accuracy			0.90	50
macro avg	0.89	0.90	0.89	50
weighted avg	0.90	0.90	0.90	50

Table 14. Model 2 Fold 1

	precision	recall	f1-score	support
0	0.91	0.94	0.92	32
1	0.88	0.83	0.86	18
accuracy			0.90	50
macro avg	0.90	0.89	0.89	50
weighted avg	0.90	0.90	0.90	50

Table 15. Model 2 Fold 2

	precision	recall	f1-score	support
0	0.97	0.90	0.93	31
1	0.85	0.94	0.89	18
accuracy			0.92	49
macro avg	0.91	0.92	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 16. Model 2 Fold 3

	precision	recall	f1-score	support
0	0.89	1.00	0.94	31
1	1.00	0.78	0.88	18

accuracy			0.92	49
macro avg	0.94	0.89	0.91	49
weighted avg	0.93	0.92	0.92	49

Table 17. Model 2 Fold 4

	precision	recall	f1-score	support
0	1.00	0.94	0.97	31
1	0.90	1.00	0.95	18
accuracy			0.96	49
macro avg	0.95	0.97	0.96	49
weighted avg	0.96	0.96	0.96	49

Table 18. Model 2 Fold 5

	precision	recall	f1-score	support
0	0.94	0.94	0.94	32
1	0.88	0.88	0.88	17
accuracy			0.92	49
macro avg	0.91	0.91	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 19. Model 2 Fold 6

	precision	recall	f1-score	support
0	0.94	0.91	0.92	32
1	0.83	0.88	0.86	17
accuracy			0.90	49
macro avg	0.88	0.89	0.89	49
weighted avg	0.90	0.90	0.90	49

Table 20. Model 2 Fold 7

	precision	recall	f1-score	support
0	0.97	0.97	0.97	32
1	0.94	0.94	0.94	17
accuracy			0.96	49
macro avg	0.95	0.95	0.95	49
weighted avg	0.96	0.96	0.96	49

Table 21. Model 2 Fold 8

	precision	recall	f1-score	support
0	0.91	0.91	0.91	32
1	0.82	0.82	0.82	17

accuracy			0.88	49
macro avg	0.86	0.86	0.86	49
weighted avg	0.88	0.88	0.88	49

Table 22. Model 2 Fold 9

	precision	recall	f1-score	support
0	0.97	0.91	0.94	32
1	0.84	0.94	0.89	17
accuracy			0.92	49
macro avg	0.90	0.92	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 23. Model 2 Fold 10

Model 3

	precision	recall	f1-score	support
0	0.97	0.97	0.97	32
1	0.94	0.94	0.94	18
accuracy			0.96	50
macro avg	0.96	0.96	0.96	50
weighted avg	0.96	0.96	0.96	50

Table 24. Model 3 Fold 1

	precision	recall	f1-score	support
0	0.97	1.00	0.98	32
1	1.00	0.94	0.97	18
accuracy			0.98	50
macro avg	0.98	0.97	0.98	50
weighted avg	0.98	0.98	0.98	50

Table 25. Model 3 Fold 2

	precision	recall	f1-score	support
0	0.97	0.90	0.93	31
1	0.85	0.94	0.89	18
accuracy			0.92	49
macro avg	0.91	0.92	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 26. Model 3 Fold 3

	precision	recall	f1-score	support
0	1.00	0.87	0.93	31
1	0.82	1.00	0.90	18
accuracy			0.92	49
macro avg	0.91	0.94	0.92	49
weighted avg	0.93	0.92	0.92	49

Table 27. Model 3 Fold 4

	precision	recall	f1-score	support
0	0.94	0.97	0.95	31
1	0.94	0.89	0.91	18
accuracy			0.94	49
macro avg	0.94	0.93	0.93	49
weighted avg	0.94	0.94	0.94	49

Table 28. Model 3 Fold 5

	precision	recall	f1-score	support
0	0.94	1.00	0.97	32
1	1.00	0.88	0.94	17
accuracy			0.96	49
macro avg	0.97	0.94	0.95	49
weighted avg	0.96	0.96	0.96	49

Table 29. Model 3 Fold 6

	precision	recall	f1-score	support
0	0.97	0.94	0.95	32
1	0.89	0.94	0.91	17
accuracy			0.94	49
macro avg	0.93	0.94	0.93	49
weighted avg	0.94	0.94	0.94	49

Table 30. Model 3 Fold 7

	precision	recall	f1-score	support
0	0.96	0.84	0.90	32
1	0.76	0.94	0.84	17
accuracy			0.88	49
macro avg	0.86	0.89	0.87	49
weighted avg	0.89	0.88	0.88	49

Table 31. Model 3 Fold 8

	precision	recall	f1-score	support
0	0.94	0.94	0.94	32
1	0.88	0.88	0.88	17
accuracy			0.92	49
macro avg	0.91	0.91	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 32. Model 3 Fold 9

	precision	recall	f1-score	support
0	0.94	0.97	0.95	32
1	0.94	0.88	0.91	17
accuracy			0.94	49
macro avg	0.94	0.93	0.93	49
weighted avg	0.94	0.94	0.94	49

Table 33. Model 3 Fold 10

Model 4

	precision	recall	f1-score	support
0	0.97	0.97	0.97	32
1	0.94	0.94	0.94	18
accuracy			0.96	50
macro avg	0.96	0.96	0.96	50
weighted avg	0.96	0.96	0.96	50

Table 34. Model 4 Fold 1

	precision	recall	f1-score	support
0	0.94	1.00	0.97	32
1	1.00	0.89	0.94	18`
accuracy			0.96	50
macro avg	0.97	0.94	0.96	50
weighted avg	0.96	0.96	0.96	50

Table 35. Model 4 Fold 2

	precision	recall	f1-score	support

0	0.96	0.87	0.92	31
1	0.81	0.94	0.87	18
accuracy			0.90	49
macro avg	0.89	0.91	0.89	49
weighted avg	0.91	0.90	0.90	49

Table 36. Model 4 Fold 3

	precision	recall	f1-score	support
0	1.00	0.87	0.93	31
1	0.82	1.00	0.90	18
accuracy			0.92	49
macro avg	0.91	0.94	0.92	49
weighted avg	0.93	0.92	0.92	49

Table 37. Model 4 Fold 4

	precision	recall	f1-score	support
0	0.94	0.97	0.95	31
1	0.94	0.89	0.91	18
accuracy			0.94	49
macro avg	0.94	0.93	0.93	49
weighted avg	0.94	0.94	0.94	49

Table 38. Model 4 Fold 5

	precision	recall	f1-score	support
0	0.97	0.97	0.97	32
1	0.94	0.94	0.94	17
accuracy			0.96	49
macro avg	0.95	0.95	0.95	49
weighted avg	0.96	0.96	0.96	49

Table 39. Model 4 Fold 6

	precision	recall	f1-score	support
0	0.94	0.94	0.94	32
1	0.88	0.88	0.88	17
accuracy			0.92	49
macro avg	0.91	0.91	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 40. Model 4 Fold 7

	precision	recall	f1-score	support

0	1.00	0.84	0.92	32
1	0.77	1.00	0.87	17
accuracy			0.90	49
macro avg	0.89	0.92	0.89	49
weighted avg	0.92	0.90	0.90	49

Table 41. Model 4 Fold 8

	precision	recall	f1-score	support
0	0.94	0.94	0.94	32
1	0.88	0.88	0.88	17
accuracy			0.92	49
macro avg	0.91	0.91	0.91	49
weighted avg	0.92	0.92	0.92	49

Table 42. Model 4 Fold 9

	precision	recall	f1-score	support
0	0.88	0.94	0.91	32
1	0.87	0.76	0.81	17
accuracy			0.88	49
macro avg	0.87	0.85	0.86	49
weighted avg	0.88	0.88	0.88	49

Table 43. Model 4 Fold 10

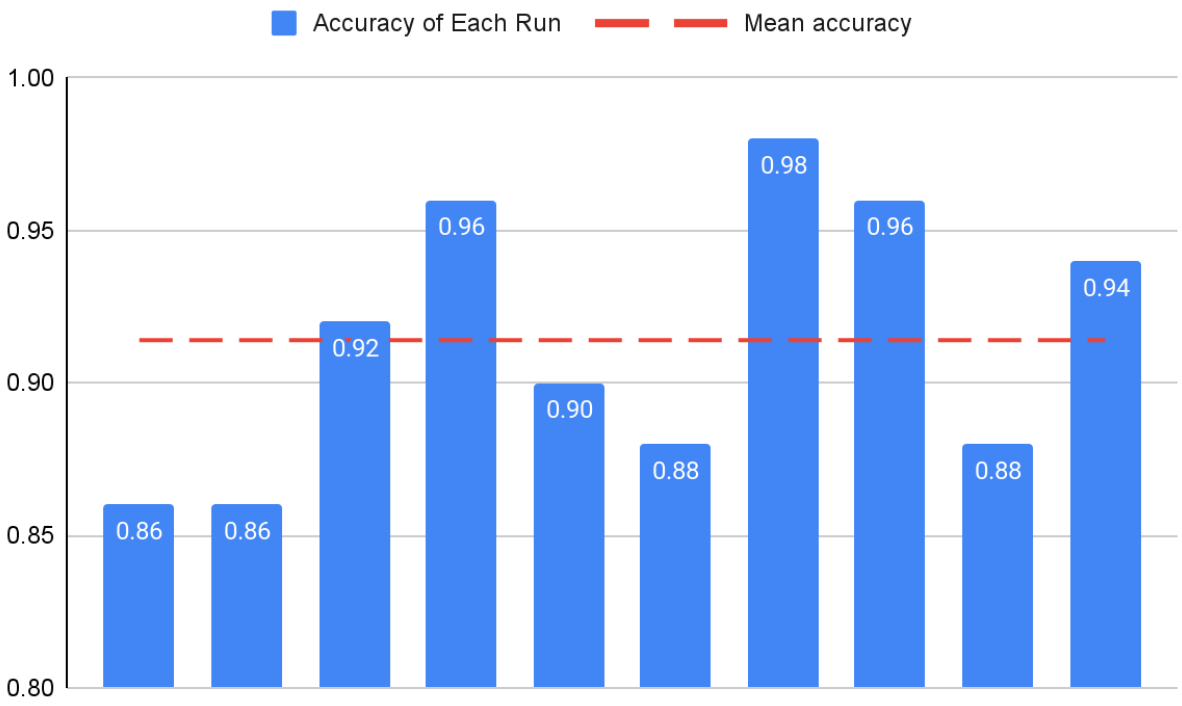

Figure 7. Model 1 10-Fold Accuracy

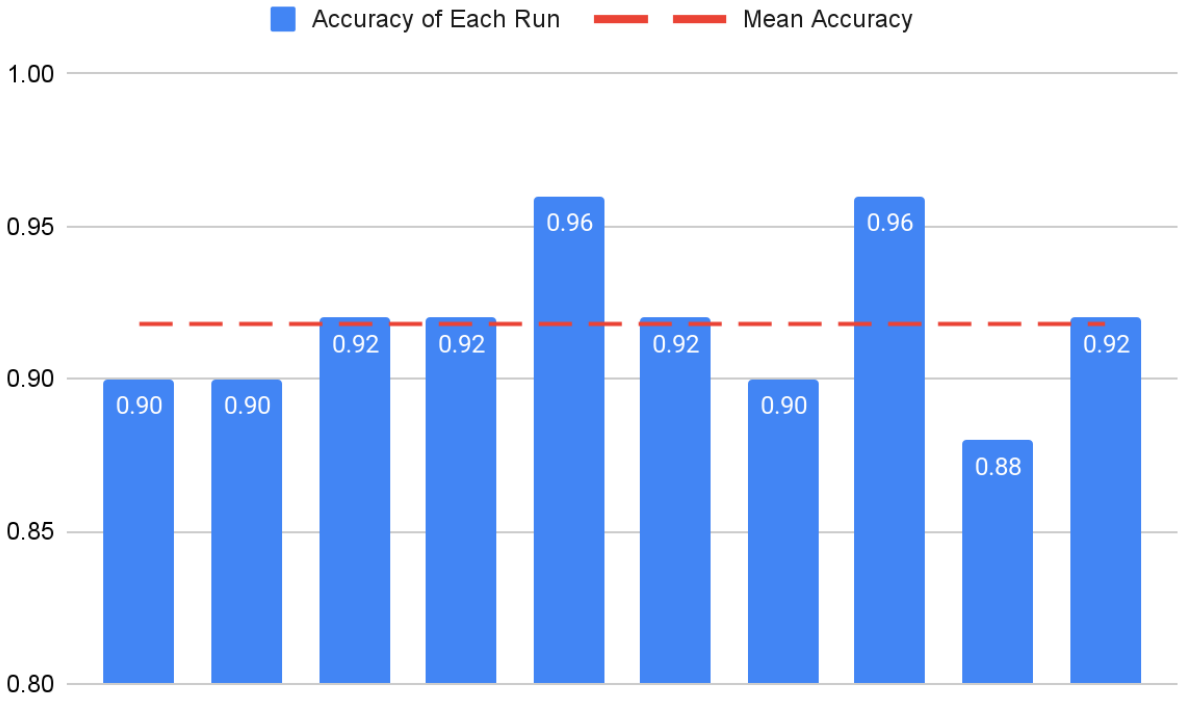

Figure 8. Model 2 10-Fold Accuracy

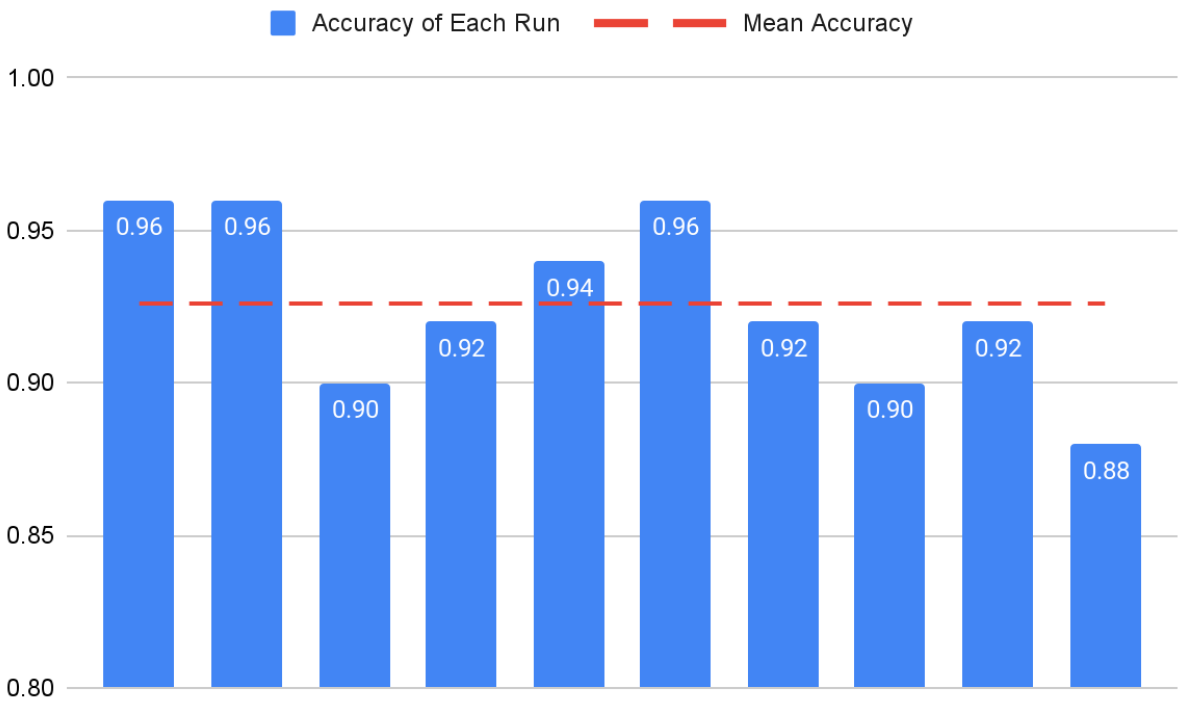

Figure 9. Model 3 10-Fold Accuracy

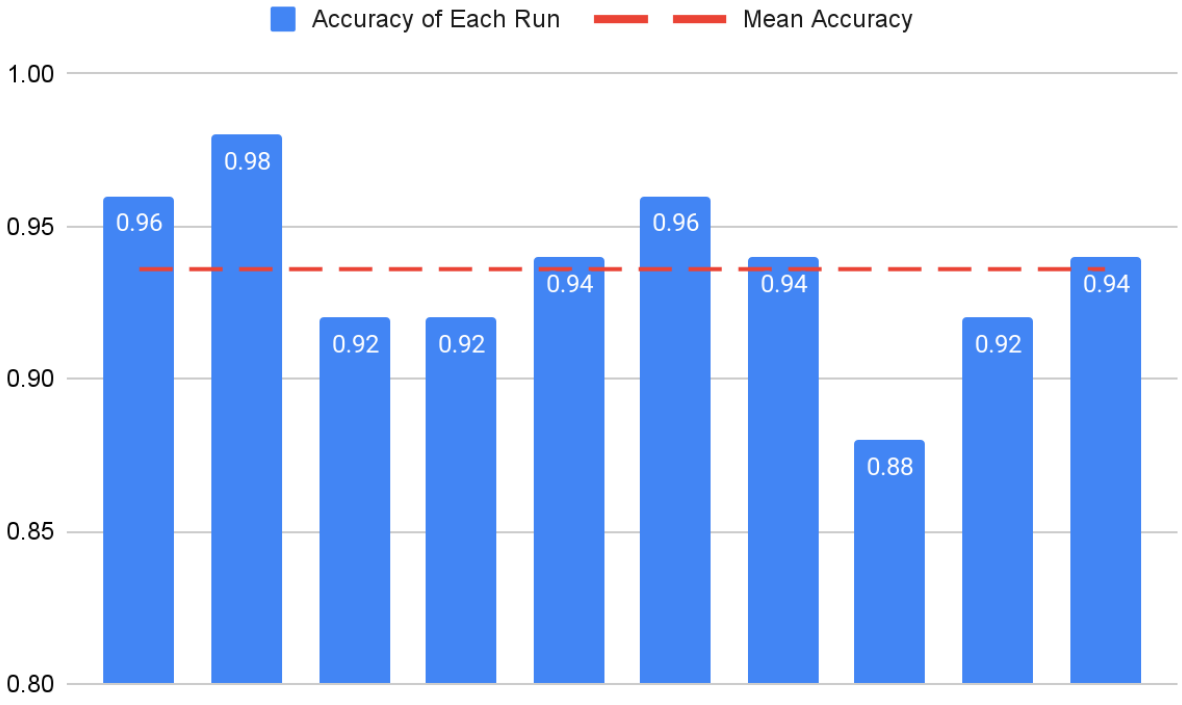

Figure 10. Model 4 10-Fold Accuracy