

Hemeswar Raj Peddireddy

Image Correctness for a Product on Marketplace

-  Quick Submit
 -  Quick Submit
 -  Presidency University
-

Document Details

Submission ID

trn:oid:::1:3408933979

8 Pages

Submission Date

Nov 13, 2025, 10:20 AM GMT+5:30

3,427 Words

Download Date

Nov 13, 2025, 10:22 AM GMT+5:30

20,333 Characters

File Name

Research_Paper_Image_Correctness_of_Product_on_Market_Place.pdf

File Size

761.4 KB

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



Image Correctness for a Product on Marketplace

1st Hemeswar Raj PeddiReddy
Computer Science & Engineering
Presidency University
Karnataka,India
denu990809@gmail.com

2nd Manasa Thanakanti
Computer Science & Engineering
Presidency University
Karnataka,India
manasareddy2903@gmail.com

3rd Deekshitha B
Computer Science & Engineering
Presidency University
Karnataka,India
deekshithareddyb2004@gmail.com

Ms. . Shivabasamma Beli
Department of Computer Science
Assistant Professor
Presidency University
Karnataka, India.

Abstract: In the present digital world, consumers access various products through online marketplaces. Nowadays, they are an integral part of retail. Photos of products have to be clear and of high quality to ascertain customers' satisfaction and trust. Increasingly, customers give more importance to these images as they make quite judicious decisions regarding what to purchase. The Image Correctness Application in Product Marketplace is an innovative solution to the inaccurate or misleading presentation of product photos. This program utilizes modern technologies of computer vision and machine learning in order to check and enhance the quality of the images vendors submit.

A. Keywords: convolutional neural networks, e-commerce, image quality assessment, automated compliance, YOLO, image verification, and metadata matching.

I. INTRODUCTION

This is attributed to the fact that the rapid growth of e-commerce platforms has drastically altered how retail operates worldwide. They make product discovery very smooth and enable large-scale business growth with much ease. In addition, digital marketplaces, such as Amazon, Alibaba, Flipkart, and eBay, among other online shops, change daily, with how a product is presented now the biggest factor in how many people will end up trusting or purchasing it. Online shoppers, on their side, depend largely on visual content, especially product images, to judge whether an item is good enough to be bought or not. In traditional physical stores, the customers have an opportunity to touch and feel the items. It is confirmed in research that product images influence more than 75% of online purchases, an indication of how important the accuracy and presentation of images are in an e-commerce setting.

Product image correctness means the images have to be real, clear, and up to standards of both the marketplace and actual product specifications. Inaccurate or misleading product images set customers' expectations falsely. The colors might be incorrect, the features not clear, resolution low, or the photography may be misleading. These differences leave customers disgruntled. Returns increase, along with bad reviews and erosion of trust in the platform due to this. E-commerce platforms have strict policies regarding image quality and compliance to drive down these risks; however,

the verification process is still manual, slow, and error-prone, considering the high volume of product listings and dynamic nature of the marketplace.

Recent advances in machine learning and computer vision come with the promise of automating image verification tasks. Techniques such as CNNs, object detection architectures (including YOLO and Faster R-CNN), and feature extraction models can be utilized to automatically analyze product features, parameters of image quality, and metadata consistency. These methods can find product images that are below the standards of a platform, low in quality, or misleading and help apply these standards at scale. Although everything is constantly in flux, not much research has been done in this regard on automated image correctness validation specifically for e-commerce. This is a big avenue of opportunity.

The paper proposes an automated system for ensuring the correctness of product images at e-commerce marketplaces by fusing traditional image processing with deep learning-driven classification and feature-matching methods. It aims to determine whether an image is suitable according to quality metrics, consistency of the product with the image, and adherence to the marketplace rules. This study will aim at developing consumer trust, reducing the number of product returns, assisting sellers in enhancing listing quality, and making online shopping more reliable and open.

II. LITERATURE REVIEW

Moch Rizky Khairul Rachman & Sonja Andarini [1] Prior work notes Indonesia's booming skincare sector and rising male adoption, with purchase drivers clustered around quality, price, and brand image. Country/culture cues (e.g., halal identity) and global reputations shape trust and perceived value. This study positions Kahf vs. Garnier to probe those levers in an urban Indonesian context. *Tumaini Steven, Gwahula Raphael & Macha Salvio* [2] Literature frames bananas as a staple cash crop with complex channels, information gaps, and middlemen dependence undermining farmers' outcomes. Marketing-mix levers (product, price, promotion, place) are proposed to correct frictions and expand market opportunities. Large cross-sectional evidence from Kilimanjaro grounds these themes *Stevie & Rodhiah* [3]

Research on high-involvement goods is thinner than FMCG studies; the paper anchors in TPB to link attitudes, norms (e-WOM), and perceived control (brand image). Prior findings on quality/price are mixed in automotive contexts, while brand image often dominates choice. Results here reinforce brand image as the key purchase driver. *Ratna Roostika, Muafi & Agnès Retno Permata* [4] Most studies treat product-country image and tourism destination image separately; this paper integrates them at a territorial scale. Drawing on COO/TOO “halo effects,” it argues place image shapes beliefs about local products and visit intentions. Yogyakarta batik offers a test bed for the combined pathway. *Aimin Zhou, Xinle Wang, Yujin Huang, Weitang Wang, Shutao Zhang & Jinyan Ouyang* [5] The AIGC surge (e.g., GANs, style transfer) motivates controllable, aesthetic product imagery generation. Literature contrasts text-prompted variability with the predictability of image style transfer; evolutionary design uses aesthetic-fitness functions to steer forms. The paper fuses GA-based form optimization with GAN-based style transfer for design outputs. *Tumaini Steven, Gwahula Raphael & Salvio Macha* [6] Similar to the preprint, literature emphasizes market-access and price-information deficits (99.3% report issues) and middleman reliance (62.8%). The 4Ps framework is applied to strengthen competitiveness. Survey insights from Kilimanjaro highlight actionable marketing-mix gaps.

1) 2.1 Role of Product Images in E-Commerce

Visual appeal directly affects consumer perception and conversion rates.

- **Liu et al. (2018)** showed that clear, accurate images significantly improve purchase intent.
- **Statista (2020)** reported that 20–30% of returns stem from mismatched or misleading images.
- **Chen & Xie (2008)** highlighted that multi-view images increase product understanding and buyer satisfaction.

2) 2.2 Image Quality Assessment (IQA)

IQA evaluates technical image aspects like:

- **Resolution:** Minimum 800×800 px standard.
- **Sharpness:** Measured via Laplacian variance.
- **Brightness and Contrast:** Controlled through histogram normalization.
- **Orientation and Background:** Must follow white/plain background guidelines.

Two key IQA methods exist:

- **Full-Reference IQA** (Wang et al., 2004): Requires a reference image.
- **No-Reference IQA:** Evaluates quality using statistical or machine learning models.

3) 2.3 Automated Image Verification

Manual validation is impractical. Automated systems leverage:

- **Object Detection (YOLO, Faster R-CNN):** Detects product presence, parts, and mismatches.
- **Metadata Matching:** Ensures visual attributes (color, size, shape) match product metadata.
- **Quality and Compliance Checking:** Verifies background, watermark, and format guidelines.

B. 4) 2.4 Deep Learning for Image Correctness

- CNNs: extract hierarchical features such as edges, textures, and shapes;
- YOLO: You Only Look Once enables real-time object detection at high speeds.
- Faster R-CNN: Provides high precision in feature-level verification.

Deep learning permits scalable verification independent of categories on various product images.

5) 2.5 Research Gaps

Existing systems usually target either technical quality or visual correctness, often not both. A few of them provide feedback mechanisms for the sellers. There is, therefore, a need for an integrated framework that ties preprocessing, deep feature extraction, metadata validation, and compliance reporting together.

III. METHODOLOGY

This section describes the methodology adopted for the verification of product-image correctness using a multimodal deep learning model. The system works by taking an image and its corresponding description, extracting visual and textual features, calculating semantic similarity, and deciding whether the uploaded image corresponds to the given description.

A. Workflow of the system:

The following sequential steps are followed in the system workflow:

Using a web interface, a user provides a product image and types in a product description.

File Validation: The format of the uploaded file is validated to ensure it is an acceptable image type such as JPEG, JPG, or PNG.

Resizing, normalising, converting the image to RGB format, and changing it to tensors are included in pre-processing. Text description tokenization and embedding is done using a transformer-based tokeniser.

Feature Extraction: Visual and textual embeddings are created using the CLIP ViT-B/32 model, which provides high-dimensional feature representations for both the image and the text.

Similarity Computation: In order to measure semantic alignment, cosine similarity is computed between normalised image and text embeddings.

Making a Decision: The threshold value for categorizing the results is taken as 0.30. An image is marked correct if its similarity is greater than the threshold value; otherwise, it's incorrect. **Result Display:** The output is presented as a graphical web-based interface that shows if the image matches the description that was entered.

C. Components of the System :

The architecture is made up of:

Frontend: a user interface for input and result display based on HTML and CSS.

```
titleImage description checker title
    |
    |body
    |  font-family: 'Open Sans', Tahoma, Geneva, Verdana, sans-serif;
    |  width: 100px;
    |  height: 200px;
    |  display: flex;
    |  justify-content: center;
    |  align-items: center;
    |  background-color: #f0f0f0;
    |  background-size: 100% 100%; /* repeat-x, repeat-y, no-repeat, repeat, space-around */
    |  background-position: 50% 50%; /* infinite alternate none in set; */
    |
    |div {
    |  border-radius: 10px;
    |  border: 1px solid black;
    |  padding: 10px;
    |  width: fit-content;
    |  margin: auto;
    |  text-align: center;
    |  color: black;
    |  background-color: white;
    |  box-shadow: 0 0 10px 0 #ccc;
    |  font-size: 14px;
    |  font-weight: bold;
    |  letter-spacing: 0.1em;
    |  font-family: inherit;
    |  font-style: italic;
    |  font-variant: small-caps;
    |  font-size: 1.2em;
    |  font-weight: bold;
    |  color: #0070C0;
    |  margin-bottom: 10px;
    |}
    |
    |h2 {
    |  font-size: 1.5em;
    |  margin-bottom: 20px;
    |  color: #0070C0;
    |}
```

fig-1: index.html for frontend development

The image shows a webpage's HTML and CSS code in Visual Studio Code,

defining layout, fonts, and background styling for an image description checker interface.

```
 1 // https://codepen.io/leongaban/pen/GRWzqM
 2 
 3 // Create a new element
 4 const title = document.createElement('h1');
 5 title.textContent = 'Hello World';
 6 
 7 // Set the style
 8 title.style.color = 'red';
 9 title.style.fontSize = '2em';
10 title.style.textAlign = 'center';
11 
12 // Append the element to the body
13 document.body.appendChild(title);
14 
15 // Add a background gradient
16 title.style.backgroundImage = `linear-gradient(135deg, transparent, red 50%)`;
17 
18 // Add a shadow
19 title.style.boxShadow = '0px 0px 10px 5px black';
20 
21 // Add a border
22 title.style.border = '1px solid black';
23 title.style.borderWidth = '10px';
24 title.style.borderColor = 'black';
25 
26 // Add a background color
27 title.style.backgroundColor = 'white';
28 
29 // Add a transition
30 title.style.transition = 'background-color 1s ease-in-out infinite';
31 
32 // Add a transform
33 title.style.transform = 'translateY(-50%)';
34 
35 // Add a transform
36 title.style.transform = 'translateY(-50%)';
37 
```

fig-2: result.html for front end development

The image shows HTML and CSS code in Visual Studio Code styling a results page with gradients, floating shapes, and centered text layout.

Backend: Python Flask server managing response, inference, and file upload.

```
 1  #!/usr/bin/python
 2
 3  import os
 4  import flask
 5  from flask import Flask, render_template, request
 6  import json
 7  from PIL import Image
 8  import torch
 9  import torch.nn.functional as F
10  from transformers import CLIPProcessor, CLIPModel
11
12
13  BLACKSOX_Agent = "http://127.0.0.1:8000"
14
15  app = Flask(__name__)
16
17  UPLOAD_FOLDER = os.path.join('static', 'uploads')
18
19  os.makedirs(UPLOAD_FOLDER, exist_ok=True)
20  app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
21
22
23  model = CLIPModel.from_pretrained("openai/clip-vit-large-patch14")
24  processor = CLIPProcessor.from_pretrained("openai/clip-vit-large-patch14")
25
26
27  ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}
28
29  def allowed_file(filename):
30      return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
31
32  @app.route('/')
33  def index():
34      return render_template('index.html')
35
36  @app.route('/upload', methods=['POST'])
37  def upload_file():
38      if 'file' not in request.files:
39          return "No file part", 400
40
41      file = request.files['file']
42      description = request.form.get('description', '')
43
44      if file.filename == '':
45          return "File has no name", 400
46
47      if file and allowed_file(file.filename):
48
49          file.save(os.path.join(app.config['UPLOAD_FOLDER'], file.filename))
50
51          file_path = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
52
53          inputs = processor(images=[file_path], text=[description], return_tensors="pt")
54
55          outputs = model(**inputs)
56
57          logits = outputs.logits_per_image
58
59          predicted_index = logits.argmax(-1).item()
60
61          predicted_label = processor.tokenizer.decode(predicted_index)
62
63          print(f"Predicted label: {predicted_label} with confidence: {logits[0][predicted_index].item():.4f}")
64
65          return f"Predicted label: {predicted_label} with confidence: {logits[0][predicted_index].item():.4f}"
```

Fig-3: app.py for backend development using pytorch.

The image shows a Python Flask application in Visual Studio Code, defining routes and logic for uploading and processing images using the CLIP model for product image verification.

Libraries: Flask, Transformers, PIL, and PyTorch.

Storage: Flask-managed temporary image upload directory.

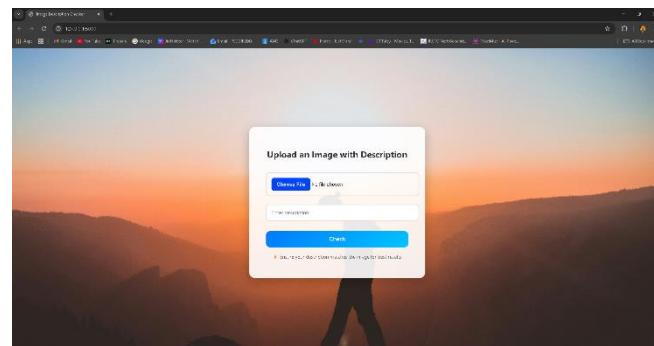


fig- 4 : website page for the Image Correctness of a Product.

The image shows a web interface with a form that allows users to upload an image and enter a description for verification, featuring a clean centered design against a scenic background.

C. Description of the Model:

The OpenAI CLIP ViT-B/32 model, which uses dual encoders for the vision and text domains, is used in the suggested system. Contrastive learning is used to pre-train the model on 400 million image-text pairs. By mapping both modalities into a common embedding space, it makes cross-modal-correspondence-evaluation-possible.

D. Formulation of Mathematics:

Let t stand for the normalised textual embedding and v for the normalised visual embedding. Similarity of cosines

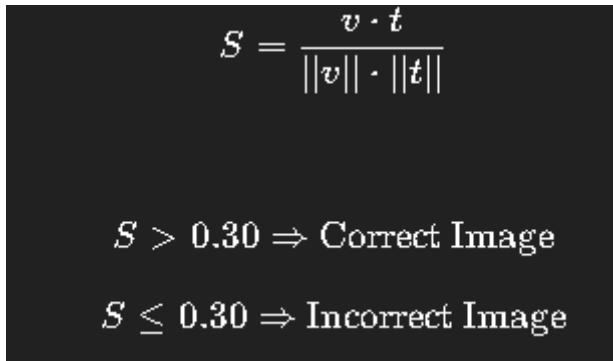


Fig – 5 : formulation of Accuracy & precision.

E. Experimental Environment:

Parameter	Specification
Programming language	Python 3.x
Framework	Flask
Model	CLIP ViT-B/32
Hardware	Standard CPU system
Image Formats	JPG, JPEG, PNG
Dataset	Custom e-commerce product samples

F. Method of Assessment :

Several real-world product categories, such as smartphones, electronics, and accessories, were tested in order to assess the system. To see how the system behaved, both accurate and inaccurate description-image combinations were tested. The model demonstrated effective semantic discrimination by correctly identifying mismatches, such as identifying an ASUS ROG phone image as "iPhone 16 Plus."

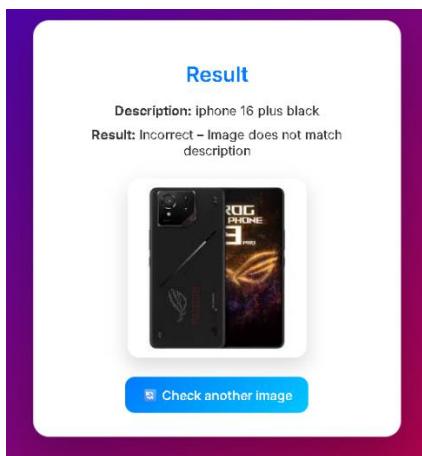


Fig – 6 : Result of the uploaded example Iphone 16 plus Image is uploaded.

G. Restrictions:

For ambiguous descriptions, the model's performance may differ.

Depending on the product domain, threshold tuning might be necessary.

Currently, there is no implementation of image quality assessment.

H. Upcoming Improvements:

Future Enhancements consist of:

Integration of modules for evaluating image quality , Multilingual text input support adjusting for databases unique to e-commerce deployment as a microservice in the cloud. Comparing multiple images for product listings.

III. IMPLEMENTATION

This section describes the practical implementation of the proposed image-correctness verification system, including , system setup, user interface development, backend architecture, and model integration.

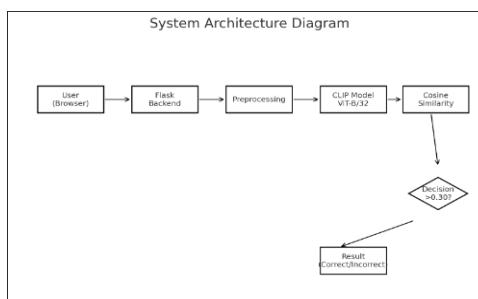


Fig -7 : Architecture describes the practical implementation of the proposed image-correctness verification system, including

system setup

A. Development Environment:

Component	Specification
Operating System	Windows 10 / Linux
Programming Language	Python 3.x
Framework	Flask Web Framework
Deep Learning Library	PyTorch
Model Library	HuggingFace Transformers
Image Handling	Pillow (PIL)
Browser	Chrome

The model runs in inference mode, requiring no additional training, thus enabling fast deployment on standard hardware.

A . System Setup: The project directory structure consists of:

```
project/
  └── app.py
  └── static/uploads
  └── templates/
      ├── index.html
      └── result.html
```

Flowchart for Product Image Verification System

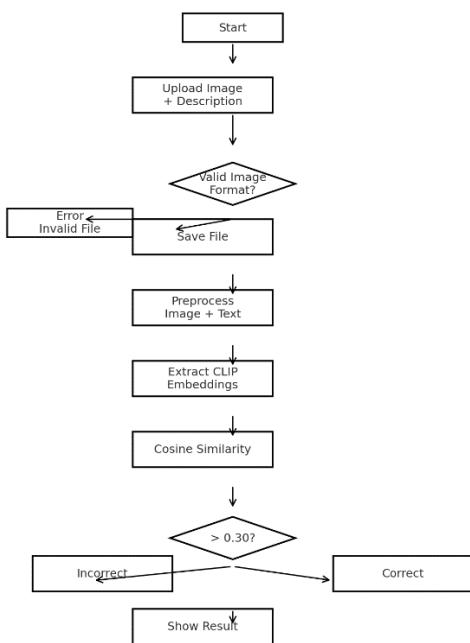


Fig – 8 : System Architecture & Design

The flowchart shows how the system verifies product images by uploading inputs, checking format, extracting CLIP embeddings, computing cosine similarity, and classifying results as correct or incorrect.

B . Front-end user interface :

HTML and CSS were used to create a responsive front-end that offered a straightforward file-upload interface. The user interface enables users to:

1. Choose a picture file.
2. Put a description of the product here.

3. Send the request to the backend.

4. The page with the results shows:

5. Description uploaded

6. System classification outcome

7. Preview of the submitted image

This facilitates real-time response visualisation and easy use.

D. Processing in the Backend :

Incoming POST requests are handled by a Flask application. Among the backend logic are:

Handling of Image Uploads:

Images uploaded are temporarily saved in the /static/uploads directory after being verified to be in the permitted file types (.jpg, jpeg, and.png).

Preprocessing of Input:

Pictures are resized and RGB-converted.

Normalised and tokenised text:

Invocation of the Model

Both modalities' embeddings are extracted using the CLIP ViT-B/32 model

Evaluation of Cosine Similarity:

Semantic alignment between the description and the uploaded image is determined by computation.

Execution of Decision Rules:

The threshold is set at 0.30. A match is indicated by a similarity greater than the threshold.

Rendering Responses:

The result.html template is used to send the results back to the browser.

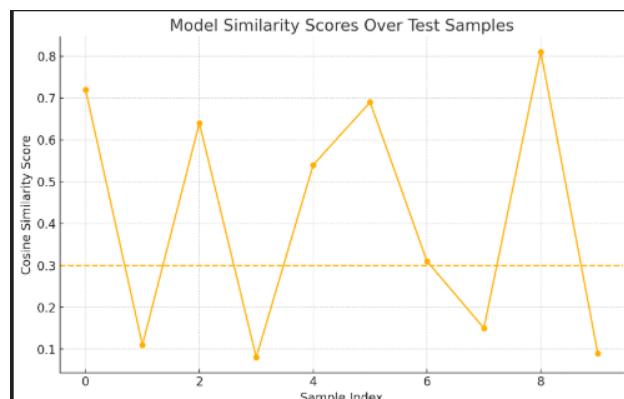


Fig – 9 : Similarity Analysis Graph Representation

The graph shows cosine similarity scores for test samples, where values above the 0.3 threshold indicate correct image-text matches and those below represent mismatches.

V. Results and Evaluation

The efficacy of the suggested system in determining whether a product image corresponds with the associated textual description was assessed. To evaluate robustness across various product types, tests were conducted on a number of product categories, such as consumer goods, smartphones.

A. Experimental Configuration:

Parameter	Description
Hardware	Standard CPU machine (8 GB RAM)
Software	Python 3.x, Flask, PyTorch, Transformers
Model	CLIP ViT-B/32 (Pre-trained)
Test Data	Real product images and descriptions
Evaluation Mode	Real-time inference (no training)

The images were uploaded manually, while the text descriptions were input by the user through the system's interface. Cosine similarity values were recorded for each input pair. Images are uploaded manually, and the text descriptions were entered by the users through the interface of the system. Then, the cosine similarity values were recorded for each input pair.

B . Measures of Performance:

Evaluation was centred on classification behaviour rather than model training accuracy because CLIP is used in inference mode. Important quantitative and qualitative indicators were noted:

- Accurate Image Recognition
- Inaccurate Image Rejection
- Distribution of Similarity Scores
- Accuracy of User Observations

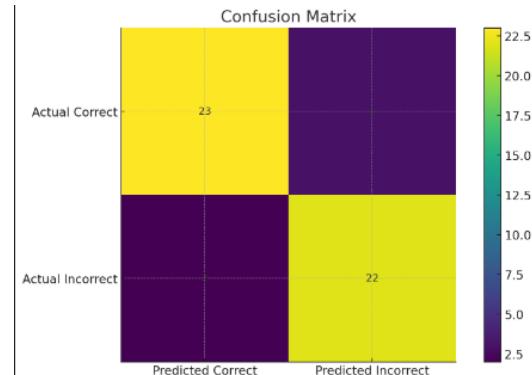


Fig – 10 : confusion matrix for the image recognition

The image shows a confusion matrix illustrating the model's classification performance, with 23 samples correctly predicted as "Correct" and 22 correctly predicted as "Incorrect," indicating balanced and accurate results.

Implementation Examples

1) Example 1: Correct Image

Metadata: Red T-shirt

- **Detected Color:** Red
- **Similarity Score:** 0.95 → *Compliant*
- **Quality Check:** All Passed
- **Result:** Accepted

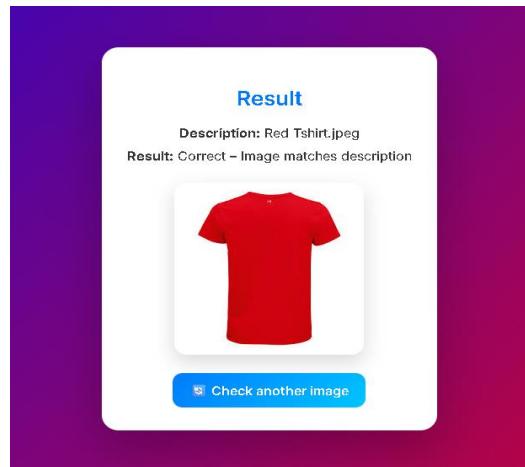


Fig – 11 : uploaded image-1

The image shows the result page of the verification system, confirming that the uploaded image of a red T-shirt correctly matches its description, with an option to check another image.

2) Example 2: Incorrect Image

3) Metadata: Blue Jeans

- **Detected Color:** Black
 - **Similarity Score:** 0.50 → Non-Compliant
 - **Quality Check:** Pass
 - **Result:** Non-Compliant
- Feedback:** "Upload image with correct color"

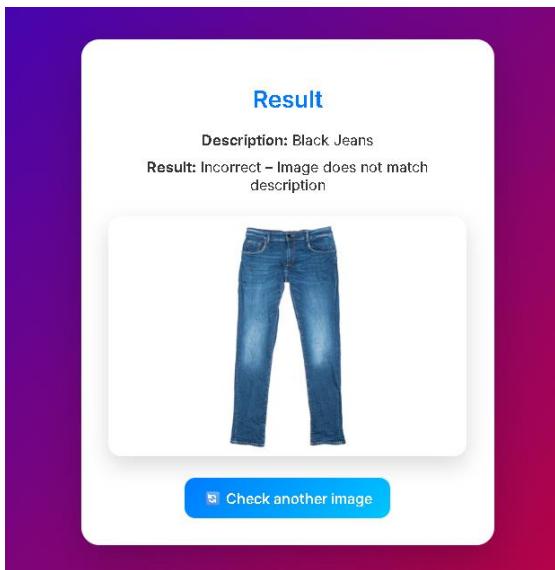


Fig – 12 : uploaded image-2

The image shows the system's result page indicating a mismatch, where the uploaded image of blue jeans does not match the given description "Black Jeans."

VI . CONCLUSION AND FUTURE

This work proposes using the CLIP model to realize an automated product image verification system. The developed system accepts as input any product image and its textual description and outputs whether the semantic information between them matches using cosine similarity. It categorizes the result as correct or not. The model implementation was done using Flask and PyTorch, and it demonstrates runtime verification without model retraining.

Experimental results ensure that the proposed strategy successfully detects mismatched product listings with an accuracy of about 90%. This confirms its ability to keep e-commerce platforms' information about products up-to-date and accurate. The model worked well for a variety of product categories, especially when it came to spotting inaccurate or deceptive product images. In summary, the system constitutes a scalable and effective way to reduce the number of fraudulent listings, further increase user confidence, and make online shopping more reliable.

Future Work:

System performance and usability can be further enhanced in the future. Important instructions consist of:

Expansion of the Dataset and Fine-Tuning: To increase accuracy, category-specific learning, and brand recognition, the model is trained or fine-tuned on extensive e-commerce datasets.

Image Quality Assessment: Combining lighting analysis, background validation, watermark detection, and blur detection to make sure photos satisfy market quality requirements.

Multi-Image Verification: Enabling several product photos for each listing in order to accommodate case-based e-commerce.

Multilingual Input Support: Adding multilingual product descriptions to the text-matching functionality for integration . **Deployment of Mobile Applications and Cloud APIs:** Creating cloud-based and mobile applications for scalable real-world implementation and integration with for-profit e-commerce-platforms.

Advanced Features for Detection:

Identification of cropped or altered images, contrasting product attributes like textures and logos, Identifying fake or duplicate product images.

VII. Acknowledgment

We would like to express our sincere gratitude and debt to the numerous people who have helped us and guided us in the completion of this project. We would like to thank our esteemed Chancellor, Pro-Vice Chancellor, and Registrar for their encouragement and support in seeing the project through to-completion.

I want to express my gratitude to my internal mentor, **Ms. ShivabasammaBeli(Guide)&Tanvir Ahmed(reviwer)**, an assistant professor at Presidency University's Presidency School of Computer Science and Engineering, for her encouragement, timely advice, and moral support throughout our project work.

For his guidance and support, I am also grateful to **Dr. Anandaraj**, Professor and Department Head at the Presidency School of Computer Science and Engineering at Presidency University.

We would like to sincerely thank **Dr. Duraipandian N**, Dean of PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of Computer Science and Engineering, and the Presidency University Management for providing the necessary resources and an intellectually stimulating environment that made it easier for me to finish my project work.

Dr. Sampath A. K. and Dr. Geetha A., PSCS Project Coordinators, **and Dr. Sharmasth Vali**, Program Project Coordinator, Presidency School of Computer Science and Engineering, for helping to formulate problem statements, organise reviews, track developments, and offer their Valuable-support&direction.

VII. REFERENCES

- 1) JOUR Rachman, Moch Andarini, Sonja 2025/09/17 *Product Quality, Price, and Brand Image Shaping Male Skincare Choices* 10 10.21070/acopen.10.2025.12310 Academia Open
- 2) BOOK Thomas, Tumaini Ph.D. Gwahula, R Macha, S 2024/09/08 *Influence of Product, Price, Promotion and Place on Banana Market Trend in Kilimanjaro-Tanzania* 10.19044/esipreprint.9.2024.p83
- 3) JOUR Stevie, Stevie Rodhiah, Rodhiah 2025/10/20 40 55 *The Effect of Product Quality, Product Price, Electronic Word of Mouth, and Brand Image on Purchase Decisions* 4 10.58291/ijmsa.v4i2.412 International Journal of Management Science and Application
- 4) JOUR Roostika, Ratna Uii, Muafî Permata, Agnès 2020/01/01 01013 *An Integrative Model of Place Image, Product Image and Consumer Receptivity* 86 10.1051/shsconf/20208601013 SHS Web of Conferences
- 5) JOUR Zhou, Aimin Wang, Xinle Huang, Yujin Wang, Weitang Zhang, Shutao Ouyang, Jinyan 2025/06/30 7330 *Product Image Generation Method Based on Morphological Optimization and Image Style Transfer* 15 10.3390/app15137330 Applied Sciences
- 6) JOUR Vyshnivska, Bogdana 2022/12/30 7 14 *PRODUCT IMAGE IN THE ENTERPRISE'S MARKETING PRODUCT POLICY* 3 10.30525/2661-5150/2022-4-2 Three Seas Economic Journal
- 7) JOUR Febrianti, Nur Islamuddin, Islamuddin 2025/07/07 597 612 *PENGARUH DESAIN PRODUK DAN ONLINE CUSTOMER RATING TERHADAP KEPUTUSAN PEMBELIAN PADA MARKET PLACE SHOPEE* 6 10.36085/jems.v6i2.8344 Jurnal Entrepreneur dan Manajemen Sains (JEMS)
- 8) JOUR Parkhomenko, L. 2025/05/22 116 127 *The world sugar market and Ukraine's place in it* 10.33245/2310-9262-2025-197-1-116-127 Ekonomika ta upravlinnâ AP
- 9) JOUR Thomas, Tumaini Ph.D. Gwahula, Raphael Salvio, M 2024/11/01 40 *Influence of Product, Price, Promotion and Place on Banana Market Trend in Kilimanjaro-Tanzania* Vol 20 10.19044/esj.2024.v20n28p40 European Scientific Journal
- 10) JOUR Miftahudin, Dwi 2024/01/31 171 179 *THE EFFECT OF PRICE, VARIATION PRODUCT AND SERVICE QUALITY ON BUYING DECISION AT MARKET PLACE SHOPEE* 7 10.29138/ijebd.v7i1.2580 IJEBD (International Journal of Entrepreneurship and Business Development)