Phishing Email and Fake logo detection using Browser Extension

Problem Statement

The Internet has become an indispensable part of our lives. Internet services are used daily by millions of people to communicate around the globe. However, it has also provided opportunities to perform malicious activities anonymously, such as phishing. Phishers attempt to deceive their victims through social engineering or by creating fake websites to steal sensitive information such as account IDs, usernames, and passwords from individuals and organizations.

Abstract

Phishing attacks via email have been increasing day by day users are often getting mislead by fake urls and counterfeit logos posing a serious threat for sensitive information of users. This project presents a browser extension that helps in enhancing security within Gmail by detecting phishing emails which consist threatening URLs and fake logos that might lead users into traps. The extension works by scanning the entire content in the email , analyzing embedded urls and brand logos displayed within the email body. This extension integrates URL scanning techniques to classify them as safe and unsafe and uses a Trained Deep Learning model that compares the visible logos against the real brand logos and classifies them into real or fake. This extension reduces the rate of phishing combining cybersecurity with computer vision offering security to users.

Introduction

Motivation for the Work

Although Gmail successfully filters most spam and phishing emails into the spam folder, there are still cases where phishing emails bypass these filters and land in the inbox. For instance, a phishing email was observed claiming that YouTube's terms and conditions had changed, urging the user to click a link, which ultimately led to a phishing attack.

Real-World Applications

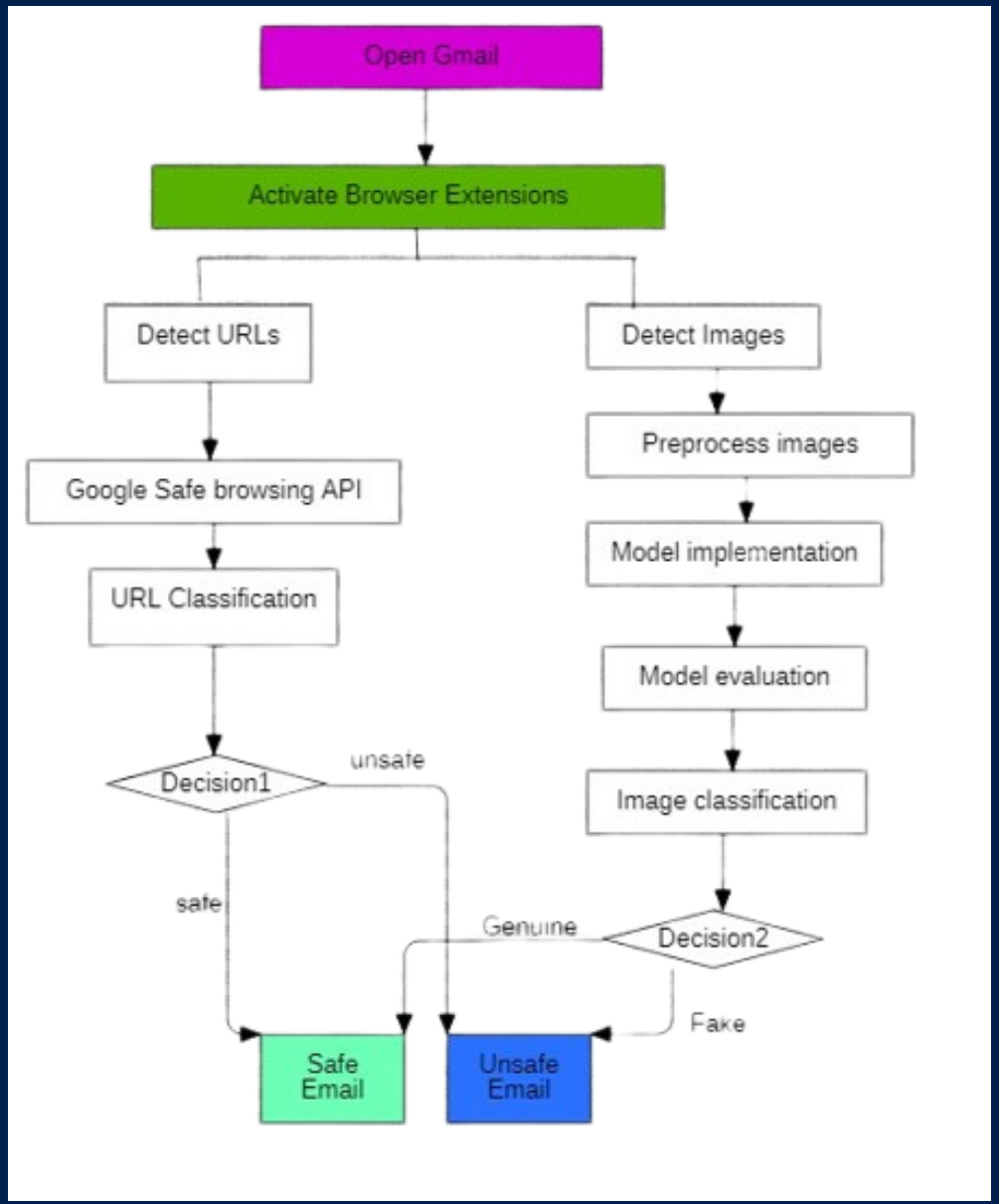
Currently, users are expected to visit separate URL scanners to verify each suspicious link received in their Gmail. This is inconvenient and time-consuming. Hence, this browser extension was developed to address this issue and provide advanced, seamless protection within the Gmail environment.

Methodology

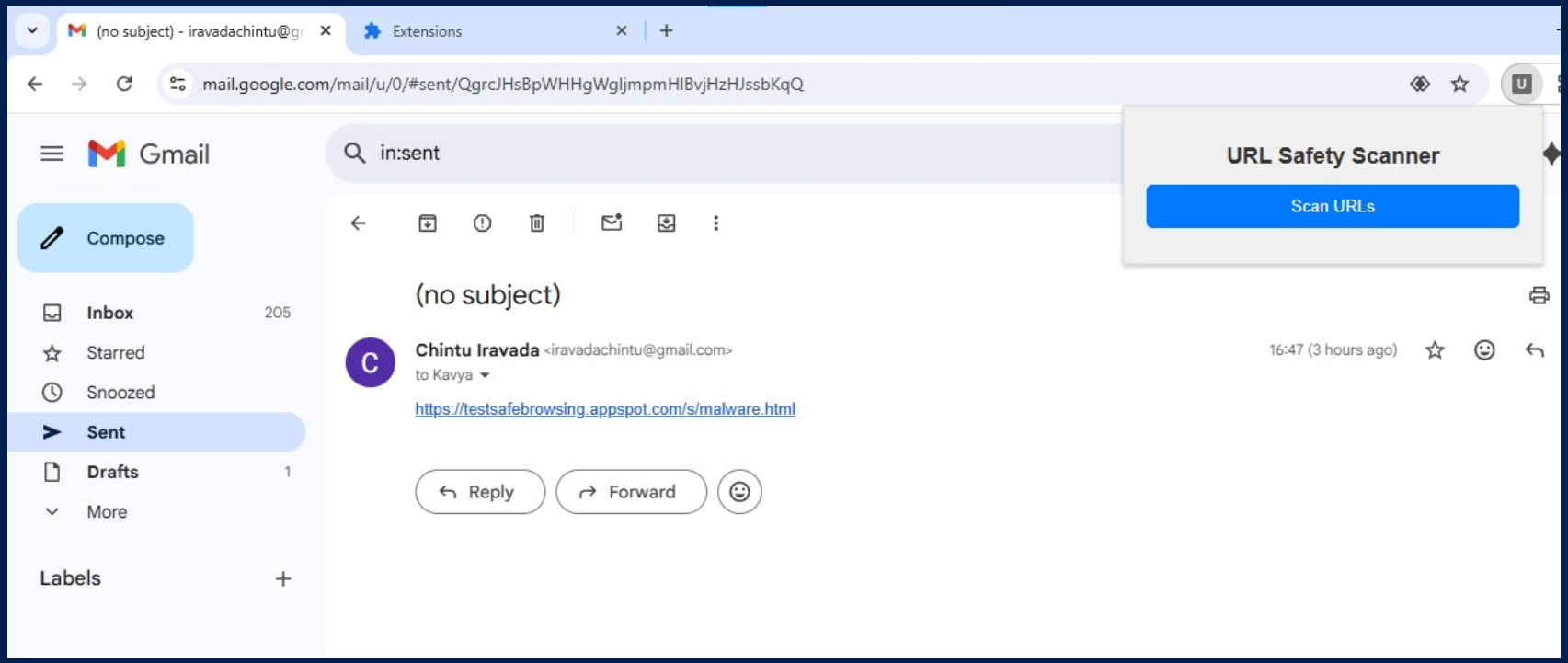
Workflow Stages

- Activate browser extensions
- Open Gmail
- Detect URLs and Images
- Model Implementation
- Model Evaluation
- Classification
- Output

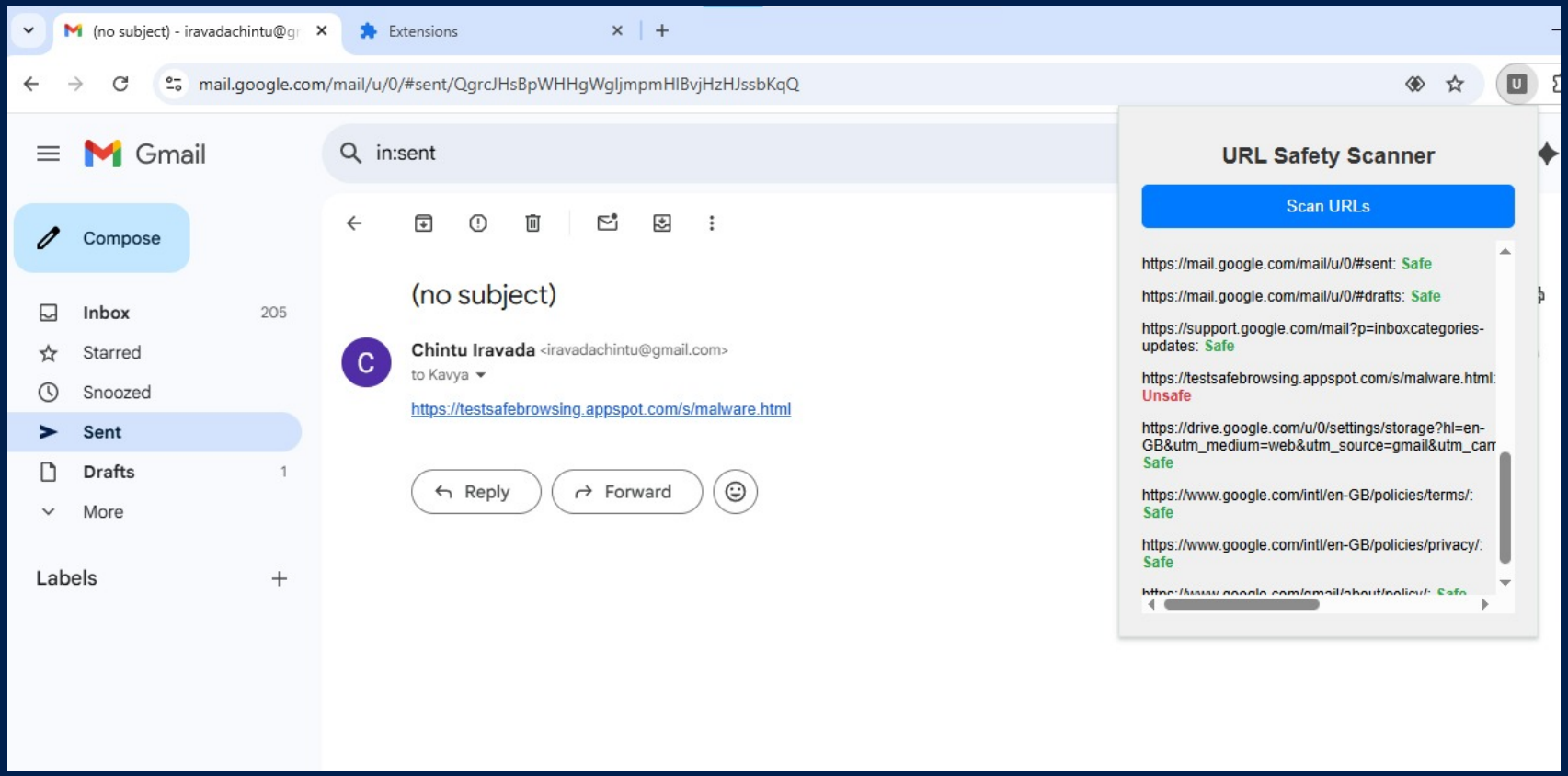
FlowChart


Open Gmail
Activate Browser Extensions
Detect URLs
Detect Images
Google Safe browsing API
Preprocess images
URL Classification
Model implementation
Model evaluation
Decision1
unsafe
Image classification
safe
Genuine
Decision2
Fake
Safe Email
Unsafe Email

Input


(no subject) - iravadachintu@g
Extensions
mail.google.com/mail/u/0/#sent/QgrcJHsBpWHHgWgljmpmHlBvjHzHJssbKqQ
Gmail
in:sent
URL Safety Scanner
Scan URLs
Compose
Inbox 205
Starred
Snoozed
Sent
Drafts 1
More
Labels
(no subject)
Chintu Iravada <iravadachintu@gmail.com>
to Kavya
16:47 (3 hours ago)
https://testsafebrowsing.appspot.com/s/malware.html
Reply
Forward

Output


(no subject) - iravadachintu@g
Extensions
mail.google.com/mail/u/0/#sent/QgrcJHsBpWHHgWgljmpmHlBvjHzHJssbKqQ
Gmail
in:sent
Compose
Inbox 205
Starred
Snoozed
Sent
Drafts 1
More
Labels
(no subject)
Chintu Iravada <iravadachintu@gmail.com>
to Kavya
https://testsafebrowsing.appspot.com/s/malware.html
Reply
Forward
URL Safety Scanner
Scan URLs
https://mail.google.com/mail/u/0/#sent: Safe
https://mail.google.com/mail/u/0/#drafts: Safe
https://support.google.com/mail?p=inboxcategories-updates: Safe
https://testsafebrowsing.appspot.com/s/malware.html: Unsafe
https://drive.google.com/u/0/settings/storage?hl=en-GB&utm_medium=web&utm_source=gmail&utm_cam
Safe
https://www.google.com/intl/en-GB/policies/terms/:
Safe
https://www.google.com/intl/en-GB/policies/privacy/:
Safe

Experimental Setup

Technologies Used:

- Frontend: HTML, CSS, JavaScript, Python.
- Backend: Flask API
- Dataset: Genuine and Fake Logo
- Core AI & ML Technologies: Pandas, torch, torchvision.models, torchvision.transforms, torch.utils.data.DataLoader, PIL.Image, os, cv2

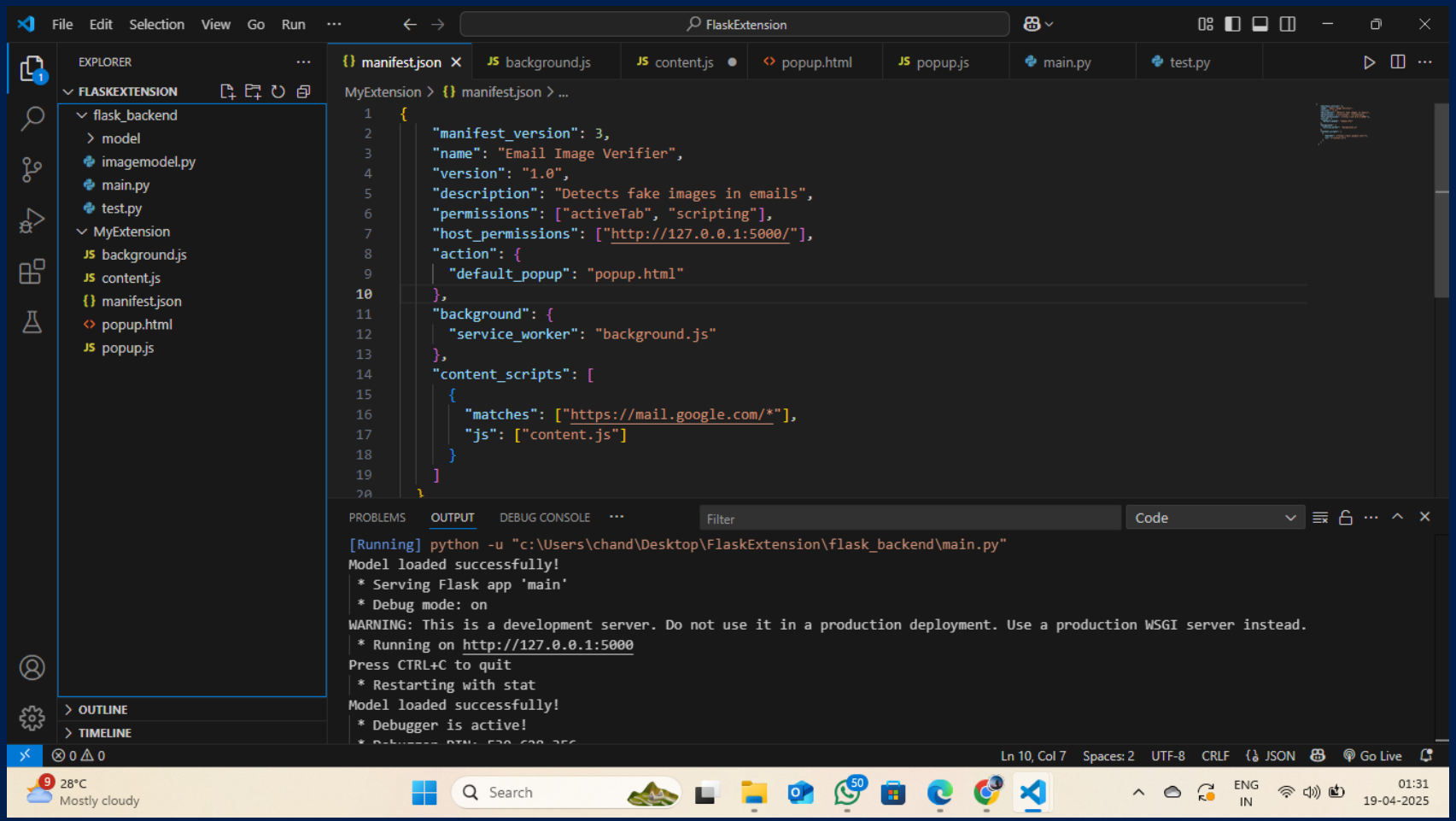
System Hardware:

- **Processor:** 64-bit, Core i5, 2.5 GHz minimum per core
- **RAM:** 8 GB or more
- **HDD:** 20 GB of available space or more
- **Display:** Dual XGA (1024×768) or higher resolution monitors
- **Keyboard:** A standard keyboard

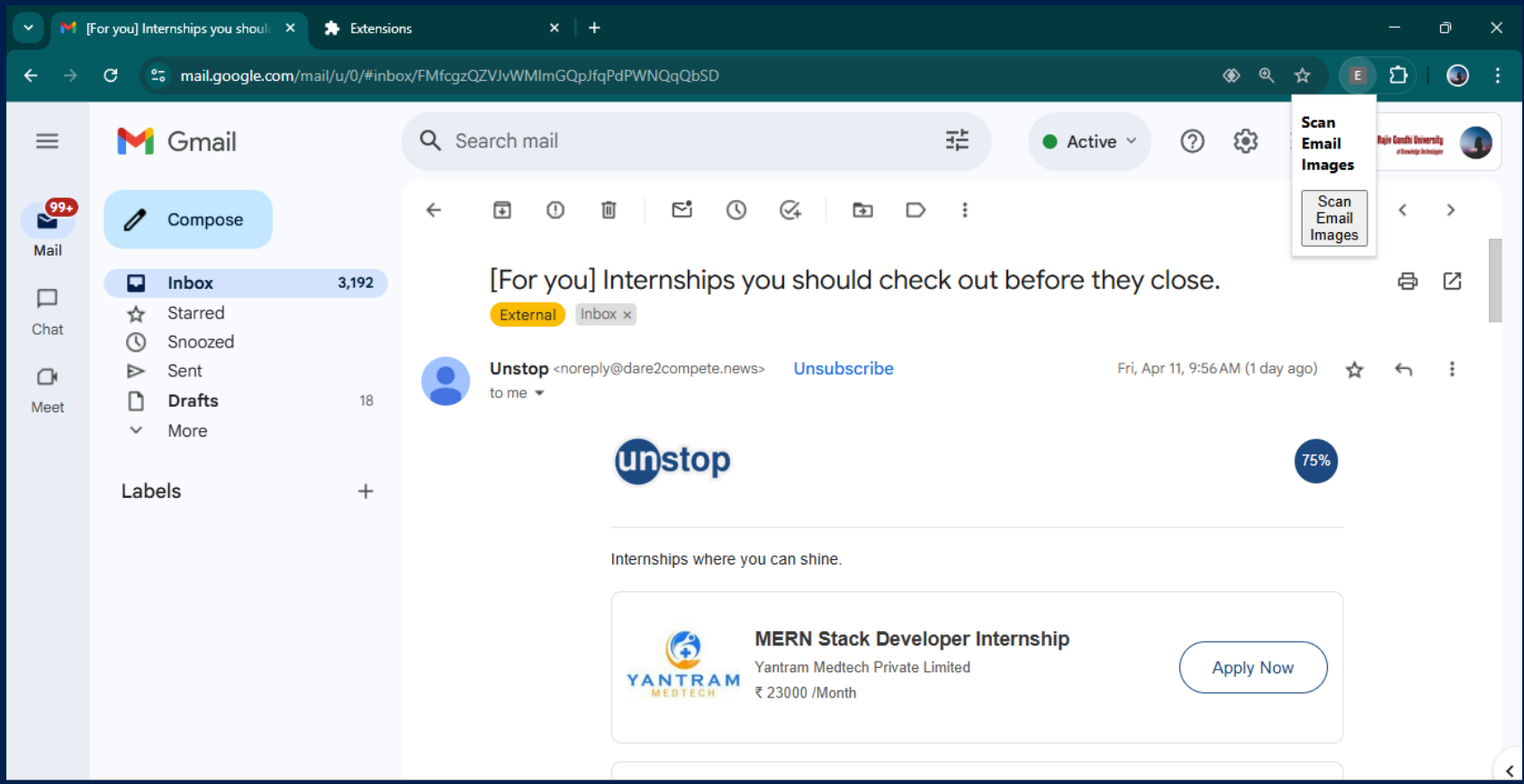
Resnet Model

ResNet-50, a deep convolutional neural network, is used to identify counterfeit logos through deep learning. It learns intricate patterns from im ages and distinguishes between authentic and counterfeit logos. This model is mostly used in image classification, object detection, facial recognition, medical imageing.
The usage of loss function and optimizers, such as adam and cross entropy, plays a vital role in enhancing model performance .

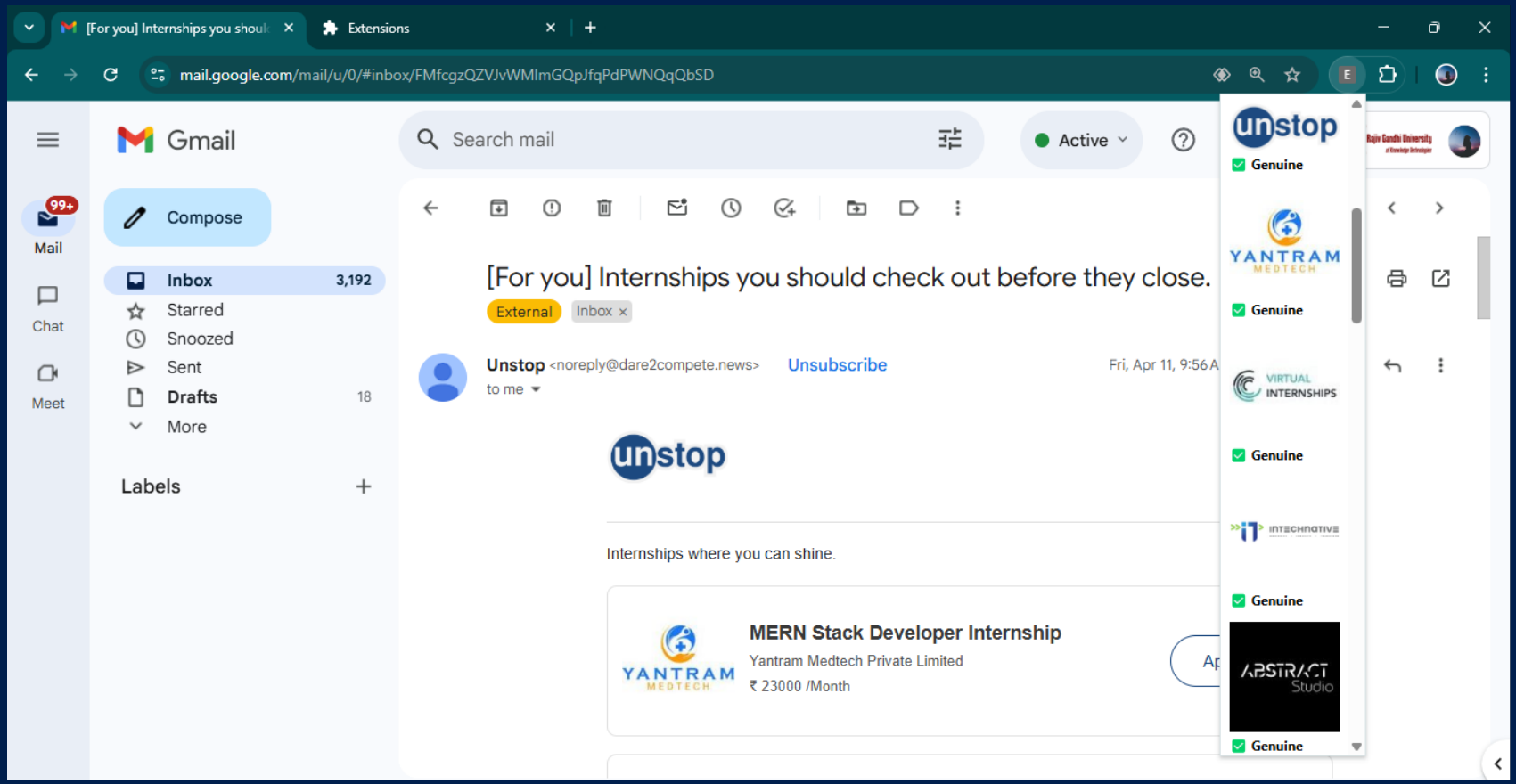
Input


File Edit Selection View Go Run
FlaskExtension
EXPLORER
FLASKEXTENSION
flask_backend
model
imagemodel.py
main.py
test.py
MyExtension
background.js
content.js
manifest.json
popup.html
popup.js
manifest.json
background.js
content.js
popup.html
popup.js
main.py
test.py
MyExtension > manifest.json > ...
{
"manifest_version": 3,
"name": "Email Image Verifier",
"version": "1.0",
"description": "Detects fake images in emails",
"permissions": ["activeTab", "scripting"],
"host_permissions": ["http://127.0.0.1:5000/"],
"action": {
"default_popup": "popup.html"
},
"background": {
"service_worker": "background.js"
},
"content_scripts": [
{
"matches": ["https://mail.google.com/*"],
"js": ["content.js"]
}
]
}
PROBLEMS OUTPUT DEBUG CONSOLE
Filter
Code
[Running] python -u "c:\Users\chand\Desktop\FlaskExtension\flask_backend\main.py"
Model loaded successfully!
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Model loaded successfully!
* Debugger is active!
OUTLINE
TIMELINE
Ln 10, Col 7 Spaces: 2 UTF-8 CRLF JSON Go Live
28°C
Mostly cloudy
Search
ENG
IN
01:31
19-04-2025

Input


[For you] Internships you shoul
Extensions
mail.google.com/mail/u/0/#inbox/FMfcgzQZVJvWMImGQpJfqPdPWNQqQbSD
Gmail
Search mail
Active
Scan Email Images
Scan Email Images
Compose
Inbox 3,192
Starred
Snoozed
Sent
Drafts 18
More
Labels
Mail
Chat
Meet
[For you] Internships you should check out before they close.
External
Inbox ×
Unstop <noreply@dare2compete.news>
Unsubscribe
Fri, Apr 11, 9:56 AM (1 day ago)
to me
unstop
75%
Internships where you can shine.
MERN Stack Developer Internship
Yantram Medtech Private Limited
₹ 23000 /Month
Apply Now
YANTRAM MEDTECH

Output


[For you] Internships you should check out before they close.
Unstop <noreply@dare2compete.news>
Unsubscribe
to me
Internships where you can shine.
MERN Stack Developer Internship
Yantram Medtech Private Limited
₹ 23000 /Month
Genuine
Genuine
Genuine
Genuine
Genuine

Conclusion

Email is an inexpensive, effective, and fast way to exchange messages using the internet. Spam email is annoying to end-users, financially damaging, and can be a security risk. The objective of spam email is to collect sensitive personal information about users. The majority of emails in internet traffic contain spam. This work uses deep learning method such as CNN to classify Spam and Not-Spam messages.

This project demonstrates the powerful capabilities of convolutional Neural Networks (CNNs) integrated with Web browser for email spam detection. Traditional intrusion detection systems often rely on manual feature extraction, which can miss crucial relationships and patterns.

Future Scope

In actuality we have made two different browser extensions, where one is used to detect threat URLS and the other is for detecting fake logos. Our URL extension works on Google safe browsing API and the fake logo detection extension is based on RESNET-50 pretrained model. We have an idea of integrating both extensions into one and some extra features where we can be able to detect deepfake images and threatening text along with threat URLs.
To make this happen we have thought of two approaches

- Either using parallel implementation of models.
- Pipeline of the extensions to integrate all models into one.

We would also like to integrate some more features that come under phishing and make the extensions useful in the real world.

References

- A. Li-Lian, "Image classification with ResNet (PyTorch)," 2022.
- D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer," 2021.
- A. Sheneamer, "Comparison of deep and traditional learning methods for email spam filtering," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 1, 2021.

Thank You!