International Journal on Engineering Technology (InJET)

www.kec.edu.np/kec-journal/
Volume 2, issue no. 1, Dec 2024, Page 106-116
Received Date: July 21, 2024
Accepted Date: Sept 23, 2024
ISSN: 3021-940X (Print)
DOI: https://doi.org/10.3126/injet.v2i1.72526

Content-Based Image Retrieval and Recommendation Based on User Reviews

Anup Kafle [1,*], Anushil Timsina [2], Rochak Sedai[3], Sandeep Subedi[4], Upendra Prasad Neupane[5*], Prakash Chandra Prasad[6]

[1]Khwopa College of Engineering,Tribhuvan University, Libali, Bhaktapur, Nepal, anupkafle24@gmail.com
[2]Khwopa College of Engineering,Tribhuvan University, Libali, Bhaktapur, Nepal, anushil.timsina1@gmail.com
[3]Khwopa College of Engineering,Tribhuvan University, Libali, Bhaktapur, Nepal, rozsedai@gmail.com
[4]Khwopa College of Engineering,Tribhuvan University, Libali, Bhaktapur, Nepal, sandeepsubedi94@gmail.com
[5]Sagarmatha Engineering College, Tribhuvan University, Sanepa, Lalitpur, Nepal, info.upendrapn@gmail.com
[6]IOE Pulchowk Campus, Tribhuvan University, Pulchowk, Lalitpur, Nepal, prakash.chandra@pcampus.edu.np

Abstract

Effective search functionality is crucial for enhancing user experience and boosting sales on e-commerce sites. However, relying on specific product names instead of common names can cause potential customers to be lost while searching for products, as many platforms still depend on text-based search engines. While text searches effectively find keywords, Search Engine fall short when customers do not know the exact product identity or only have an image of desired product. With the rise of multiple E-commerce sites, customers often feel confused about where to buy the best product. This paper proposes a novel architecture for recommending the desired product based on image search. The input image provided by the user is processed through YOLO models to extract the brand, category, and color of the shoe, which serve as keywords. Data about shoe with similar keywords is retrieved from multiple e-commerce sites using APIs. The SIFT (Scale-Invariant Feature Transform) algorithm is employed to compare the input image with the extracted images and assign a score. Additionally, reviews, ratings, and price analyses are conducted on the extracted data, and scores are assigned accordingly. The TextBlob library is used for sentiment analysis of reviews. Based on these scores, the best product is recommended to the customer. The architecture is deployed on an EC2 instance on the AWS platform, demonstrating its practical applicability. The result is a model that is able to suggest similar product wanted by the user which has the most positive reviews and ratings from other users and has the most competitive price in the market. The project suggests an easier and comparatively better way to improve the online shopping experience of users by providing what they want which is liked by others as well and are priced competitively.

Keywords: E-commerce, Text-based search, Deep learning, Sentiment analysis, YOLO.

1. Introduction

Online shopping, along with e-commerce sites, has been on the rise lately in Nepal as well as all around the world. The fact that people get what they want at their homes has made online shopping a convenient market for buyers and a good marketplace for sellers. This online shopping has generated a lot of data, which is ideal for people to research and improve the general shopping experience for customers, making it easier for sellers to sell their products. The data has been in the form of text, which may be product names or descriptions of the product, like its features. Also, the data can be in the form of images showing the product to prospective buyers. The product searching and finding process is thorough. A proper product name may be required for the algorithm to find the product we are looking for, and having the product's image is next to no use, as using images for search is rare in e-commerce websites. As most people still prefer offline markets for shopping because they can get a feel for what they are buying, it is necessary to have a platform to convince users that the products they are about to buy online are the best they can get and are not getting ripped off.

There has been a lot of work in image recognition and interpretation of the recognized image. (Singh, 2020)

**Corresponding Author*

proposed an efficient bi-layer CBIR system that extracts image features in terms of color, texture, and shape, comparing query images with dataset images to retrieve the most similar ones. Our model purposes a simpler way to extract features of color, category, and brand compared to them with increase in performance. Real-time web scrapping is a common practice. Users can sort all scraped products according to prices in descending order. Sentiment analysis from product reviews determines the gist. Finally, the system implements product recommendations based on reviews and purchase history. But our system combines all these methods to give the user the best shopping experience, and they can get the best product verified by other users at the best possible price.

1.1.Problem Definition

As online shopping grows daily, it significantly impacts the Nepali market, with e-commerce usage at an all-time high. However, finding the desired product remains challenging due to the limited use of photo-based searches on Nepali e-commerce sites. This issue forces customers to spend excessive time searching for specific products, even when images are available. Additionally, customers must switch between different sites to compare prices for the same product, further complicating the shopping process. Reviews can be vague or manipulated by sellers, making it difficult for customers to trust the information provided.

Moreover, there are few mechanisms to suggest similar products with better reviews within the same price range. These problems make online shopping cumbersome and time-consuming for customers, potentially discouraging future use and resulting in significant losses for e-commerce sites and online businesses. This paper aims to address these issues by proposing a solution that enhances the product search experience, ensuring customers can find the best products efficiently and reliably.

2. Literature Review

In recent decades, significant research has been conducted on Content-Based Image Retrieval (CBIR) and Recommendation Systems. (Resnick, 2006) introduced a virtual search engine for product images using CBIR technology and analyzed various visual descriptors. (Miyazawa, 2013) developed a context-aware recommendation system using CBIR, which improves image-matching precision by incorporating contextual information.

(Singh, 2020) proposed an efficient bi-layer CBIR system that extracts image features in terms of color, texture, and shape, comparing query images with dataset images to retrieve the most similar ones. (Chan, 2004) demonstrated the effectiveness of linguistic processing in the automatic sentiment classification of product reviews using a Support Vector Machine (SVM) on text features. (Fang, 2015) conducted sentiment analysis on product reviews, categorizing sentiment polarity at both sentence and review levels.

In collaborative filtering, (Su, 2009) surveyed memory-based, model-based, and hybrid CF techniques, analyzing their predictive performance. (He, 2017)proposed neural collaborative filtering (NCF) to enhance recommendation systems using neural networks, leveraging a multi-layer perceptron to learn user-item interactions.

3. Methodology

3.1. Data Collection

The required data, such as footwear images, reviews, ratings, and prices, were collected through web scraping from multiple e-commerce websites such as Amazon, Daraz, Sastodeal, Flipkart, etc. The images were then are rotated to give the best view, and labelled using makesense.ai website. 3 different datasets were built for category, color, and brand models by selecting images that fitted the models the most. A total of 4873 images were taken and was done 80:20 split with 3912 images for training, and 961 images for testing.

3.2. System Block Diagram

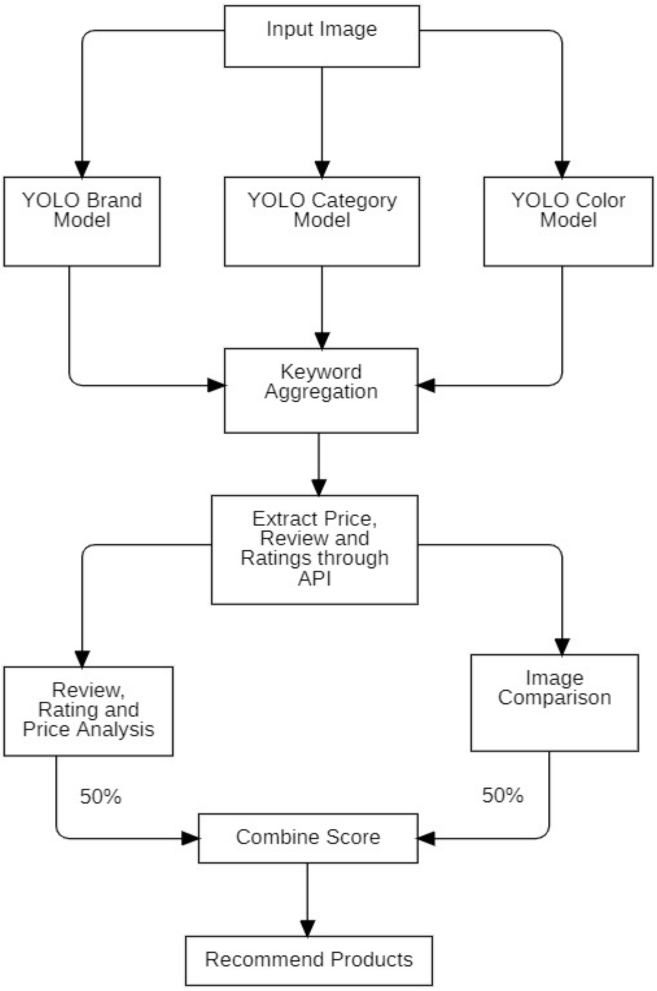


Figure 1. System Block Diagram

The system takes input in the form of an image. Three YOLO models run concurrently to find the Category, Brand, and Color of footwear in the given input. YOLO, or "You Only Look Once," is a highly efficient model for real-time object detection that analyzes an entire image in a single pass, allowing it to quickly identify and classify multiple objects simultaneously. By dividing the image into a grid and predicting bounding boxes and class probabilities for each section, YOLO can accurately detect various objects and their locations, streamlining the process of pattern recognition. Its speed and effectiveness enable machines to interpret their surroundings in real time, enhancing decision-making in dynamic environments.

The Category model labels images as per category, such as sports, sneakers, boots, etc. The brand model labels footwear images based on brands such as Goldstar, Converse, Peak, etc. The color model labels footwear images by color, such as red, blue, black, etc. The above three models generate keywords based on the footwear category, brand, and color. The system extracts data about footwear with similar keywords from multiple e-commerce sites using APIs. The data includes reviews, ratings, prices, and images. Using the SIFT algorithm, the system compares the input image with the images extracted from e-commerce sites to find a similar product.

Scoring is also done for similarity, representing 50% of the overall score—review, rating, rating, and price analysis parallel the above step. Using sentiment analysis, scoring is done for a review of the product. Rating ratings and prices are also considered when assigning scores. An item having more positive reviews, high ratings, and low prices is given more scores. It carries 50% of the overall score. The score from the above two steps is combined and arranged in descending order. As per the score, the products are recommended to the user.

3.3. SIFT Algorithm

The Scale Invariant Fourier Transform (SIFT) is a powerful tool used in image processing to identify and analyze distinct features in images, no matter their size or orientation. Essentially, it allows computers to recognize objects consistently, whether they appear small or large. By using a method called the Fourier Transform, SIFT converts the visual information of an image into a different form that highlights patterns and structures more clearly. This makes it especially useful for tasks like recognizing objects, stitching together images, and creating 3D models. Overall, SIFT enhances how machines understand and interpret visual data.

Implementation:

1) Scale-space peak selection:

Potential location for finding features. The scale space of an image is a function L(x,y,σ) that is produced from the convolution of a Gaussian kernel(Blurring) at different scales with the input image. Scale-space is separated into octaves, and the number of octaves and scale depends on the size of the original image. Several octaves of the original image are generated. Each octave's image size is half the previous one. Within an octave, images are progressively blurred using the Gaussian Blur operator. Mathematically, "blurring" is the convolution of the Gaussian operator and the image.

$$L(x,y,\sigma) = G(x,y,\sigma)*I(x,y) \qquad \text{(Equation 1)}$$

Those blurred images are used to generate another set of images, the Difference of Gaussians (DoG). These DoG images are great for finding out interesting key points in the image. The difference between Gaussian and Gaussian blurring of an image is obtained. One pixel in an image is compared with its 8 neighbors as well as 9 pixels in the next scale and 9 pixels in previous scales. This way, a total of 26 checks are made. If it is a local extremum, it is a potentially key point. It basically means that the key point is best represented in that scale.

2) Key point Localization:

Accurately locating the feature key points. Key points generated in the previous step produce a lot of key points. Some of them lie along an edge, or they don't have enough contrast. In both cases, they are not as helpful as features. So, get rid of them. Their intensities should be checked for low-contrast features. Taylor's series expansion of scale space was used to get a more accurate location of the extrema. If the intensity of these extremes is less than a threshold value (0.03 as per the paper), it is rejected. DoG has a higher response for edges, so edges must also be removed. A 2x2 Hessian matrix (H) was used to compute the principal curvature.

3) Orientation Assignment:

Describing the key points as a high-dimensional vector. Compute a descriptor for the local image region about each key point that is as highly distinctive and invariant as possible to variations in viewpoint and illumination. To do this, a 16x16 window around the key point is taken. It is divided into 16 sub-blocks of 4x4 size. So, 4 X 4 descriptors over 16 X 16 sample arrays were used in practice. 4 X 4 X 8 directions give 128 bin values. It is represented as a feature vector to form a key point descriptor. This feature vector introduces a few complications. Get rid of them before finalizing the fingerprint.

4) Key Point Descriptor:

Describing the key points as a high-dimensional vector. Compute a descriptor for the local image region about each key point that is as highly distinctive and invariant as possible to variations such as changes in viewpoint and illumination. To do this, a 16x16 window around the key point is taken. It is divided into 16 sub-blocks of 4x4 size. So, 4 X 4 descriptors over 16 X 16 sample arrays were used in practice. 4 X 4 X 8 directions give 128 bin values. It is represented as a feature vector to form a key point descriptor. This feature vector introduces a few complications. Get rid of them before finalizing the fingerprint.

5) Key Point Matching:

Critical points between two images are matched by identifying their nearest neighbors. However, in some cases, the second closest match may be near the first, possibly due to noise or other reasons. In that case, the ratio of the most relative distance to the second-closest distance is taken. If it is more significant than 0.8, it is rejected. As per the paper, it eliminates around 90% of false matches while discarding only 5% of correct games.

3.4. Implementation of Scoring

In this process, the previously obtained data frame is used. Review is taken as input to perform sentiment analysis using Text Blob. It classified the review in the range [-1,1] where -1 means negative and +1 means positive. 10% score value is assigned to the review:

Review score= Review score of the product /Highest Review score in the data frame. (Equation 2)

Rating is obtained in the form of stars [0.5-5] and stored in the data frame as per the number of stars. Rating is given a 10% score value.

Rating score= Rating of given image /Highest rating among the extracted data (Equation 3)

For price overall 30% score value is assigned. Among the extracted data, the footwear score for the price is given by using the formula:

Price score = (Max price – Price) / Max price (Equation 4)

A similarity comparison is done using the SIFT algorithm to find a similar image between the input and extracted image. It carries a 50% score value. The SIFT algorithm gives the similarity score in ascending order. The footwear is given a score:

Similarity score= Similarity / Max similarity (Equation 5)

The system of comparing images with those existing on the website is done using this algorithm. The input from the user is taken to the YOLO model, which gives the footwear's color, brand, and category. Similar footwear is searched on the websites, and those that match the labels are collected. Then, the collected pictures are compared against the given input image, and similarity is calculated. This comparison is done with the help of the SIFT algorithm. The result of this comparison is how similar the given image is to those in the database in percentage. The image with the highest similarity is displayed to the user as the most similar product to the input they gave. The scale factor is the hyperparameter that can be changed in the SIFT algorithm. The scale factor is a parameter that controls the scale at which features are detected and described. It refers to the ratio between the scale of the original image and the scale of the Gaussian blurred images used to create a scale space. The SIFT algorithm constructs a scale space to detect features at different scales by repeatedly convolving the original image with Gaussian kernels of increasing standard deviations. The result is a series of blurred images at various scales. The scale factor determines the ratio between the standard deviation of the Gaussian kernel used to generate each level in the scale space and that of the previous level. By adjusting the scale factor, the SIFT algorithm can detect features at different scales and resolutions, making it more robust to image size and orientation changes.

Overall score = Similarity score*50%+ Price Score*30% +Rating score*10%+ Review score*10% (Equation 6)

The division of percentages to calculate the overall score depended on multiple factors. We went with the thinking that facts should have more value than opinions. Whenever someone wants to buy footwear as they have referenced, the first thing to check is whether the provided footwear to the customer is similar to the recommended one. So, the project focuses more on the actual content of the 24-footwear and is given 50% weight on the overall score. Then, 30% weight is assigned to the footwear price as the customer is looking for cheaper products during shopping. Finally, 10% each is given to ratings and reviews of the product. Human sentiment is easy to manipulate and does not represent the product only during the review. Reviews contain customer services, delivery time, etc. So, a good product may have a lousy delivery time, affecting the rating and customer reviews.

To validate the choice for the percentages to be given to different models, we tried using different sets of scores for each model. Looking at the outputs provided by the different sets of hyperparameters, these were the best sets of results. High percentage for review and ratings gave more popular products than similar products while high percentage for price always gave the cheapest footwear. According to (Resnick, 2006), customers are likely to buy expensive items compared to cheaper ones if they match their description. So, this strengthens the belief of giving higher priority to similarity score on the overall score.

4. Result

Each model was trained with epochs 50, 100, 150, and 200. The model with the best results were selected which were model with 150 epochs for category and color, and with 100 epochs for brand model.

All the models were underfitting at 50 epochs but started to overfit at 200 epochs.

4.1. Category Model

In this model, 9 types of footwear namely: Sports, Sneakers, Formal, Loafers, Slippers, Boot, Heels, Crocks, and Kitto, were included. 2025 training instances and 512 validation instances were considered for these 9 categories trained for 150 epochs. The F1 score of the model increased to 0.84 and the PR score to 0.911.

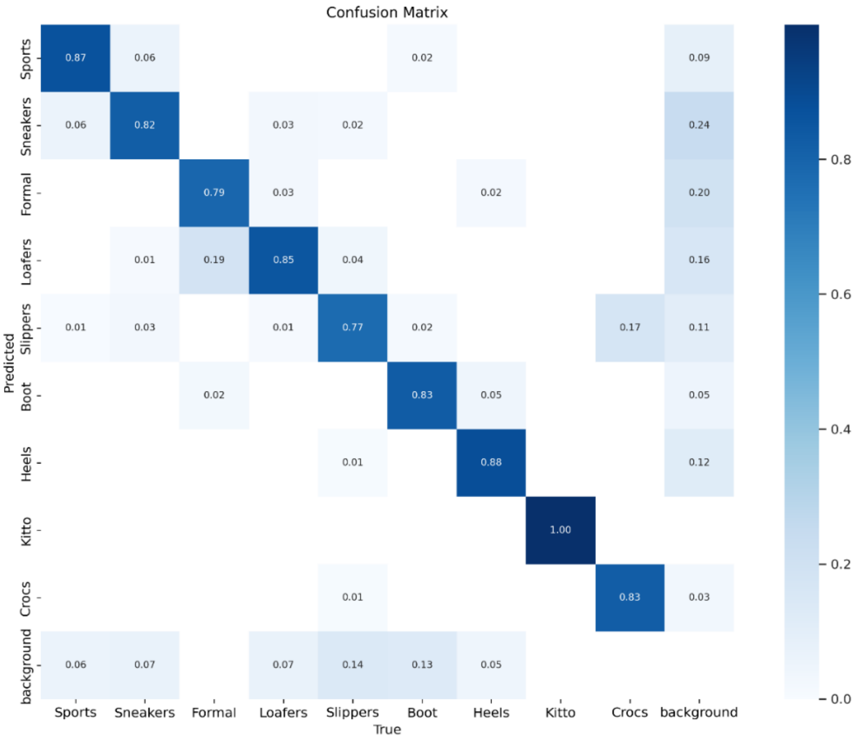


Figure 2. Confusion Matrix for Category Model

Here, we can see the model's confusion matrix. It is a good result as all the values are along the diagonal. The data is distinct to a particular category after removing confusing categories like Slippers and kids. Also, the model's accuracy increased by relabeling the input images to include clear photos and proper bounding boxes, as shown in the matrix above.

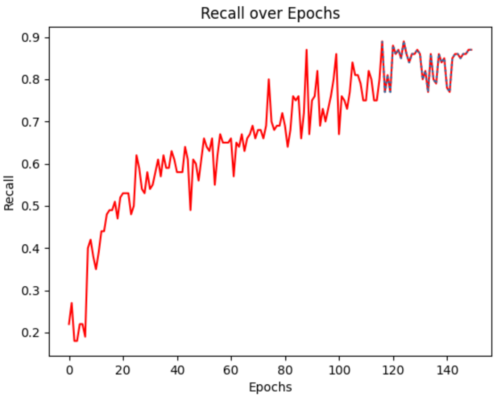


Figure 3. Recall Curve for Category Model

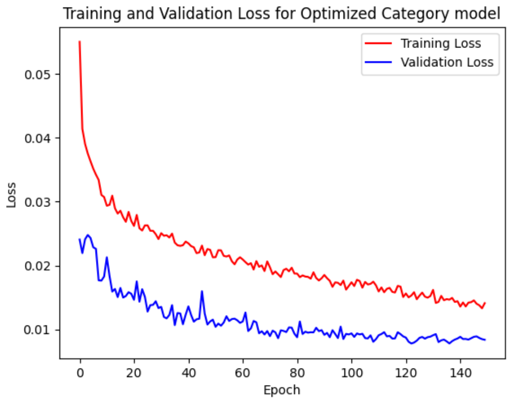


Figure 4. Train and Validation Loss Curve for Category Model

The recall is low at the start. It shoots up exponentially but is distorted. The peak value is seen at 125 epochs. Then, the recall value is almost stable but oscillates. The training loss decreases with the increase in epochs. The decrease is gradual and distorted. The validation loss starts from a lower point than the training loss. It decreases almost exponentially for about 80 epochs. Then, the decrease is gradual and slow. The validation loss is not as smooth as the training loss. It reaches its lowest value at about 130 epochs and is constant afterward.

4.2. Brand Model

In this model, 14 brands of footwear, namely Adidas, Converse, Goldstar, Nike, New Balance, Erke, Vans, Puma, Peak, Reebok, Magic, Asics, Lotto, and SG, were included. 1050 training instances and 247 validation instances were considered for these 14 categories. The training was conducted with parameters of 100 epochs and 32 batch sizes.

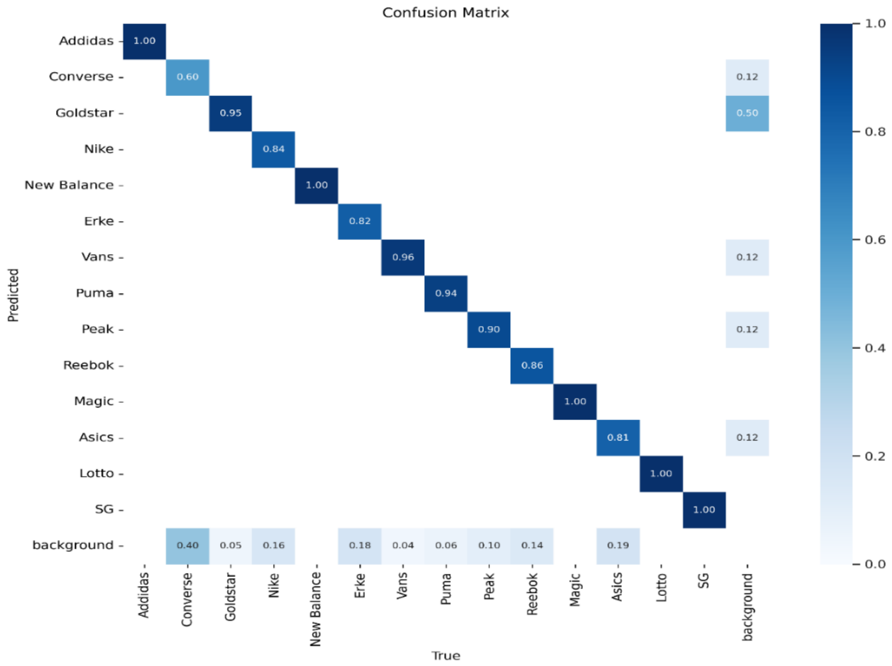


Figure 5. Confusion Matrix for Brand Model

Here, we can see the model's confusion matrix. It is a good result as all the values are along the diagonal. Brands like Adidas, Magic, Lotto, New Balance, and SG, having distinct and clear logos, have a score of 1. No classes are misclassified. Converse has the lowest score of 0.6, and the loss is background loss, as the bounding box could not be made around the Converse logo.

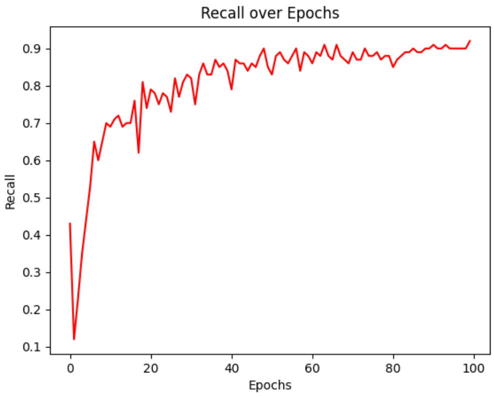


Figure 6. Recall Curve for Brand Model

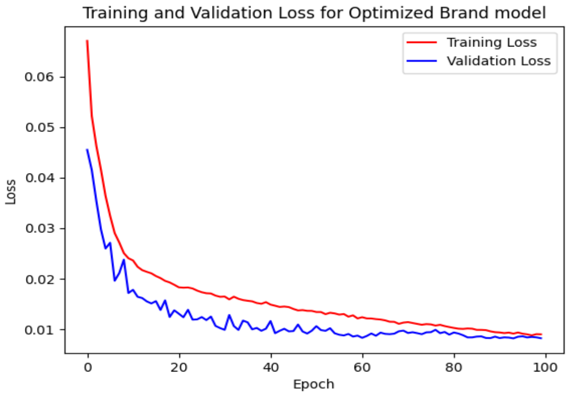


Figure 7. Training and Validation Loss Curve for Brand Model

The recall is surprisingly high at the start. It gets to its lowest point on the net epoch and then increases exponentially with the increase in epochs, but the shape is distorted. After about 50 epochs, the increment is gradual. The peak value is seen at 100 epochs. But, the training loss smoothly decreases exponentially with an increase in epochs. The validation loss starts from a lower point than the training loss. It decreases almost exponentially for about 50 epochs. Then, the decrease is gradual and slow. It reaches the lowest value at about 60 epochs, is constant after that, and nearly intersects with the training loss at about 100 epochs.

4.3. Color Model

In this model, 11 colors of footwear, namely Black, Blue, Brown, White, Grey, Red, Cream, Orange, Pink, Green, and Yellow, were included. 837 training instances and 202 validation instances were considered for these 11 categories. The training was conducted with parameters such as 150 epochs and 64 batch sizes. The F1 score of the model was 0.78, and the PR score was close to 0.85.

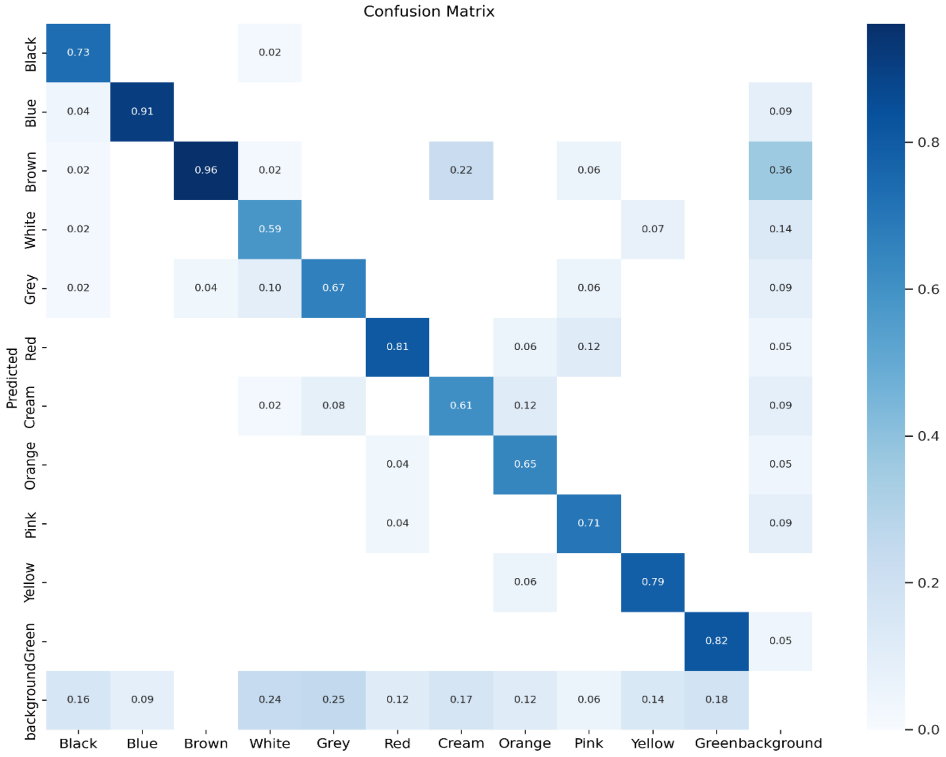


Figure 8. Confusion Matrix for Color Model

Here, we can see the model's confusion matrix. Most of the values are along the diagonal, which is a good result. Almost all the categories are well recognized. However, the model seems confused by similar colors like red and pink, cream and yellow. Also, there have been losses as the objects are not identified and considered background.

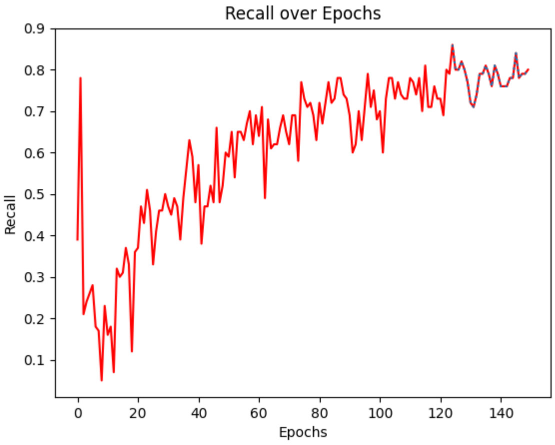


Figure 9. Recall Curve for Color Model

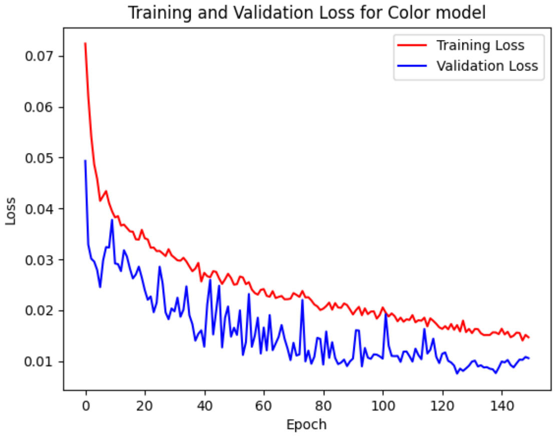
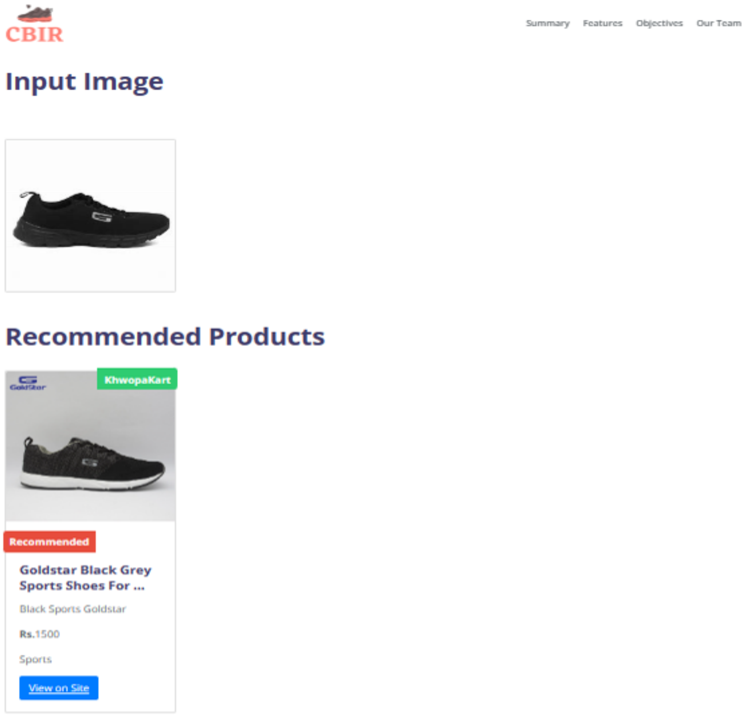


Figure 10. Training and Validation Loss Curve for Color Model

The recall is surprisingly high at the start. It shoots up to nearly 0.8 at epoch two and then drastically falls to 0.25. The lowest recall is at about ten epochs. After ten epochs, recall gradually increases in a highly distorted manner, and the peak value is seen at 125 epochs. Then, the recall value is almost stable.

The training loss decreases exponentially with increased epochs but is not smooth. The validation loss starts from a lower point than the training loss. It decreases gradually for about 140 epochs. The validation loss is not as smooth as the training loss. It reaches its lowest value at about 140 epochs and then is seen increasing.

An example of the output is shown below:



Input Image

Recommended Products

Goldstar Black Grey Sports Shoes For ...

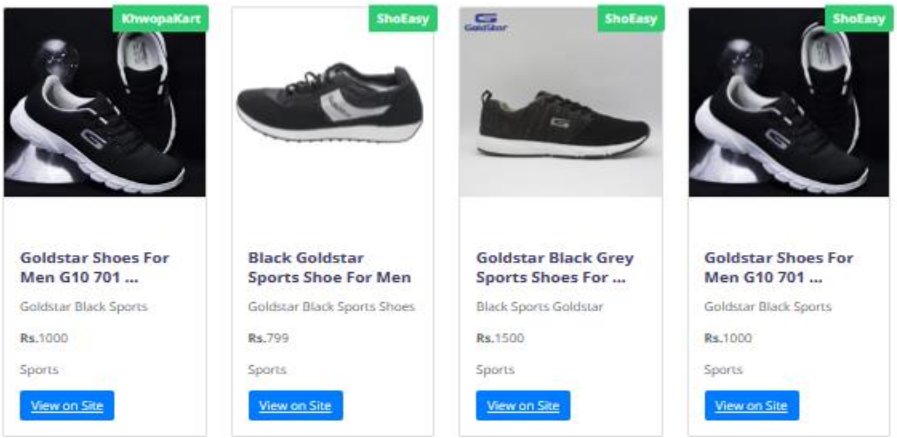
Black Sports Goldstar

Rs.1500

Sports

Figure 11. Recommended Product Based on the Input Image

The single product here gets the highest overall score and is recommended by the engine. Also, other similar and recommended product is shown as:



Goldstar Shoes For Men G10 701 ...
Goldstar Black Sports
Rs.1000
Sports

Black Goldstar Sports Shoe For Men
Goldstar Black Sports Shoes
Rs.799
Sports

Goldstar Black Grey Sports Shoes For ...
Black Sports Goldstar
Rs.1500
Sports

Goldstar Shoes For Men G10 701 ...
Goldstar Black Sports
Rs.1000
Sports

Figure 12. Other Similar Recommended Products

Comparing the results to Singh, S., & Batra, S (2020), our model slightly underperforms using precision as a metric but significantly better if recall is used. But they have used different objects while training.

Table 1. Result Comparison between Bi-Layer CBIR and Purposed Model

Bi-layer Content CBIR Models (Singh et al)	**Peak Precision Score**	**Peak Recall Score**	**Peak F1 Score**
Buildings	0.8	0.17	0.28
Buses	1	0.2	0.33
Food	0.9	0.19	0.31

Purposed CBIR Models	**Peak Precision Score**	**Peak Recall Score**	**Peak F1 Score**
Category	0.618	0.89	0.73
Brand	0.943	0.9	0.92
Color	0.72	0.85	0.78

Interpreting the results, such high improvement in the F1 score means the model significantly outperforms the competition in terms of the result. Also, the time taken by the YOLO model to process the data is way less compared to their models.

4.4. Conclusion

The project is able to classify any given input footwear according to its brand, category, and color. The result also depends on the amount of data present on the website as a new type of footwear does not have results in comparison to footwear present abundantly on the website. If the model cannot find the exact product given by all three keywords, it gives output for two keywords or for one keyword. If no keywords can be generated, the user is asked to input the image again. Content-Based as well as Collaborative Filtering was carried out, making this model a Hybrid Filtering model for the optimal result. One potential disadvantage of this model is the huge time and resource required to collect the data, label it, train the model, and retrain it using different hyperparameters like changing epochs and batch size. Improving the model is the key step to improve the performance of the system. Further optimization of project includes training the model with datasets to increase the precision and removing the false negatives. Also, finding stability in the ever-fluctuating recall score is a future goal.

References

Chan, S. and Na, J.-C. (2004) 'Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews', *Advances in Knowledge Organization*, 9, pp. 49-54.

Fang, X. and Zhan, J. (2015) 'Sentiment analysis using product review data', *Journal of Big Data*, 2(1), pp. 1-14.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.-S. (2017) 'Neural collaborative filtering', in *Proceedings of the 26th International Conference on World Wide Web*, pp. 173-182

Lin, X., Gokturk, B., Sumengen, B. and Vu, D. (2008) 'Visual search engine for product images', in *Proceedings of SPIE Multimedia Content Access: Algorithms and Systems II*, pp. 22-60.

Miyazawa, Y., Yamamoto, Y. and Kawabe, T. (2013) 'Context-aware recommendation system using content-based image retrieval with dynamic context considered', in *2013 International Conference on Signal-Image Technology & Internet-Based Systems*, IEEE, pp. 779-783.

Resnick, P., Zeckhauser, R., Swanson, J. and Lockwood, K. (2006) 'Price versus similarity: An empirical study of consumer behavior in online product search', *Electronic Commerce Research and Applications*, 5(1), pp. 40–50.

Singh, S. and Batra, S. (2020) 'An efficient bi-layer content-based image retrieval system', *Multimedia Tools and Applications*, 79(25), pp. 17731-17759.

Su, X. and Khoshgoftaar, T. M. (2009) 'A survey of collaborative filtering techniques', *Advances in Artificial Intelligence*, 2009, pp. 1-19.