Project Report Format

1. INTRODUCTION

1.1 Project Overview

Fake/Real logo detection is a computer vision problem that can be tackled using Convolutional Neural Networks (CNNs). One such CNN architecture is VGG19, which has pre-trained layers and a great understanding of what defines an image in terms of shape, color, and structure 1. The CNN architecture contains a combination of layers that transform an image into output the model can understand. The convolutional layer creates a feature map by applying a filter that scans the image several pixels at a time. The pooling layer scales down the information generated by the convolutional layer to effectively store it. The fully connected input layer flattens the outputs into a single vector. The fully connected layer applies weights over the inputs generated by the feature analysis. The fully connected output layer generates final probabilities to determine the image class.

The process of fake/real logo detection involves forward and backward propagation iterating through all of the training samples in the network until the optimal weights are determined and only the most powerful and predictive neurons are activated to make a prediction. The model trains throughout many epochs by taking one forward and one backward pass of all training samples each time. Forward propagation calculates the loss and cost functions by comparing the difference between the actual and predicted target for each labeled image. Backward propagation uses gradient descent to update the weights and bias for each neuron, attributing more impact on the neurons which have the most predictive power, until it arrives at an optimal activation combination. As the model sees more examples, it learns to better predict the target causing the loss measure to decrease. The cost function takes the average loss across all samples indicating overall performance.

To deploy the model, Flask, a Python web framework, can be used. Flask provides a simple way to deploy machine learning models as web applications. The model can be loaded into the Flask application and the user can upload an image to the web application. The image is then passed through the model and the result is displayed on the web page. Flask also provides a way to host the web application on a server so that it can be accessed from anywhere.

1.2 Purpose

The purpose of the project is to develop a system that can automatically analyze visual characteristics and patterns in logo images, enabling the classification of logos as either genuine or fake. The project uses Convolutional Neural Networks (CNNs) to detect fake/real logos. One such CNN architecture is VGG19, which has pre-trained layers and a great understanding of what defines an image in terms of shape, color, and structure.

The CNN architecture contains a combination of layers that transform an image into output the model can understand. The process of fake/real logo detection involves forward and backward propagation iterating through all of the training samples in the network until the optimal weights are determined and only the most powerful and predictive neurons are activated to make a prediction.

To deploy the model, Flask, a Python web framework, can be used. Flask provides a simple way to deploy machine learning models as web applications. The model can be loaded into the Flask application and the user can upload an image to the web application. The image is then passed through the model and the result is displayed on the web page. Flask also provides a way to host the web application on a server so that it can be accessed from anywhere.

2. LITERATURE SURVEY

2.1 Existing problem and References

1) Deep Fake Image Detection Based on Pairwise Learning

In this paper, a fake feature network-based pairwise learning is proposed to detect the fake face and general images generated by the state-of-the-art GANs. The proposed CFFN can be used to learn the middle- and high-level and discriminative fake features by aggregating the cross-layer feature representations. The proposed pairwise learning strategy enables the fake feature learning, which allows the trained fake image detector to have the ability to detect the fake image generated by a new GAN, even it was not included in the training phase. The experimental results demonstrated that the proposed method outperformed other state-of-the-art methods in terms of precision and recall rate. The fake video detection is also an important issue, so in our future work, we will extend the proposed method to fake video detection, incorporating the object detection and Siamese network structure.

Reference - Hsu, C.-C.; Zhuang, Y.-X.; Lee, C.-Y. Deep Fake Image Detection Based on Pairwise Learning. Appl. Sci. 2020, 10, 370. https://doi.org/10.3390/app10010370

2)Aesthetic Text Logo Synthesis via Content-Aware Layout Inferring

Reference - Yizhi Wang, Guo Pu, Wenhan Luo, Yexin Wang, Pengfei Xiong, Hongwen Kang, Zhouhui Lian; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 2436-2445

In this paper, we proposed a layout generation network
for aesthetic text logo synthesis. Both the linguistic and visual information of input texts and glyph images were taken
into account for layout prediction. Moreover, a novel dualdiscriminator module was designed to capture both the placing trajectories and detailed shapes of characters for synthesizing high-quality layouts. We built a large-scale dataset,
TextLogo3K, on which extensive experiments (both quantitative and qualitative) were conducted to verify the effectiveness of our method.
Finally, a prototype system was developed to automatically synthesize aesthetic text logos by combing our layout generation network with existing font
generation and texture transfer models.

3) Deep Learning for Logo Detection: A Survey

Reference - Sujuan Hou, Member, IEEE, Jiacheng Li, Weiqing Min, Senior Member, IEEE, Qiang Hou, Yanna Zhao, Member, IEEE, Yuanjie Zheng, Member, IEEE, and Shuqiang Jiang, Senior Member, IEEE

With deep learning's ongoing advancement, deep learning-based solutions have also been effectively used for logo categorization in recent years. Using the DCNN logo identification algorithm, Karimi et al. first extracted features from a pre-trained model and then classified logos using SVM. They also enhanced pre-existing pre-trained models for logo identification using transfer learning. Ultimately, a more effective deep model for logo identification is obtained by applying the refined deep model to the parallel structure. Creating effective networks with few resources is the newest trend in logo categorization A Discriminative Region Navigation and Augmentation Network (DRNA-Net) was proposed by Wang et al. it may enlarge these regions for logo discovery and find additional informative regions.

4) Identification of Fake VS Original logos using Deep Learning

Reference - Ranjith K C1,Sharath Kumar Y H2
1Assistant Professor, Department of Computer Science & Engineering,Mandya, (Karnataka), INDIA
2Professor, Department of Information Science & Engineering,Mandya, (Karnataka), INDIA

In this paper, we have presented a method for Fake logo detection using fusion approaches of SIFT and SURF features. Features of the logo images are extracted using the SIFT and SURF techniques and the features are fused to obtain the significant features to detect the query image. Further the features were reduced with the use of PCA. The obtained features are stored for the database for querying the input image to determine whether it is a valid or fake logo. The experimentation is done with the own dataset of 144 image of six different brand car ogos were considered out of which 20 images were of fake logos.

5)Online Fake Logo Detection System

DOI - https://doi.org/10.21203/rs.3.rs-2492597/v1

Reference - Vivek Tanniru Department of Computer Science Auburn university at Montgomery Montgomery, Alabama,USA vtanniru@aum.edu
Dr.Tathagata Bhattacharya,Asst.Professsor Department of Computer Science Auburn University at Montgomery Montgomery,Alabama,USA tbhatta1@aum.edu

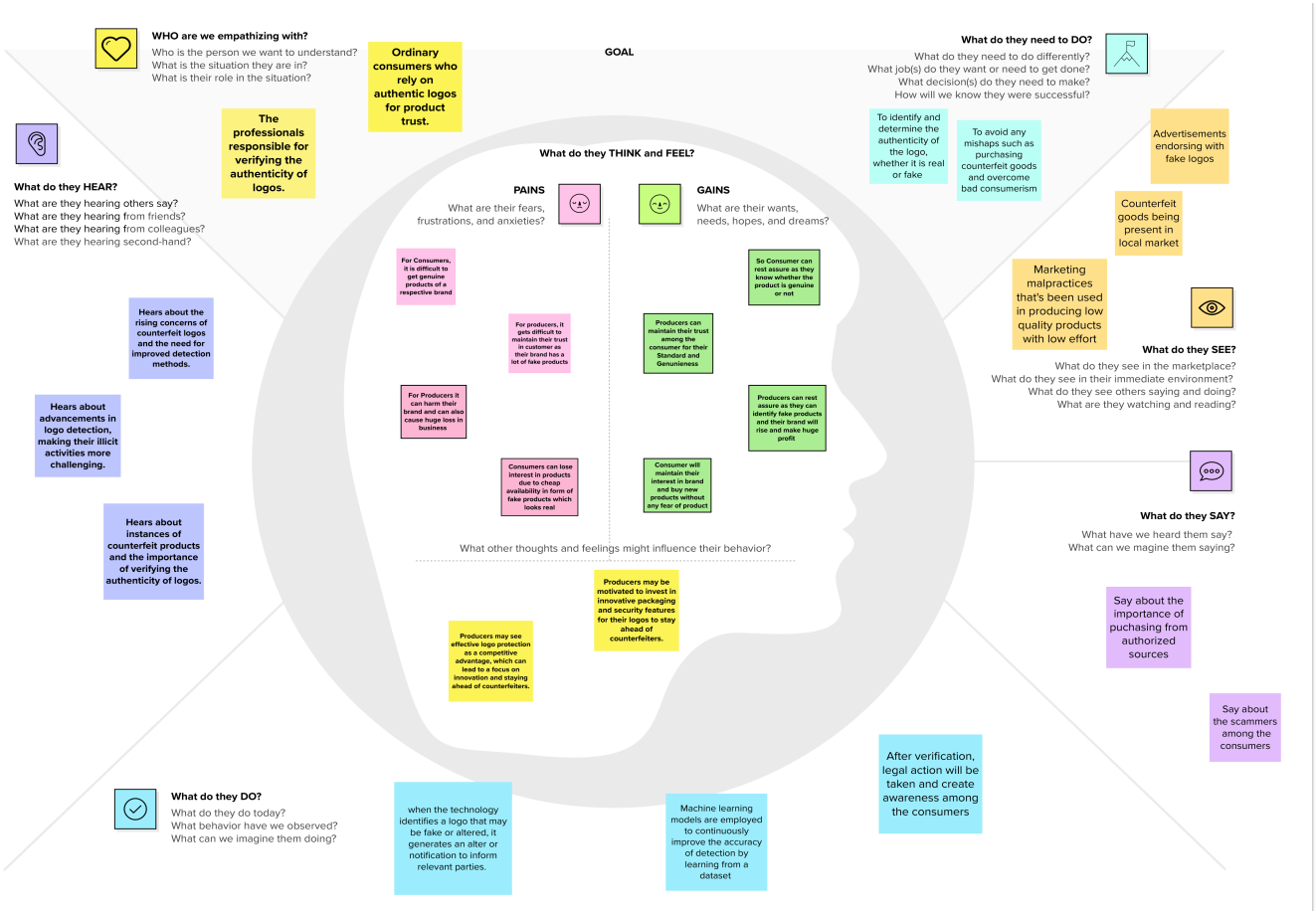
I ran into two main issues when creating a logo detection system: I didn't know how to train a model for some of the necessary techniques, and I had trouble producing photos with non-standard resolutions. These issues compelled me to create my own technique for differentiating logos based on resolution differences. The foundation of this implementation is convolutional neural networks. I was able to develop this technique and obtain an accuracy rate of more than 60 on the test data provided by Yahoo logo detection as well as the MNIST dataset. Compared to alternative solutions that require 40 hours or more for training, this system simply required 37 minutes for training. This method, I believe, may be utilized to automatically identify

2.2 Problem Statement Definition

Counterfeit logos pose a significant threat to brand integrity and consumer trust. With the rise of fake products flooding the market, there is a pressing need for an effective and automated system to differentiate between genuine and fraudulent logos. Current methods are often manual, time-consuming, and prone to errors. This project aims to address these challenges by employing deep learning algorithms to create a reliable logo detection system, offering a proactive solution to identify and combat the proliferation of fake logos in the digital and physical realms.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

Utilization of VGG19-based deep learning for the detection and authentication of logos to tackle counterfeit and fraudulent activities. This encompasses training the model for logo differentiation and leveraging it for automated inference, ultimately enhancing brand security and product authenticity.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Harsh

Develop an advanced logo detection tool with a user-friendly interface, allowing specialists to quickly upload and verify logos.

Create a mobile app that enables logo authentication specialists to verify logos in real-time using their smartphones, making the process more efficient and accessible.

Create a platform where consumers can report and share their experiences with counterfeit products, fostering a community that promotes awareness.

Kunal

We can use CNNs as they are the most popular choice for image-based tasks, including logo detection.
You can train a CNN for binary classification to distinguish between real and fake logos.
Popular CNN architectures like VGG, ResNet, and Inception are commonly used as a starting point.

We can also use Siamese networks which are used for one-shot image recognition task and it's embedded with a pre-trained model that is fine-tuned on a challenging logo dataset.and we get a encoded outputs for each image are afterwards which is compared using a trained metrics and thresholded get defined matches and mismatches. So the proposed approach gives an accuracy of 77.07% under the one-shot constraints in the QMUL-OpenLogo dataset

In our Platform we can categorize the logo in different types like fashion brand logo, different car companies logo,etc and then will study that data and try to find patterns like which brand has most fake logo , which country has the most use of these fake brands and assist the those companies providing the effective counterfeit measures this service will be exclusive for the MNC's only

Rueben

Create a web app that can use the R-CNN or Faster R-CNN algorithm to quickly detect any anomalies that are present on the logo when it is uploaded by the user to recognize its authenticity.

Create a tool that utilizes OpenCV technology to detect and understand the authenticity of the logo

Create a fake logo detection feature that can be used by third-party services in order to inspect the logo design elements, whether they are original or imitated.

Indu

Handling changes in logos over time as companies update their logos.

Creating models that can identify misuse, even when logos are resized, rotated, or incorporated into new designs

Making models more interpretable and transparent for users that provide clear explanations for their decisions

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Create a mobile app that enables logo authentication specialists to verify logos in real-time using their smartphones, making the process more efficient and accessible.

We can use CNNs as they are the most popular choice for image-based tasks, including logo detection.
You can train a CNN for binary classification to distinguish between real and fake logos.
Popular CNN architectures like VGG, ResNet, and Inception are commonly used as a starting point.

Develop an advanced logo detection tool with a user-friendly interface, allowing specialists to quickly upload and verify logos.

Create a tool that utilizes OpenCV technology to detect and understand the authenticity of the logo

Handling changes in logos over time as companies update their logos.

Creating models that can identify misuse, even when logos are resized, rotated, or incorporated into new designs

4

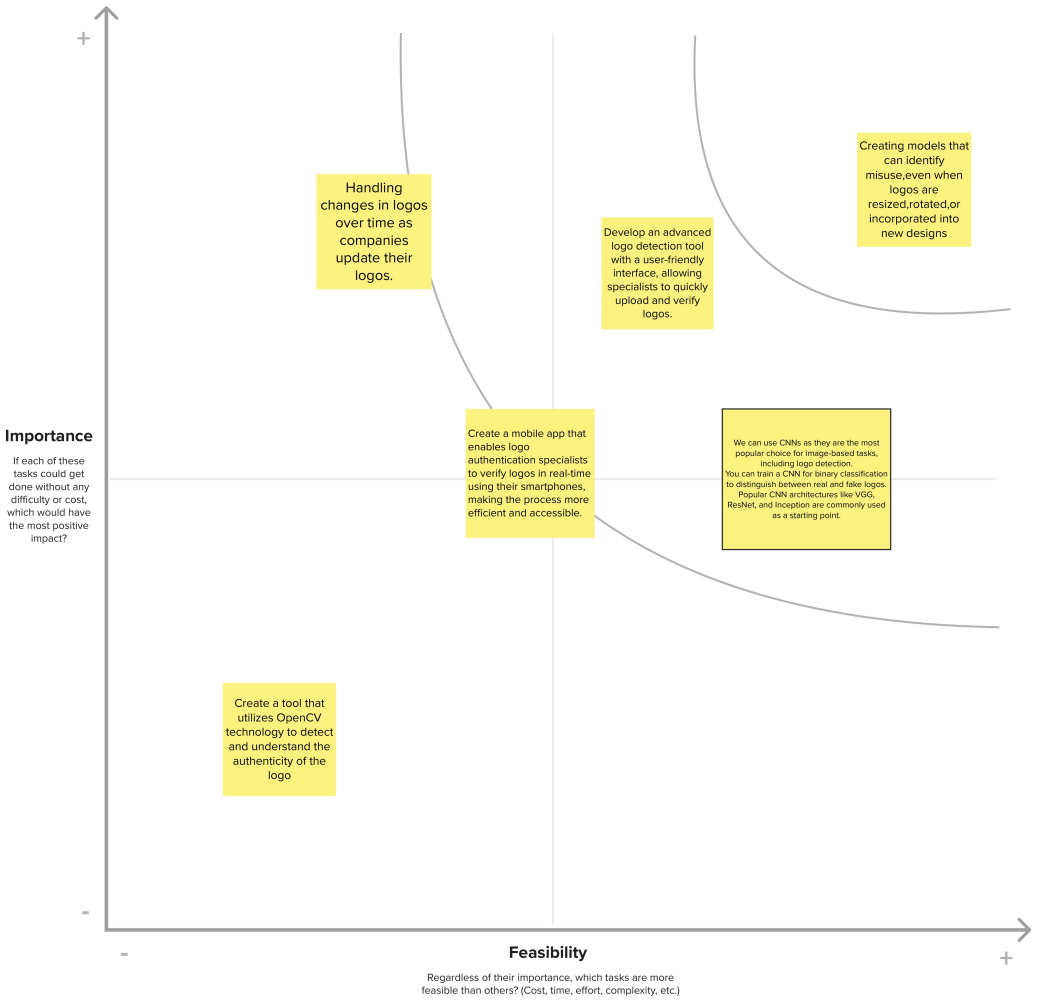
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

1. Image Input Handling: The system should be able to accept image files as input for logo detection.
2. Logo Detection Accuracy: The model should be able to accurately detect fake logos in images.
3. Real Logo Identification: The model should also be able to correctly identify genuine logos and distinguish them from fake ones.
4. Real-time Processing: If real-time processing is a requirement, specify the expected response time for logo detection.
5. Scalability: If the system is expected to handle a large number of images simultaneously, it should be able to scale efficiently.
6. Multi-Class Classification: If there are multiple classes of logos (e.g., different brands), the model should be able to classify them correctly.
7. Model Training and Updating: Include functionality to retrain or update the model with new data periodically to improve performance.
8. Logging and Reporting: The system should maintain logs of processed images, detection results, and any relevant metrics for analysis and auditing.

4.2 Non-Functional requirements

1. Accuracy and Precision: Specify the desired accuracy and precision levels for logo detection, based on the project's requirements.
2. Performance: Define the expected response time for the logo detection process, and ensure it meets the project's performance goals.
3. Scalability: Determine how the system will handle an increase in the number of images to process simultaneously, and set scalability requirements accordingly.
4. Resource Utilization: Define acceptable levels of CPU, memory, and storage usage for the system.
5. Robustness and Reliability: The system should be able to handle different types of images, lighting conditions, and common image variations.
6. Security: Specify security measures to protect against potential attacks or misuse of the system.
7. Privacy and Compliance: Ensure compliance with data protection regulations and outline measures taken to protect user privacy.
8. User Interface: Define user interface requirements, such as ease of use, accessibility,

and user feedback.

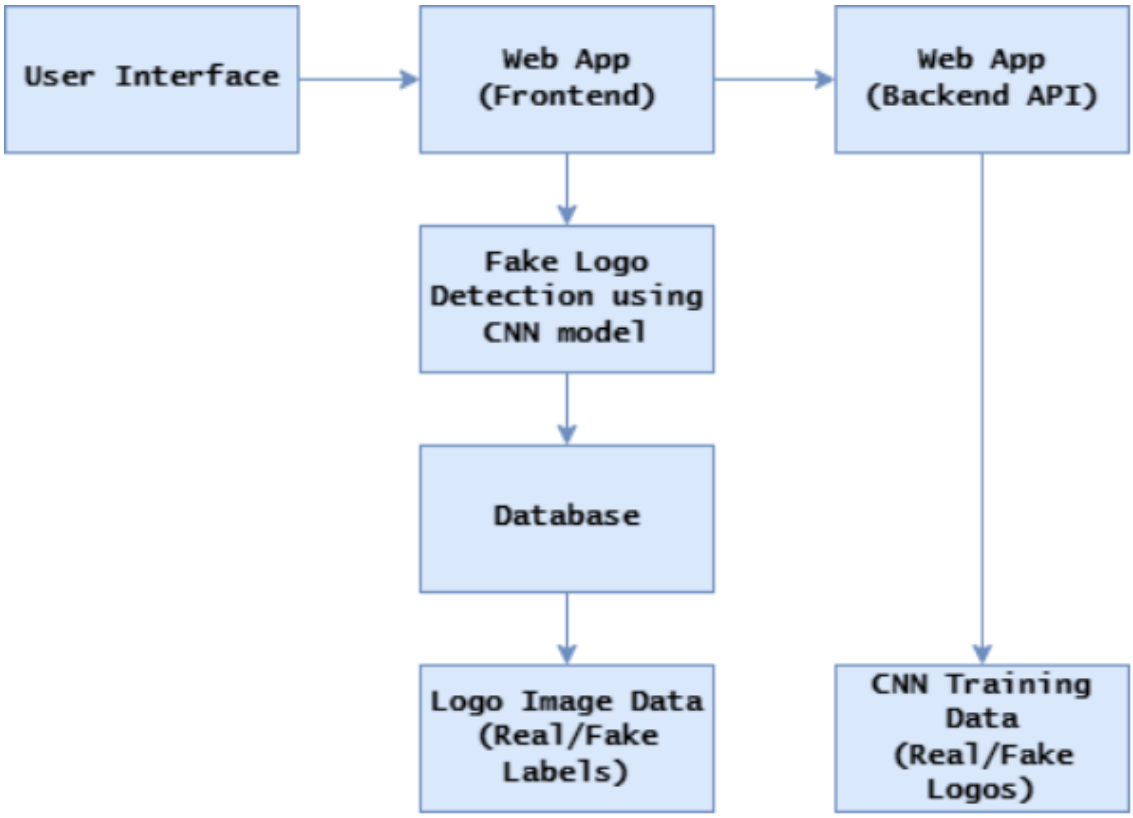
9. Documentation and Support: Provide comprehensive documentation for system installation, configuration, and usage. Include user manuals and troubleshooting guides.
10. Ethical Considerations: Consider and document any ethical considerations related to the use of the fake logo detection system, such as potential biases or fairness issues.

5. PROJECT DESIGN

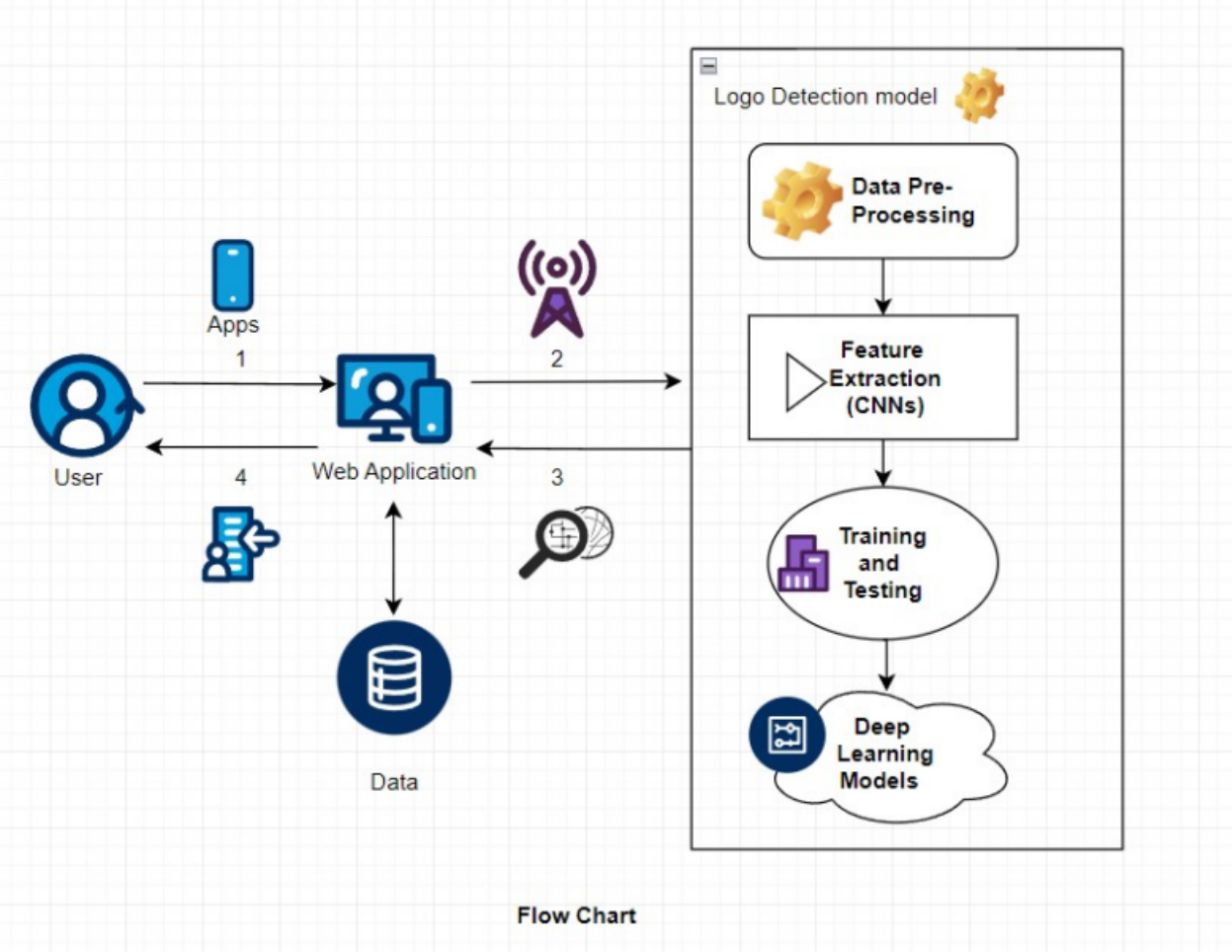
5.1 Data Flow Diagrams & User Stories

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Data Flow Diagiam:
Level 0:



Level 1:



The process of Data Flow Diagram:

1. The process begins with the user accessing the application and entering their login credentials.This provides safe access to the service. After logging in, the user uploads the image to be verified.
2. After the image is uploaded, the system performs data preprocessing to improve its quality andprepare it for analysis. This might include tasks like scaling, noise reduction, and normalization. Convolutional Neural Networks (CNNs) are used to extract detailed information from images after preprocessing. CNNs are ideal for image analysis because they can recognize patterns andstructures in visual data.
3. The preprocessed picture is now enhanced with extracted features and is sent into the deep learning models. These models have been trained to distinguish between real and counterfeit trademarks. They evaluate numerous factors such as color

schemes, forms, proportions, and other unique features. This is an important step in assessing the validity of the submitted logo orproduct.

4. The output of the deep learning models is compared with the data included in the dataset file after that. This dataset provides a large number of genuine logos and their associated attributes.The comparison ensures that the evaluation is thorough and that the system can correctly identify any inconsistencies between the uploaded image and the reference data.

5. The system produces a definite result based on its analysis when the comparison procedure is finished. This result is then displayed to the user via the application interface. The final result typically provides a clear indication of whether the logo is genuine or whether there are indicators of potential counterfeit features. The method may also produce a confidence score toassess the degree of certainty in the determination.

User Stories:

User Type	**Functional Requirement**	**User Story No.**	**User story**	**Acceptance criteria**	**priority**	**release**
General Consumer	Consumer Trust	1	As a general consumer, I want to verify the authenticity of a product's logo before making a purchase.	The system offers aneasy-to-use app. I can scan a product's logo usingmy camera. The app returns a classification result indicating whether the logo is real or fake.	High	Sprint-1
E-commerce Platform Administrator	Verify the authenticity of product listings by detecting and validating logos	2	As a E-commerce Platform Administrator, I want to ensure that all product listings on our platform accurately represent	I can review the verification results and take appropriateaction, such as removing or flagginglistings.	High	Sprint-1

			products, including the logos and branding.			
Marketing Manager	Ensure that logos in Marketing materials are used in compliance with branding guidelines.	3	As a Marketing Manager, I want to verify that logos used in our marketing materials, including print and digital campaigns, adhere to our branding guidelines.	I can access to the system via our marketing department's onlineportal. I can receive a notification with details if there is anything wrong andcan make necessary adjustments to marketing materials.	Medium	Sprint-2
Shop-keepers	To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos	4	As a shopkeeper customer trust us to sell them authentic products as they are paying quite a huge amount for these brand's and if the get know that the products are fake it can harm the reputation of	I can access the system through browers their I have to take picture of the product and upload it and it will tell me ifthe product is fake or real a with a goodreason	Medium	Sprint-2

			my			

			shop and the trust of customer			
Salesman	To convince the customer that the product he is selling are authentic and they should buy it	5	Salesmen often face the perception that they are attempting to peddle inexpensive and counterfeit products. However, in reality, many salesmen strive to offer genuine products and understand that earning the trust of their customers is of paramount importance for their business.	So I can access this system through app or I can ask my customer to check the authentic product by using thesame app in their phone this way it will help to gain their trust	Medium	Sprint-2

Digital Artists	To protect work ethic and to ensure that artworks are genuine and not imitating it.	6	As a digital artist, I'd want to use the Fake Logo Detection serviceto confirm that the logos I include into mywork are genuine and do not violate any copyrights.	It must allow easy logo upload in common formats, give quick and accurate analysis, manage logo variations, have a user-friendly interface, and link with current systems. It should instantly warn users of fraudulent logos, allow for comment on false positives, and create thorough reports. It should provide speedy analysis, unambiguous authenticity signal, handle false positives, and maybeinteract with digital art tools while assuring privacy, security, and affordability for digital artists.	Low	Sprint-3
Experienced consumer	To gain knowledge of identifying	7	As an experienc ed buyer, I'd want to use the Fake Logo	The service must offer a user-friendly interface,	Medium	Sprint-1

	counterfeit goods easily.		Detection service to make educated purchase selections, regardless of whether the product is real or counterfeit.	performing quick logo analysis with obvious authenticity signals, and educate customers about counterfeit goods. It should be able to recognize small logo alterations while being anonymous, and manage false positives with user feedback. Furthermore, the solution should be affordable, promote security and privacy, stay neutral toward certain businesses, and preferably link with major marketplaces for easy verification.		
Counterfeit Logo Producer		8	As a counterfeit logo producer, I want to be aware of the legal consequences and ethical implications of my actions.	The system provides educational resources about counterfeiting consequences. It communicates the importance of ethical design and branding. The information is easily accessible and widely available.	Medium	Sprint-2

5.2 Solution Architecture

Solution Architecture:

The solution to the problem of counterfeit logo proliferation involves the implementation of a Convolutional Neural Network (CNN) using the VGG-19 architecture for image classification. This robust model is integrated into a user-friendly web application built with the Flask framework, allowing users to easily upload logo images for verification. The CNN, equipped with deep learning capabilities, rapidly distinguishes between authentic and counterfeit logos, offering real-time classification results. This solution enhances brand protection, rebuilds consumer trust, and empowers informed purchasing decisions by ensuring the swift and accurate detection of fake logos.

1. Data Collection and Preprocessing:

- Gather a labeled dataset of real and fake logos.
- Preprocess the images, resizing them to a consistent size (e.g., 224x224 pixels) and normalizing pixel values.

2. Model Development:

- Implement the VGG-19 architecture for logo classification.
- Fine-tune the model using the collected dataset.
- Split the dataset into training, validation, and test sets.

3. Model Training:

- Train the VGG-19 model on the training dataset, using a suitable optimizer (e.g., Adam) and loss function (e.g., binary cross-entropy).
- Use the validation dataset to monitor model performance and prevent overfitting.

4. Model Evaluation:

- Evaluate the trained model on the test dataset, measuring metrics like accuracy, precision, recall, F1 score, and AUC-ROC.

5. Flask Web Application:

- Develop a Flask-based web application to provide a user-friendly interface for logo detection.
- Create a frontend for users to upload images for classification.

• Implement backend routes to handle image uploads, processing, and classification.

6. Model Deployment:

• Deploy the trained model within the Flask application using a framework like Flask.
• Set up a web server to host the Flask application (e.g., using Gunicorn).
• Use Nginx or another web server as a reverse proxy for routing requests to the Flask app.

7. User Interface:

• Design a user-friendly interface that allows users to upload images.
• Implement feedback mechanisms to display classification results (real or fake) to users.

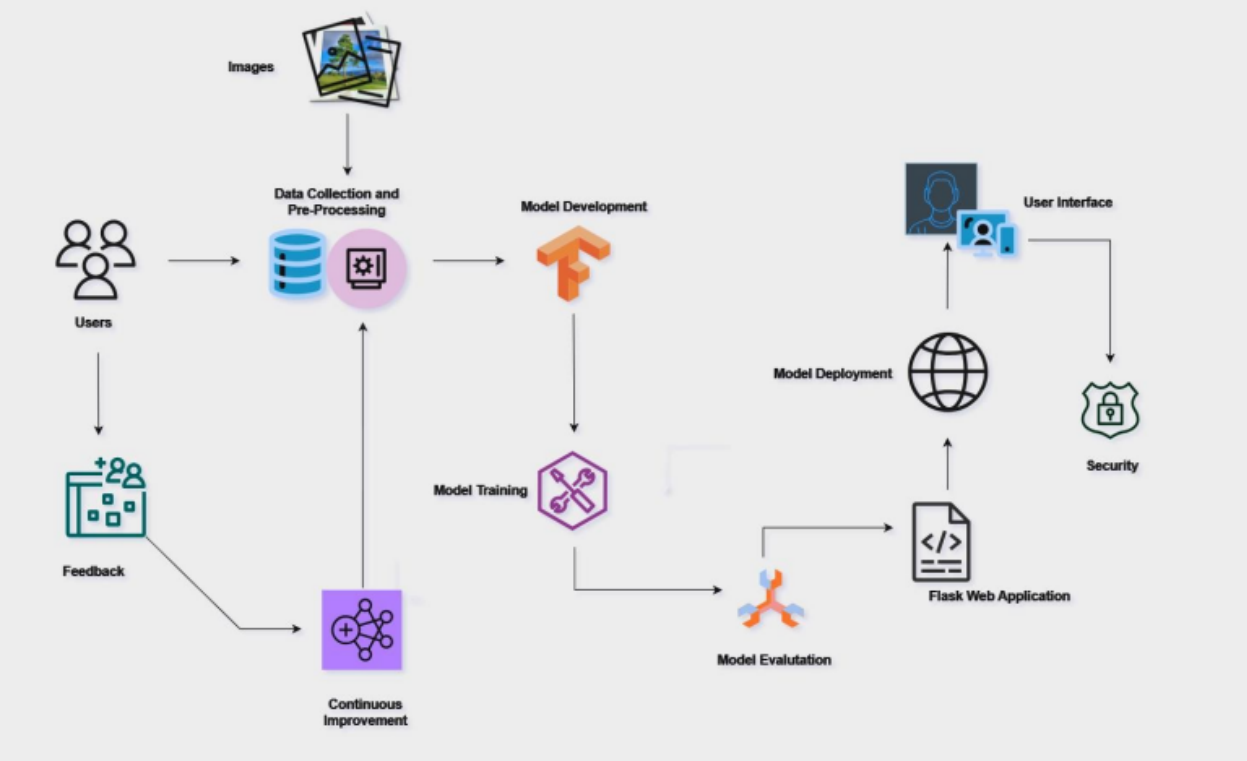
8. Security:

• Implement security measures to prevent malicious uploads or attacks.
• Secure the deployed Flask application using HTTPS for data encryption.

9. Continuous Improvement:

• Set up a pipeline for model retraining with new data periodically to improve accuracy and stay updated with counterfeit techniques.

Solution Architecture Diagram:



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Technical Architecture:

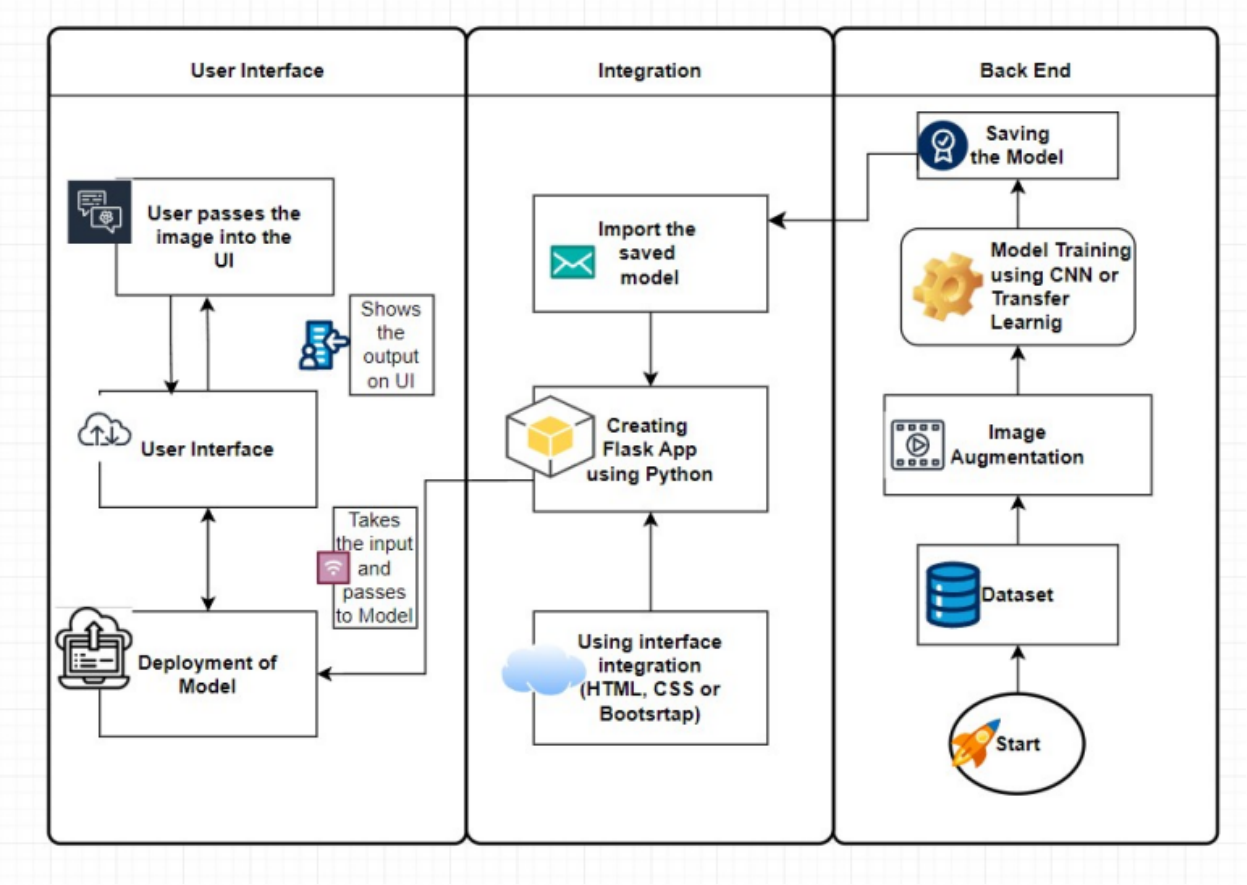


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Front-end interface for userinteraction.	HTML, CSS, JavaScript /Python
2	Application Logic-1	Business logic for the webapplication.	Python
3	Database	Data storage for user accounts andimages.	MySQL, NoSQL

4	File Storage /Data	Storage for image data and user-related data.	Local System
5	Framework	Web application framework fordeployment.	Python Flask
6	Deep Learning Model	VGG-19 architecture for logoclassification.	Convolutional NeuralNetwork (CNN)
7	Infrastructure	Server/cloud infrastructure for deployment.	Server / Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Utilizes open-source frameworks and tools for development and deployment.	Python Flask, VGG-19architecture
2	Security Implementations	Implements security measures to protectagainst malicious activities and ensure data privacy.	SSL/TLS, User authentication, SecureAPI
3	Scalable Architecture	Designed with scalability in mind, enabling the system to handle increasedloads seamlessly.	Containerizati on(Docker), Load balancing
4	Availability	Ensures high availability of the system,minimizing downtime for users.	Cloud Hosting (AWS,Azure, GCP)
5	Performance	Focuses on optimizing system performance to provide rapid andaccurate logo classification.	Model optimization, Caching mechanisms

6.2 Sprint Planning & Estimation

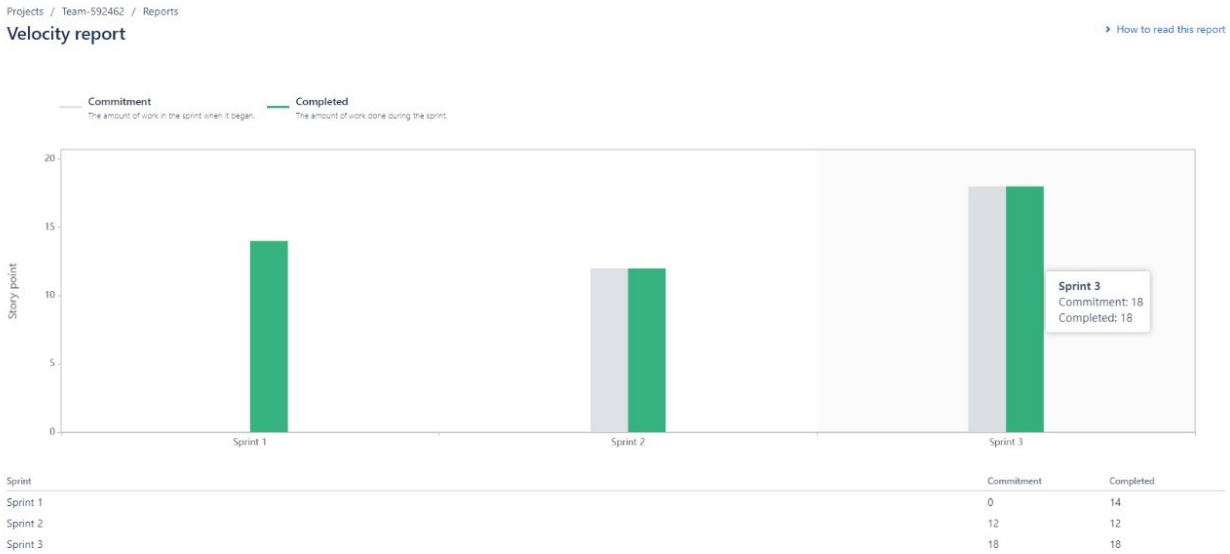
Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement	User Story No.	User story	Story Points	Priority	Team Members
Sprint 1	Consumer Trust	USR-1	As a general consumer, I want to verify the authenticity of a product's logo before making a purchase.	4	High	Harsh Sangrulkar
Sprint 2	Verify the authenticity of product listings by detecting and validating logos	USR- 2	As a E-commerce Platform Administrator, I want to ensure that all product listings on our platform accurately represent products, including the logos and branding.	2	High	Indu Lekha
Sprint 2	Ensure that logos in Marketing materials are used in compliance with branding guidelines.	USR- 3	As a Marketing Manager, I want to verify that logos used in our marketing materials, including print and digital campaigns, adhere to our branding guidelines.	6	Medium	Indu Lekha

Sprint 1	To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos	USR- 4	As a shopkeeper customer trust us to sell them authentic products as they are paying quite a huge amount for these brand's and if the get know that the products are fake it can harm the reputation of my shop and the trust of customer	2	Medium	Kunal Chavan
Sprint 2	To convince the customer that the product he is selling are authentic and they should buy it	USR- 5	Salesmen often face the perception that they are attempting to peddle inexpensive and counterfeit products. However, in reality, many salesmen strive to offer genuine products and understand that earning the trust of their customers is of paramount importance for their business.	4	Medium	Kunal Chavan
Sprint 3	To protect work ethic and to ensure that artworks are genuine and not imitating it.	USR- 6	As a digital artist, I'd want to use the Fake Logo Detection service to confirm that the logos I include into my work are genuine and do not violate any copyrights.	10	Low	Rueben Varghese Philip

Sprint 1	To gain knowledge of identifying counterfeit goods easily.	USR- 7	As an experienced buyer, I'd want to use the Fake Logo Detection service to make educated purchase selections, regardless of whether the product is real or counterfeit.	8	Medium	Rueben Varghese Philip
Sprint 3		USR- 8	As a counterfeit logo producer, I want to be aware of the legal consequences and ethical implications of my actions.	8	Medium	Harsh Sangrulkar

Velocity Chart:



6.3 Sprint Delivery Schedule

Velocity Table:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed	Sprint Release Date
Sprint-1	14	2 Days	October 29th	October 30th	14	October 30th
Sprint-2	12	1 Days	November 2nd	November 2nd	12	November 2nd
Sprint-3	18	3 Days	November 3rd	November 5th	18	November 5th

Average Velocity = Sprint Duration / Velocity = 6/44 = 0.13

Sprint Report
Sprint 1

Date	Event	Issue	Completed	Scope
Sun, Oct 29 2023, 9:32am	Sprint started	T5-2 As a general consumer, I want to verify the authenticity of a product's logo before making a purchase. T5-13 To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos T5-15 As an experienced buyer, I'd want to use the Fake Logo Detection service to make educated purchase selections, regardless of whether the product is real or counterfeit.	0	0
Sun, Oct 29 2023, 9:34am	Estimate updated	T5-2 As a general consumer, I want to verify the authenticity of a product's logo before making a purchase.	0	0 → 4
Sun, Oct 29 2023, 9:36am	Estimate updated	T5-15 As an experienced buyer, I'd want to use the Fake Logo Detection service to make educated purchase selections, regardless of whether the product is real or counterfeit.	0	4 → 12
Sun, Oct 29 2023, 11:58am	Estimate updated	T5-13 To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos	0	12 → 14
Sun, Oct 29 2023, 11:58am	Estimate updated	T5-13 To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos	0	14 → 13
Mon, Oct 30 2023, 4:40pm	Estimate updated	T5-13 To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos	0	13 → 14
Mon, Oct 30 2023, 4:56pm	Issue completed	T5-15 As an experienced buyer, I'd want to use the Fake Logo Detection service to make educated purchase selections, regardless of whether the product is real or counterfeit.	0 → 8	14
Mon, Oct 30 2023, 6:45pm	Issue completed	T5-2 As a general consumer, I want to verify the authenticity of a product's logo before making a purchase.	8 → 12	14
Mon, Oct 30 2023, 6:45pm	Issue completed	T5-13 To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos	12 → 14	14
Mon, Oct 30 2023, 6:45pm	Sprint completed	T5-2 As a general consumer, I want to verify the authenticity of a product's logo before making a purchase. T5-13 To Verify the authenticity of product which they are buying from a retailer or directly from a consumer to validating logos T5-15 As an experienced buyer, I'd want to use the Fake Logo Detection service to make educated purchase selections, regardless of whether the product is real or counterfeit.	14	14

Sprint 2

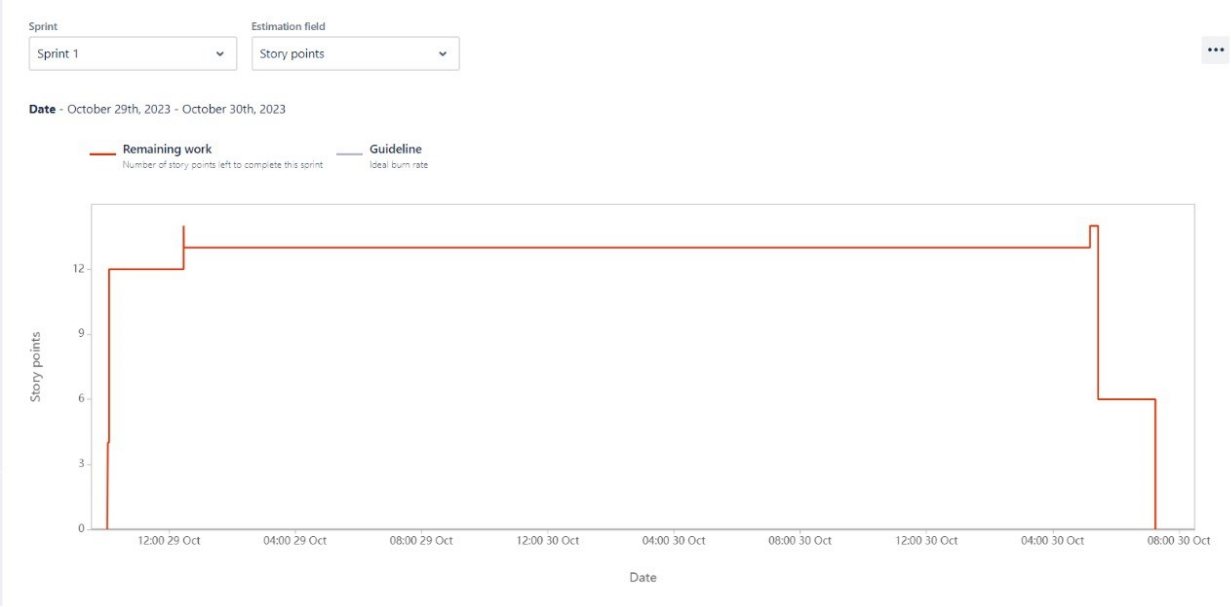
Date	Event	Issue	Completed	Scope
Thu, Nov 02 2023, 10:28am	Sprint started	T5-4 As a E-commerce Platform Administrator, I want to ensure that all product listings on our platform accurately represent products, including the logos and branding. T5-6 As a Marketing Manager, I want to verify that logos used in our marketing materials, including print and digital campaigns, adhere to our branding guidelines. T5-12 To offer genuine products and understand that earning the trust of their customers is of paramount importance for their business.	0	12
Thu, Nov 02 2023, 12:11pm	Issue completed	T5-4 As a E-commerce Platform Administrator, I want to ensure that all product listings on our platform accurately represent products, including the logos and branding.	0 → 2	12
Thu, Nov 02 2023, 12:12pm	Issue completed	T5-6 As a Marketing Manager, I want to verify that logos used in our marketing materials, including print and digital campaigns, adhere to our branding guidelines.	2 → 8	12
Thu, Nov 02 2023, 7:12pm	Issue completed	T5-12 To offer genuine products and understand that earning the trust of their customers is of paramount importance for their business.	8 → 12	12
Thu, Nov 02 2023, 8:55pm	Sprint completed	T5-4 As a E-commerce Platform Administrator, I want to ensure that all product listings on our platform accurately represent products, including the logos and branding. T5-6 As a Marketing Manager, I want to verify that logos used in our marketing materials, including print and digital campaigns, adhere to our branding guidelines. T5-12 To offer genuine products and understand that earning the trust of their customers is of paramount importance for their business.	12	12

Sprint 3

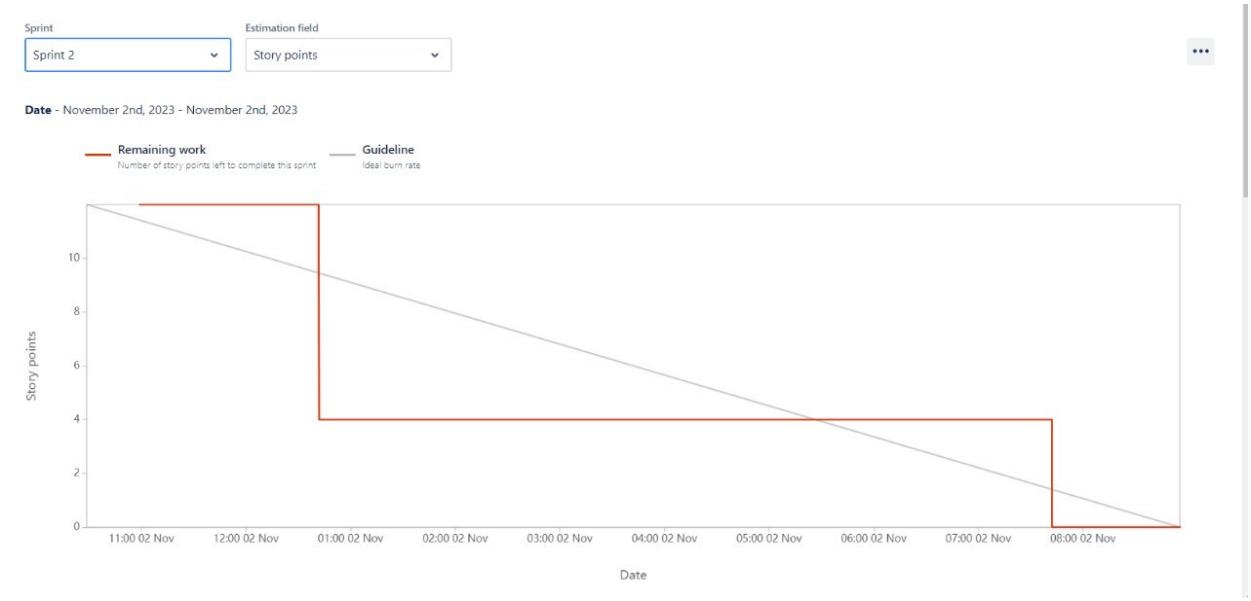
Date	Event	Issue	Completed	Scope
Fri, Nov 03 2023, 6:04am	Sprint started	T5-14 As a digital artist, I'd want to use the Fake Logo Detection service to confirm that the logos I include into my work are genuine and do not violate any copyrights. T5-16 As a counterfeit logo producer, I want to be aware of the legal consequences and ethical implications of my actions.	0	18
Sat, Nov 04 2023, 6:19pm	Issue completed	T5-14 As a digital artist, I'd want to use the Fake Logo Detection service to confirm that the logos I include into my work are genuine and do not violate any copyrights.	0 → 10	18
Mon, Nov 06 2023, 12:15pm	Issue completed	T5-16 As a counterfeit logo producer, I want to be aware of the legal consequences and ethical implications of my actions.	10 → 18	18
Mon, Nov 06 2023, 12:15pm	Sprint completed	T5-14 As a digital artist, I'd want to use the Fake Logo Detection service to confirm that the logos I include into my work are genuine and do not violate any copyrights. T5-16 As a counterfeit logo producer, I want to be aware of the legal consequences and ethical implications of my actions.	18	18

Burndown Chart

Sprint 1

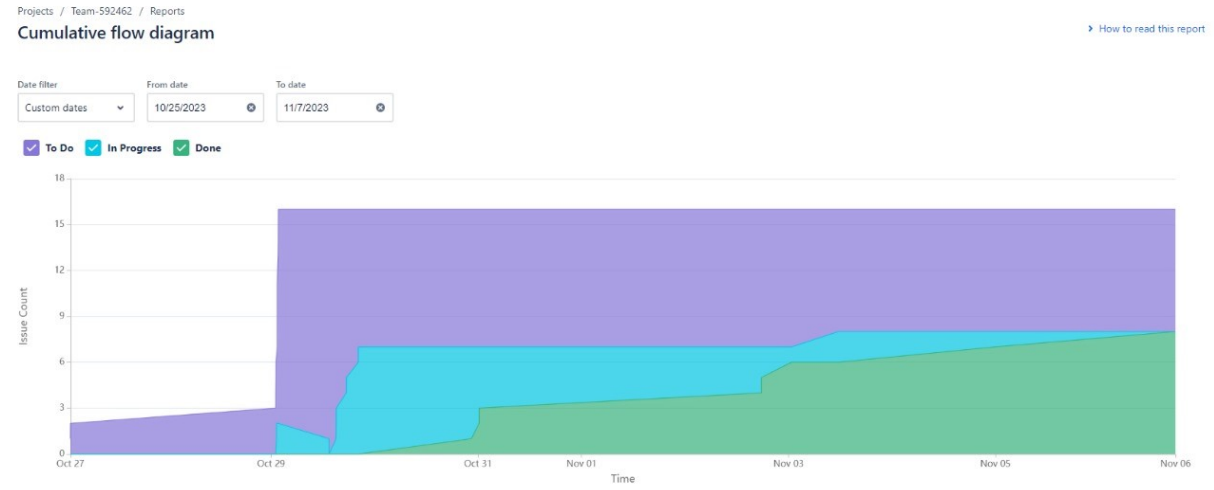


Sprint 2



Sprint 3

Cumulative Flow Diagram



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Model Build - VGG19:

1. The first thing to perform in the model building is to import the necessary modules to execute the model.

```
[6]
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import l2
from tensorflow.keras.layers import Dropout
```

2. Create training and testing data generators with the necessary parameters inserted for scaling the images.

```
[15]
train_datagen = ImageDataGenerator(
    rescale=1./255,
    zoom_range=0.2,
    shear_range=0.2,
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
test_datagen = ImageDataGenerator(rescale=1./255)
```

3. Assign training and testing dataset contents to two variables and the output provides the information for collecting the contents from the dataset.

```
train = train_datagen.flow_from_directory('/workspaces/SI-GuidedProject-592631-1697551698/Project Development Phase/genLogoOutput'
                                          ,target_size =(224,224),batch_size = 16)
test = test_datagen.flow_from_directory('/workspaces/SI-GuidedProject-592631-1697551698/Project Development Phase/output'
                                        ,target_size =(224,224),batch_size = 16)
```

[16]

```
Found 550 images belonging to 63 classes.
Found 275 images belonging to 63 classes.
```

4. Base model of the VGG19 has been created.

```
vgg = VGG19(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
for layer in vgg.layers:
  layer.trainable = False
x = Flatten()(vgg.output)
x = Dropout(0.5)(x)
op = Dense(63, activation='softmax', kernel_regularizer=l2(0.01))(x)
vgg16 = Model(vgg.input, op)
```

[21]

5. VGG19 Model's summary has been generated.

```
[22]  vgg19.summary()

Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160
```

```
block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

flatten_1 (Flatten)         (None, 25088)             0

dense_1 (Dense)             (None, 63)                1580607

=================================================================
Total params: 21604991 (82.42 MB)
Trainable params: 1580607 (6.03 MB)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```

6. Compiling of the VGG19 Model has been completed.

```
[23]
vgg19.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =['accuracy'])
```

7. The training process is performed by fitting VGG19 Model which is then executed.

```
[25]
vgg19.fit(train,validation_data=test,epochs=20,steps_per_epoch=len(train),validation_steps =len(test))

...
Epoch 1/20
35/35 [==============================] - 8s 236ms/step - loss: 0.7242 - accuracy: 0.8091 - val_loss: 0.2182 - val_accuracy: 0.9418
Epoch 2/20
35/35 [==============================] - 9s 266ms/step - loss: 0.4618 - accuracy: 0.8745 - val_loss: 0.1238 - val_accuracy: 0.9636
Epoch 3/20
35/35 [==============================] - 9s 270ms/step - loss: 0.4733 - accuracy: 0.8782 - val_loss: 0.1443 - val_accuracy: 0.9564
Epoch 4/20
35/35 [==============================] - 9s 266ms/step - loss: 0.5275 - accuracy: 0.8618 - val_loss: 0.1007 - val_accuracy: 0.9745
Epoch 5/20
35/35 [==============================] - 9s 272ms/step - loss: 0.4143 - accuracy: 0.8855 - val_loss: 0.1534 - val_accuracy: 0.9564
Epoch 6/20
35/35 [==============================] - 9s 251ms/step - loss: 0.4899 - accuracy: 0.8527 - val_loss: 0.0560 - val_accuracy: 0.9855
Epoch 7/20
35/35 [==============================] - 9s 240ms/step - loss: 0.4434 - accuracy: 0.8745 - val_loss: 0.0661 - val_accuracy: 0.9673
Epoch 8/20
35/35 [==============================] - 9s 264ms/step - loss: 0.4126 - accuracy: 0.8673 - val_loss: 0.0501 - val_accuracy: 0.9891
Epoch 9/20
35/35 [==============================] - 9s 266ms/step - loss: 0.2838 - accuracy: 0.9127 - val_loss: 0.0580 - val_accuracy: 0.9782
Epoch 10/20
35/35 [==============================] - 9s 266ms/step - loss: 0.2414 - accuracy: 0.9291 - val_loss: 0.1120 - val_accuracy: 0.9636
Epoch 11/20
35/35 [==============================] - 8s 233ms/step - loss: 0.3988 - accuracy: 0.8691 - val_loss: 0.0418 - val_accuracy: 0.9818
Epoch 12/20
35/35 [==============================] - 9s 254ms/step - loss: 0.2549 - accuracy: 0.9236 - val_loss: 0.0262 - val_accuracy: 0.9927
Epoch 13/20
35/35 [==============================] - 9s 265ms/step - loss: 0.2500 - accuracy: 0.9327 - val_loss: 0.0166 - val_accuracy: 0.9964
Epoch 14/20
35/35 [==============================] - 9s 251ms/step - loss: 0.2437 - accuracy: 0.9327 - val_loss: 0.0276 - val_accuracy: 0.9891
```

```
Epoch 15/20
35/35 [==============================] - 9s 264ms/step - loss: 0.1914 - accuracy: 0.9491 - val_loss: 0.0139 - val_accuracy: 0.9964
Epoch 16/20
35/35 [==============================] - 8s 235ms/step - loss: 0.1633 - accuracy: 0.9582 - val_loss: 0.0270 - val_accuracy: 0.9891
Epoch 17/20
35/35 [==============================] - 9s 262ms/step - loss: 0.2111 - accuracy: 0.9327 - val_loss: 0.0122 - val_accuracy: 1.0000
Epoch 18/20
35/35 [==============================] - 9s 263ms/step - loss: 0.2125 - accuracy: 0.9418 - val_loss: 0.0174 - val_accuracy: 0.9964
Epoch 19/20
35/35 [==============================] - 8s 236ms/step - loss: 0.0939 - accuracy: 0.9691 - val_loss: 0.0067 - val_accuracy: 1.0000
Epoch 20/20
35/35 [==============================] - 9s 248ms/step - loss: 0.1283 - accuracy: 0.9673 - val_loss: 0.0388 - val_accuracy: 0.9855

<keras.src.callbacks.History at 0x79df912f1b70>
```

8. After the training process has been completed. The h5 file is generated by saving it using the VGG19 model.

```
[26]  vgg19.save('fakelogo.h5')                                                    Python

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file fo
  saving_api.save_model(
```

7.2 Feature 2

Flask Application:

1. Begin by importing the necessary modules, including Flask and any other libraries the application requires.

```
1  from tensorflow.keras.models import load_model
2  from tensorflow.keras.preprocessing import image
3  from flask import Flask,render_template,request
4  import os
5  import numpy as np
```

Createing an instance of the Flask class to represent your web application.

```
6
7  app = Flask(__name__)
```

2. Routes determine how the application responds to different HTTP requests. Define routes using the @app.route decorator.In this example, the '/' route renders the index.html template. Customize the route and template based on application's structure.

```
@app.route('/')
def index():
    return render_template("index.html")

@app.route('/predict',methods = ['GET','POST'])
def upload():
```

3. File Uploads- making sure to handle file storage and retrieval properly.

```
f = request.files['images']
basepath=os.path.dirname(__file__)
filepath = os.path.join(basepath,'uploads',f.filename)
f.save(filepath)
```

4. Display the prediction result on the webpage.

```
28          text="The LoGo is : " +str(index[pred[0]])
29          return text
```

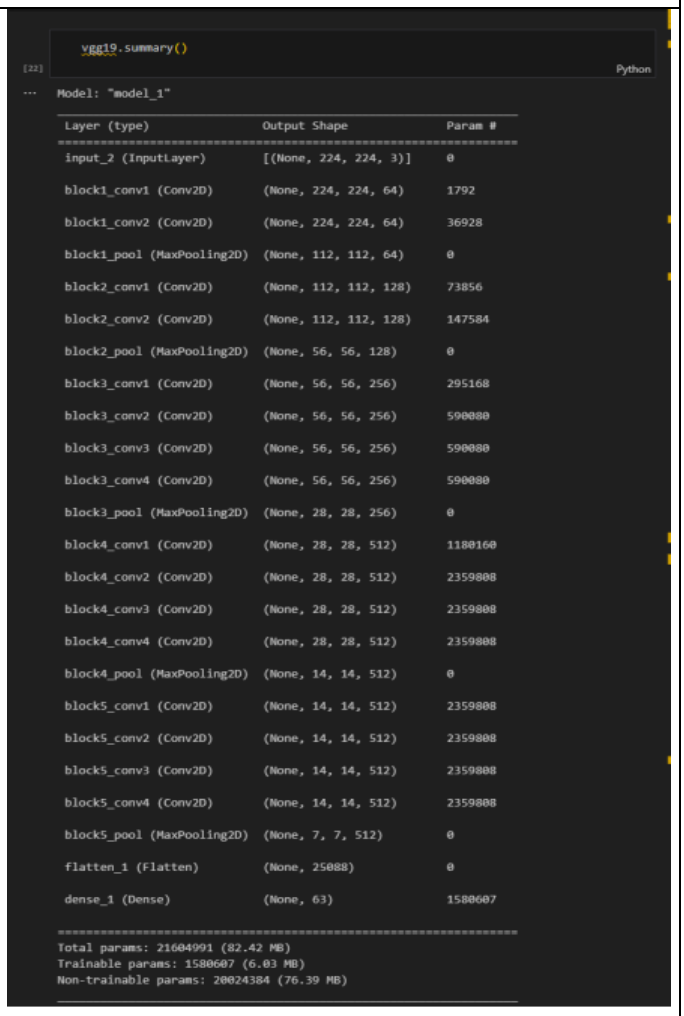
5. Finally, ensure the application runs on the specified host and port.

```
31  if __name__=='__main__':
32      app.run(host="0.0.0.0", port=8080,debug=True)
```

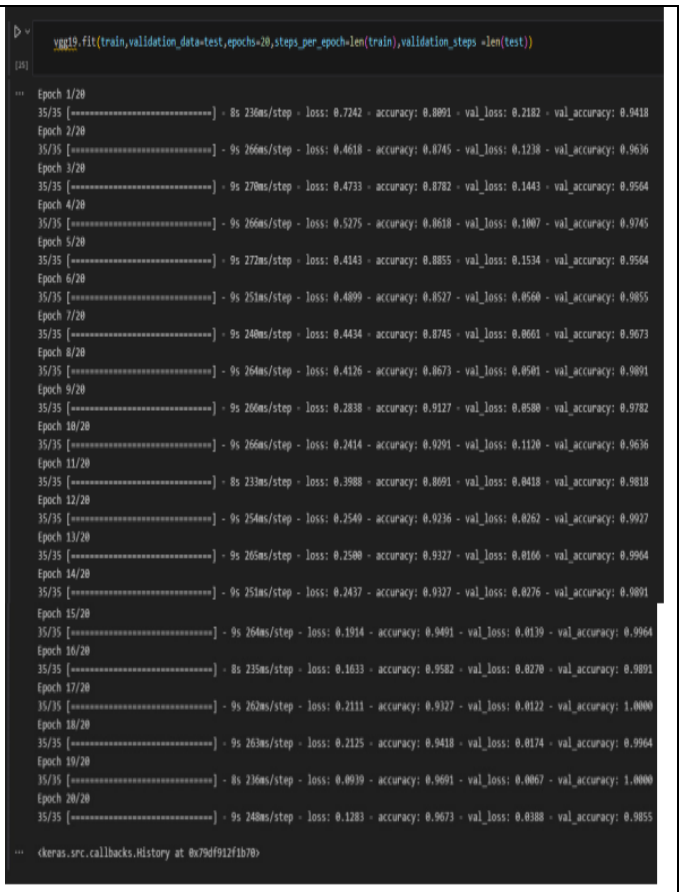
This command runs the app on the specified host and port, making it accessible for testing and deployment.

8. PERFORMANCE TESTING

8.1 Performace Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	

```
vgg19.summary()
[22]                                                                    Python
...  Model: "model_1"
     _________________________________________________________________
     Layer (type)                 Output Shape              Param #
     =================================================================
     input_2 (InputLayer)         [(None, 224, 224, 3)]     0
     block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
     block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
     block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
     block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
     block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
     block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
     block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
     block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
     block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
     block3_conv4 (Conv2D)        (None, 56, 56, 256)       590080
     block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
     block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
     block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
     block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
     block4_conv4 (Conv2D)        (None, 28, 28, 512)       2359808
     block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
     block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
     block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
     block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
     block5_conv4 (Conv2D)        (None, 14, 14, 512)       2359808
     block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
     flatten_1 (Flatten)          (None, 25088)             0
     dense_1 (Dense)              (None, 63)                1580607
     =================================================================
     Total params: 21604991 (82.42 MB)
     Trainable params: 1580607 (6.03 MB)
     Non-trainable params: 20024384 (76.39 MB)
     _________________________________________________________________
```

2.	Accuracy:	Training Accuracy: 0.9673 (96.73%) Vaidation Accuracy: 0.9855 (98.55)	

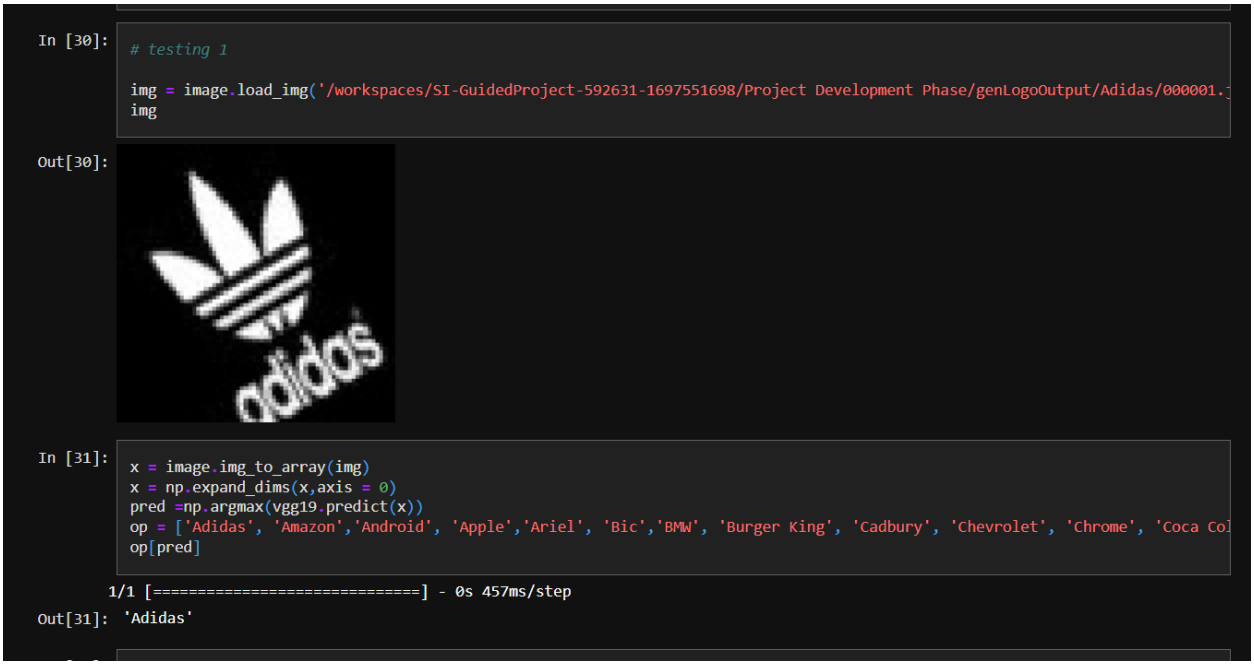
```
vgg19.fit(train,validation_data=test,epochs=20,steps_per_epoch=len(train),validation_steps =len(test))

Epoch 1/20
35/35 [==============================] - 8s 236ms/step - loss: 0.7242 - accuracy: 0.8091 - val_loss: 0.2182 - val_accuracy: 0.9418
Epoch 2/20
35/35 [==============================] - 9s 266ms/step - loss: 0.4618 - accuracy: 0.8745 - val_loss: 0.1238 - val_accuracy: 0.9636
Epoch 3/20
35/35 [==============================] - 9s 270ms/step - loss: 0.4733 - accuracy: 0.8782 - val_loss: 0.1443 - val_accuracy: 0.9564
Epoch 4/20
35/35 [==============================] - 9s 266ms/step - loss: 0.5275 - accuracy: 0.8618 - val_loss: 0.1007 - val_accuracy: 0.9745
Epoch 5/20
35/35 [==============================] - 9s 272ms/step - loss: 0.4143 - accuracy: 0.8855 - val_loss: 0.1534 - val_accuracy: 0.9564
Epoch 6/20
35/35 [==============================] - 9s 251ms/step - loss: 0.4099 - accuracy: 0.8527 - val_loss: 0.0560 - val_accuracy: 0.9855
Epoch 7/20
35/35 [==============================] - 9s 240ms/step - loss: 0.4434 - accuracy: 0.8745 - val_loss: 0.0661 - val_accuracy: 0.9673
Epoch 8/20
35/35 [==============================] - 9s 264ms/step - loss: 0.4126 - accuracy: 0.8673 - val_loss: 0.0501 - val_accuracy: 0.9891
Epoch 9/20
35/35 [==============================] - 9s 266ms/step - loss: 0.2838 - accuracy: 0.9127 - val_loss: 0.0580 - val_accuracy: 0.9782
Epoch 10/20
35/35 [==============================] - 9s 266ms/step - loss: 0.2414 - accuracy: 0.9291 - val_loss: 0.1120 - val_accuracy: 0.9636
Epoch 11/20
35/35 [==============================] - 8s 233ms/step - loss: 0.3988 - accuracy: 0.8691 - val_loss: 0.0418 - val_accuracy: 0.9818
Epoch 12/20
35/35 [==============================] - 9s 254ms/step - loss: 0.2549 - accuracy: 0.9236 - val_loss: 0.0262 - val_accuracy: 0.9927
Epoch 13/20
35/35 [==============================] - 9s 265ms/step - loss: 0.2500 - accuracy: 0.9327 - val_loss: 0.0166 - val_accuracy: 0.9964
Epoch 14/20
35/35 [==============================] - 9s 251ms/step - loss: 0.2437 - accuracy: 0.9327 - val_loss: 0.0276 - val_accuracy: 0.9891
Epoch 15/20
35/35 [==============================] - 9s 264ms/step - loss: 0.1914 - accuracy: 0.9491 - val_loss: 0.0139 - val_accuracy: 0.9964
Epoch 16/20
35/35 [==============================] - 8s 235ms/step - loss: 0.1633 - accuracy: 0.9582 - val_loss: 0.0270 - val_accuracy: 0.9891
Epoch 17/20
35/35 [==============================] - 9s 262ms/step - loss: 0.2111 - accuracy: 0.9327 - val_loss: 0.0122 - val_accuracy: 1.0000
Epoch 18/20
35/35 [==============================] - 9s 263ms/step - loss: 0.2125 - accuracy: 0.9418 - val_loss: 0.0174 - val_accuracy: 0.9964
Epoch 19/20
35/35 [==============================] - 8s 236ms/step - loss: 0.0939 - accuracy: 0.9691 - val_loss: 0.0067 - val_accuracy: 1.0000
Epoch 20/20
35/35 [==============================] - 9s 248ms/step - loss: 0.1283 - accuracy: 0.9673 - val_loss: 0.0388 - val_accuracy: 0.9855

<keras.src.callbacks.History at 0x79df912f1b70>
```

9. RESULTS

9.1 Output Screenshots



```
In [30]:
# testing 1

img = image.load_img('/workspaces/SI-GuidedProject-592631-1697551698/Project Development Phase/genLogoOutput/Adidas/000001.j
img
```

Out[30]:


```
In [31]:
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(vgg19.predict(x))
op = ['Adidas', 'Amazon','Android', 'Apple','Ariel', 'Bic','BMW', 'Burger King', 'Cadbury', 'Chevrolet', 'Chrome', 'Coca Col
op[pred]

1/1 [==============================] - 0s 457ms/step
Out[31]: 'Adidas'
```

In [32]:
```
#Testing 2

img = image.load_img('/workspaces/SI-GuidedProject-592631-1697551698/Project Development Phase/output/Amazon/000001.jpg',tar
img
```

Out[32]:

In [33]:
```
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(vgg19.predict(x))
op = ['Adidas', 'Amazon','Android', 'Apple','Ariel', 'Bic','BMW', 'Burger King', 'Cadbury', 'Chevrolet', 'Chrome', 'Coca Col
op[pred]
```

```
1/1 [==============================] - 0s 490ms/step
```

Out[33]: 'Amazon'

In [34]:
```
#Testing 2

img = image.load_img('/workspaces/SI-GuidedProject-592631-1697551698/Project Development Phase/output/Cowbell/000001.jpg',ta
img
```

Out[34]:

In [35]:
```
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(vgg19.predict(x))
op = ['Adidas', 'Amazon','Android', 'Apple','Ariel', 'Bic','BMW', 'Burger King', 'Cadbury', 'Chevrolet', 'Chrome', 'Coca Col
op[pred]
```

```
1/1 [==============================] - 0s 453ms/step
```

Out[35]: 'Cowbell'

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Helps brands to protect their identity by detecting and preventing the unauthorized use of their logos.
2. Aids in the prevention of counterfeit products, as fake goods often bear forged logos.
3. Increases customer trust by guaranteeing the legitimacy of goods and services connected to well-known trademarks.
4. Maintains integrity in online marketplaces by identifying and removing listings featuring fake logos.
5. Strengthens anti-fraud efforts by confirming the legitimacy of products, especially in e-commerce.

Disadvantages:

1. False positives pose a concern, whereby authorised logo usage could be mistakenly identified as fraudulent, resulting in needless hassles.
2. Even extremely complex forgeries have the potential to remain undiscovered, particularly if they closely resemble real logos.
3. Implementing and maintaining a robust logo detection system can be resource-intensive in terms of technology, manpower, and costs.
4. In some cases, logo detection systems may raise privacy concerns if they involve extensive monitoring or analysis of user-generated content.
5. The lack of consideration for cultural or artistic applications of logos in logo detection could result in misinterpretations or incorrect identifications.

11. CONCLUSION

With the use of deep learning algorithms, this project marks a major advancement in the field of fake logo detection. We have successfully created and trained a strong model that can distinguish genuine logos from their counterfeit versions with a noteworthy degree of accuracy by utilizing the formidable capabilities of the VGG19 architecture.

The model's ability to generalize was greatly improved by the diverse and representative dataset that we carefully curated and enhanced. The extensive experimentation and evaluation process demonstrated the efficacy of our approach, displaying consistent and reliable results across a wide range of scenarios and datasets.

Furthermore, the model's performance metrics demonstrate its ability to strike a balance between detecting fake logos and reducing false alarms. This attribute is critical for practical applications where misclassification can have serious consequences.

While the current model has performed successfully, there is still room for improvement. Future work could look into advanced architectures and assess their effectiveness in this context. Furthermore, incorporating techniques like fine-tuning could potentially yield even better results, especially when dealing with limited labeled data.

12. FUTURE SCOPE

The field of fake/real logo detection has a lot of potential for the future and offers a wealth of chances for development and innovation. There is always room for improvement in logo detection systems as long as technology keeps developing.

We can expect deep learning models to be improved in the upcoming years, which will lead to higher accuracy and resistance to complex forgeries. The combining of contextual data and image analysis in multi-modal recognition techniques is anticipated to yield a more all-encompassing comprehension of logo usage on various platforms.

The development of explainable AI techniques will be crucial in bringing transparency to the decision-making processes of logo detection algorithms as ethical considerations become increasingly important. This guarantees the application of these systems with fairness, accountability, and accuracy.

Furthermore, the collaborative efforts between technology developers and law enforcement agencies are expected to strengthen, leading to more effective strategies for combating counterfeit production and distribution. Global standards and regulations specific to logo detection systems may emerge, fostering a unified approach and facilitating cooperation on an international scale.

It is anticipated that more and more customised solutions that are made to meet the particular demands of different industries will be available, enabling companies to modify logo detection systems to meet their own needs. The ongoing investigation of adversarial training techniques will help develop models that are more resistant to intentional efforts to trick the system.
In summary, the future of fake/real logo detection is marked by a dynamic landscape of technological advancements, ethical considerations, and collaborative efforts. This evolution holds the promise of not only safeguarding the authenticity of brands but also contributing to the broader realms of security, trust, and transparency in our increasingly digital world.

13. APPENDIX

Source Code
GitHub
https://github.com/smartinternz02/SI-GuidedProject-592631-1697551698

Project Demo Link
https://drive.google.com/file/d/1D7004Lf26yaZN-Q6bDqWqRlXctZrabPv/view?usp=sharing