



AI-Based Skin Condition Detection and Personalized Cosmetic Recommendation Using the YOLO Deep Learning Model

¹Shruthi M K, ²Ramesh B E, ³Khazi Mujtaba Abdul Kabeer, ⁴Chinmayee U, ⁵Dhanush H, ⁶Karthik E H

¹ & ² Professor and ^{3,4,5} & ⁶ Final year students, Department of Computer Science and Engineering
SJM Institute of Technology, Chitradurga, Karnataka, India
Affiliated to Visvesvaraya Technological University (VTU), Belagavi

Abstract—Choosing a cosmetic product without knowing the actual condition of one's skin is a common problem, and most people end up buying on advertisement or on a friend's word. In this work we present a system that looks at a face photograph, finds the visible skin problems, and then suggests suitable cosmetics. The detection part is built on YOLO (You Only Look Once), a single-stage deep learning object detector that localises acne, dark spots and wrinkles in one forward pass. The detected issues are passed to a recommendation layer that maps them to products stored in a cosmetic product database, and the result is shown to the user through a simple interface. The paper explains the layered system architecture, the data flow design from Level 0 to Level 1, the role of the two data stores, and the way the detector output is converted into product suggestions. The design is modular, so that the detector, the database and the interface can be improved separately. The proposed system is meant to give quick, low-cost and non-clinical guidance to a user who has only a camera or a smartphone.

Keywords—skin classification, skin problem detection, YOLO, deep learning, cosmetic recommendation, data flow diagram, computer vision.

I. INTRODUCTION

Skin is the first thing a person notices about another person, and problems such as acne, dark spots and wrinkles affect both appearance and confidence. A trained dermatologist can judge these conditions correctly, but a visit is not always possible, particularly in small towns where specialists are few and the cost is high. Because of this, many people choose skincare and cosmetic products by trial and error. A product that does not suit the skin can make the problem worse.

Deep learning has changed how images are analysed. Convolutional neural networks (CNN) learn features directly from pixels and have reached very good accuracy in image classification and object detection [1], [2]. In the medical imaging field, the work of Esteva et al. showed that a deep network can classify skin lesions at a level comparable to dermatologists [3]. This encouraged us to ask a simpler, everyday question: can the same kind of technology help an ordinary user to understand the visible condition of the face and pick a suitable cosmetic?

In this project we have combined two ideas. The first is skin problem detection using a YOLO model, and the

second is a recommendation engine that links the detected problem to cosmetic products. We chose YOLO because it treats detection as a single regression problem and therefore runs fast enough for use on ordinary hardware [4]. The main contributions of this paper are listed below.

- (a) A four-layer architecture that takes a face image as input and returns cosmetic suggestions as output.
- (b) A data flow design, described at Level 0 and Level 1, with two data stores for skin images and cosmetic products.
- (c) A rule for connecting the output of an object detector to a product recommendation step, which keeps the detection and the recommendation parts independent of each other.

The remainder of the paper is arranged as follows. Section II reviews the related work. Section III states the problem and the objectives. Section IV describes the system architecture, Section V gives the data flow design, and Section VI explains the implementation approach. Section VII discusses evaluation measures, Section VIII lists the advantages and limitations, and Section IX concludes the paper.

II. RELATED WORK

Object detection methods are usually divided into two groups. Two-stage detectors such as Faster R-CNN (Region-based Convolutional Neural Network) first propose candidate regions and then classify each of them [5]. They are accurate, but the two steps make them slower. Single-stage detectors, of which YOLO is the best known, predict the boxes and the class probabilities together from the full image [4]. The later versions of this family improved the speed and the accuracy step by step, through better backbones, multi-scale prediction and training tricks [6]–[9]. A broad survey of two decades of progress in object detection is given by Zou et al. [10].

For skin analysis, most published work concentrates on medical diagnosis. Esteva et al. trained a deep network on a large set of clinical images and compared it with dermatologists for skin cancer classification [3]. Such systems are designed for disease screening and need carefully labelled medical data. Our target is different. We are interested in common cosmetic concerns that are visible on the face, and we do not attempt to diagnose a disease.

Recommender systems are the second area that is related to our work. A recommender filters a large collection of items and presents the few that suit the user's need [11], [12]. Content-based methods match the properties of an item with the needs of the user, and this is close to what we do when a detected skin problem is matched with a product that is meant for it. Deep residual networks and very deep convolutional networks have also become standard feature extractors for the image part of such pipelines [13], [14]. The present work therefore stands between skin image analysis and product recommendation, and it joins the two with a simple, modular design.

III. PROBLEM STATEMENT AND OBJECTIVES

The problem addressed here is stated as follows: given a face image captured by a camera or uploaded from storage, identify the skin problems that are visible in it and suggest cosmetic products that are suitable for those problems. The objectives of the work are given below.

- 1) To accept a skin image from the user through camera capture or file upload.
- 2) To detect and classify visible skin problems with a YOLO based detection model.
- 3) To maintain a cosmetic product database and a recommendation logic that maps detected problems to products.
- 4) To present the suggested cosmetics to the user in a clear and easy to read form.
- 5) To keep the design modular so that every layer can be modified without disturbing the others.

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed system follows a layered design, shown in Fig. 1. A user supplies an image at the top, and the image moves downwards through four layers: the input layer, the YOLO model, the recommendation layer and the results display. The first two layers together form the processing layer, the recommendation layer holds the logic and the product data, and the results display is the user interface layer. Table I summarises what each module receives and what it hands over to the next one.

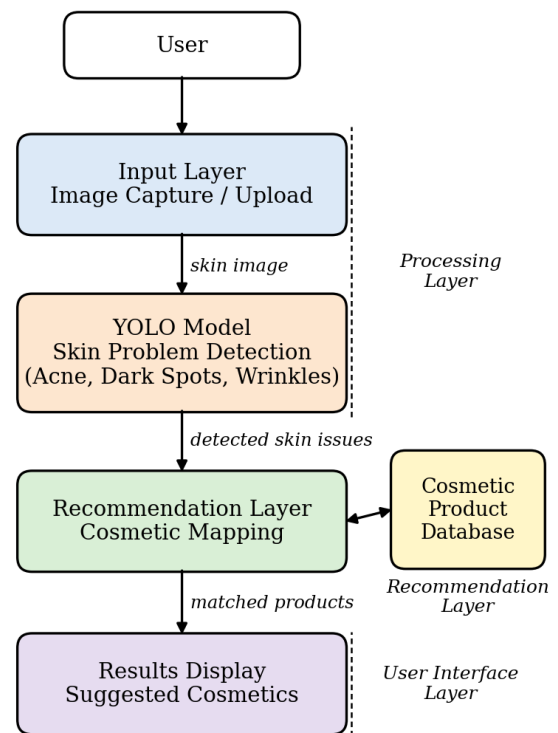


Fig. 1. Layered system architecture of the proposed skin classification and cosmetics suggestion system.

A. Input Layer

The input layer is the entry point of the system. It lets the user either capture a photograph with the device camera or upload an existing image from storage. The layer only collects the image and passes it on. Keeping this job small has a practical benefit, since the same layer can later be connected to a web page, a mobile application or a kiosk camera without changing anything that comes after it.

B. YOLO Model for Skin Problem Detection

The second layer is the heart of the system. The YOLO model divides the incoming image into a grid and, in one forward pass, predicts bounding boxes together with confidence scores and class probabilities for each cell [4]. In our system the classes correspond to the visible facial skin problems, namely acne, dark spots and wrinkles. Each box tells the system where on the face the problem is, and the class label tells it what the problem is. Overlapping boxes for the same region are reduced to one by non-maximum suppression, so that a single spot is not reported many times.

A single-stage detector suits this application for two reasons. The first is speed, because the user expects the answer within a short time after pressing the button. The second is that a detector gives both the location and the type of the problem, which a plain image classifier cannot do. The output of this layer is the list of detected skin issues, and this list is the only information that the recommendation layer needs.

C. Recommendation Layer

The recommendation layer performs cosmetic mapping. It reads the list of detected issues and consults the cosmetic product database, which holds the products together with the skin problems for which each product is meant. The recommendation logic then selects the entries that match the detected issues. When more than one problem is found in the same image, the logic collects the products for every one of them, so that the user receives a combined suggestion and not just a single product. Since the product knowledge is

kept in a database and not inside the neural network, new products can be added or old ones removed without retraining the detector.

D. Results Display

The last layer shows the suggested cosmetics to the user. It presents the matched products in a readable format, so that the user can see which skin problem was detected and which cosmetics are proposed for it. This layer works as the user interface of the system and returns the final answer to the person who gave the image.

TABLE I. Summary of the modules in the proposed system

Module	Input	Processing	Output
Input layer	Camera image or uploaded photograph	Image acquisition and hand-over to detector	Skin image
YOLO model	Skin image	Detection and classification of skin problems	Detected skin issues with class labels
Recommendation layer	Detected skin issues	Mapping of issues to entries of the cosmetic product database	Matched cosmetic products
Results display	Matched cosmetic products	Formatting for the user interface	Suggested cosmetics shown to the user

V. DATA FLOW DESIGN

The movement of data through the system is described with data flow diagrams (DFD). We use two levels. The Level 0 diagram treats the whole system as a single process and shows only its relation with the outside world: the user gives an image to the system, and the system returns the recommended output.

The Level 1 diagram opens this single process into its four sub-processes and is shown in Fig. 2. The user supplies a skin image to Process 1.0, image capture or upload. The image data goes to Process 2.0, skin problem detection, which runs the YOLO model. The analysis results are passed to Process 3.0, the recommendation system, which produces the recommended products. These are handed to Process 4.0, cosmetic suggestions, and finally the suggestions are displayed to the user. In short, the flow is user, image input, YOLO detection, classification, recommendation engine, cosmetic suggestions and user interface.

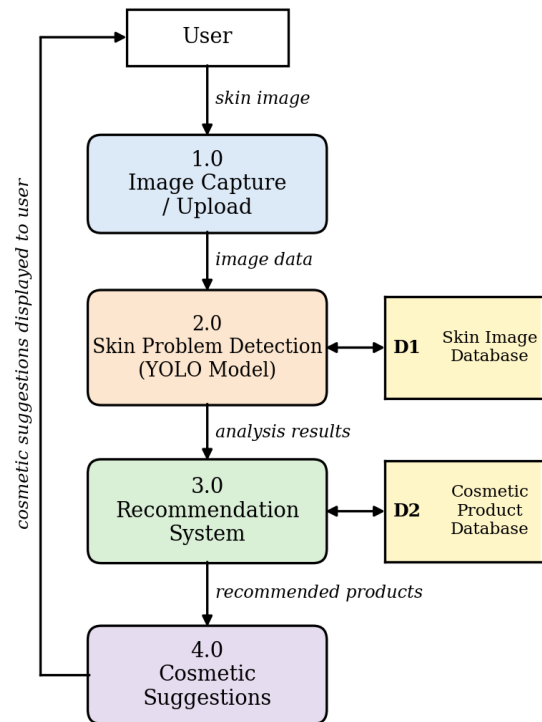


Fig. 2. Level 1 data flow diagram showing processes 1.0 to 4.0 and data stores D1 and D2.

Two data stores support the processes. The data store D1 is the skin image database. It is connected to the detection process and holds the skin images that the model works with. The data store D2 is the cosmetic product database. It is connected to the recommendation process, which reads the product data from it while preparing the suggestions. Both connections are drawn as two-way flows in Fig. 2, because a process can read from a store and can also write to it when the database is updated.

It is worth noting that the data flow design and the layered architecture describe the same system from two different points of view. The architecture in Fig. 1 explains what the building blocks are, and the DFD in Fig. 2 explains what travels between them. Reading both figures together makes it clear that the only data that crosses from the detector to the recommender is the list of detected skin issues.

VI. IMPLEMENTATION APPROACH

A. Data Preparation

The detector needs images in which every visible skin problem is marked with a bounding box and a class label. We collect face images that show the problems of interest and annotate them in the format that is accepted by the YOLO training pipeline. The annotated set is divided into training, validation and test parts so that the model is always judged on images that it has not seen during learning. Standard benchmark sets such as MS COCO (Microsoft Common Objects in Context) are useful for pre-training the backbone [15], after which the network is fine-tuned on the skin data.

B. Model Training

Training uses the annotated skin images to adjust the weights of the network so that the predicted boxes move towards the marked boxes and the predicted classes towards the true classes. Image processing operations from the OpenCV (Open Source Computer Vision) library [16] are used for reading, resizing and, where needed, augmenting

the images. Augmentation by flipping, brightness change and small rotation increases the variety of the data and reduces overfitting. The resulting weights are saved and loaded by the detection process at run time.

C. Inference and Recommendation

At run time the user image is resized to the input size of the network and passed through it. The detections whose confidence is below a chosen threshold are dropped, and non-maximum suppression removes duplicate boxes. The remaining class labels form the list of detected skin issues. This list is given to the recommendation logic, which queries the cosmetic product database for each issue and returns the matched products to the results display.

D. Software Tools

The system is written in Python, because the deep learning libraries and the image processing tools are all available in that language. The YOLO implementation can be taken from an open source project such as Ultralytics [17], and the product database can be kept in any simple relational store or even a structured file in a small deployment.

VII. PERFORMANCE EVALUATION MEASURES

A detector of this kind is judged by how well its boxes agree with the marked ones and by how many problems it finds or misses. The usual measures are precision, recall and the F1 score, which are calculated from the number of true positives (TP), false positives (FP) and false negatives (FN).

$$\text{Precision} = TP / (TP + FP) \quad (1)$$

$$\text{Recall} = TP / (TP + FN) \quad (2)$$

$$F1 = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (3)$$

A predicted box is counted as correct when its intersection over union (IoU) with the marked box is above a chosen limit. The mean average precision (mAP) is then obtained by averaging the area under the precision-recall curve over all classes [15]. These values are measured on the test part of the data. For the recommendation layer, the useful check is whether every detected issue returns at least one suitable product from the database, and whether the products shown to the user are consistent with the issues that were found. The system should also be tested with images taken under different lighting and with different skin tones, because the quality of a face image changes a good deal in real use.

VIII. ADVANTAGES AND LIMITATIONS

The main advantage of the proposed system is that it is simple and quick. A user needs only a camera, and the answer comes in a short time. The modular layers make the system easy to maintain, since the detector, the product database and the interface can be changed one at a time. The product knowledge sits outside the neural network, so the catalogue can be updated every week if required.

The system also has limits, and we state them plainly. It looks only at visible features of the skin in a photograph and cannot take the place of a clinical examination. Its quality depends on the variety and the labelling quality of the training images, and poor lighting or a blurred photograph can reduce the detection quality. The recommendations are only as good as the product database behind them. For these reasons the output should be treated as guidance and not as medical advice.

IX. CONCLUSION AND FUTURE SCOPE

In this paper we have described a skin classification and cosmetics suggestion system that joins a YOLO based skin

problem detector with a recommendation layer. The layered architecture and the Level 1 data flow design show clearly how a face image becomes a list of detected skin issues and then a list of suggested cosmetics. The two data stores keep the images and the products separate from the detection model, which makes the system easy to extend.

There is scope for several improvements. The set of skin problems can be enlarged with more classes, such as pores, redness and dark circles. The recommendation logic can use the skin type, the age and the allergy information given by the user, in addition to the detected problem. A lighter model can be deployed on a mobile phone so that the analysis runs on the device itself. Finally, a feedback option where users rate the suggestions would help to improve the recommendation step over time.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, SJM Institute of Technology, Chitradurga, for the facilities and the encouragement given during this project work.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1097–1105.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [3] A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 7263–7271.
- [7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv:1804.02767*, 2018.
- [8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, 2020.
- [9] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2023, pp. 7464–7475.
- [10] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.
- [11] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
- [12] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
- [14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
- [15] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 740–755.
- [16] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–125, 2000.
- [17] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>