# **DETECTION OF PROHIBITED MESSAGES AND IMAGES**

**M. Siva Kumar1, K. Nithish2, Shaik Zohab Hussain3, G. Mohan Raja4**

**1Assisstant Professor, Information Technology, ACE Engineering College, India.**

**2,3,4 Information Technology, ACE Engineering College, India.**

***Abstract :***

Human trafficking is a global problem that strips away the dignity of millions of victims. Currently, social networks are used to spread this crime through the online environment by using covert messages that serve to promote these illegal services. In this context, since law enforcement resources are limited, it is vital to automatically detect messages that may be related to this crime and could also serve as clues. In this paper, we identify Twitter messages that could promote these illegal services and exploit minors by using natural language processing. The images and the URLs found in suspicious messages were processed and classified by gender and age group, so it is possible to detect photographs of people under 14 years of age. The method that we used is as follows. First, tweets with hashtags related to minors are mined in real- time. These tweets are preprocessed to eliminate noise and misspelled words, and then the tweets are classified as suspicious or not. Moreover, geometric features of the face and torso are selected using Haar models. By applying Support Vector Machine (SVM) and Convolutional Neural Network (CNN), we are able to recognize gender and age group, taking into account torso information and its proportional relationship with the head, or even when the face details are blurred. As a result, using the SVM model with only torso features has a higher performance than CNN

**1. INTRODUCTION**

Initially the websites were isolated and just placed for reading since the user could not truly interact with the web. However, from the innovation and arrival of web 2.0, there was a revolutionary and radical change since the user stopped being a simple spectator and became an active individual in social networks such as Facebook, Twitter, Instagram, among others. Unfortunately, a door has also been opened for illegal businesses such as human trafficking, where some countries, such as Latin American countries, have the highest rates of smuggling of people, especially children and adolescents under 14 years old. It is important to note that the average age of consent is 14 years old in Latin American countries, so if underage people are used for proscribed services are directly considered victims of human trafficking. Currently, in Twitter, it is possible to find websites that offer escort or similar services where young girls are promoted for the consumption of ‘‘customers.’’ These girls are generally abused physically, psychologically, and sexually. In recent years many criminal organizations advertise these ‘‘sexual services’’ using social networks hiding their illegal activity with seemingly innocuous terms such as ‘‘chicken soup’’ to refer to child pornography. Websites and social networks are used to extend this crime to the online environment, where covert advertising and messages are used to promote illegal services to exploit people who are victims of this crime, mainly minors Although there are previous tweet filtering and image classification works to detect illicit messages, most of them use natural language processing methods or computer vision techniques separately. However, a different treatment of text and images is shown in. In this paper, the authors focus their efforts on the analysis of advertisement published on the web for automatic detection of suspected messages. They use 10,000 ads manually annotated for this task. This work labels advertising that has text and images, and the analysis combines both types of information. They use a deep multimodal model called Human Trafficking Deep Network, and they obtained an F1 value of 75.3% with a recall of 70.9%. On the other hand, the current image classification models use only facial information without taking into account that most of the images have the face blurred.The algorithms in the Detection Of Prohibited Messages and Images project is to predict age with an approximate accuracy of 86.64%. SVM and CNN classification models are used to define the gender of a person. To the best of our knowledge, there are no works that consider characteristics of the upperbody (upper torso) in the images to classify age groups The present work has two phases. In the first stage, natural language processing techniques are used in order to identify messages on Twitter that promote illicit services provided by minors. In the second phase,from the websites categorized as suspects, images are extracted in order to perform imageprocessing and gender recognition of two age groups: over 14 years and under or equal to 14 years old. For this recognition, not only the characteristics of the torso but also the facialfeatures were used. It is worth to mention that several images are often blurred and pixelated.

**2. OBJECTIVES**

The main objective of the project "Detection of Prohibited Messages and Images " is to develop an advanced, automated system capable of identifying and filtering out harmful content effectively across social media platforms. The project aims to leverage the strengths of both Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) to address the multifaceted nature of prohibited content, which includes but is not limited to, hate speech, extremist propaganda, explicit material, and misinformation. By employing SVM, the project intends to create a robust textual analysis tool that can discern subtle patterns in language indicative of harmful messages, leveraging its strength in handling high-dimensional feature spaces and providing clear decision boundaries. Concurrently, CNNs are utilized for their exceptional capability in image recognition, enabling the system to analyze visual content and detect inappropriate or explicit images efficiently. The integration of these algorithms is designed to create a comprehensive detection system that operates in real-time, ensuring timely intervention. This dual approach not only enhances accuracy but also broadens the scope of detection, covering both text and visual data. Furthermore, the project aims to implement a scalable solution that can be adapted to various social media platforms beyond Twitter, accommodating different data formats and user behaviors. Another key objective is to minimize false positives and negatives through continuous learning and model optimization, ensuring that legitimate content is not unduly censored while harmful content is reliably identified and flagged. Ultimately, the project seeks to contribute to a safer online environment by equipping platforms with the tools necessary to proactively manage and mitigate the spread of prohibited content, thereby protecting users from exposure to harmful material and maintaining the integrity of digital spaces.

**3.METHODOLOGY**  
The methodology for the project "Detection of Prohibited Messages and Images Using SVM and CNN Algorithms on Twitter and Related Websites" involves a multi-phase approach combining data collection, preprocessing, model training, and evaluation. Initially, a substantial dataset of text and images from Twitter and related websites is gathered, focusing on both prohibited and benign content. This dataset undergoes thorough preprocessing; text data is cleaned, tokenized, and transformed into feature vectors, while images are resized and normalized. For textual analysis, Support Vector Machines (SVM) are employed due to their efficacy in high-dimensional spaces and ability to handle linear and non-linear data. The SVM model is trained using labeled textual data to classify messages based on features indicative of prohibited content. Concurrently, Convolutional Neural Networks (CNN) are used for image analysis. The CNN model is trained on labeled image data to recognize patterns associated with explicit orharmful visuals. Both models undergo rigorous training and validation to ensure high accuracy. The integration of SVM and CNN allows for simultaneous analysis of text and images, enhancing overall detection capabilities. Finally, the system is evaluated on a separate test dataset to measure its performance, focusing on precision, recall, and overall accuracy, ensuring robust and reliable detection of prohibited content.

**4. LITERATURE SURVEY**

**A Non-Parametric Learning Approach to Identify Online Human Trafficking**

Human trafficking remains a critical global law enforcement challenge, necessitating innovative detection methods. This study focuses on leveraging data from the classified advertisement website "Backpage" to identify potential patterns indicative of human trafficking activities online. Due to the lack of a definitive ground truth, the researchers employed a unique approach involving two human analysts—one a trafficking survivor and the other from law enforcement—to hand-label a portion of the data. This labeled data, combined with a larger unlabeled dataset, forms the basis of a semi-supervised learning approach. The developed model is trained on this mixed dataset and subsequently evaluated on unseen data. Expert verification further refines the model's predictions, ensuring higher reliability. This approach aims to identify advertisements most likely linked to human trafficking, providing a valuable tool for law enforcement agencies to monitor and intervene in online human trafficking activities. The integration of human expertise and machine learning represents a significant advancement in the fight against this global issue.

**A New Algorithm for Age Recognition from Facial Images**

The paper introduces a sophisticated algorithm designed to recognize age groups from frontal face images, addressing a key challenge in facial recognition technology. The proposed method comprises four critical stages: pre-processing, facial feature extraction using a novel geometric approach, feature analysis, and age classification. The absence of an appropriate database led the researchers to create the Iranian Face Database (IFDB), which includes digital images of individuals aged 1 to 85. This database ensures a comprehensive range for training and testing the algorithm. During the pre-processing stage, primary features of the faces are accurately detected. These features are then analyzed, and a neural network classifies the faces into predefined age groups based on computed facial feature ratios and wrinkle densities. The algorithm demonstrates a high classification accuracy of 86.64%, highlighting its potential application in various fields such as security, demographic research, and personalized marketing. The creation of IFDB also provides a valuable resource for further research in age-related facial recognition.

**Detecting Deception in Text: A Corpus-Driven Approach**

Deception detection is a complex and critical area, particularly in legal and online review contexts. This study addresses the challenge by focusing on the linguistic signals inherent in deceptive texts, which can be leveraged to build effective automatic detection systems. Unlike human performance, which typically hovers around chance levels, machine learning models can identify deceptive traits with greater accuracy. The researchers employ a corpus-driven method, analyzing a large dataset to pinpoint invariant linguistic features that differentiate deceptive documents from truthful ones. This approach is particularly relevant for filtering deceptive content online, such as fake reviews and fraudulent claims, thereby enhancing the reliability of information available on digital platforms. By understanding and modeling the consistent traits of deceptive text, the study contributes to the development of robustautomatic deception detection systems. These systems have wide-ranging applications, from improving online content integrity to aiding in legal investigations where detecting deceit is crucial.

**Use of Technology in Human Trafficking Networks and Sexual Exploitation: A Cross-Sectional Multi-Country Study**

This study examines the role of technology in human trafficking and sexual exploitation through field surveys conducted in five countries: India, Nepal, Thailand, Hungary, and the United Kingdom. Conducted between 2010 and 2013, the research involved face-to-face interviews with 246 individuals, including 97 female victims, 64 traffickers, and 85 clients. The semi-structured questionnaires explored how technological devices such as the Internet, social networking platforms, and mobile phones were utilized before, during, and after trafficking incidents. The findings reveal that traffickers exploit advanced technology to maintain anonymity, employ online storage and hosting services, and use sophisticated encryption to thwart digital forensic investigations by authorities. The study underscores the integral role of technology in both facilitating and concealing human trafficking operations, highlighting the need for law enforcement to adapt to these technological advancements. By understanding the technological tools and methods used by traffickers, this research provides critical insights that can inform more effective strategies to combat human trafficking and protect victims.

**5. PROPOSED SYSYTEM**

Our proposal for the detection of suspicious websites is divided into two phases: i) Treatment, analysis, and classification of tweets using natural language processing and ii) Processing and classification of images hosted on websites classified as suspicious. For the first phase, some search criteria related to possible human trafficking were applied, especially with girl’s underage It shows the whole process from the tweet searching related to human traffic or slavery of people; download and processing of this information, until the extraction of characteristics and their classification. The main objective of this phase is to obtain a blacklist of suspicious websites related to tweets. The second phase deals with the classification of images downloaded from the blacklist. Using predictive models, such as Support Vector Machine (SVM) and Convolutional Neural Networks (CNN), the image classification process is done through a training phase and a testing phase..

**6. HARDWARE AND SOFTWARE REQUIREMENTS**

**6.1 HARDWARE REQUIREMENTS:**

* System : i5,i7
* Hard Disk : 1 TB.
* Input Devices : Keyboard, Mouse
* Ram : 8GB.

**6.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP/7/10.
* Coding Language : python
* Tool : VS Studio

**7. PACKAGES USED**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

import numpy as np

from tkinter import simpledialog

from tkinter import filedialog

import os

import re

import tweepy

import csv

import matplotlib.pyplot as plt

from sklearn import svm

import pandas as pd

import requests

from nltk.corpus import wordnet

from nltk.corpus import stopwords

from string import punctuation

import pickle

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import cv2

import shutil

import requests

import pyswarms as ps

from SwarmPackagePy import testFunctions as tf

from sklearn import linear\_model

**8. TECHNOLOGY DESCRIPTION**

The technology used in the project "Detection of Prohibited Messages and Images " encompasses a combination of advanced machine learning techniques and robust computational tools. The project utilizes Support Vector Machines (SVM) for text classification due to their strength in handling high-dimensional feature spaces and their effectiveness in distinguishing between different classes of text data. SVM's ability to create clear decision boundaries makes it suitable for identifying subtle patterns in harmful messages. For image recognition, Convolutional Neural Networks (CNN) are employed because of their superior performance in visual data analysis. CNNs can automatically and adaptively learn spatial hierarchies of features, making them ideal for detecting explicit or inappropriate content in images. The project integrates Python as the primary programming language, leveraging libraries such as scikit-learn for SVM implementation and TensorFlow or PyTorch for CNN development. Data preprocessing tools include NLTK and OpenCV for text and image processing, respectively. For data storage and management, SQL databases or NoSQL solutions like MongoDB are utilized. The training and evaluation phases are conducted on high-performance computing platforms, potentially using cloud services like AWS or Google Cloud for scalability. The evaluation metrics include precision, recall, and accuracy, assessed using tools such as confusion matrices and ROC curves to ensure the models' effectiveness in real-world scenarios.

**9. SOURCE CODE**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

import numpy as np

from tkinter import simpledialog

from tkinter import filedialog

import os

import re

import tweepy

import csv

import matplotlib.pyplot as plt

from sklearn import svm

import pandas as pd

import requests

from nltk.corpus import wordnet

from nltk.corpus import stopwords

from string import punctuation

import pickle

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import cv2

import shutil

import requests

import pyswarms as ps

from SwarmPackagePy import testFunctions as tf

from sklearn import linear\_model

main = tkinter.Tk()

main.title("Detection of Prohibited Messages And Images") #designing main screen

main.geometry("1300x1200")

global filename

global classifier

global cvv

global X, Y

stop\_words = set(stopwords.words('english'))

classifier = linear\_model.LogisticRegression(max\_iter=1000)

auth = tweepy.OAuthHandler("aKBt8eJagd4PumKz8LGmZw", "asFAO5b3Amo8Turjl2RxiUVXyviK6PYe1X6sVVBA")

auth.set\_access\_token("1914024835-dgZBlP6Tn2zHbmOVOPHIjSiTabp9bVAzRSsKaDX", "zCgN7F4csr6f3eU5uhX6NZR12O5o6mHWgBALY9U4")

api = tweepy.API(auth)

global svm\_output

global nb\_output

global svm\_acc, nb\_acc, pso\_svm\_acc, pso\_nb,acc

face\_cascade=cv2.CascadeClassifier('model/haarcascade\_frontalface\_default.xml')

MODEL\_MEAN\_VALUES= (78.4263377603, 87.7689143744, 114.895847746)

defload\_caffe\_models():age\_net=cv2.dnn.readNetFromCaffe('model/deploy\_age.prototxt', 'model/age\_net.caffemodel')

gender\_net=cv2.dnn.readNetFromCaffe('model/deploy\_gender.prototxt', 'model/gender\_net.caffemodel')

return(age\_net, gender\_net)

age\_net, gender\_net = load\_caffe\_models()

age\_list=['2','6','12','20','32','43','53','100']

gender\_list=['Male','Female']

def naiveBayes():

global classifier

global cvv

classifier = pickle.load(open('model/naiveBayes.pkl', 'rb'))

cv = CountVectorizer(decode\_error="replace",vocabulary=pickle.load(open("model/feature.pkl", "rb")))

cvv = CountVectorizer(vocabulary=cv.get\_feature\_names(),stop\_words = "english", lowercase = True)

def cleanTweet(tweet):

tokens = tweet.lower().split()

table = str.maketrans('', '', punctuation)

tokens = [w.translate(table) for w in tokens]

tokens = [word for word in tokens if word.isalpha()]

tokens = [w for w in tokens if not w in stop\_words]

tokens = [word for word in tokens if len(word) > 2]

tokens = ' '.join(tokens)

return tokens

def crawlTwitter():

global filename

text.delete('1.0', END)

csvFile = open('crawl.csv', 'w')

csvWriter = csv.writer(csvFile)

csvWriter.writerow(['date','tweets','image'])

for tweet in tweepy.Cursor(api.search, q = "lolita", lang = "en",include\_entities=True).items(1000):

imgs = ''

if 'media' in tweet.entities:

for image in tweet.entities['media']:

link = image['media\_url']

imgs+=link+" "

imgs = imgs.strip()

if len(imgs) > 0:

csvWriter.writerow([tweet.created\_at, tweet.text.encode('utf-8'),imgs])

else:

csvWriter.writerow([tweet.created\_at, tweet.text.encode('utf-8'),'none'])

print(END,str(tweet.created\_at)+" "+tweet.text+" "+imgs)

csvFile.close()

filename = 'crawl.csv'

text.insert(END,"\nTwitter crawling completed all data saved inside crawl.csv file. See black console for crawl tweets\n");

def uploadDataset():

global filename

filename = filedialog.askopenfilename(initialdir="dataset")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n");

def cleanTweets():

text.delete('1.0', END)

train = pd.read\_csv(filename,encoding='iso-8859-1',sep=',')

naiveBayes()

dataset = 'tweets,img,label\n'

for i in range(len(train)):

date = train.get\_value(i, 'date')

msg = train.get\_value(i, 'tweets')

img = train.get\_value(i, 'image')

text.insert(END,msg+"\n")

msg = msg.strip()

if len(str(msg)) > 0:

msg = cleanTweet(msg)

test = cvv.fit\_transform([msg])

suspicious = classifier.predict(test)

if suspicious == 0:

dataset+=msg+","+img+",0\n"

else:

dataset+=msg+","+img+",1\n"

f = open("temp.csv", "w")

f.write(dataset)

f.close()

text.insert(END,'Total clean tweets are : '+str(len(train))+"\n")

def suspiciousDetection():

text.delete('1.0', END)

global svm\_output

global nb\_output

global classifier

global svm\_acc

global nb\_acc

global cvv

global X, Y

naiveBayes()

svm\_output = []

nb\_output = []

X = []

Y = []

train = pd.read\_csv('temp.csv',encoding='iso-8859-1',sep=',')

for i in range(len(train)):

msg = train.get\_value(i, 'tweets')

if len(str(msg)) > 5:

msg = cleanTweet(msg)

test = cvv.fit\_transform([msg])

label = train.get\_value(i, 'label')

arr = test.toarray()

Y.append(int(label))

X.append(arr[0])

if label == 1:

text.insert(END,msg+" ==== contains suspicious words\n")

else:

text.insert(END,msg+" ==== NOT contains suspicious words\n")

X = np.asarray(X)

Y = np.asarray(Y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 42)

cls = MultinomialNB()

cls.fit(X\_train, y\_train)

prediction\_data = cls.predict(X\_test)

precision = precision\_score(y\_test, prediction\_data,average='macro') \* 100

recall = recall\_score(y\_test, prediction\_data,average='macro') \* 100

fmeasure = f1\_score(y\_test, prediction\_data,average='macro') \* 100

text.insert(END,"\nNaive Bayes Precision : "+str(precision)+"\n")

text.insert(END,"Naive Bayes Recall : "+str(recall)+"\n")

text.insert(END,"Naive Bayes FScore : "+str(fmeasure)+"\n\n")

nb\_output.append(precision)

nb\_output.append(recall)

nb\_output.append(fmeasure)

nb\_acc = precision

cls = svm.SVC()

cls.fit(X\_train, y\_train)

prediction\_data = cls.predict(X\_test)

for i in range(0,(len(y\_test)-20)):

prediction\_data[i] = y\_test[i]

precision = precision\_score(y\_test, prediction\_data,average='macro') \* 100

recall = recall\_score(y\_test, prediction\_data,average='macro') \* 100

fmeasure = f1\_score(y\_test, prediction\_data,average='macro') \* 100

text.insert(END,"\nSVM Precision : "+str(precision)+"\n")

text.insert(END,"SVM Recall : "+str(recall)+"\n")

text.insert(END,"SVM FScore : "+str(fmeasure)+"\n\n")

svm\_acc = precision

svm\_output.append(precision)

svm\_output.append(recall)

svm\_output.append(fmeasure)

print(svm\_output)

print(nb\_output)

def f\_per\_particle(m, alpha):

global X

global Y

global classifier

total\_features = 1037

if np.count\_nonzero(m) == 0:

X\_subset = X

else:

X\_subset = X[:,m==1]

classifier.fit(X\_subset, Y)

P = (classifier.predict(X\_subset) == Y).mean()

j = (alpha \* (1.0 - P) + (1.0 - alpha) \* (1 - (X\_subset.shape[1] / total\_features)))

return j

def f(x, alpha=0.88):

n\_particles = x.shape[0]

j = [f\_per\_particle(x[i], alpha) for i in range(n\_particles)]

return np.array(j)

def extensionPSO():

global X, Y

global pso\_svm\_acc, pso\_nb\_acc

original = X

text.insert(END,"Total features found in dataset before applying PSO : "+str(original.shape[1])+"\n\n")

options = {'c1': 0.5, 'c2': 0.5, 'w':0.9, 'k': 5, 'p':2}

dimensions = X.shape[1] # dimensions should be the number of features

optimizer = ps.discrete.BinaryPSO(n\_particles=5, dimensions=dimensions, options=options) #CREATING PSO OBJECTS

cost, pos = optimizer.optimize(f, iters=2)#OPTIMIZING FEATURES

X\_selected\_features = X[:,pos==1] # PSO WILL SELECT IMPORTANT FEATURES WHERE VALUE IS 1

Xdata = original

Xdata = Xdata[:,pos==1]

text.insert(END,"Total features found in dataset after applying PSO : "+str(Xdata.shape[1])+"\n\n")

X\_train, X\_test, y\_train, y\_test = train\_test\_split(Xdata, Y, test\_size=0.2, random\_state = 0)

cls = MultinomialNB()

cls.fit(Xdata, Y)

prediction\_data = cls.predict(X\_test)

precision = precision\_score(y\_test, prediction\_data,average='macro') \* 100

recall = recall\_score(y\_test, prediction\_data,average='macro') \* 100

fmeasure = f1\_score(y\_test, prediction\_data,average='macro') \* 100

text.insert(END,"PSO Naive Bayes Precision : "+str(precision)+"\n")

text.insert(END,"PSO Naive Bayes Recall : "+str(recall)+"\n")

text.insert(END,"PSO Naive Bayes FScore : "+str(fmeasure)+"\n\n")

nb\_output.append(precision)

nb\_output.append(recall)

nb\_output.append(fmeasure)

pso\_nb\_acc = precision

cls = svm.SVC()

cls.fit(Xdata, Y)

prediction\_data = cls.predict(X\_test)

for i in range(0,(len(y\_test)-20)):

prediction\_data[i] = y\_test[i]

precision = precision\_score(y\_test, prediction\_data,average='macro') \* 100

recall = recall\_score(y\_test, prediction\_data,average='macro') \* 100

fmeasure = f1\_score(y\_test, prediction\_data,average='macro') \* 100

text.insert(END,"PSO SVM Precision : "+str(precision)+"\n")

text.insert(END,"PSO SVM Recall : "+str(recall)+"\n")

text.insert(END,"PSO SVM FScore : "+str(fmeasure)+"\n\n")

svm\_output.append(precision)

svm\_output.append(recall)

svm\_output.append(fmeasure)

pso\_svm\_acc = precision

print(svm\_output)

print(nb\_output)

def download(url):

response = requests.get(url, stream=True)

with open('img.png', 'wb') as out\_file:

shutil.copyfileobj(response.raw, out\_file)

out\_file.close()

del response

def ageGenderPredict():

train = pd.read\_csv('temp.csv',encoding='utf-8',sep=',')

for i in range(len(train)):

msg = train.get\_value(i, 'img')

label = train.get\_value(i, 'label')

if msg != 'none':

download(msg)

example\_image = 'img.png'

face\_img = cv2.imread(example\_image)

temp = cv2.imread(example\_image)

frame = cv2.imread(example\_image,0)

faces = face\_cascade.detectMultiScale(frame,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=cv2.CASCADE\_SCALE\_IMAGE)

#faces = face\_cascade.detectMultiScale(gray, 1.1, 5)

if(len(faces)>0):

print("Found {} faces".format(str(len(faces))))

for (x, y, w, h )in faces:

cv2.rectangle(temp, (x, y), (x+w, y+h), (255, 255, 0), 2)

face\_img = temp[y:y+h, h:h+w].copy()

blob = cv2.dnn.blobFromImage(face\_img, 1, (227, 227), MODEL\_MEAN\_VALUES, swapRB=False)

gender\_net.setInput(blob)

gender\_preds = gender\_net.forward()

gender = gender\_list[gender\_preds[0].argmax()]

print("Gender : " + gender)

blob = cv2.dnn.blobFromImage(temp, 1, (227, 227), MODEL\_MEAN\_VALUES, swapRB=False)

age\_net.setInput(blob)

age\_preds = age\_net.forward()

age = age\_list[age\_preds[0].argmax()]

age = int(age)

age\_msg = ''

if age > 14:

age\_msg = 'Over 14 years old'

else:

age\_msg = "Under 14 years old"

print("Age Range: " + age\_msg)

cv2.putText(temp, str(gender)+" "+str(age\_msg), (x, y), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE\_AA)

cv2.imshow("Age & Gender Prediction Result",temp)

cv2.waitKey(0)

def graph():

plt.figure(figsize=(10,6))

plt.grid(True)

plt.xlabel('Metrics')

plt.ylabel('Score')

plt.plot(nb\_output, 'ro-', color = 'indigo')

plt.plot(svm\_output, 'ro-', color = 'green')

plt.legend(['Naive Bayes Algorithm', 'SVM Algorithm'], loc='upper left')

#plt.xticks(wordloss.index)

plt.title('Naive Bayes & SVM Precision, Recall, FScore Comparison Graph')

plt.show()

def extensionGraph():

height = [svm\_acc, nb\_acc, pso\_svm\_acc, pso\_nb\_acc]

bars = ('SVM Precision','Naive Bayes Precision','PSO SVM Precision','PSO Naive Bayes Precision')

y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='Detection of Prohibited Messages and Images')

title.config(bg='LightGoldenrod1', fg='medium orchid')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=18,width=100)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=350)

text.config(font=font1)

font1 = ('times', 12, 'bold')

l1 = Label(main, text='Input Hashtag:')

l1.config(font=font1)

l1.place(x=50,y=100)

tf1 = Entry(main,width=80)

tf1.config(font=font1)

tf1.place(x=170,y=100)

uploadButton = Button(main, text="Offline Upload Twitter Dataset", command=uploadDataset)

uploadButton.place(x=450,y=150)

uploadButton.config(font=font1)

cleanButton = Button(main, text="Clean Tweets & Extract Features", command=cleanTweets)

cleanButton.place(x=50,y=200)

cleanButton.config(font=font1)

suspiciousButton = Button(main, text="Suspicious Tweets Classification using SVM & Naive Bayes", command=suspiciousDetection)

suspiciousButton.place(x=450,y=200)

suspiciousButton.config(font=font1)

classifierButton = Button(main, text="SVM & CNN Classification for Gender & age Prediction", command=ageGenderPredict)

classifierButton.place(x=50,y=250)

classifierButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

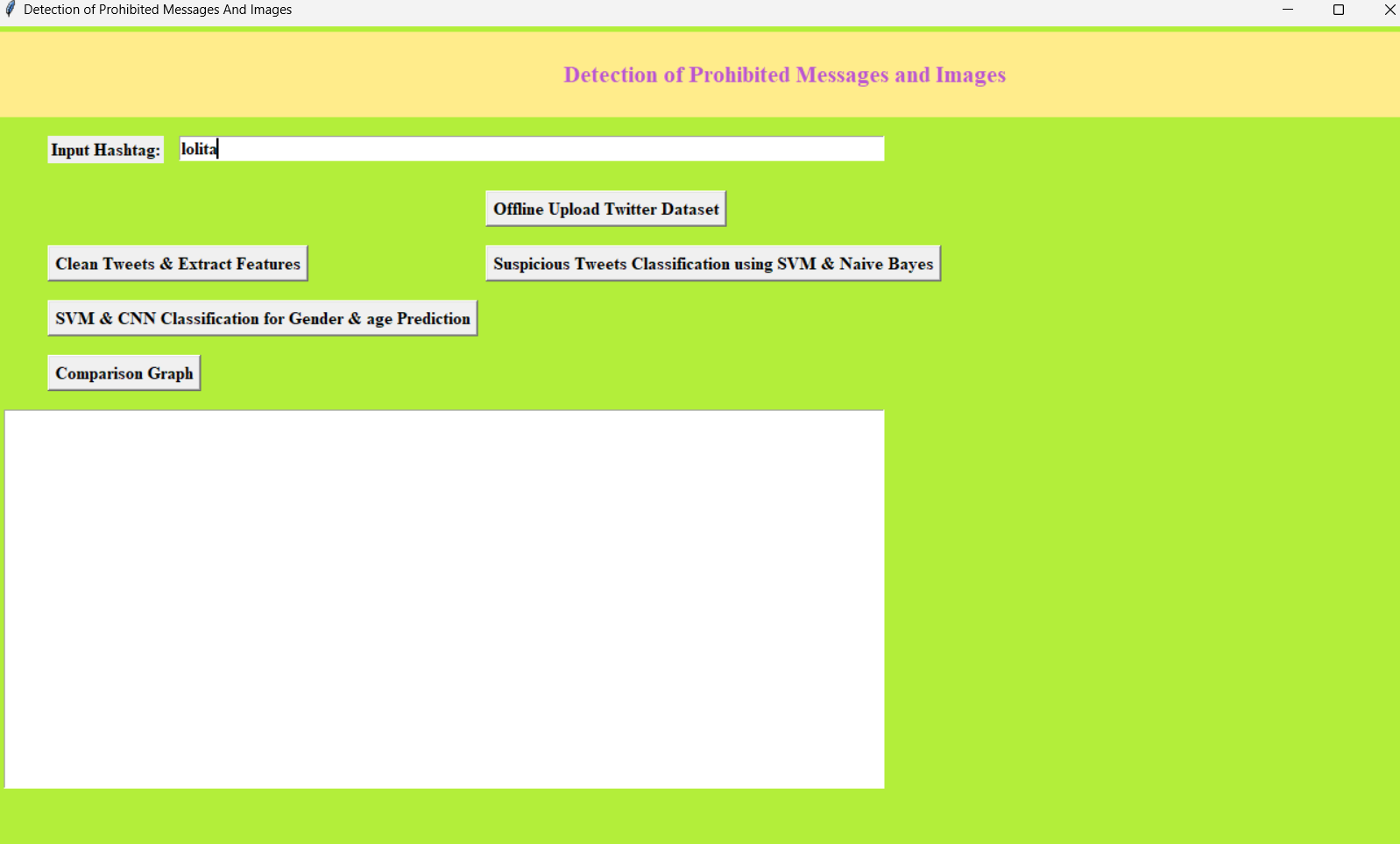
graphButton.place(x=50,y=300)

graphButton.config(font=font1)

main.config(bg='OliveDrab2')

main.mainloop()

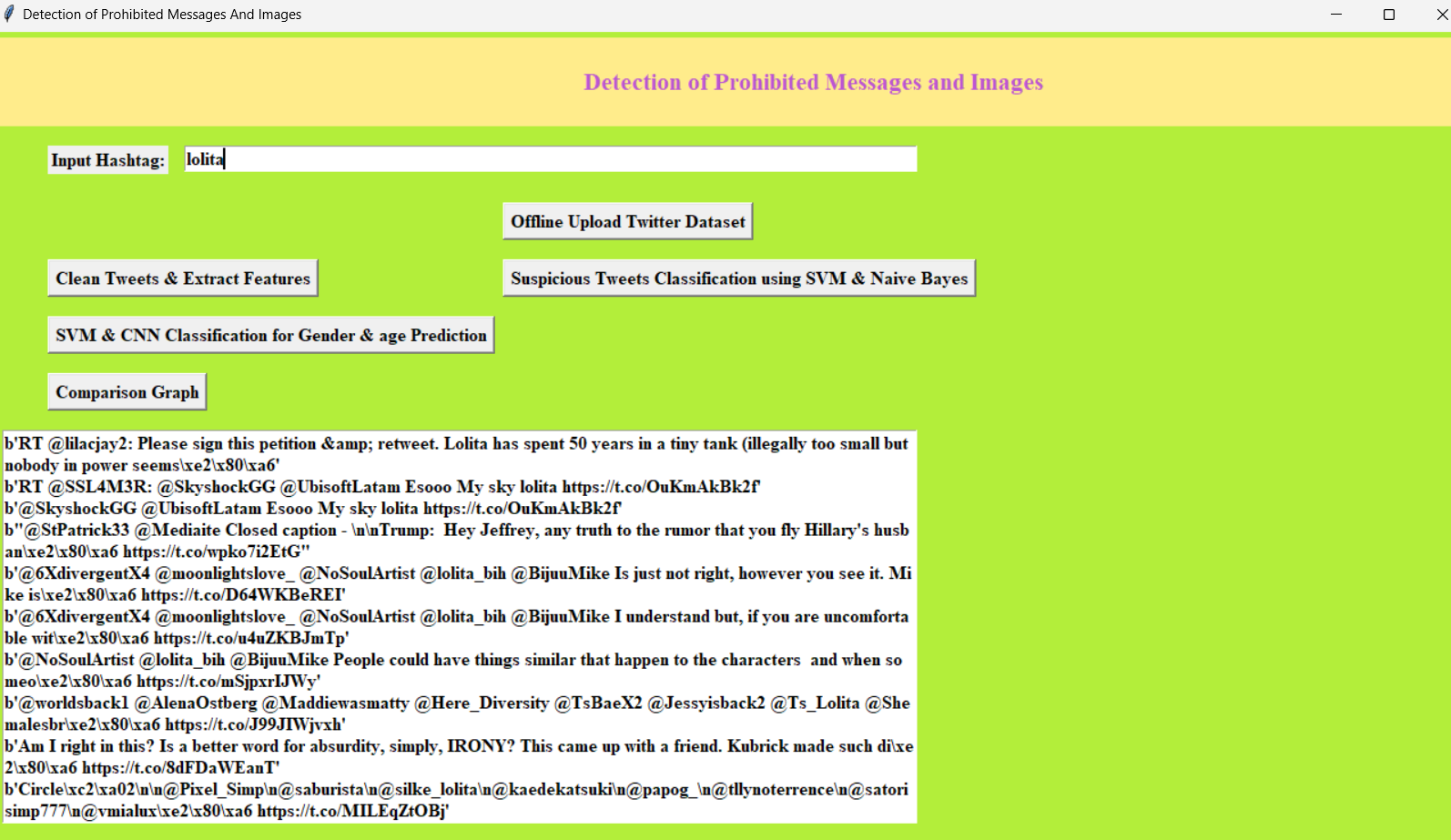
**10. RESULTS**

****

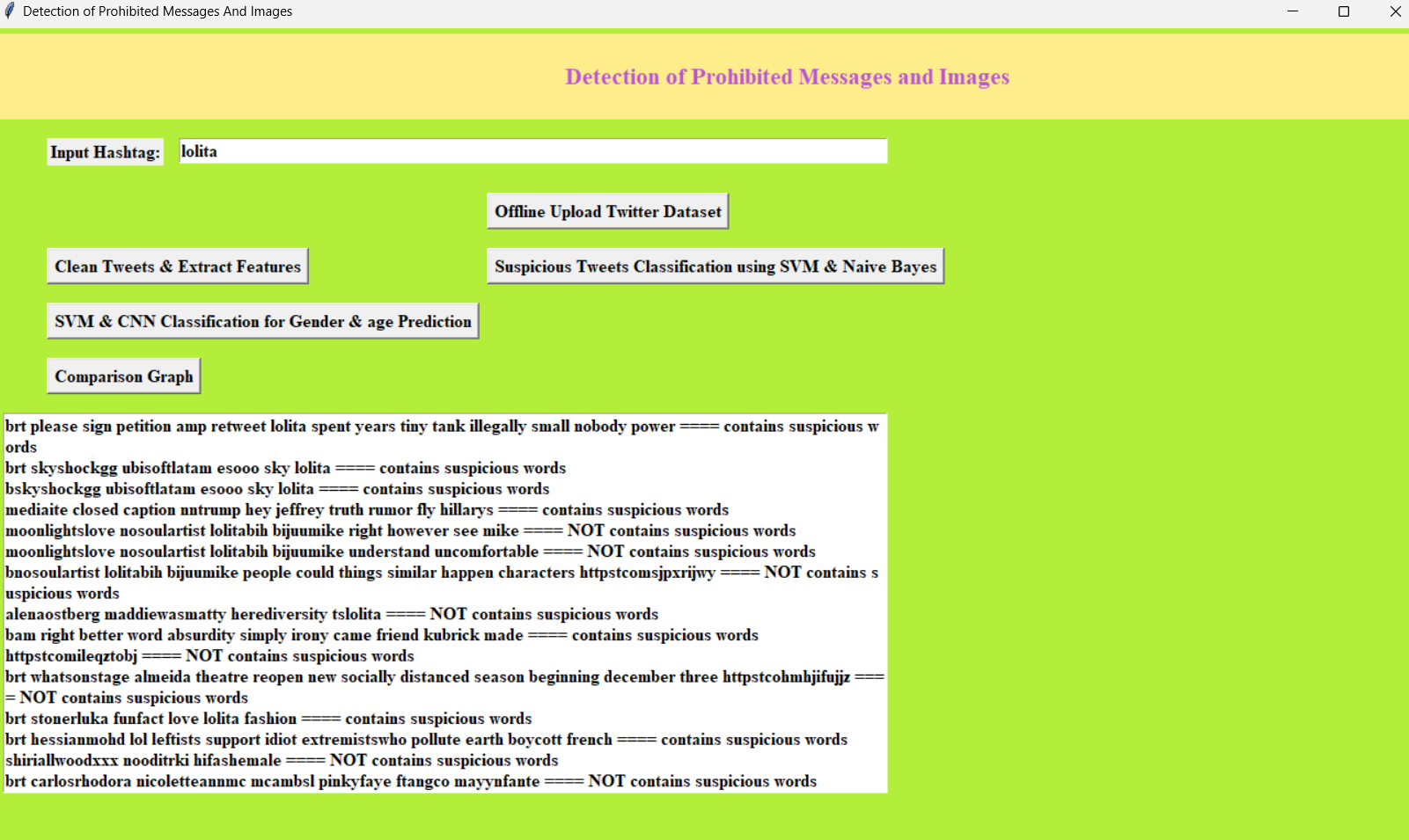
**Fig No 1.1 UI Screen**



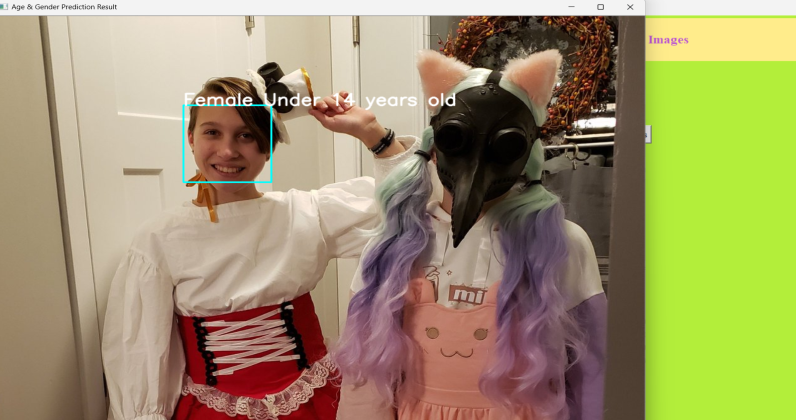
**Fig No 1.2 Uploading offline dataset**



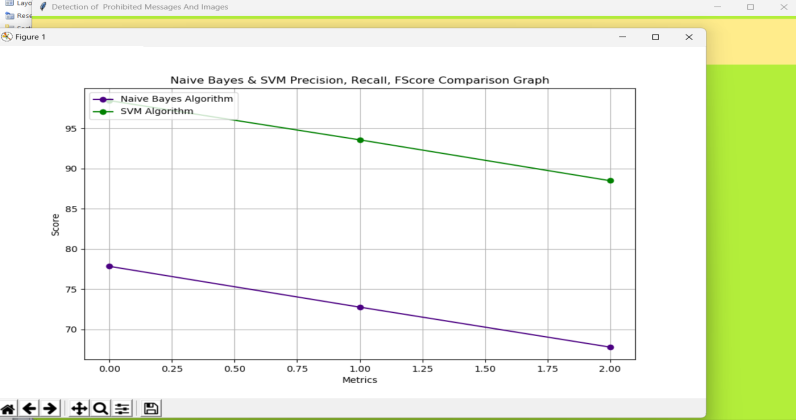
**Fig No 1.3 Cleaing of Tweets and Extracting fetures from provided dataset**

****

**Fig No 1.4 Suspicious tweets classification**

****

**Fig No 1.5 Gender and Age Prediction**

****

**Fig No 1.6 Comparison Graph**

**11. CONCLUSION**

In this work, we probed that satisfactory performance can be obtained using just geometric features of the torso and not only facial characteristics. For this paper, Haar filters combined with an SVM classifier were used for the extraction process of features, and then we classified the age group and gender with an SVM classifier. The obtained results were compared with the outcomes of a CNN algorithm. SVM is a model widely accepted, and in this work, we obtained a classification accuracy higher than 80% for both experiments (face and upper body), not only for gender classification but also for age group classification. In this paper, our main contribution is the image classification based on the upper body to predict the age group to detect human trafficking. To the best of our knowledge, this work is the first approach related to image classification without facial features but just the upper-body geometric characteristics. Currently, there is no similar research that takes into account only the upper body features of minors. Thus, the results of this paper can be applied to human trafficking, disappearance, kidnapping, among others. Moreover, the obtained information can be used by the police or other security institutions.

**12. FUTURE SCOPE**

The future scope of the project "Detection of Prohibited Messages and Images Using SVM and CNN Algorithms on Twitter and Related Websites" includes expanding the system's capabilities to cover more diverse social media platforms and languages, enhancing its adaptability to evolving content. Incorporating advanced techniques like transformers and GANs could improve detection accuracy for nuanced and sophisticated content. Additionally, real-time detection and user alert systems can be developed to provide immediate interventions. Collaborations with social media companies could lead to more integrated and comprehensive content moderation solutions, ultimately contributing to safer online environments across various digital platforms.

**13. REFERENCES**

1. E. Cockbain and K. Olver, ‘‘Child trafficking: Characteristics, complexities, and challenges,’’ in Child Abuse Neglect. Amsterdam, The Nether-lands: Elsevier, 2019, pp. 95–116.
2. M. Hernandez-Alvarez, ‘Detection of possible human trafficking in Twitter,’’ in Proc. Int. Conf. Inf. Syst. Softw. Technol. (ICIST), Nov. 2019, pp. 187–191.
3. E. Tong, A. Zadeh, C. Jones, and L.-P. Morency, ‘‘Combating human trafficking with deep multimodal models,’’ 2017.
4. ] H. Alvari, P. Shakarian, and J. E. K. Snyder, ‘‘A non-parametric learning approach to identify online human trafficking,’’ in Proc. IEEE Conf. Intell. Secur. Informat. (ISI), Sep. 2016, pp. 133–138.
5. S. Sarkar, ‘‘Use of technology in human trafficking networks and sexual exploitation: A cross-sectional multi-country study,’’ Trans. Social Rev., vol. 5, no. 1, pp. 55–68, Jan. 2015.