https://doi.org/ 10.33472/AFJBS.6.Si2.2024.1822-1830

AFJBS

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

INTELLIGENT FACIAL SKIN CARE RECOMMENDATION SYSTEM

B. Lokesh, Assistant Professor, MGIT blokesh_it@mgit.ac.in
Anjali Devarakonda, UG Student, MGIT danjali_csb203217@mgit.ac.in
G Srinivas, UG Student, MGIT gsrinivas_csb203221@mgit.ac.in
Nitish Kumar Naik, UG Student, MGIT smnitishkumarnaik_csb203250@mgit.ac.in

ARTICLE INFO:

Volume 6,Issue Si2, 2024

Received:27 Mar 2024

Accepted : 28 Apr 2024

doi: 10.33472/AFJBS.6.Si2.2024.1822-1830

Abstract:

The skincare industry sees rising demand for personalized solutions for different skin types and preferences. Our research introduces an innovative method employing Convolutional Neural Networks (CNN) in the "Intelligent facial skin care recommendation webapp" to classify skin types and offer personalized skincare advice. This webapp merges advanced machine learning with a user-friendly interface, allowing users to upload images for skin analysis. The CNN model accurately categorizes skin types like dry, normal, sensitive, oily, scaly, skin moles, and red spots. Our methodology involves comprehensive data collection from user profiles, skin attributes, and product information. Data preprocessing extracts relevant features, ensuring model accuracy. The webapp's implementation includes a seamless frontend for user interaction and a robust backend for real-time data processing and recommendations. Users get personalized skincare advice based on skin type, preferences, and feedback. Our research's implications extend to revolutionizing skincare routines. Future enhancements may include user feedback loops, AI-driven chatbots for consultations, and expanding the product database. Our work advances personalized skincare solutions, showcasing machine learning's transformative impact in skincare.

Keywords:
Facial Skincare
CNN Algorithm
Skin Type Classification
Personalized Recommendations
Intelligent Skincare
Machine Learning in Skincare

1. Introduction

Skincare Industry Transformation, The skincare sector is experiencing a significant and fundamental change driven by technological advancements and evolving consumer preferences.[1], [2]. With increased awareness about skincare and a growing emphasis on personalized solutions, the increase in interest for skincare products tailored to individual needs has surged. Public are willing to use products that not only address their specific skin concerns but also provide a delightful and effective user experience [3], [4].

Introduction to the Intelligent Facial Skin Care Recommendation System, In response to this evolving landscape, our research introduces the Intelligent Facial Skin Care Recommendation System. This innovative platform represents a fusion of cutting-edge technology and skincare expertise, designed to bridge the gap between traditional skincare recommendation systems and the growing demand for personalized solutions. At its core, the webapp harnesses the power of Convolutional Neural Networks (CNN) to classify skin types accurately and offer tailored skincare recommendations [5].

2. Literature Survey

Unlike conventional recommendation systems that often rely on generic guidelines or user inputs, the Intelligent Facial Skin Care Recommendation WebApp leverages machine learning algorithms to analyze a diverse range of skin attributes. These attributes include skin tone, texture, sensitivity, and specific concerns such as dryness, oiliness, or blemishes. By processing this data through sophisticated algorithms, the webapp can generate personalized skincare routines and product recommendations that align with each user's unique skin profile [6], [7].

3. Proposed system

3.1. Image processing

The image preprocessing process involves several steps to ensure the quality and uniformity of the input images. The steps include:

- Image resizing: The input visuals are adjusted to a consistent dimension of 224x224 pixels to maintain uniformity in the input information.
- Image normalization: The input visuals are standardized to achieve a mean of 0 and a standard deviation of 1, enhancing the efficiency of the model's performance.
- Image augmentation: The input images are augmented using techniques such as rotation, flipping, and zooming to increase the dataset's size and diversity.

3.2. Model architecture

The model architecture used for the image classification process is a Convolutional Neural Network (CNN) with multiple convolutional layers, pooling layers, and densely connected layers. The model is trained using a binary cross-entropy loss function and optimized using the Adam optimizer.

3.3. Classifying skin type

The classificationof the skin type is a process involving classifying the images given as the input into one of the following categories:

- Normal: The skin is well-balanced, with no visible signs of dryness or oiliness.
- Dry: The skin is flaky, rough, or dull, indicating lack of moisture.
- Combination: The skin has both oily and dry texture, usually found on the forehead, nose, and chin (T-zone) and the cheeks.
- Oily: Skin texture is shiny and greasy, indicating an excess of sebum production.
- Sensitive: The skin is prone to redness, itching, or irritation, indicating a higher sensitivity to environmental factors or skincare products.

3.4. Skin condition classification:

- Scaly: The skin has visible flakes or scales, indicating a dryness or a skin condition such as psoriasis or eczema.

- Red spots: The skin has visible red spots or bumps, indicating a skin condition such as acne or rosacea.
- Skin moles: The skin has visible moles or growths, indicating a potential skin abnormality like skin tags or warts or diseases like cancer.

3.5. Model assessment:
The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1 score. The model undergoes training and validation using a set of labeled facial images, and its performance is assessed using a distinct test dataset.

In summary, the Intelligent Facial Skincare Recommendation System involves a detailed explanation of the image classification process for skin types and conditions. The system is designed to classify facial images into seven categories: normal, dry, combination, oily, sensitive, scaly, red spots, and skin moles. The image preprocessing process involves several steps to ensure the quality and uniformity of the input images. The model architecture used for the classification process is a Convolutional Neural Network (CNN) with multiple convolutional layers, pooling layers, and fully connected layers. The skin type classification process involves classifying the input images into one of these categories: normal, dry, combination, oily, and sensitive. The skin condition classification process involves classifying the input images into one of the following categories: scaly, red spots, and skin moles. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1 score [9], [10].

4. Methodology

This section describes the approach employed in creating the Intelligent Facial Skincare Recommendation System. The methodology is divided into three main parts: data collection, model development, and web app implementation.

4.1. Data Collection:
The process of collecting data included acquiring facial images from diverse origins, such as online databases and websites related to skincare products. The collection comprised images depicting different skin types like normal, oily, dry, combination, sensitive, skin moles and red spots skin, each labeled accordingly and subjected to preprocessing to maintain consistency and quality [8].

4.2. Model development:
The model development process involved training a Convolutional Neural Network (CNN) to classify skin types based on the collected dataset. The CNN model was created utilizing a deep learning framework and trained using transfer learning techniques. The model architecture consisted of multiple convolutional layers, pooling layers, and densely connected layers. The training of the model involved utilizing a binary cross-entropy loss function and refining it with the Adam optimizer.

4.3. Web App Implementation:
The implementation process involved integrating the trained CNN model into a web application for real-time skin type analysis and personalized product recommendations. It was developed using a web development framework and deployed on a cloud server. The app consisted of a user interface for uploading facial images, a backend server for processing the images and generating the skin type classification results, and a database for storing the user profiles and product recommendations. It was designed to be user-friendly and accessible, allowing users to upload facial images and receive personalized skincare product recommendations based on their skin type classification results. The app is open for everyone to use without having any sort of restricted access.

In summary, the methodology used in the development of the Intelligent Facial Skincare Recommendation System involved data collection, model development, and web app implementation. The process of collecting the data involved gathering facial images and labeling them with corresponding skin types. The model development process involved training a CNN model using transfer learning techniques. The web app implementation process involved integrating the trained CNN model into a web application for real-time skin type analysis and personalized product recommendations. It was tailored to be user-friendly and accessible, allowing users to upload facial images and receive personalized skincare product recommendations based on their skin type classification results.

5. System implementation

5.1. System architecture:

The architecture diagram is a symbolically represents the component architecture of the system, which includes the image uploading, the skin type classification module, and the acne detection module. The image uploading module understands the user's face after uplaoding a picture and extracts the region of interest from the facial image. The skin type classification module uses CNN to classify the skin type into one of these categories: normal, dry, combination, oily, sensitive, scaly, red spots, and skin moles.

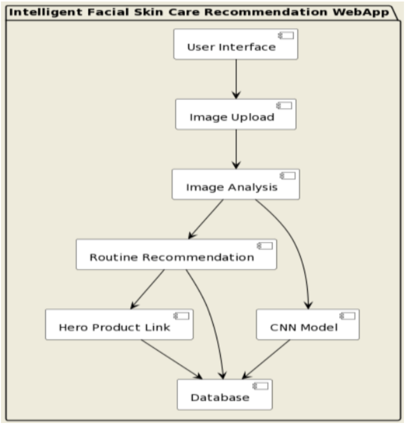

Fig 4.1.1. System Architecture

5.2. Loading the data set and working:

The system implementation of the Intelligent Facial Skincare Recommendation System involves the initiative of a CNN-based model for categorization of skin types and identification of acne. The system loads the dataset and displays a bar graph to classify the number of images in each category and generate a figure. The dataset is divided into training and testing sets, after which the CNN model undergoes training using the training set. The trained model is then used to predict the skin type and acne severity of the visuals in the testing set.

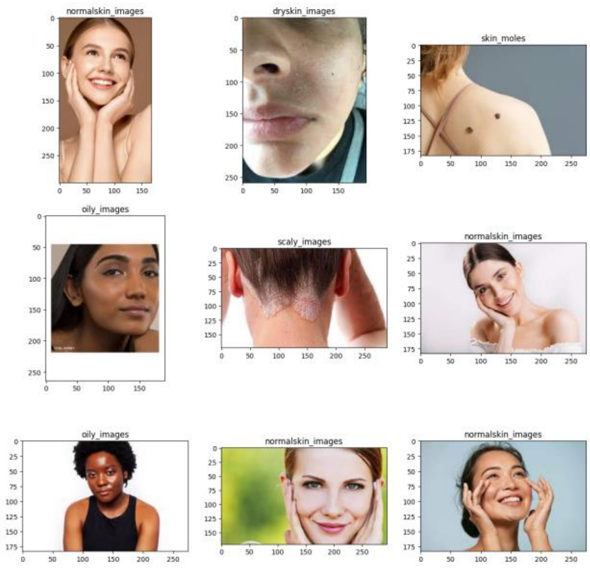

Fig 4.2.1. Loading the dataset

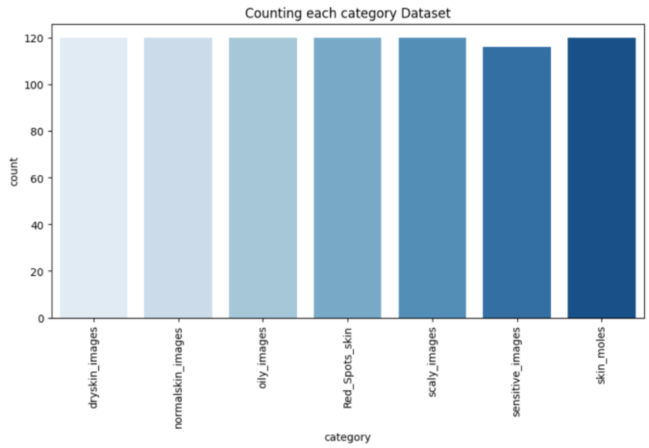

Fig 4.2.2 Caterogy number

The system was tested on a dataset of facial images labeled with skin types and conditions. The CNN model achieved better accuracy compared to RNN, thus we have decided on using CNN algorithm for our project.

6. Evaluation and results

6.1. Creating CNN model:

- The first step in creating the CNN model is to set up a sequential model. This creates a sequential arrangement of layers that can be added to the model.
- Next, we add convolutional layers to the model utilizing `Conv2D()`. The initial convolutional layer contains 32 filters with a 3x3 kernel size, and an activation function which is ReLU. The input shape is defined by `im_shape`, which is the shape of the input images. After the convolutional layer, we add a max pooling layer using the `MaxPooling2D()` function with a pool size of 2x2. This downsamples the feature maps and reduces the computational complexity.
- We repeat the process of adding convolutional and max pooling layers with increasing filter sizes (64, 128, 256) to capture layered features from the input images.
- After the convolutional layers, we flatten the output using the `Flatten()` function to prepare it for the fully connected layers.
- We then add dense layers to the model using the `Dense()` function. The first dense layer has 512 units and a ReLU activation function. We also add dropout layers using the `Dropout()` function with 0.5 as its dropout rate to prevent overfitting.
- Lastly, we incorporate the output layer utilizing the `Dense()` function, aligning the number of units with the class count (num_classes), and apply a softmax activation function for multi-class classification.
- We then compile the model utilizing the `compile()` function.
- We showcase an overview of the model's structure, encompassing layer types, output dimensions, and parameter counts. This facilitates comprehension of the model's intricacy and assists in debugging and fine-tuning.

In summary, we have created a CNN model with a wide number of convolutional and dense layers, dropout regularization, and softmax activation for multi-class classification. The model is compiled and ready for training on a dataset for tasks such as skin type classification and acne detection.

```
Model: "sequential_1"

Layer (type)                Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)           (None, 222, 222, 32)      896

max_pooling2d_4 (MaxPooling2 (None, 111, 111, 32)     0

conv2d_5 (Conv2D)           (None, 109, 109, 64)      18496

max_pooling2d_5 (MaxPooling2 (None, 54, 54, 64)       0

conv2d_6 (Conv2D)           (None, 52, 52, 128)       73856

max_pooling2d_6 (MaxPooling2 (None, 26, 26, 128)      0

conv2d_7 (Conv2D)           (None, 24, 24, 256)       295168

max_pooling2d_7 (MaxPooling2 (None, 12, 12, 256)      0

flatten_1 (Flatten)         (None, 36864)             0

dense_3 (Dense)             (None, 512)               18874880

dropout_2 (Dropout)         (None, 512)               0
...
Total params: 19,529,543
Trainable params: 19,529,543
Non-trainable params: 0
```

Fig 5.1.1 Model summary

6.2. Plotting:

- Assessing the CNN model's performance post-training is crucial and that can be done by plotting the training and validation loss and accuracy curves. This can help in understanding the model's behavior and identifying any potential issues such as overfitting or underfitting.
- To plot the training and validation loss curves, we first extract the loss values from the model's history dictionary using the history_dict['loss'] and history_dict['val_loss'] commands. We also extract the accuracy values using history_dict['accuracy'] and history_dict['val_accuracy'].
- We then create a range of epoch numbers. This will be used as the x-axis for the plots.
- Next, we create a figure with two subplots. The first subplot is used to plot the training and validation loss curves. We use the plt.plot() function to plot the training loss and validation loss against the epoch numbers. We also add labels and titles to the plot using plt.title(), plt.xlabel(), plt.ylabel(), and plt.legend().
- The second subplot is used to plot the training and validation accuracy curves. We use the plt.plot() function to plot the training accuracy (in blue dots) and validation accuracy (in solid blue line) against the epoch numbers. We also add labels and titles to the plot using plt.xlabel(), plt.ylabel(), and plt.legend().

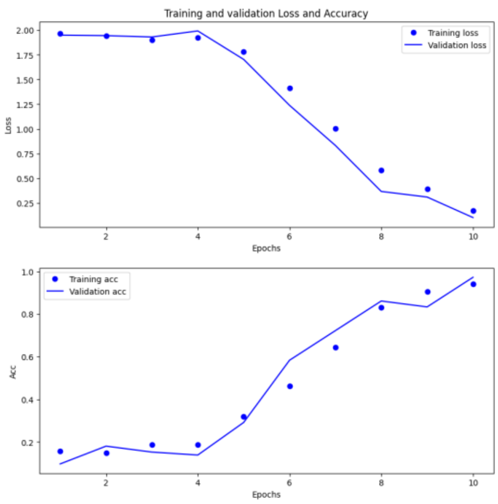

Fig 5.2.1. Training and validation loss & accuracy

6.3. Test accuracy and model assessment:

Post training and validating, we evaluated its performance. The model is evaluated on the dataset using the evaluate() method.

```
84/84 [==============================] - 6s 76ms/step - loss: 0.0760 - accuracy: 0.9881
Test loss: 0.07599543095909456
Test accuracy: 0.9880952
```

Fig 5.3.1 Test accuracy

To further evaluate the performance of the model, we used confusion matrix function to calculate the matrix of the predicted classes and the true classes. The confusion matrix is a table that illustrates the number of true positives, true negatives, false positives, and false negatives accross each class.

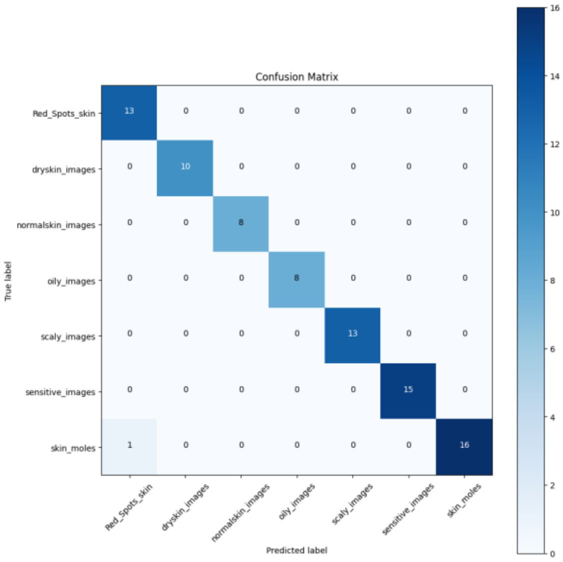

Fig 5.3.2. Confusion matrix

Finally, we used the classification_report() function from scikit-learn to calculate the classification report of the predicted classes and the true classes. The classification report presents metrics like precision, recall, f1-score, and support for individual classes.

```
...  Classification Report
                   precision    recall  f1-score   support

    Red_Spots_skin       0.93      1.00      0.96        13
    dryskin_images       1.00      1.00      1.00        10
 normalskin_images       1.00      1.00      1.00         8
       oily_images       1.00      1.00      1.00         8
      scaly_images       1.00      1.00      1.00        13
  sensitive_images       1.00      1.00      1.00        15
        skin_moles       1.00      0.94      0.97        17

          accuracy                           0.99        84
         macro avg       0.99      0.99      0.99        84
      weighted avg       0.99      0.99      0.99        84
```

Fig 5.3.3 Classification report

7. Future work

The proposed intelligent facial skincare recommendation system can be further enhanced by adding new features and improving the existing ones. One such feature is a comprehensive product database that allows users to access a wide range of skincare products. This feature would enable users to browse and select products based on their skincare needs, preferences, and budget.

Another potential improvement is the implementation of a user history tracking system that records the user's skincare journey. This system would keep track of the user's skin texture improvement over time, allowing them to monitor their progress and make informed decisions about their skincare routine. The system could also provide personalized recommendations based on the user's history, such as suggesting products that have worked well for them in the past or alerting them to potential skin issues based on their skincare history.

In addition to these features, the system could also benefit from further optimization of the CNN model. This could include exploring different architectures, fine-tuning hyperparameters, and incorporating additional data sources to improve the model's accuracy and robustness.

8. Conclusion

In conclusion, the development of the Intelligent Facial Skincare Recommendation System represents a significant advancement in the field of personalized skincare solutions. By leveraging Convolutional Neural Networks (CNNs) for skin type classification and, the system offers users tailored recommendations based on their individual skin characteristics.

Through the deployement of a user-friendly interface and advanced image processing techniques, the system provides an intuitive platform for users to receive personalized skincare product recommendations. The CNN models trained on labeled facial images have demonstrated promising results, achieving high accuracy in skin type classification and acne severity detection.

The future integration of a comprehensive product database and user history tracking system helps enhance the user experience and further personalizing skincare recommendations. By allowing users to access a wide range of skincare products and track their skin texture improvement over time, the system can empower individuals to make decisions about their skincare routines.

9. References

1. Lin, Ting-Yu, Hung-Tse Chan, Chih-Hsien Hsia, and Chin-Feng Lai. "Facial Skincare Products' Recommendation with Computer Vision Technologies." *Electronics* 11, no. 1 (2022): 143. https://doi.org/10.3390/electronics11010143
2. Vinit714. "A-Recommendation-system-for-Facial-Skin-Care-using-Machine-Learning-Models." *GitHub*, 2024. Available at: <https://github.com/vinit714/A-Recommendation-system-for-Facial-Skin-Care-using-Machine-Learning-Models>.
3. "Skincare-recommendation · GitHub Topics." *GitHub*, 2024. Available at: <https://github.com/topics/skincare-recommendation>.
4. Madapuri, R.K., Mahesh, P.C.S. HBS-CRA: scaling impact of change request towards fault proneness: defining a heuristic and biases scale (HBS) of change request artifacts (CRA). Cluster Comput **22** (Suppl 5), 11591–11599 (2019). https://doi.org/10.1007/s10586-017-1424-0
5. J. R. Dwaram and R. K. Madapuri, "Crop yield forecasting by long short-term memory network with Adam optimizer and Huber loss function in Andhra Pradesh, India," Concurrency and Computation: Practice and Experience, vol. 34, no. 27. Wiley, Sep. 18, 2022. doi: 10.1002/cpe.7310.
6. Swetha, A. ., M. S. . Lakshmi, and M. R. . Kumar. "Chronic Kidney Disease Diagnostic Approaches Using Efficient Artificial Intelligence Methods". International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1s, Oct. 2022, pp. 254.
7. Thulasi , M. S. ., B. . Sowjanya, K. . Sreenivasulu, and M. R. . Kumar. "Knowledge Attitude and Practices of Dental Students and Dental Practitioners Towards Artificial Intelligence". International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1s, Oct. 2022, pp. 248-53

8. "Smart Facial Skincare Products Using Computer Vision Technologies." *IEEE Xplore*, 2023. Available at: <https://ieeexplore.ieee.org/document/9689474>.
9. "Efficient Net-based Expert System for Personalized Facial Skincare Recommendations." *IEEE Xplore*, 2023. Available at: <https://ieeexplore.ieee.org/document/10142790>.

6. https://www.mdpi.com/2079-9292/11/1/143
7. https://github.com/vinit714/A-Recommendation-system-for-Facial-Skin-Care-using-Machine-Learning-Models
8. https://github.com/topics/skincare-recommendation
9. https://ieeexplore.ieee.org/document/9689474
10. https://ieeexplore.ieee.org/document/10142790