

Skin Cancer Detection

(Group-2) Akash Parmar, Nihaal Chowdary Surpani, Sharath Sen Reddy Vellanki , Pranav Reddy Gunda

Abstract—The "SKIN CANCER DETECTION" project integrates dermatoscopy with advanced neural networks to enhance automated diagnosis of skin conditions. Drawing inspiration from a 1994 study on melanoma-mole differentiation, our project overcomes limitations by utilizing a more diverse dataset and modern deep learning techniques, especially convolutional neural networks (CNNs). Despite technological advancements, the project highlights the critical challenge of a limited supply of high-quality images for training precise diagnostic computer programs in the realm of skin diseases.

I. INTRODUCTION

Our SKIN CANCER DETECTION project involves integrating dermatoscopy, a technique for detailed skin lesion examination, with advanced neural networks to automate diagnosis. Dermatoscopy offers high-quality images for machine learning training. While a 1994 study used dermatoscopic images for melanoma-mole differentiation, limitations like a small dataset persist. Our project addresses this by expanding with a more diverse dataset and modern deep learning, especially convolutional neural networks (CNNs). Despite technological strides, the scarcity of high-quality images for varied skin diseases remains a critical challenge in training computer programs for precise diagnosis using technology and machine learning. Released to address the challenge of limited and homogeneous dermatoscopic image datasets, the HAM10000 ("Human Against Machine with 10000 training images") dataset comprises 10,015 diverse images from varied populations and acquisition modalities. Covering essential diagnostic categories like actinic keratoses, basal cell carcinoma, and melanoma, the dataset includes over 50% of lesions confirmed through histopathology. Ground truth for other cases is determined by follow-up examinations, expert consensus, or in-vivo confocal microscopy confirmation. While the test set remains non public, an evaluation server ensures fair method comparisons using the official test set on the challenge website, fostering standardized assessments in the development of automated pigmented skin lesion diagnosis.

II. DATA

The HAM10000 training set encompasses pigmented lesions from diverse populations, providing a rich dataset for skin lesion analysis. Leveraging TensorFlow and Keras, we employ the ResNet50 architecture to implement a Convolutional Neural Network (CNN) for image classification. The dataset consists of seven columns, each serving a distinct purpose. In total there are 7 features describing the metadata of the HAM10000 dataset. Further there are 2 different folders that contains the images for the dataset with each folder containing approximately 5000 images.

Feature	Feature Details
'lesion_id'	serves as a unique identifier for each skin lesion
'image_id'	corresponds to the identifier for the lesion's image
'dx'	denotes the diagnosis category (e.g., nv for nevi, mel for melanoma)
'dx_type'	indicates the diagnosis method (e.g., histopathology, follow-up)
'age'	records the age of the individual with the skin lesion
'sex'	denotes the gender (male or female)
'localization'	specifies the body location of the lesion

TABLE I
DATA DESCRIPTION

III. RELATED WORKS

Several significant contributions to the exploration of the HAM10000 dataset have been documented in the existing literature. One such endeavor, conducted by Ardan Adi Nugroho[11], employed Convolutional Neural Networks (CNNs) to effectively classify image data. Another notable research initiative led by Nour Aburaed [12] utilized the VGG-16 AND VGG-19 architectures. for the purpose of categorizing distinct classes within the dataset. Noteworthy is the commendable accuracy demonstrated by both models, showcasing superior performance particularly on the validation data. This synthesis underscores the importance and efficacy of these methodologies within the context of our project.

IV. EXPLORATORY DATA ANALYSIS

During our initial analysis, we observed a few insights from the data through visualization. Bar graph 1 represents the age of the people in relation to the count, with each bar having different colors indicating the types of diseases. We can observe that people in the age range of 40 to 60 were most affected by these diseases. Additionally, the second bar graph illustrates the localization of the diseases with respect to count. From these graphs, we can understand that most of the diseases are affecting the back and lower trunk region.

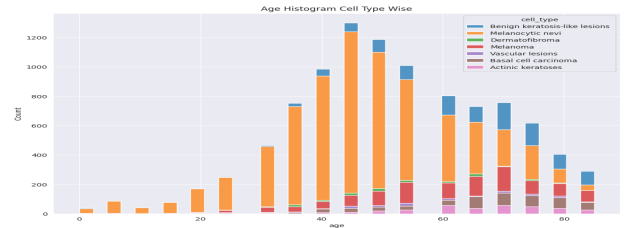


Fig. 1. AGE vs COUNT.

Through visualization , we found that the data is distributed unequally. We can observe this from Fig. 3, where a particular class of the data, such as [Melanocytic niv], have more data points, while other classes, such as [Vascular lesions,

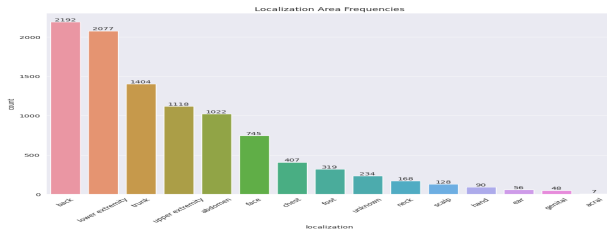


Fig. 2. LOCALIZATION OF DISEASES.

Dermatofibroma], have fewer data points. We thought that this data imbalance might affect the learning of the model, leading to overfitting on classes with more data and underfitting on classes with less data. To address this issue, we considered implementing data augmentation to balance the data for each class. Performing data augmentation significantly increased the accuracy of the model.

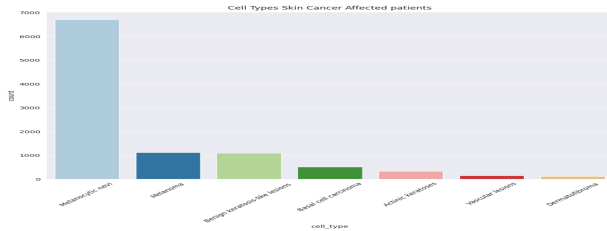


Fig. 3. TYPES OF DISEASES VS COUNT .

For this model, we considered using 500 images for each class. The augmentation is done as follows: if a class originally has more than 500 images, we randomly select 500 images from that class. If a class originally has fewer than 500 images, we include all existing images and perform data augmentation by selecting random images from that class to reach the target size of 500. Techniques such as random flipping, rotation, and random zoom are employed to augment the data.

V. METHODOLOGY

A. Transfer Learning

Transfer Learning is a technique in Deep Learning in which knowledge learned from a task is re-used to boost performance on a related task. The steps involved in Transfer Learning are as follows:

- The initial step is typically to instantiate the base model using one of the architectures such as ResNet or Inception. As a result, when generating the base model, you must remove the final output layer. You will later add a final output layer compatible with your problem.
- Freeze the pre-trained model's layers. This is because the weights of these models are updated in such a way that given any image, they can extract the relevant features such as edges, etc. So we could utilize this knowledge to extract features from images of the project we're working on.

- The last stage is to add new trainable layers, which will convert existing features into predictions on the new dataset. Because the pre-trained model is loaded without the final output layer, this is significant.
- The ultimate output of the pre-trained model will almost certainly differ from the output of your model. Pre-trained models trained on the ImageNet dataset, for example, will produce 1000 classes. Your model, on the other hand, may only have two classes. In this instance, you must train the model using a new output layer. As a result, you will add various more dense layers as you see fit, but most crucially, a final thick layer with units equal to the amount of outputs required by your model.
- By fine-tuning the model, we can improve its performance. Fine-tuning is accomplished by unfreezing the underlying model or a portion of it and training the entire model on the entire dataset again at a very low learning rate. The modest learning rate will improve the model's performance on the new dataset while limiting overfitting.

B. Inception V3 Architecture

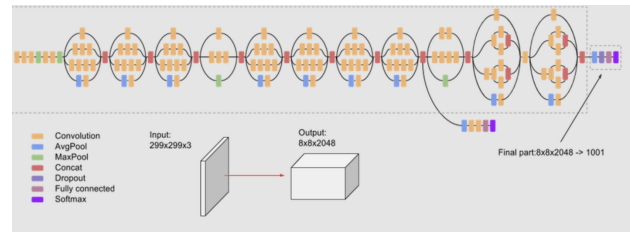


Fig. 4. Inception V3 Architecture.

Inception V3 is a convolution neural network for assisting in image analysis and object detection and got its start as a module for GoogLeNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge. The design of Inceptionv3 was intended to allow deeper networks while also keeping the number of parameters from growing too large.

C. ResNet50 Architecture

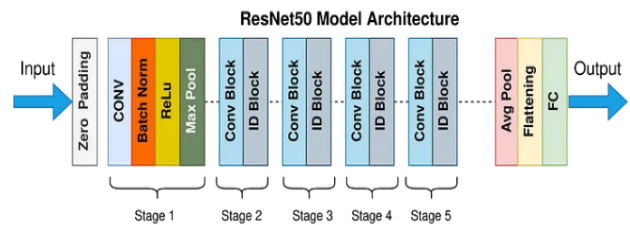


Fig. 5. ResNet50 Architecture.

This CNN architecture was first introduced in 2015 by researchers from Microsoft. It is trained on imagenet dataset with over million images and more than 1000 classes. The suffix 50 refers to the number of layers in the CNN model.

D. Components of the Models

Some components of the Models we used are as follows:

- Convolutional Layers : These are the building blocks of the CNN and are mainly used with the images. These layers mainly extract the features from the images by applying the convolutional filters. The filters are applied over a small region of the image and it covers the entire image resulting in creation of the feature maps. These feature maps are inputs to the next layers in the series.
- Batch Normalization : It is mainly used after the convolutional or fully connected layers but before the activation function. It serves multiple purposes such as faster convergence , improved generalization and regularization.
- ReLU Activation Function : It mainly introduces non-linearity into the system.
- Max Pooling : It reduces the dimensions of the outputs such as feature maps. It is useful in reducing the computational cost and even counter the problem of overfitting.
- Flatten Layer : It converts the input into a 1D vector as an output.
- Fully Connected Layer : The neurons in this layer are connected to each of the neurons in the previous layer . It is mainly used as the final layer and is responsible for making the prediction.
- Identity Block : An identity block consists of 3 convolutional layers , each followed by Batch Normalization and ReLU Activation function. It is mainly used to keep the dimensions of both input and output similar.
- Projection Block : It is used when the input and output dimensions differs . It has 2 convolutional layers with the first layer having stride (2,2) to downsample the input and the second layer has filter size(1,1) to adjust depth of input as per the depth of output.

The dataset HAM10000 is an imbalanced dataset with one of the classes having almost 70% of the data points . The classification labels were the names of the cancer cell types and to convert them into the numerical labels we have used label encoder . So now we had 7 different labels representing the 7 different cancer cell types . Creating a model on such a dataset would not have been effective in classifying the data so we resorted to the data augmentation . Data Augmentation technique is most commonly used in machine learning especially with the images. In this technique the data is artificially increased by making some transformation to the existing dataset. In the data augmentation , for the classes that have data points less than the target number (number of samples per class we want) we are creating the additional data points using data augmentation while for the classes that have equal or more data points than the target number we are doing the random sampling. We took 500 samples initially , from each of the 7 classes which in total was 3500 data points. After performing the data augmentation we performed one hot encoding for the labels. It is mainly used to convert the numerical labels into the vectors of 0 and 1 . The data points were split into training , validation and the testing set. 70% of

the data was for training , 15% data was for the validation and the remaining 15% was used for the test set. Before passing the data to the model the normalization of the data was performed. Upon training and validating the model we tested the model on test data and in order to evaluate the model we used the accuracy score , confusion matrix , training & validation accuracy and loss plots.

Algorithms	Accuracy Score
Inception_v3	0.5892443208159481
ResNet50	0.5016179879462216

TABLE II
ACCURACY SCORES

VI. EXPERIMENT

As mentioned above the models were trained with the dataset of 3500 points (500 data points per sample). The accuracy of both the models was very less. On the investigation from the previous model using the confusion matrix we found that for some classes the models were performing badly. And as was advised by professor Vu we decided to reduce the number of the classes and decided to increase the samples per class. Therefore for the new experiment we only considered 4 classes with 2000 samples from each class ('Melanocytic nevi', 'Dermatofibroma', 'Vascular lesions', 'Actinic keratoses') . Now we had a dataset with 8000 images. So initially the number of epochs for Inception-v3 and ResNet50 were 10 and 20 but for the new dataset we decided to increase for the Inception_v3 by 10 and for the ResNet-50 we increased the batch size to 35 which was 30 previously. The loss function for the models was set at categorical cross entropy which is mainly used for multiclass classification problems and we used Adam optimization algorithm which is based on the stochastic gradient descent. Both the models were trained on GPU T4 x2. We saw improvement in the final results as there was significant increase in the accuracy score of both the models.

Algorithms	Accuracy Score
Inception_v3	0.8083333333333333
ResNet50	0.5783333333333334

TABLE III
IMPROVED ACCURACY SCORES

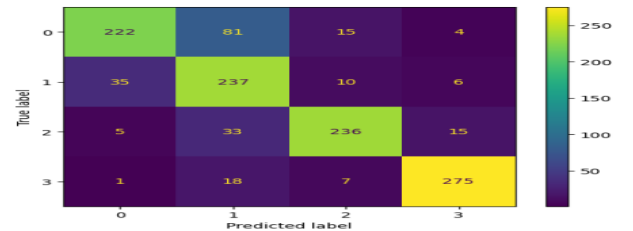


Fig. 6. Confusion Matrix(Inception_v3)

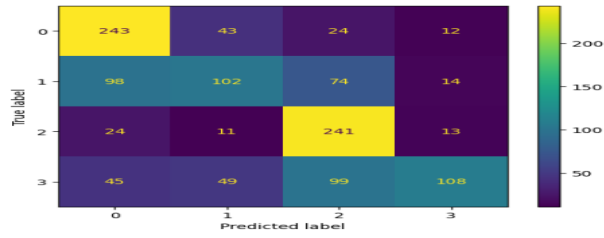


Fig. 7. Confusion Matrix(ResNet50)

VII. CONCLUSION

Initially we tried to train the models with less number of data points where we found that the accuracy scores were very less. Even for some of the classes the models were not performing well. Therefore we decided to reduce the number of classes and to increase the epochs for both and batch size for ResNet50. It resulted in significant increase in the accuracy scores of both the models. For resnet50 there was an increase of 8% while for inception_v3 there was an increase of 22%.

VIII. ACKNOWLEDGEMENT

We would like to express our gratitude to the authors as for our report we have used the sample report "Predicting Disaster" authored by Kathryn Beck, Lindsay Jacobs, Nicole Manno, and Becca Shipper. Their report acted as a guide to put our content together. We have divided content into 7 sections thus helping us to describe our work in an efficient manner.

IX. REFERENCES

REFERENCES

- [1] <https://medium.com/@sharma.tanish096/detailed-explanation-of-residual-network-resnet50-cnn-model-106e0ab9fa9e>
- [2] <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>
- [3] https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000?select=HAM10000_metadata.csv
- [4] <https://derma.jmir.org/2023/1/e42129>
- [5] <https://almarefa.net/blog/how-to-implement-batch-normalization-in-a>
- [6] Szegedy, Christian (2015). "Going deeper with convolutions"
- [7] <https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras>
- [8] <https://medium.com/@suryatejamenta/upsampler-and-convolution-transpose-layers-7e4346c05bb6>
- [9] <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- [10] <https://vucdinh.github.io/S23/sample.pdf>
- [11] <https://pubs.aip.org/aip/acp/article/2202/1/020039/962411/Skins-cancer-identification-system-of-HAM10000>
- [12] N. Aburaed, A. Panthakkan, M. Al-Saad, S. A. Amin and W. Mansoor, "Deep Convolutional Neural Network (DCNN) for Skin Cancer Classification," 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Glasgow, UK, 2020, pp. 1-4, doi: 10.1109/ICECS49266.2020.9294814.