Major Project Report

On

Skin Lesion Classification for Disease Detection

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence & Machine Learning)

by

20WH1A6603 B.Tanmayee
20WH1A6604 P.Abheesta
20WH1A6605 G.Spurthy Vahini
20WH1A6624 M.Devi Sri Chandana
21WH5A6605 M.Ashwini

Under the esteemed guidance of

B.Kishore Kumar
Assistant Professor, CSE (AI&ML)

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with A Grade
Bachupally, Hyderabad – 500090

2023-24

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with A Grade
Bachupally, Hyderabad – 500090
2023-24

CERTIFICATE

This is to certify that the project entitled "Skin Lesion Classification for Disease Detection" is a bonafide work carried out by **Ms. B. Tanmayee (20WH1A6603), Ms. P. Abheesta(20WH1A6604), Ms. G. Spurthy Vahini (20WH1A6605), Ms. M. Devi Sri Chandana(20WH1A6624), Ms. M. Ashwini(21WH5A6605)** in partial fulfillment for the award of B.Tech degree in **Computer Science & Engineering (AI&ML)** , **BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**,affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision.

Supervisor **Head of the Department**

External Examiner

DECLARATION

We hereby declare that the work presented in this project entitled **"Skin lesion classification for disease detection"** submitted towards completion of Project work in IV Year of B.Tech of CSE(AI&ML) at **BVRIT HYDERABAD College of Engineering for Women,** Hyderabad is an authentic record of our original work carried out under the guidance of **Mr. B. Kishore Kumar, Asst. Prof, Dept of CSE(AI&ML)**

Sign with Date:
B.Tanmayee
(20WH1A6603)

Sign with Date:
P.Abheesta
(20WH1A6604)

Sign with Date:
G.Spurthy Vahini
(20WH1A6605)

Sign with Date:
M.Devi Sri Chandana
(20WH1A6624)

Sign with Date:
M.Ashwini
(21WH5A6605)

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr.K.V.N.Sunitha**, **Principal, BVRIT HYDERABAD College of Engineering for Women,** for her support by providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr B Lakshmi Praveena, Head of the Department, Department of CSE(AI&ML) , BVRIT HYDERABAD College of Engineering for Women,** for all timely support and valuable suggestions during the period of our project.

We are extremely thankful to our Internal Guide, **Mr.B.Kishore Kumar, Asst Professor, CSE(AI&ML), BVRIT HYDERABAD College of Engineering for Women,** for his constant guidance and encouragement throughout the project.

Finally, We would like to thank our Major Project Coordinator, all Faculty and Staff of CSE department who helped us directly or indirectly. Last but not least, we wish to acknowledge our **Parents** and **Friends** for giving moral strength and constant encouragement.

B. Tanmayee(20WH1A6603)

P. Abheesta(20WH1A6604)

G. Spurthy Vahini(20WH1A6605)

M. Devi Sri Chandana(20WH1A6624)

M. Ashwini(21WH5A6605)

ABSTRACT

Skin cancer, encompassing various types, presents a diagnostic challenge due to diverse morphological features. Timely identification of skin lesions is crucial for effective medical intervention. Traditional methods, reliant on visual inspection and invasive procedures, may result in delayed or inaccurate diagnoses. This project proposes a solution using deep learning, specifically Convolutional Neural Networks (CNNs), to automate skin lesion classification. Leveraging diverse datasets and transfer learning, the model learns intricate patterns, facilitating accurate disease identification. The objective is to provide a scalable, objective, and accessible tool, potentially reducing dependence on subjective interpretations and enabling early intervention for dermatological conditions.

LIST OF FIGURES

LIST OF CONTENTS

1.INTRODUCTION AND BACKGROUND

A skin lesion is a growth or appearance of the skin that is abnormal concerning the surrounding skin. Primary and secondary skin lesions are the two types of skin lesions. Primary skin lesions are abnormal skin conditions that can develop over time or be present at birth. Secondary skin lesions can develop from primary skin lesions that have been exacerbated or altered. When a mole is scraped until it bleeds, the crust that forms, as a result, develops a secondary skin lesion. Dermatologists propose one of three treatments for afflicted skin, depending on the type of lesion: home care, medicines, or surgery. Regardless of ways innocent they appear; a few sorts of skin lesions may be pretty risky to the patients, since they will indicate the presence of malignancy and require surgical removal. Melanoma is the most dangerous type of skin cancer; as soon as it has spread, it's deadly, however, it is treatable in its early stages. As a result, a precise diagnosis of skin patches is essential to protect patients' growths and ensure that they receive timely treatment.

Machine Learning methods could be used to automate the analysis, resulting in a system and framework in the medical field that would aid in providing contextual relevance, improving clinical reliability, assisting physicians in communicating objectively, reducing errors related to human fatigue, lowering mortality rates, lowering medical costs, and more easily identifying diseases. A machine learning method that can categorize both malignant and benign pigmented skin lesions is a step toward achieving these goals

1.1 Problem statement:

The manual interpretation of skin lesions poses significant challenges in terms of accuracy, timeliness, and accessibility to specialized care. Dermatologists, while possessing expertise, may face difficulties in differentiating between visually similar benign and malignant lesions. Furthermore, the subjectivity in visual assessments and the limitations in accessing dermatological expertise can lead to delayed diagnoses, impacting patient outcomes. Addressing these challenges requires an innovative approach that combines advanced technologies, such as deep learning, to enhance the efficiency and reliability of skin lesion classification.

1.2 Objectives:

1. Develop a deep learning model, specifically a Convolutional Neural Network (CNN), for the automated and accurate classification of skin lesions.

2. Implement transfer learning techniques to expedite the training process and optimize the deep learning

model's performance in accurately identifying and classifying skin lesions.

3. Curate a diverse and comprehensive dataset of skin lesion images, encompassing various dermatological conditions, to train and validate the deep learning model.

4. Enhance the interpretability of the model by incorporating attention mechanisms or Grad-CAM to highlight regions of interest within the images that contribute most to the classification decision.

5.Conduct comparative analyses with existing state-of-the-art models and validate the developed system on an independent test set to demonstrate its effectiveness in real-world scenarios.

1.3 Dataset

The HAM10000 dataset which consists of 10015 images has been used in the proposed work.The HAM10000 dataset is a vast collection of dermoscopic images of pigmented skin lesions which are very common from multiple sources. Datasets with significant class imbalances are fairly common in the medical industry. It is the same with this data set. In the proposed work, it proved to be a significant challenge. The dataset images have a resolution of 600 × 450 pixels and are saved as JPEG formats.

They are manually cropped and cantered around the lesion, as well as modified for visual contrast and color reproduction, at first. Each image and patient had seven features, namely, age of the patient, sex of the patient, lesion id which is a unique identifier for a particular type of lesion, image id which is a unique identification number for an image, dx type for technical validation, Skin lesion's geographical location, and a diagnostic skin lesion category which is a classification of skin lesions that can be used to diagnose a condition.

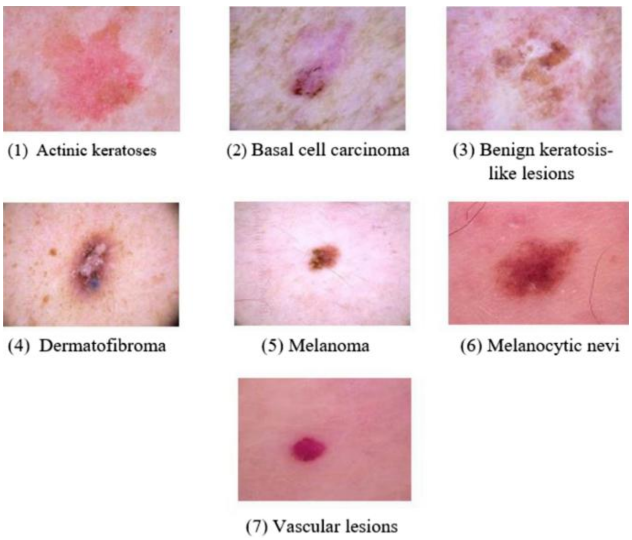

Figure 1.1: Dataset

1.4 Background

Skin lesion classification has emerged as a pivotal field, blending dermatology with artificial intelligence (AI) to address the escalating challenges in diagnosing skin disorders accurately. Traditional diagnostic approaches heavily rely on visual assessment by dermatologists, but the integration of digital imaging and machine learning, especially deep learning, has ushered in a new era. These advancements have led to the creation of automated models proficient in discerning various types of skin lesions, distinguishing between benign and malignant cases. The synergy of medical expertise and computational capabilities holds considerable promise, offering improved efficiency and precision in dermatological diagnoses. This integration has the potential to significantly enhance healthcare outcomes and broaden access to dermatological care.

2.LITERATURE SURVEY

S.No	Title of the Paper	Author	Description
1	Method to Cassify skin lesions using dermoscopic images.	Dusa Sai Charan,Hemanth Nadipineni,Subin Sahayam,Umarani Jayaraman	The paper proposed different models where the output of the first model acted as the input to the other.All the models use different input strategies,preprocessing techniques and training techniques.
2	Skin Tone Detection and Debiasing for Skin Lesion Classification .	Peter J. Bevan and Amir Atapour-Abarghouei	This paper primarily focuses on using ResNeXt-101 for their experiments, and they also employ different debiasing techniques (such as 'Learning Not To Learn' and 'Turning a Blind Eye') alongside this feature extractor to mitigate skin tone bias in melanoma classification.The main output of the model is a binary classification result: either benign or malignant.
3	Skin Lesions Classification Using Convolutional Neural Networks in Clinical Images.	Danilo Barros Mendes and Nilton Correia da Silva.	This research paper presents a deep learning model, specifically a ResNet-152 architecture, for classifying skin lesions, including Malignant Melanoma and Basal Cell Carcinoma, using clinical images. The study addresses the critical need for early detection of skin diseases, given the global impact of such conditions. The authors leverage transfer learning to overcome data scarcity in the medical field and report promising results, with an area under the curve (AUC) of 0.96 for Melanoma and 0.91 for Basal Cell Carcinoma.

Figure 2.1: Literature survey

3.DATASET

1. Dataset Size:
   - The dataset consists of 10,015 dermatoscopic images.
   - Each image is labeled with one of seven diagnostic categories.

2. Categories:
   - The dataset includes images of seven different skin lesion categories, including melanoma, nevus, seborrheic keratosis, basal cell carcinoma, actinic keratosis, benign keratosis (solar lentigo/lichen planus-like keratosis), and vascular lesions.

3. Dermatoscopic Images:
   - The images are captured using dermatoscopes, which are devices that allow detailed examination of the skin lesions.
   - Dermatoscopy involves examining skin lesions using a specialized magnifying lens and light source to detect patterns not visible to the naked eye.

4. Clinical Relevance:
   - The dataset is significant for research in the field of dermatology and computer-aided diagnosis.
   - It provides a valuable resource for developing and testing machine learning models for the classification of skin lesions.

5. Data Availability:
   - The dataset is publicly available, making it accessible for researchers and developers interested in dermatology and medical image analysis.

6. Metadata:
   - Each image in the HAM10000 dataset is associated with metadata, including patient information, lesion type, and other relevant details.

7. Challenges and Competitions:
   - The dataset has been used in various machine learning competitions and challenges related to dermatology.
   - Competitions often focus on developing algorithms for accurate and early detection of skin cancer.

8. Research Impact:

- The dataset has contributed to numerous research papers and studies aimed at improving the understanding and diagnosis of skin diseases.

9. Normalization Challenges:

- Normalizing the dataset poses challenges due to variations in imaging conditions, lighting, and patient characteristics.

It's important to note that the field of medical datasets, including dermatology datasets, evolves over time. Yet this dataset seems to remain intact.

4. PROPOSED SYSTEM

Our proposed system revolutionizes disease detection through advanced medical imaging technologies. Integrating data augmentation, deep learning, and transfer learning, our model achieves superior accuracy by training on an extensive dataset. Optimized with cutting-edge techniques, the system features a user-friendly interface for efficient interaction, benefiting both medical professionals and patients.

Employ advanced data augmentation techniques to systematically expand the dataset, thereby facilitating comprehensive model training on a larger and more diverse set of examples.Leverage sophisticated deep learning architectures to improve disease detection accuracy and unravel intricate features inherent in medical imaging datasets. The application of deep learning models ensures a nuanced understanding of complex patterns within the data.

Integrate transfer learning methodologies by harnessing the pre-trained weights of established models, such as ResNet. This approach optimizes model performance and facilitates superior learning by leveraging the knowledge acquired during the pre-training phase on large-scale datasets.

Implement state-of-the-art optimization techniques to enhance the model's overall accuracy. Optimization strategies will be employed to fine-tune model parameters and improve convergence, ensuring the model achieves optimal performance on disease detection tasks.

Develop a user-friendly interface tailored for both medical professionals and patients. The interface design prioritizes ease of use, ensuring a seamless experience for doctors and patients alike. This user-centric approach enhances accessibility and facilitates efficient interaction with the disease detection system.

5. LIST OF MODULES

Modules play an important role in the flow of the project. System comprises several crucial modules designed to synergistically enhance disease detection capabilities in medical imaging.Together, these modules form a cohesive and innovative framework poised to significantly advance disease detection in medical diagnostics

List of Modules:

5.1 Data Collection

In the pursuit of advancing medical diagnostics, our project hinges on a meticulous data collection process sourced from various challenges specializing in medical image analysis. Our primary dataset, the HAM10000, has been carefully curated and encompasses a rich repository of 10,000 medical images. Each image within this dataset is accompanied by pertinent labels, crucial for our classification endeavors. This strategic approach to data collection not only underscores the project's commitment to leveraging diverse and high-quality datasets but also positions us at the forefront of innovative solutions in the realm of medical image diagnostics.

5.2 Data preprocessing

In the realm of medical image analysis, effective data preprocessing is paramount to ensure the quality and relevance of the dataset. Initially sourced from various challenges and curated meticulously, our dataset, notably the HAM10000 dataset with 10,000 images, undergoes a series of preprocessing steps to optimize its utility for subsequent tasks.

1. Train-Test Split:

The implementation of a train-test split involves partitioning the dataset into distinct training and testing subsets. This procedural division is essential for evaluating the model's performance on independent data, fostering robustness and generalization.

2. Class Balancing:

The endeavor to balance classes within the dataset is a strategic measure aimed at facilitating model training. By mitigating class imbalances, we aim to prevent the model from developing biases toward a particular class, thereby enhancing its capacity for unbiased and comprehensive learning.

3. Data Augmentation:

Data augmentation represents a pivotal technique involving the augmentation of the dataset through the introduction of artificially generated data derived from existing training samples. This practice serves to diversify the dataset, fortifying the model against overfitting and promoting heightened adaptability.

4. Image Resize, Normalization, and Label Encoding:

A meticulous preprocessing regimen involves resizing and normalizing images, ensuring uniform dimensions and standardized pixel values. Concurrently, label encoding is applied to represent categorical labels in a numerical format, streamlining subsequent model training and evaluation processes.

5.3 Deep learning Model Selection

The primary focus is on crafting a sophisticated deep learning model meticulously tailored to the intricate demands of image data processing. Employing advanced transfer learning techniques, the project strategically utilizes a pre-trained ResNet model, ensuring optimal adaptation to the targeted image-related task. Rigorous hyperparameter fine-tuning is executed with meticulous attention to detail, aiming to optimize the model's performance against the specified image dataset. This approach is geared towards achieving predefined objectives with a high degree of precision. The neural network architecture is systematically configured, entailing the discerning selection and arrangement of layers specifically curated for image data manipulation. This includes the strategic integration of convolutional and pooling layers, instrumental in extracting and processing pertinent features with a professional level of sophistication.

5.4 Integration with Explainability Module

Leveraging ResNet-152 architecture with integrated Class Activation Maps (CAM) significantly enhances the interpretability of the neural network's predictions. CAM provides insights into the network's focus during predictions, improving model interpretability. By highlighting influential image regions, CAM facilitates object localization, contributing to a more transparent understanding of the network's decision-making process in image classification tasks. This strategic integration of CAM into ResNet-152 not only bolsters model transparency but also aids in refining the network's performance for nuanced image analyses.

5.5 Performance Evaluation

Test Data Quality Assurance: Ensuring the reliability of test data is paramount. This involves meticulous attention to data quality, necessitating high-quality, accurately labeled, and representative test data. Furthermore, diversity within the test data is affirmed, encompassing a comprehensive range of skin types, lesion types, and demographics. This rigorous approach guarantees the robustness and generalizability of the model's performance.

Assessment of Low-Quality Images: Addressing low-quality images involves a comprehensive evaluation of image preprocessing techniques. The effectiveness of these techniques in enhancing image quality is systematically analyzed, ensuring that the model can accommodate and perform optimally even with less-than-ideal input conditions.

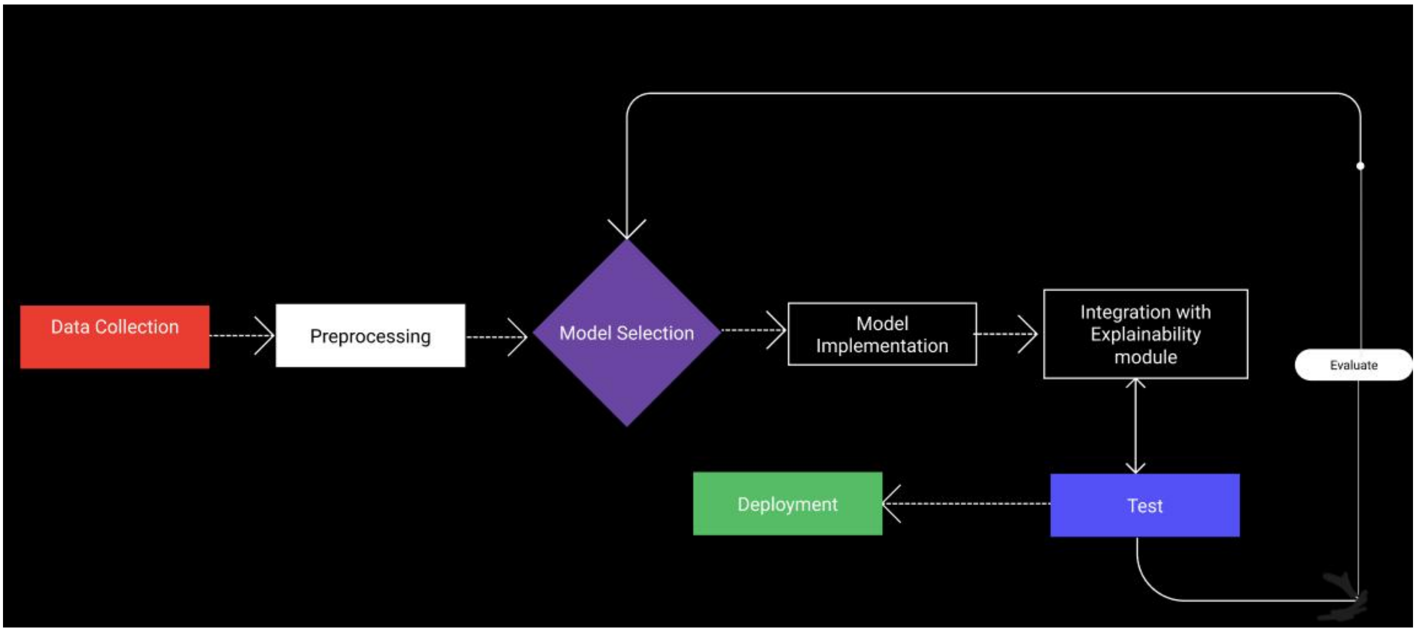

Figure 5.1 Modules

6. ARCHITECTURE

The proposed architecture consists of ResNet152 along with the integration of CAM.The architectural framework integrates ResNet-152 as its foundational backbone, comprising 152 layers. Commencing with an input layer and an initial convolution layer, it progresses through subsequent layers featuring MaxPool, convolution, and ReLU operations organized into residual blocks. Notably, this architecture is augmented with the incorporation of Class Activation Maps (CAM), strategically enhancing interpretability by revealing the network's focus during predictions. The final layer consists of Softmax, contributing to precise class predictions. This combined architecture synergizes the depth of ResNet with the interpretability afforded by CAM, establishing a comprehensive framework for robust and insightful image analysis.

It has 2 major components:

6.1. ResNet-152 Architecture:

ResNet-152 is employed as the foundational framework, featuring 152 layers. It begins with an input layer, followed by an initial convolution layer, MaxPool, and a sequence of convolutional and ReLU layers structured into residual blocks. ResNet-152 is a deep convolutional neural network architecture introduced in 2016 as part of the ResNet family. Developed by Kaiming He et al., ResNet-152 is known for its remarkable depth, featuring 152 layers. The key innovation lies in its residual learning blocks, which include skip connections to ease the training of deep networks by learning residual functions. This architecture has been particularly successful in image recognition tasks, demonstrating state-of-the-art performance in tasks like image classification, thanks to its ability to effectively handle very deep networks and mitigate issues like the vanishing gradient problem.

The key innovation in ResNet is the introduction of residual learning blocks. These blocks contain skip connections (or shortcuts) that allow the network to learn residual functions, making it easier to train very deep networks. The residual blocks help to mitigate the degradation problem, where the accuracy of a deep neural network saturates and then degrades as the network gets deeper.

In the case of ResNet-152, the architecture is built upon a series of residual blocks. It starts with a traditional convolutional layer, followed by multiple residual blocks organized into different stages. Each stage gradually reduces spatial dimensions while increasing the number of filters. The final global average pooling layer and fully connected layer produce the network's output.

The major characteristics of this architecture are:

- ResNet-152 comprises 152 layers in its architecture.
- The initial layers include the input layer followed by an initial convolution layer.
- A MaxPool layer follows the convolution layer.
- A series of convolutions with ReLU layers are structured into layers, implementing the concept of residual blocks.
- The final layer is the Softmax layer, facilitating class predictions.

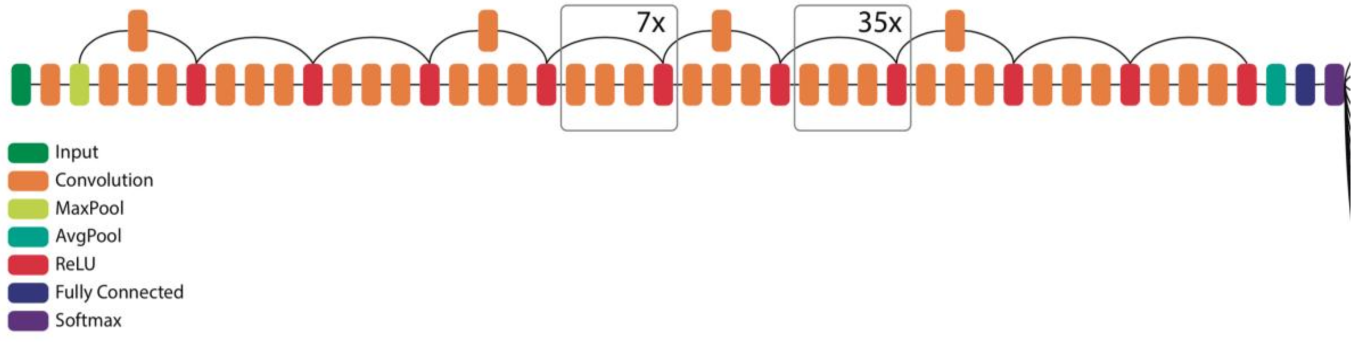

Figure 6.1 ResNet 152

6.2 Class Activation Maps (CAM) Integration

The architecture is augmented with the addition of Class Activation Maps (CAM). CAM enhances interpretability by revealing the network's focus during predictions, providing valuable insights into the decision-making process for precise image analysis.

Class Activation Maps (CAM) serve as a valuable technique within convolutional neural networks (CNNs) for enhancing model interpretability. Applied typically to the final convolutional layer, CAM generates heatmaps that highlight crucial regions in an input image, offering spatial localization insights. By employing a gradient-based approach, CAM transparently reveals the decision-making process, attributing importance scores to different image regions. Particularly useful in tasks like object localization, CAM aids in both understanding and validating model predictions, providing a detailed visual representation of where the network focuses during classification. Its integration, especially with architectures like ResNet, contributes to a nuanced and insightful analysis of feature importance in image classification tasks.

- ResNet-152, when combined with Class Activation Maps (CAM), provides insight into the neural network's focal points during predictions.
- The utilization of CAM enhances model interpretability by emphasizing influential regions within images.
- CAM aids in object localization, contributing to a detailed understanding of where the network focuses during predictions. The integration of CAM enhances transparency, providing insights into the decision-making process of the network in image classification tasks.

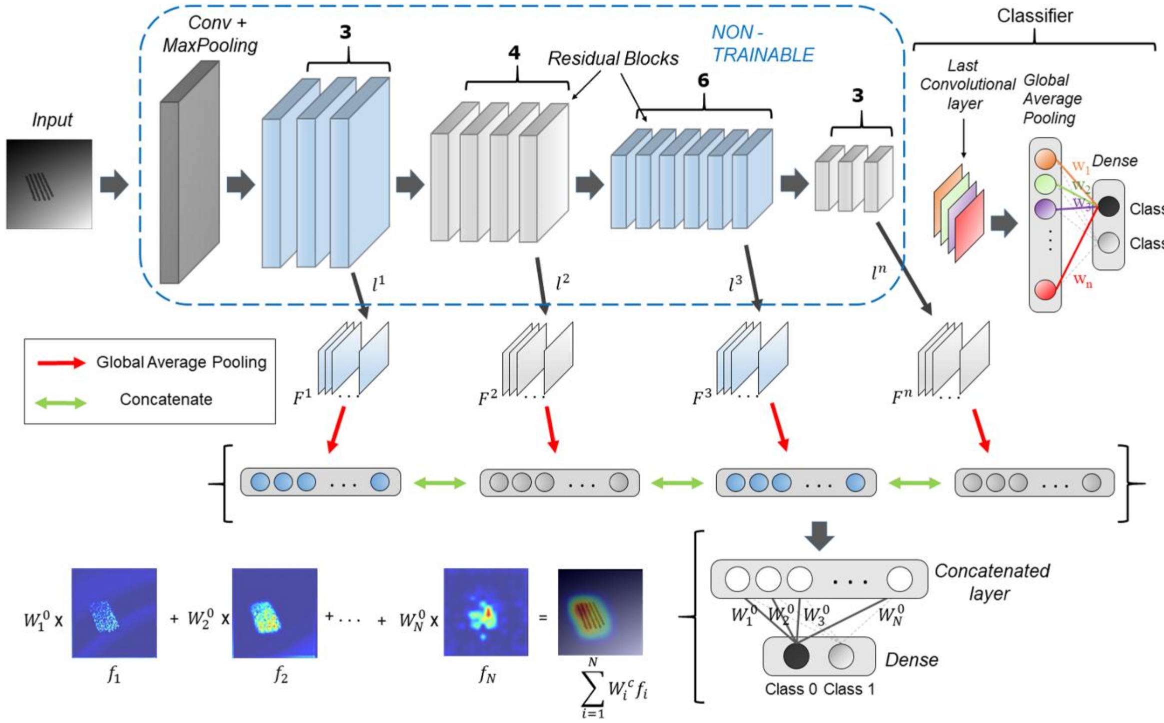

Figure 6.2 Class Activation Maps(CAM)

7. PSEUDOCODE

7.1 Data extraction:

Obtaining a diverse dataset of skin lesion images is the initial step. This dataset should encompass various types of skin disorders, categorized into benign and malignant cases. The dataset is then split into training and testing sets to facilitate model training and evaluation.

Algorithm - Data Extraction and merging
Input : Kaggle API key
Output: Dataset D

Start
1. Initialize Kaggle API:
   - Set Kaggle API key.
   - Authenticate Kaggle API.

2. Download HAM10000 Dataset:
   - Use Kaggle API to download HAM10000 dataset.
   - Specify the download path.

3. Combine Image Folders:
   - Combine two image folders into one.
   - Specify the paths for the two folders and the combined folder.
   - Store this as Dataset D
End

return D

7.2 Data Augmentation:

To enhance the model's robustness and prevent overfitting, data augmentation techniques such as shear, zoom, and horizontal flip are applied during the training phase. This helps the model generalize better to diverse skin lesion patterns. After augmentation, the data is normalized to ensure consistency and facilitate convergence during training.

To bolster the model's robustness and counteract overfitting, various data augmentation techniques, including rotate, and horizontal flip, are systematically applied during the training phase. These operations introduce diversity into the dataset, exposing the model to a broader array of skin lesion patterns. Following augmentation, the data undergoes normalization to ensure consistent pixel values, facilitating convergence during training. This dual approach not only enhances the model's adaptability to diverse lesion characteristics but also contributes to its ability to generalize effectively, fostering accurate and resilient skin lesion classification.

Algorithm - Data Augumentation

Input : Dataset D

Output: Augumented Dataset D_a

Start

1. Set Output Directory:
   - Specify an output directory path for storing augmented images.

2. Create Output Directory:
   - Create the output directory if it does not exist.

3. Set Original Images Directory:
   - Specify the path to the original images directory.

4. Set Classes to Augment and Target Sample Count:
   - Specify the classes to augment.
   - Set the target sample count for augmentation.

5. Initialize Augmented Entries List:
   - Create an empty list to store augmented entries.

6. Iterate Through Metadata :

- For each row in the metadata:
  - Check if the class is in the classes to augment.
  - Read the original image.
  - Normalize pixels if the image is not None.

    $image_{norm} = image/255.0$
  - Set augment count.
  - For each augmentation:
    - Rotate and flip the image.
      - Rotation

        If (x,y) are the original pixel coordinates and (x', y') are the rotated coordinates by an Angle θ

$$x' = x.\cos(\theta)-y.\sin(\theta)$$
$$y' = x.\sin(\theta)+y.\cos(\theta)$$

      - Flip

        Flipping involves change of sign of the coordinates

$$Flip_x(x,y) = (-x,y)$$
$$Flip_y(x,y) = (x,-y)$$

    - Save the augmented image.
    - Create a new image ID.

      $Image\_id_{new} = Image\_id_{old} + 1$
    - Create an augmented entry.

7. Create Augmented Metadata DataFrame :

- Create a DataFrame from the augmented entries.

8. Merge Metadata:

- Merge the original metadata with the augmented metadata into one DataFrame.

9. Store All Images:

- Store all the images, including augmented ones, in the specified output directory.
- Save it as Dataset D_a

End

Return D_a

7.3 Merge all the datasets

Merging datasets is a crucial step in consolidating diverse sources of information into a cohesive and comprehensive dataset. This process involves combining individual datasets, each representing a unique aspect or source of data, to create a unified and enriched dataset that captures a broader spectrum of information. It ensures a more holistic representation of the underlying phenomena or entities of interest. Merging can be particularly valuable in scenarios where data is collected from multiple sources, such as different experiments, sensors, or diverse data sources, leading to a more nuanced and robust dataset for subsequent analysis. This amalgamation facilitates a more holistic understanding of the domain under consideration and provides a solid foundation for more accurate and generalized model training or analysis.

Algorithm - Merge Datasets
Input:
  - List of datasets (datasets_list)
Output:
  - Merged dataset (merged_dataset)

Start
1. Initialize an empty dataset to store the merged data:
   merged_dataset = []

2. For each dataset Da in the list datasets_list:
   a. Append each element in Da to the merged_dataset.

End
Return merged_dataset

7.4 Normalization:

Normalization is a crucial step post-augmentation, ensuring that pixel values across images are standardized. This process helps the model converge faster during training and improves its ability to generalize across different skin lesion images.

In the continuum of data processing post-augmentation, normalization assumes paramount importance. This pivotal step is instrumental in standardizing pixel values across all images, a prerequisite for fostering accelerated convergence during model training. By mitigating variations in pixel intensities, normalization not only expedites the optimization process but also significantly bolsters the model's generalization capabilities. This heightened adaptability across diverse skin lesion images ensures a more resilient and effective model for accurate classification, underlining the critical role that normalization plays in refining the model's performance post-augmentation.

Algorithm - Normalize and Resize Images

Input:
- List of image paths (image_paths)
- Dataset D_a
- Target size for resizing (target_size)

Output:
- List of normalized and resized images (normalized_resized_images)

Start

1. Initialize an empty list to store normalized and resized images:
   Normalized_resized_images = []

2. For each image path in the list image_paths:
   a. Read the image from the file path.
   b. Convert the image to a format suitable for processing (e.g., RGB).
   c. Resize the image to the target size:
      resized_image = cv2.resize(normalized_image, target_size)
   d. Convert them into grayscale images:
      $I=0.299\times R+0.587\times G+0.114\times B$
   e. Convert them into numpy arrays:
      If I is the intensity of the pixels at (x,y) then
      Img_array[x,y] = I
   d. Normalize the pixel values of the image:
   - If the pixel values are in the range [0, 255], apply normalization:
      normalized_image = image/255.0

e. Append the normalized and resized image to the list normalized_resized_images.
End

Return normalized_resized_images

7.5 Choose the best model:

Examining three prevalent deep learning architectures—ResNet, MobileNet, and Inception—reveals distinctive features crucial for optimal model selection. ResNet's deep and residual learning design proves advantageous for tasks demanding high accuracy and intricate feature learning. However, its computational demands may pose challenges in resource-constrained environments. MobileNet, tailored for edge and mobile applications, strikes a balance between efficiency and accuracy, making it suitable for real-time scenarios. Inception, with its diverse inception modules, achieves equilibrium between accuracy and computational efficiency, though its implementation complexity and model size warrant careful consideration. The nuanced decision-making process involves evaluating task-specific needs, dataset characteristics, and deployment constraints, ensuring an informed selection aligns with the dynamic landscape of deep learning applications.

Algortihm -Model Comparison:

Input:
- Dataset (D_a) with labeled samples for training and evaluation.
- Models: ResNet-152, MobileNet, InceptionV3.

Output:
- Best performing model($model_{best}$)

Begin
1. Data Preprocessing:
   - Split the dataset into training and evaluation sets.

2. Model Training:
   - Train each model (ResNet-152, MobileNet, InceptionV3) on the training set using appropriate hyperparameters.
   - Utilize a common evaluation metric (e.g., accuracy, F1-score) to assess the models' performance during

training.Utilizing accuracy

$$\text{Accuracy} = \frac{\text{Total number of true positives and true negatives}}{\text{Total number of samples}}$$

-Utilizing F1 Score

$$\text{F1 Score} = \frac{2*\text{number of true positives}}{2*\text{number of true positives} + \text{number of false positives}+\text{number of false negatives}}$$

3. Model Evaluation:

- Evaluate the trained models on the evaluation set using the chosen metrics.
- Calculate the performance metric for each model.

4. Model Selection:

- Identify the model with the highest performance based on the evaluation metric.
- If multiple metrics are considered, use a weighted combination or prioritize the most critical metric based on the application.

5. Result Output:

- Output the selected model as the best-performing model($model_{best}$) for the given dataset and evaluation metric.

return $model_{best}$

7.6 Add explainability factor to the best model:

This algorithm, designed to enhance interpretability, adds a Class Activation Map (CAM) explainability module to the best-performing model. It begins by loading the saved best model and the image designated for CAM visualization. The image undergoes preprocessing to align with the model's input requirements. The algorithm then identifies the last convolutional layer and classifier layer indices in the model. By leveraging TensorFlow's GradientTape, it calculates gradients of the predicted class concerning the output feature map. Subsequently, the algorithm computes the CAM through global average pooling, normalizes it to values between 0 and 1, and creates a heatmap using these values. The final step involves overlaying this heatmap onto the original image, producing the CAM visualization. The outcome, consisting of the original image and the overlaid CAM, is then displayed, providing valuable insights into the model's decision-making process.

Algorithm - Add Explainability module(CAM) to Best Model:

Input:
- Best performing model ($model_{best}$)
- Image for CAM visualization

Output:
- CAM visualization overlaid on the original image

Begin:

1. Load Best Model:
   - Load the saved best-performing model($model_{best}$).

2. Load Image for CAM Visualization:
   - Load the image on which CAM will be visualized.

3. Preprocess Image:
   - Preprocess the image to match the input requirements of ($model_{best}$).

4. Get Relevant Layers:
   - Identify the last convolutional layer and classifier layer indices in ($model_{best}$).

5. Calculate Gradients:
   - Using TensorFlow GradientTape, calculate gradients of the predicted class with respect to the output feature map.

6. Obtain CAM:
   - Perform global average pooling to obtain the CAM.

7. Normalize CAM:
   - Normalize the CAM to values between 0 and 1.

8. Heatmap Visualization:

- Create a heatmap using the CAM values.

9. Overlay CAM on Original Image:
   - Superimpose the heatmap on the original image to create the CAM visualization.

10. Display Result:
    - Display the original image and the CAM visualization.

End

Return visualization

7.7 Deployment:

Deploying a machine learning model involves a multi-faceted approach encompassing both technical deployment and user interface/user experience (UI/UX) design considerations. The initial step is to select a deployment platform, whether cloud-based or on-premises, ensuring compatibility with the model's infrastructure requirements. Once deployed, creating an effective UI/UX is crucial for user interaction. The UI design should be intuitive, offering a seamless experience for users to interact with the model. This involves designing clear and user-friendly interfaces that accommodate input parameters, showcase model outputs, and provide feedback. A well-crafted UX ensures a smooth and engaging user journey, facilitating users in understanding the model's functionality and making informed decisions. It's imperative to iteratively test and refine the UI/UX design, taking user feedback into account to enhance usability and overall satisfaction. The combination of robust deployment strategies and thoughtful UI/UX design contributes to a successful integration of the model into real-world applications. Once deployed, the model performs its prediction.

Algorithm - Disease Prediction

Input
- image of the skin lesion
- Deployed model($model_{deployed}$)

Output
- Prediction of the disease along with the visualization of where it is

1. Load Deployed Model:

- Load the deployed mode($model_{deployed}$)l.

2. Load Image for Prediction and CAM Visualization:

- Load the image for which predictions and CAM will be visualized.

3. Preprocess Image:

- Preprocess the image to match the input requirements of

img_array = Preprocess(img)

4. Make Prediction:

- Obtain model predictions for the input image:

predictions = $model_{deployed}$(img_array)

5. Obtain CAM:

- Obtain CAM visualization

6. Display Result:

- Display the original image, predictions, and the CAM visualization.

End

Return results

8. EXTENSION PLAN

The primary goal of project extension is to enhance the functionalities and introduce new features. This extension plan encompasses various aspects, including model selection, integration of CAM, performance optimization, improvements in user interface/experience, and thorough testing. Additionally, documentation and will be prioritized to ensure a smooth rollout of the extended features.

4.1 Model Selection:

Choosing an appropriate model that caters to the needs of the problem in hand the different models which are prospects are RESNET, MobileNet,Inception.

4.2 Integration of CAM

Once the model is created the class activation maps(CAM) are to be integrated in the architecture of the model.

8.3 Performance Optimization

Apply optimization techniques to yield better results all while not compromising on the user experience.

8.4 User interface

Create a user interface which is handy and easy to use.It has to provide the explanation to the output generated.

9. REFERENCES

- Dusa Sai Charan, Hemanth Nadipineni, Subin Sahayam, Umarani Jayaraman. Method to Classify Skin Lesions Using Dermoscopic Images.
- Peter J. Bevan and Amir Atapour-Abarghouei. Skin Tone Detection and Debiasing for Skin Lesion Classification.
- Danilo Barros Mendes and Nilton Correia da Silva. Skin Lesions Classification Using Convolutional Neural Networks in Clinical Images.