Optimizing Skin Cancer Detection Neural Networks: SVD for Storage Reduction and Model Pruning for Compute Efficiency

Dr. Bahubali Shiragapur
Supervisor, School of Computer Science Engineering and Applications
DY Patil International University, Akurdi, Pune 411035
Email: bahubali.shiragapur@dypiu.ac.in

Prathamesh Deshpande
Student, School of Computer Science Engineering and Applications
DY Patil International University, Akurdi, Pune 411035
Email: deshpandeprathamesh137@gmail.com

Aditya Ambre
Student, School of Computer Science Engineering and Applications
DY Patil International University, Akurdi, Pune 411035
Email: adityaambre17feb@gmail.com

Prathamesh Dhas
Student, School of Computer Science Engineering and Applications
DY Patil International University, Akurdi, Pune 411035
Email: 20200802212@dypiu.ac.in

***Abstract*—Malignant and non-malignant forms of skin cancer constitute a significant global issue. The most dangerous kind, melanoma, can spread quickly and has a high mortality rate if it is not detected in its early stages. Although non-melanoma skin malignancies are more common, untreated cases of squamous and basal cell carcinomas can result in deformity and morbidity. Among the primary issues at hand are late detection, low doctor-to-patient ratios, lack of delivery mechanisms, and lack of awareness. Many Deep Neural Networks and Deep Convolution Neural Networks have been developed with the aim of early skin cancer diagnosis. Large designs like these demand a lot of computation and storage, which might be expensive or unavailable in settings with limited resources. Further, these Models require millions of high resolution dermatoscopic Images for training resulting in increased storage requirements. This Article proposes a scalable and reusable approach to compress and reconstruct these images with feature preservation and minimal loss using Singular Value Decomposition (SVD) compressing a dataset of 2.72 GB to less than 1GB achieving a compression ratio of 3.77 and saving 73% storage space. It also discusses a simple, uninformed and naive Model pruning method which reduces convolution filters by a pruning factor iteratively until a minimal model is achieved. We Prune 4 developed baseline architectures and achieve a best reduction in learnable parameters of approximately 5% of original baseline model and model size reduction from 444MB to less than 10MB.**

***Index Terms*—Convolution Neural Networks, Kiosk, Learnable Parameters, Model Pruning, Pruning Factor, Singular Value Decomposition**

1. Introduction

The World Health Organisation (WHO) estimates that there are approximately 132,000 new cases of melanoma detected every year globally. Countries have different rates of skin cancer; those that have high UV exposure and a majority of fair-skinned populations—such as Australia, New Zealand, North America, and portions of Europe—have the highest rates. In tropical nations with strong UV indices, like India, the number of cases is rising, especially in fair-skinned people and those who spend a lot of time outside. According to the World Cancer Research Fund International, in 2020 there were over 150,000 new cases of melanoma worldwide, making it the 17th most common cancer globally. The highest incidence rates of melanoma are observed in countries with predominantly fair-skinned populations and high UV exposure, such as Australia and New Zealand. [1]

With the Final Objective being diagnosing Skin Cancer in Early Stages, there have been numerous methods proposed utilizing Machine Learning Techniques and Deep Neural Networks and Convolutional Neural Networks. Deep Learning Architectures such as Alexnet, Resnet-50, Resnet-101 and many custom architectures have been trained and performed remarkably. DNNs and CNNs work on Raw Images and Data and identify and learn the discriminating Features during training and hence reduce the overhead of feature extraction before model training. However, developing and training of these models is very resource intensive due to complex architecture, high resolution and dimensionality of Images, High Compute Requirements. High Resolution Images also demand large storage space and these models are trained on millions of

images to achieve benchmark performance. In the Medical Field, Images have very High Resolution and dimensionality especially Dermatoscopic images of Skin lesions where fine-detailed patterns are very important. This increases the storage Requirements of the Medical Storage system and the Number of Features to be learned by the Models to focus on the Fine-detailed Patterns. This increases training time and computes the efficiency required for achieving benchmark performances. Since the number of Features to be learned increases, the number of learnable parameters(in 10s and 100s of millions), the model size, and training time for the model and inference time for predicting an image also increases. To train a CNN model effectively, powerful computational approaches are required like GPUs or TPUs. [2]

2. Literature Review

A Number of algorithms have been suggested for skin lesion classification, including decision trees, ensemble learning, optimization, feature augmentation and random forests. One of the suggested approach makes use of CNNs' special qualities in conjunction with machine learning to increase interpretability and accuracy. This approach uses CNNs ability to learn discriminatory patterns while training and using extracted features such as Color Histograms, Hu Moments for Shape and Haralick Textures for accurate diagnosis providing a condensed view of the images. [3]

The next approach increases the precision and effectiveness of skin lesion categorization by utilizing the benefits of both data augmentation and transfer learning. Skin lesion photos are resized to fit the AlexNet input size, color space converted, and noise removed. On the skin lesion datasets, AlexNet's pre-trained weights serve as a foundation for fine-tuning. In doing so, the training time and complexity are decreased while using the deep learning model's built-in feature learning capabilities. The skin lesion datasets are used to train AlexNet's final layers, which then modify the pre-trained features to suit the particular lesion classification job. [4].

Improving patient outcomes depends on early and precise skin cancer identification. For this purpose, deep convolutional neural networks (CNNs) have shown great promise because of their high accuracy and automation potential. This research presents an improved method for classifying skin cancer using transfer learning and a bespoke DCNN. They suggested a deep learning-based deep convolutional neural network (DCNN) model for the precise categorization of benign and malignant skin lesions.With a big dataset, our suggested DCNN model outperforms other deep learning (DL) models in terms of accuracy. Compared to other transfer learning models like AlexNet, ResNet, VGG-16, DenseNet, and MobileNet, the suggested DCNN model requires a significantly shorter execution time to process the output results.The HAM10000 dataset is used to evaluate the model, and the results show that we have the greatest training and testing accuracy (93.16% and 91.93%, respectively. [5]

The next approach combines LSTM with Mobilenet v2 for classification.The type of skin illness is classified using MobileNet V2, and the model's performance is improved using LSTM, which keeps track of the features' state information from its previous iteration of picture classification. MobileNet is a CNN-based model that is widely used for image classification, in contrast to MobileNet V2 . The primary benefit of utilizing the MobileNet architecture is that, in comparison to the traditional CNN model, it requires significantly less computing work, which makes it appropriate for use with mobile devices and PCs with less processing power.The MobileNet model is a convolution layer-based simplified structure that can be used to identify the detail based on two controllable characteristics that efficiently transition between the parameter's accuracy and latency. Reducing the size of the network is a benefit of the MobileNet paradigm. [6]

Patient Metadata and Images have also been used together to create a CNN plus ANN Combined model creating an efficient and resource-efficient model deployed on smaller devices. It achieved an accuracy of up to 92% and also reduced the over-fitting of the model. Both these models process respective data in parallel and later combine the results. [7]

In conclusion a) There have been Number of Algorithms and Architectures proposed for diagnosis of Skin Cancer. Deep Neural Networks and Convolutional Networks have been very effective due to their complex architectures able to learn discriminatory patterns from high resolution images. Their complex architectures, high compute and Storage requirements demand for use of expensive hardware like GPUs and TPUs which may incur heavy costs in these resource-constrained environments (like rural healthcare settings). Complex and Deep Architectures lead to increased latency and large inference time after deployment if no specialized Hardware like GPUs is used. Larger Training Time also mean an increase time in overall training lifecycle which requires a lot of iterative fine-tuning. b) These Models are trained on Millions of High resolution Images to improve accuracy, scaling them into a medical application means storage of these images, which can lead to very large storage requirements. For example HAM10000 Dataset consisting of 10,000 Images is 2GB in Size. Hence, this approach of storage hinders scalability. c) Transfer Learning Approaches have been proposed to reduce model size, but it also poses challenges such as it can stifle the development of new, specialized models designed specifically for medical applications. d) There also have been many pruning methods for pruning models based on magnitute, similarity, sensitivity analysis which reduce computational load and are also informed methods. However, these are complex implementations and require Pre-trained model weights, expertise and need for fine-tuning to avoid a significant reduction in parameters which increases overhead in terms of training time in model lifecycle.

3. Proposed Methodology

In this research study, we use the HAM10000 Dataset consisting of biopsy-proven dermatoscopic images [8] and 164 New Images for training captured and verified by Dermatologists. The New Images are Negative Samples, meaning they are images of Non-Cancerous Skin Lesion. This shall help the Model in differentiating between patterns of Skin Cancer Lesions and Non-Cancerous Skins. The New Samples are simple abrasions and the spots that develop subsequently from them. They were captured in a High Resolution Camera. The HAM10000 Dataset provided by International Skin Imaging Collaboration (ISIC) containing dermotoscopic images from various populations for seven classes of cancer: Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv) and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc).

Figure 1. Sample Images as Negative Samples

The Overall Dataset is Highly Imbalanced and Consists of More than 8 classes including our new negative samples. We simplify this by keeping classes Melanoma and Melanocytic Nevi and rest as Non-Melanoma Cancer. The negative class is kept as it is. This is to ensure we keep our model architectures simple and limited to identifying Melanoma and Non-Melanoma Cancer. This also reduces the variety and diversity generated by the textual features combined with the images. All the Images are sized 450x600 Pixels as per HAM10000 Dataset.

After this we apply Singular Value Decomposition (SVD) on all images one by one along the Y-Dimension and reconstruct the images using Top-K Singular Values and their corresponding singular vectors in (U) and (V^T), truncating all vectors and values below K. The K-varies from 100, 200, 300, 350, 400,450.The initial step size is big (100) and as we get closer to half of the vertical axis(600) i.e. 300 we decrease step size by half(50).We Reconstruct the entire datasets at all K values. After this, we need to select the best k-value such that we get a good compression ratio but it also preserves the image characteristics and has minimal loss. Then, we study various metrics such as Peak-Signal-to-Noise Ratio, Structural Similarity, Histogram Intersection and Bhattacharya distance between Color Histograms, Euclidean Distance between Hu Moments and Haralick Textures, percentage of pixel values changed more than a threshold, Variety of Information and Normalized Mutual Information, the Mean Squared Error Between High and Low Frequency Components, Spectral Loss Coefficient (for frequency analysis) of Original and Reconstructed Data. We also analyse the mean values and standard deviations for the metrics across different values of K for reconstructed dataset. We select the optimal value of K and use it for training our Models such that Compression ratio is good, Essential Features are preserved and storage size reduction is maximum.

Models Trained on HAM10000 Have resized the input images to reduced dimensions for reducing latency and reduce memory consumption. [3] Hence, we also consider some reduced standard dimension sizes like (256,256) and also try reducing it down by one axis- across y-axis (450,150) , across x axis (300,600), across both axis - (200,100), (300,300). The selection of Resolution of Pixel size is random but less pixels mean less information resulting in less feature maps in convolution and faster training times and reduced inference times. We Compare the above mentioned metrics after resizing our reconstructed dataset to select the best performing resolution with optimal metrics. We also study the effect of how Image Size affects the number of learnable parameters of the model since less resolution means less number of feature maps passed onto subsequent layers.

Next we start pre-processing the dataset, applying normalization and scaling to Numeric Attribute Age and Label Encoding to Sex and Localization. We fill Null Values in Age using Median of same column since it is a right-skewed distribution in the dataset. We also apply data augmentation to balance the dataset and enhancing generalization ability and robustness of the task.

We have already designed 4 Custom Neural Architectures on original Dataset with benchmark accuracy and precision in training, testing and validation stages. The 4 architectures are - a) Lenet-5 based Neural Network, b) Alexnet Based Neural Network, c) Custom Neural Network - 1 d) Custom Neural Network - 2(utilizing Residual Skip Connections). The Key Concepts we use here is the Text Model a Branch of Neural Network that processes the Metadata features, and the Concatenation of Final Features by Combining Image and Text Model Outputs for making a final Diagnosis.

Next in the research study, we train the same model using the reconstructed image dataset and observe the testing , training and validation metrics. We compare these with baseline model and observe the results if model is able to capture features from reconstructed data and achieve similar baseline performance metrics. We train the same model on resized images where resolution is selected above after analysis to reduce memory consumption and reduce amount of learnable parameters. The benchmarks for Lenet-5 based Neural Network in validation metrics is 75% and for other models is 80%. For Training it remains 90% for all Models. For Testing Metric Comparison , we select the

TABLE 1. SAMPLE ARCHITECTURE OF THE LENET-5 BASED NEURAL NETWORK WE DESIGNED FOR SKIN CANCER PREDICTION

Layer	Type	Filters/Units	Kernel Size
Text (MLP) Model			
1	Dense	4	-
2	Dropout	-	-
3	Dense	6	-
4	Dense	2	-
5	Dropout	-	-
6	Dense	1	-
Image (CNN) Model			
1	Conv2D	6	5x5
2	MaxPooling2D	-	2x2
3	BatchNormalization	-	-
4	Conv2D	16	5x5
5	MaxPooling2D	-	2x2
6	BatchNormalization	-	-
7	Flatten	-	-
8	Dense	84	-
9	Dense	120	-
Combined Model			
1	Concatenate	-	-
2	Dense	16	-
3	Dense	4	-

model with values closest to baseline metrics with accepted error range of -2%.

After this we select a pruning factor as 50%, we reduce the number of convolution filters and dense units in subsequent fully connected layers by this factor in an iterative way till we achieve a model under-performing in training, validation and testing metrics. Then, we make the pruning factor 33.33 %, repeating the same procedure until a minimal model is achieved. If still facing issues, reduce pruning factor by half in every iteration. 50% and 33% are selected as staring pruning factors since they cause exponential reduction in learnable parameters of model in iterative iterations and polynomial reduction if only some convolutional filters and dense layers are reduced. Training Precision and Accuracy, Validation Accuracy and Precision, Testing Metrics like accuracy, recall and precision are compared to the baseline model metrics to select the optimal model minimizing model size and maximizing performance.

The Storage Optimization using SVD significantly compresses data while capturing most of the variance and discriminatory features of the images. This reduces storage requirements on any platform significantly and also be performed at scale without much loss in information. There is also another observation here, that reconstructed images now have psycho-visual and inter-pixel redundancy introduced so comparatively less number of convolution filters can identify the discriminatory patterns.The disadvantage is the increased overhead time and cost of analysis of what number of singular values needed for reconstruction so there is no compromise in image quality and essential features are preserved. Model Pruning reduces number of hidden neurons using a pruning factor for convolution layers and the next 2 fully connected layers. This helps in reduction of model size, reduction in number of filters, number of learnable parameters and faster training and inference times. This also makes it easy to deploy the model on small scale devices. The disadvantage here is this is a quite simple, uninformed and naive approach on model pruning and requires lots of iterations in fine-tuning it.

4. Analysis and Design

4.1. Experimental Design

TABLE 2. ARCHITECTURE OF THE LENET-5 BASED NEURAL NETWORK

Layer	Type	Filters/Units	Kernel Size
Text (MLP) Model			
1	Dense	4	-
2	Dense	6	-
3	Dense	2	-
4	Dense	1	-
Image (CNN) Model			
1	Conv2D	6	5x5
2	Conv2D	16	5x5
3	Flatten	-	-
4	Dense	84	-
5	Dense	120	-
Combined Model			
1	Concatenate	-	-
2	Dense	16	-
3	Dense	4	-

TABLE 3. ARCHITECTURE OF THE ALEXNET-BASED NEURAL NETWORK

Layer	Type	Filters/Units	Kernel Size	Strides
Text (MLP) Model				
1	Dense	4	-	-
2	Dense	6	-	-
3	Dense	2	-	-
4	Dense	1	-	-
Image (CNN) Model				
1	Conv2D	64	5x5	2x2
2	ZeroPadding2D	-	2x2	-
3	Conv2D	192	5x5	1x1
4	ZeroPadding2D	-	1x1	-
5	Conv2D	384	3x3	-
6	ZeroPadding2D	-	1x1	-
7	Conv2D	384	3x3	-
8	ZeroPadding2D	-	1x1	-
9	Conv2D	256	3x3	-
10	Flatten	-	-	-
11	Dense	1024	-	-
12	Dense	1024	-	-
13	Dense	512	-	-
Combined Model				
1	Concatenate	-	-	-
2	Dense	8	-	-
3	Dense	4	-	-

Custom Neural Network-1 - The standard CNN model in our mixed neural network framework focuses on a straightforward, deep convolutional approach for processing image data. It begins with convolutional layers that capture the intricacies of the input images using progressively increasing filters: starting with 16 filters at a

7×7 kernel, moving to 48 filters at 5×5, and further layers with 92, 192, 192, and 256 filters each with a 3×3 kernel. Each convolutional layer is followed by max-pooling layers to downsample the feature maps, batch normalization to maintain stability and enhance training speed, and dropout to prevent overfitting by randomly deactivating neurons during training. These layers work together to create detailed feature maps from the input images. The CNN then uses fully connected layers with 1024, 1024, and 512 units to further refine and map the learned features into a compact form suitable for classification.

Custom Neural Network-2 - The ResNet-based CNN model for our mixed neural network leverages the power of deep residual learning to process image data efficiently. This architecture starts with a sequence of convolutional layers where the filters increase in size progressively: 16 filters with a 5×5 kernel, 32 filters with a 7×7 kernel, followed by layers with 64, 96, and 192 filters each using a 3×3 kernel. Each convolutional layer is paired with max-pooling to reduce the spatial dimensions, batch normalization to stabilize and accelerate training, and dropout to mitigate overfitting. This design extracts multi-scale feature hierarchies from the images. The model incorporates ResNet blocks—specifically, with 192 and 256 filters—that include skip connections to preserve the flow of gradients, thus enabling deeper networks to be trained without suffering from vanishing gradients

We begin by defining the architectures of our 4 Custom Neural Networks and the design choices in the common elements. The elements and building blocks of our custom defined neural networks have been described below. We can also view the number of filters, number of neurons , kernel size for two of the architectures in the Figures 2 and 3. The Other two architectures have been described due to their deep network architecture.

1) Two Separate Branches for Text and Image Processing: We create two separate model - Image and Text that learns features from respective input since patient metadata is also important in providing a nuanced and personalized diagnosis along with the Skin Lesions
2) Max pooling - We use MaxPooling for dimensionality reduction and to learn the most prominent features such as fine textures, edges which have high activation. It also helps in reduction of noise and smoothens out the surrounding lesion area since there are not many important and prominent edges outside of lesion boundary. An Example can be visible in **??**
3) Concatenation - We Concatenate the Feature Maps Extracted from both the Models and combine them into a single feature map for subsequent Layers to Learn the patterns from both of them before prediction.
4) Activation Functions : We use ReLU Activation Function as it is simple and computationally efficient. It also introduces a non-linearity in features and patterns.
5) Residual Blocks - Residual blocks, introduced in ResNet architectures, address the vanishing gradient problem by providing shortcut connections that bypass one or more layers. This design allows gradients to flow directly through these shortcuts, ensuring that deeper networks can be trained effectively.
6) Alexnet and Lenet-5 Architectures - We use them in two of our architectures and in the CNN Image Model as they are established and standard CNN Architectures that have performed well on Skin Cancer Dataset.

We then preprocess our Original Dataset, applying standardisation on age attribute, One-Hot encoding for class labels, Label encoding for sex and localization. We then Split the Dataset into training and validation, where 3300 samples are for training and 1100 samples are for validation. We also use a testing dataset of 360 Images. In each split, there is equal distribution of the classes.

The Models are trained on this data achieving good performance. The baseline benchmarks are setup for training and validation here, 90% in training and 80% in validation. Only for Lenet-5, we create 75% validation benchmark since the model depth is low and only 2 convolution layers are present with less neurons. All Architectures are trained for 75 Epochs , with a learning rate of 0.002 and Adaptive Moment Estimation Optimizer. The best performing model achieving baseline performance are saved at Custom Checkpoints.

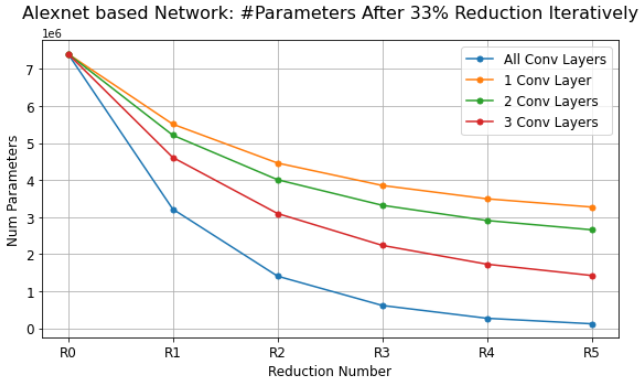

Figure 2. Number of Learnable Parameters (in millions) vs Reduction Iteration at different pruning factors

We now study the effect of reducing filters and neurons in next 2 Fully connected Layers by the pruning factor iteratively on all our architectures. We first reduce all convolution filters in all layers, then of only first Convolution layers, then first two layers and then first 3 convolution layers. We reduce the Neurons in Fully Connected Layers in All cases .We Model the Graph using a Linear Regression, a Polynomial and exponential Model to study the coefficient of determination Value. As it can be observed in graphs of 2 architectures from 2, and R^2 is maximum and greater than 0.9 for Exponential Model when we reduce all convolution layer's number of filters iteratively and in polynomial time in other cases. Thus, reducing maximum convolution filters is desirable.

Now, First we train and compare the same architecture on reconstructed dataset and measure its performance with baseline metrics. Then, we apply Model Pruning

using numerous pruning factors from Pruning factor array [50%, 33.33%, 16.5%, 8.25%] till we achieve a minimal model without much performance degradation. The Pruning Factor reduces filter and neurons in following fashion.

$$N_{\text{new}} = \lfloor N_{\text{old}} \times (1 - P[p]) \rfloor$$

where:

- N_{new} is the number of filters or neurons after pruning.
- N_{old} is the original number of filters or neurons before pruning.
- P is the Pruning Factor, expressed as a percentage. It represents the proportion of filters or neurons to be retained after pruning.
- $\lfloor \cdot \rfloor$ denotes the floor function, which rounds down to the nearest integer.

4.2. Algorithm

We apply Singular Value Decomposition (SVD) on all images one by one along the Y-Dimension and reconstruct the images using Top-K Singular Values and their corresponding singular vectors in (U) and (V^T), truncating all vectors and values below K. The K-varies from 100, 200, 300, 350, 400,450.The initial step size is big (100) and as we get closer to half of the vertical axis(600) i.e. 300 we decrease step size by half(50).We Reconstruct the entire datasets at all K values. After this, we need to select the best k-value such that we get a good compression ratio but it also preserves the image characteristics and has minimal loss. Then, we study various metrics such as Peak-Signal-to-Noise Ratio, Structural Similarity, Histogram Intersection and Bhattacharya distance between Color Histograms, Euclidean Distance between Hu Moments and Haralick Textures, percentage of pixel values changed more than a threshold, Variety of Information and Normalized Mutual Information, the Mean Squared Error Between High and Low Frequency Components, Spectral Loss Coefficient (for frequency analysis) of Original and Reconstructed Data. We also analyse the mean values and standard deviations for the metrics across different values of K for reconstructed dataset. We select the optimal value of K and use it for training our Models such that Compression ratio is good, Essential Features are preserved and storage size reduction is maximum.

We have already designed 4 Custom Neural Architectures on original Dataset with benchmark accuracy and precision in training, testing and validation stages. The 4 architectures are - a) Lenet-5 based Neural Network, b) Alexnet Based Neural Network, c) Custom Neural Network - 1 d) Custom Neural Network - 2(utilizing Residual Skip Connections). The Key Concepts we use here is the Text Model a Branch of Neural Network that processes the Metadata features, and the Concatenation of Final Features by Combining Image and Text Model Outputs for making a final Diagnosis.

Next in the research study, we train the same model using the reconstructed image dataset and observe the testing , training and validation metrics. We compare these with baseline model and observe the results if model is able to capture features from reconstructed data and achieve similar baseline performance metrics. We train the same model on resized images where resolution is selected above after analysis to reduce memory consumption and reduce amount of learnable parameters. The benchmarks for Lenet-5 based Neural Network in validation metrics is 75% and for other models is 80%. For Training it remains 90% for all Models. For Testing Metric Comparison , we select the model with values closest to baseline metrics with accepted error range of -2%.

Algorithm 1 Image Compression using SVD and Quality Metrics Evaluation

Input: List of images I, SVD compression levels $K = \{100, 200, 300, 350, 400, 450\}$
Output: Best compression level k^* based on quality metrics
for each image I in I **do**
  Initialize empty lists for PSNR, SSIM, RMSE,
  Initialize empty lists for Color, Hu Moments, Texture Features
  for each $k \in K$ **do**
    Compute SVD of image I: $U, \Sigma, V^T \leftarrow \text{SVD}(I)$
    Retain top k singular values in Σ: Σ_k
    Compute compressed image $\hat{I}_k = U \cdot \Sigma_k \cdot V^T$
    Compute quality metrics:
    Calculate PSNR, SSIM, RMSE for $\hat{I}_k$ compared to I
    Extract Color, Hu Moments, Texture Features from $\hat{I}_k$
    Extract Color, Hu Moments, Texture Features from I
    Compare Color, Hu Moments, Texture Features from I with $\hat{I}_k$
    Append quality metrics and features to respective lists
  end for
  Determine k^* based on best performance across metrics
end for
return Best compression level k^*

The neural networks, excluding Lenet-5, are trained on the GPUP100 available via Kaggle API with a training benchmark of 90%, a validation target of 80%, and a testing tolerance of 2% for errors, whereas Lenet-5 has a special case with a validation baseline set at 75% due to its fewer convolutions and shallower network depth

Algorithm 2 Iterative Model Training with SVD-Reconstructed Data and Pruning

Input: Reconstructed data $\hat{X}_k$ for a specific k, Pruning factors $P = \{50\%, 33.33\%, 16.5\%, 8.25\%\}$, Baseline model metrics, Training, Validation, and Testing data
Output: Minimal pruned model achieving desired metrics
Initialize best model metrics
Initialize variable p as index for pruning factors
$baseline\_met \leftarrow$ **false**
$baseline\_not\_met\_counter \leftarrow 0$
while true **do**
    Train model on $\hat{X}_k$ using training data
    Validate model on validation data
    Calculate precision and accuracy metrics
    if metrics meet baseline and testing error acceptance up to 2% **then**
        Save model and metrics as best model
        return Best model and metrics
    else
        $baseline\_met \leftarrow$ **false**
        break
    end if
    $p \leftarrow 0$
    while true **do**
        if $p \geq$ length(P) **then**
            break
        end if
        Prune Convolution Layers, and next 2 Fully Connected Layers of model with factor $P[p]$
        Update number of filters/neurons using

$$N_{\text{new}} = \lfloor N_{\text{old}} \times (1 - P[p]) \rfloor$$

        Retrain pruned model on training data
        Validate pruned model on validation data
        Calculate precision and accuracy metrics
        if metrics meet baseline and testing error acceptance up to 2% **then**
            Save model and metrics as best model
            $baseline\_met \leftarrow$ **true**
            $baseline\_not\_met\_counter \leftarrow 0$
        else
            $baseline\_met \leftarrow$ **false**
            $baseline\_not\_met\_counter \leftarrow baseline\_not\_met\_counter + 1$
            if $baseline\_not\_met\_counter \geq 2$ **then**
                break
            end if
            Increment p by 1
        end if
    end while
end while

4.3. Feature Extraction and Analysis

This section deals with how we select the best k value for Singular Value Decomposition after comparing Multiple Image Quality Metrics and Minimizing Error between Features, and maintaining a significant compression ratio.

In image quality assessment, PSNR provides a straightforward measure of fidelity based on pixel-wise differences, MSE quantifies the average error, and

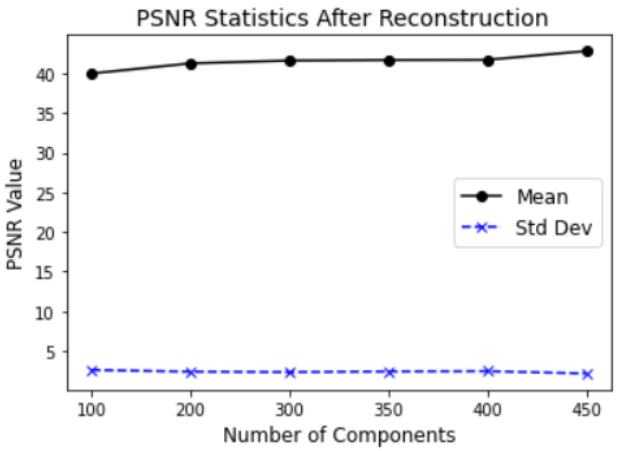

Figure 3. PSNR Comparison after reconstruction for various values of K

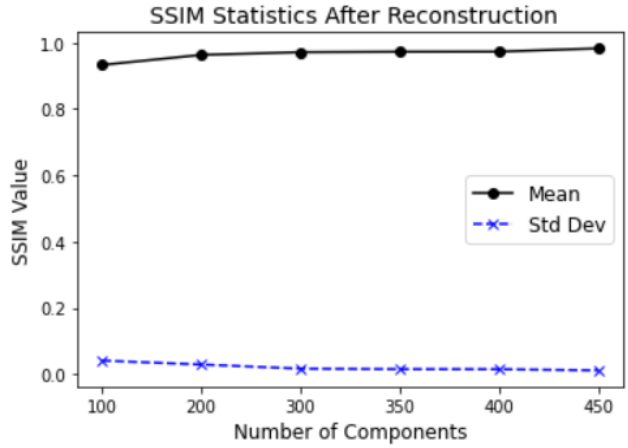

Figure 4. SSIM Comparison after reconstruction for various values of K

SSIM offers a perceptual evaluation considering structural information. Together, these metrics provide comprehensive insights into the accuracy and perceptual quality of reconstructed images, essential for various applications such as medical imaging, surveillance, and digital photography.We observe Image Quality Metrics like PSNR, SSIM and RMSE of reconstructed dataset compared to original at various values of K in Figure 3 and observe for different values of K on x-axis, depicted as number of components used in reconstruction. As we increase number of components, we observe a better values of RMSE between the pixels and slightly better PSNR Values around 40 dB. The SSIM Value improves from 0.97 to 0.99 improving at a very slow rate as we improve number of singular vectors and values. The standard deviation for these metrics also follows the same trend.

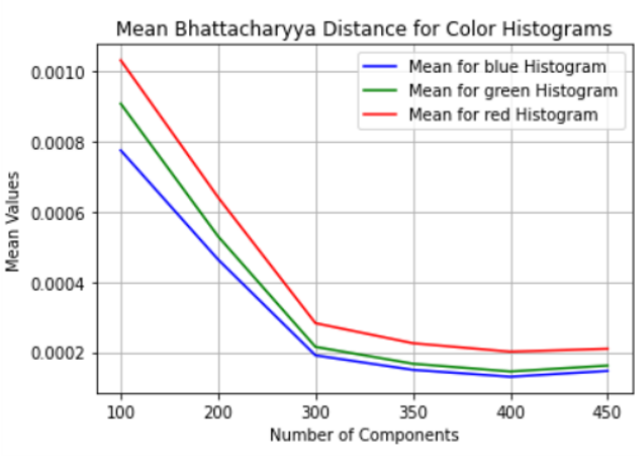

Figure 5. Comparing Average Bhattacharya Distance Between Color Histograms at different k-singular values

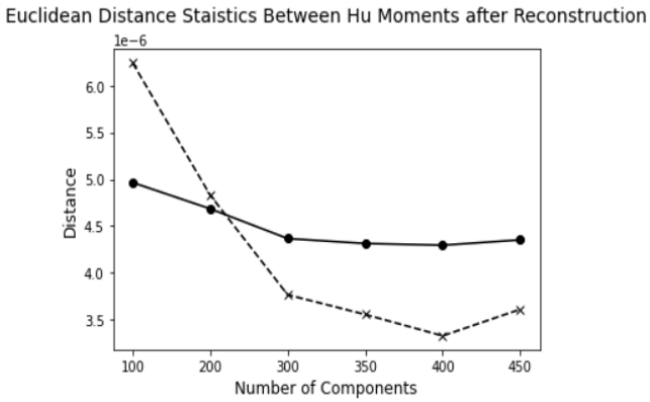

Figure 6. Comparing Euclidean Distance between Hu Moments for reconstructed dataset at different k-singular values a)Dotted Line represents the Avg. std deviation and Solid Line Average Mean

The Color, Shape and Texture are the most essential features of the skin lesions, so we analyze how these features are changed and compare them with features from original images , to see how much essential information we preserve and how much we lose.

Bhattacharya Distance- This is a statistical measure used to determine the similarity between two probability distributions. In the context of image processing, it is used to compare color histograms, which represent the distribution of colors in an image. A smaller Bhattacharya distance indicates a higher similarity between the original and reconstructed images in terms of their color distribution.

In Figure 5,we calculate the Bhattacharya distance between the original and reconstructed dataset images' color histograms in the three channels. This helps us understand how well the color information is preserved during reconstruction. The Bhattacharya distance measures the similarity between two probability distributions, making it an effective tool for comparing color histograms. As we increase the number of singular values (SVD components), the distance minimizes, indicating improved color preservation, and stabilizes after k=350, suggesting that using 350 singular values captures the essential color information effectively

Hu Moments- Hu Moments are a set of seven invariant derived from the central moments of an image, which capture the geometric properties or shape information of objects within the image. These moments remain unchanged under image transformations such as translation, scaling, and rotation, making them highly effective for shape analysis.

In Figure 6, we compute the Euclidean distance between the Hu Moments of the original and SVD-reconstructed images to assess how well the shape information is preserved. We observe that the average Euclidean distance remains stable, but the standard deviation decreases until $k = 400$ and then slightly increases.This suggests that the shape information is preserved at $k = 350$ onwards due to stability in std deviation, balancing between retaining essential components and minimizing the number of singular values used for reconstruction. The correlation coefficient between the hu moments at all components is 0.999 ensuring high degree of similarity in shape between reconstructed and original dataset.

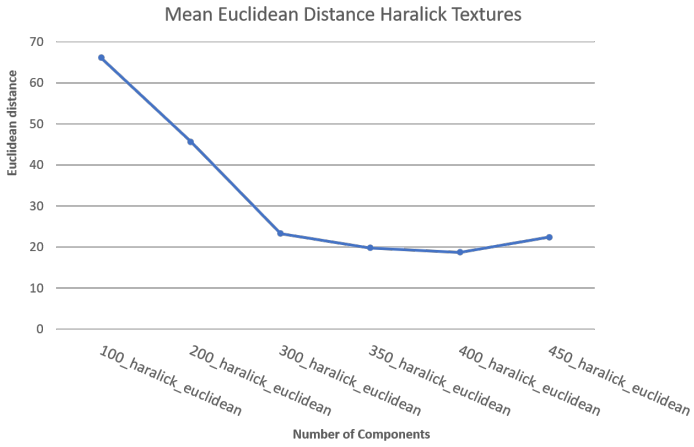

Figure 7. Comparing Euclidean Distance Between Haralick Texture Features at different k-singular values

Haralick Texture Features are a set of statistical measures derived from the Gray-Level Co-occurrence Matrix (GLCM) of an image, which quantifies the spatial relationships between pixel intensities. The GLCM calculates how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, capturing the texture properties by examining the frequency and distribution of pixel pairs. Haralick features computed from this matrix include metrics such as contrast, correlation, energy, and homogeneity, which describe the texture's smoothness, coarseness, and regularity. [9]

In our study, we observe the Euclidean distance between the Haralick texture features of the original and reconstructed images in Figure 7. This distance helps us understand how well the texture is preserved during the reconstruction process. We find that as the number of sin-

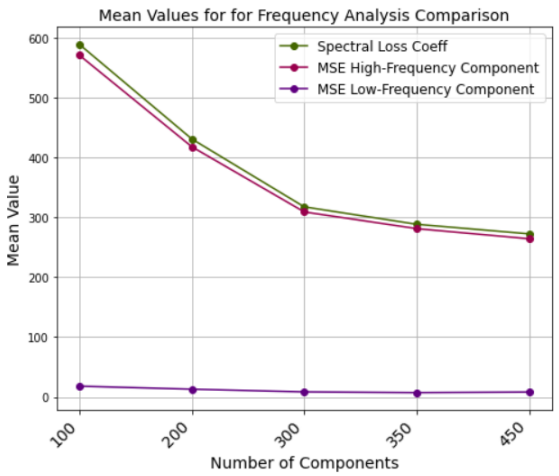

Figure 8. Comparison of Reconstructed Images at Various K Singular Values in Frequency Domain

gular values k increases, the Euclidean distance decreases, indicating that the texture of the reconstructed image becomes more similar to the original. This trend stabilizes at $k = 350$, suggesting an optimal balance between data reduction and texture preservation. However, for $k > 400$, we notice an increase in the Euclidean distance, likely due to noise introduced by the smaller singular values, which were previously truncated. Texture features, especially in sensitive areas like medical imaging of skin lesions, are crucial for accurate analysis. Even slight noise can significantly affect these metrics, as texture in skin lesions includes fine details that are essential for diagnosis. Therefore, maintaining the fidelity of these texture features without introducing noise is critical for the effectiveness of the reconstructed images.

Frequency Domain Analysis In image processing using methods such as the Fourier Transform, an image requires being transformed from the spatial domain to the frequency domain in order to be analysed in the frequency domain for image processing. We can examine the frequency components of an image by transforming it so that high frequencies capture the small details and low frequencies represent the broad structures. We can assess how well various frequency components are maintained during reconstruction by comparing the frequency domains of the original and rebuilt images.

In Figure 8, we observe the spectral correlation coefficient between the original and reconstructed images, which remains consistently high at 0.99 across all components. This high spectral correlation indicates that the reconstructed images retain their frequency characteristics almost perfectly, ensuring that both low and high-frequency details are well-preserved.

Furthermore, we examine the Mean Squared Error (MSE) between the high-frequency coefficients and the low-frequency coefficients of the original and reconstructed images. As the number of singular values k increases, we find that the MSE between these coefficients decreases. This trend suggests that increasing the number of components enhances the reconstruction's accuracy, especially in capturing the essential frequency details without losing fine details or introducing artifacts. The MSE also stabilizes at $k = 350$ as observed in Figure 8

From these observations, we infer that the reconstruction process efficiently preserves the image's frequency characteristics, crucial for maintaining the visual and structural integrity of images. In applications like medical imaging, particularly in analyzing skin lesions, preserving high-frequency details is vital as they often contain critical diagnostic information. Thus, selecting an optimal number of singular components, around $k = 350$, ensures that we retain these important details while minimizing unnecessary noise.

Models Trained on HAM10000 Have resized the input images to reduced dimensions for reducing latency and reduce memory consumption. [3] Hence, we also consider some reduced standard dimension sizes like (256,256) and also try reducing it down by one axis- across y-axis (450,150) , across x axis (300,600), across both axis - (200,100), (300,300). The selection of Resolution of Pixel size is random but less pixels mean less information resulting in less feature maps in convolution and faster training times and reduced inference times. We Compare the above mentioned metrics after resizing our reconstructed dataset to select the best performing resolution with optimal metrics. We find that at (200,100), we get least number pixel resolution and optimal values for PSNR, RMSE, SSIM. In Conclusion, All our metrics maximize/minimize at k = 350 and k=400, and stabilize afterwords. In some metrics like Haralick textures where even minimum of noise can affect the feature, the euclidean distance between haralick features starts to increase after k = 400. Hence, we select the minimum of the two, k = 350 to compress image along y-axis and perform reconstruction. The Explained Variance Ratio at k =350 is also greater than 98%.

5. Results and Discussion

Let us review the basic classification metrics and their significance before we dive into the results.

5.1. Metrics Definition

$$\text{Precision} = \frac{TP}{TP + FP}$$

The precision of the model is defined as the proportion of true positive predictions (accurately predicted positive cases) among all positive predictions. It is particularly useful in scenarios where false positives are costly, such as in fraud detection or medical diagnosis.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall quantifies the proportion of true positive predictions correctly recognized by the model out of all actual

positive cases. It is crucial in applications where identifying all positive examples is critical, like disease identification or customer churn prediction.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy measures the proportion of correct predictions (both true positives and true negatives) out of all predictions made by the model. It provides an overall assessment of how well the model performs across all classes, but it may not be suitable for imbalanced datasets.

5.2. Tables and Discussion

From Table 4 and Table 5, we observe classification metrics for the four architectures in both the training and testing phases, comparing performance on the original dataset and the dataset reconstructed using the top 350 singular values and vectors obtained through Singular Value Decomposition (SVD). The metrics indicate that the models achieve or even outperform the baseline performance of models trained on the original dataset, as evidenced by the comparison of accuracy, precision, and recall in both phases. This observation suggests that the SVD-based reconstruction method effectively preserves essential information from the original data. Despite reducing the dimensionality of the data through SVD, the reconstructed dataset retains sufficient discriminative features, allowing the models to perform comparably well.

In technical terms, accuracy measures the proportion of correctly classified instances among all instances evaluated. Precision measures the proportion of true positive predictions among all positive predictions made by the model, indicating how precise the model is when it predicts a positive class. Recall measures the proportion of true positive predictions among all actual positive instances in the dataset, indicating the model's ability to correctly identify positive instances.

The higher precision, recall, and accuracy achieved by the AlexNet-based Neural Network compared to the model trained on the original dataset suggest that the reconstructed dataset maintains or even enhances discriminative features important for classification tasks.

$$\text{Compression Ratio} = \frac{\text{Size before compression}}{\text{Size after compression}}$$

$$\text{Storage Saved (\%)} = \left(1 - \frac{\text{Size after compression}}{\text{Size before compression}}\right) \times 100$$

Achieving a compression ratio of approximately 3.77 and saving 73% of storage space through dataset compression and reconstruction represents a substantial advancement in efficient data utilization for machine learning applications. This reduction is particularly crucial in resource-constrained environments where storage costs and capacity limitations are significant concerns. By leveraging techniques like Singular Value Decomposition (SVD) for compression, the approach effectively retains critical data features while reducing redundant information. This not only optimizes storage usage but also enhances computational efficiency during model training and inference.

Furthermore, the ability to maintain or even improve model performance post-compression underscores the methodology's effectiveness in preserving essential information. Models trained on the reconstructed dataset demonstrate comparable or enhanced accuracy, precision, and recall compared to those trained on the original dataset. This outcome highlights the robustness of the compression technique in capturing discriminative patterns and minimizing data loss. Consequently, the compressed dataset not only facilitates streamlined deployment across various platforms but also supports sustainable data management practices by reducing storage footprint and operational costs. As such, this approach not only addresses immediate scalability challenges but also aligns with broader efforts towards optimizing resource utilization in machine learning and enhancing environmental sustainability in data-intensive applications.

Now , we analyse the results of Model Pruning on Model architectures presented and the reconstructed dataset. The classification metrics for pruned models are compared. For displaying reduction of a pruned model we use convention R-I, Prune Factor, which means the model has been pruned for I Number of iterations using same prune factor. If reduction is present as an array (R-I1, RI2) - (X1 %, X2%), it means for iteration I1 , X1 was used as pruning factor and X2 for I2 Iteration. We stop the pruning when the model architecture becomes very simple or when baseline performance is not met after 2 iterations of Pruning factor of 33.3%.

The baseline benchmarks are setup for training and validation here, 90% in training and 80% in validation. Only for Lenet-5, we create 75% validation benchmark since the model depth is low and only 2 convolution layers are present with less neurons. All Architectures are trained for 75 Epochs , with a learning rate of 0.002 and Adaptive Moment Estimation Optimizer. The best performing model achieving baseline performance are saved at Custom Checkpoints.

Observing Table 6, we can see pruned architectures of the proposed models and classification metrics evaluated on the reconstructed Dataset.Let's Look at the Models one by one, for Lenet-5 based Neural Network we see, we can achieve baseline metrics at Training and Validation but exceeds the 2% error rate in testing metrics. This is due to the simple, shallow architecture of Lenet-5 Architecture used in Image Model with only 2 convolution layers.For Alexnet Based Neural Network, we achieve baseline performance in training, testing and validation in comparison with the original model at iteration 3 with pruning factor 50%. We keep reducing in with same pruning factor in iteration 4 and 5 but not achieve baseline performance on testing and training, but achieve it on validation. This suggests although the model is learning patterns quickly, it needs more fine-tuning of hyper-parameters and a longer time to converge to learn more patterns effectively. Also, it implies the model is not learning the patterns quickly

TABLE 4. TRAINING AND VALIDATION METRICS FOR DIFFERENT ARCHITECTURES AND DATASETS

Training and Validation Metrics (in %) - Original vs Reconstructed Dataset					
Architecture	Dataset	Training		Validation	
		Precision	Accuracy	Precision	Accuracy
Lenet-5 Based Neural Network	Original	99.84	99.84	73.72	74.06
	Reconstructed	98.72	98.66	78.31	77.55
Alexnet Based Neural Network	Original	97.79	97.90	80.24	80.00
	Reconstructed	96.94	97.05	83.55	83.87
Custom Neural Network-1	Original	97.61	97.66	80.55	81.03
	Reconstructed	97.03	97.20	80.18	80.38
Custom Neural Network-2	Original	97.12	97.35	77.09	78.13
	Reconstructed	97.15	97.26	80.36	80.44

TABLE 5. TESTING METRICS (PRECISION, ACCURACY, AND RECALL) FOR DIFFERENT ARCHITECTURES AND DATASETS

Testing Metrics (in %) - Original vs Reconstructed Dataset				
Architecture	Dataset	Metrics		
		Precision	Accuracy	Recall
Lenet-5 Based Neural Network	Original	71.73	72.22	71.76
	Reconstructed	70.69	69.72	69.25
Alexnet Based Neural Network	Original	74.73	73.88	73.43
	Reconstructed	77.77	77.50	77.07
Custom Neural Network-1	Original	76.48	75.83	75.35
	Reconstructed	74.91	73.05	72.62
Custom Neural Network-2	Original	72.64	71.66	71.13
	Reconstructed	77.63	77.22	77.03

and has become simple. For Custom Neural Network -2 we achieve pruned model after one reduction iteration 2 with prune factor of 33.33% and for Custom Neural Network- 1 we achieve it with first reduction using factor of 50% and second iteration of 33.33%.

Achieving a substantial reduction in the number of learnable parameters and model size, particularly evident in the AlexNet-based neural network, marks a pivotal advancement in model optimization for small-scale applications. By significantly shrinking the model's footprint, the pruned AlexNet variant boasts a reduction of learnable parameters to just 5% of those in the original model. This drastic reduction not only conserves computational resources but also facilitates faster inference times, making it ideal for deployment on resource-constrained hardware platforms. The streamlined model size, which now measures less than 10MB, further enhances its suitability for deployment in environments where memory constraints are a critical consideration.

Across all architectures, similar trends are observed with reductions in learnable parameters ranging from 5% to 35% of those found in their original counterparts. This reduction in parameter complexity underscores the efficacy of the pruning technique in distilling models down to their most essential components without compromising performance metrics like accuracy and precision. Notably, the Lenet-5 architecture demonstrates a comparatively moderate reduction of 59% in learnable parameters due to its simpler structure, yet still achieving meaningful efficiency gains.

Moreover, the impact on training time is equally noteworthy, with all architectures experiencing significant reductions in training duration. Training times across the board are consistently completed in under 10 minutes, showcasing an impressive 10 to 20-minute reduction on high-performance GPU platforms like the GPUP100. This efficiency gain is particularly critical in contexts where rapid model iteration and deployment are essential, such as in real-time analytics or on embedded systems with limited computational resources. Overall, the combined benefits of reduced parameter complexity, minimized model size, and accelerated training times not only enhance operational efficiency but also extend the practical applicability of these optimized models across diverse hardware ecosystems, thereby fostering broader accessibility and scalability in machine learning deployments.

Reducing the number of learnable parameters and model size yields profound benefits that extend beyond immediate computational efficiency. Firstly, the streamlined models significantly expedite the training phase of the machine learning lifecycle. With fewer parameters to optimize during training, each epoch completes more swiftly, leading to reduced overall training times. This acceleration not only

TABLE 6. TESTING METRICS FOR DIFFERENT ARCHITECTURES AND DATASETS

Testing Metrics (in %) - Iterative Neuron and Convolutional Filter Reduction				
Architecture	Model	Metrics		
		Accuracy	Precision	Recall
Lenet-Based architecture	Original	72.22	73.08	71.89
	R1: 50%	67.22	66.90	66.74
	R1: 33%	68.88	69.68	68.33
	R2: 33%	63.88	61.94	63.38
Alexnet Based Neural Network	Original	73.33	74.81	73.08
	R1: 50%	75.00	74.95	74.58
	R2: 50%	73.33	75.65	73.13
	R3: 50%	73.88	73.89	73.58
	R4: 50%	70.83	76.88	70.64
	R5: 50%	72.77	77.91	72.74
Custom Neural Network-1	Original	76.94	77.42	76.33
	R1: 50%	76.66	77.27	76.23
	R1: 33%	71.66	75.22	70.83
	R1, R2: 50%, 33%	74.72	75.07	74.05
Custom Neural Network-2	Original	73.88	76.14	73.61
	R1: 50%	66.94	69.55	66.11
	R1: 33%	70.55	69.79	69.96
	R2: 33%	71.94	78.19	72.03

enhances developer productivity by enabling faster experimentation and iteration but also lowers the operational costs associated with training models on cloud-based GPU instances or dedicated hardware. The efficiency gains translate into shorter development cycles, allowing researchers and engineers to explore a wider range of model architectures and hyperparameters within a given timeframe.

Moreover, the decreased model complexity and size have a substantial impact on memory consumption during both training and inference phases. By requiring less memory to store model parameters and intermediate computations, the optimized models alleviate potential bottlenecks in memory-constrained environments. This advantage is particularly crucial for deploying machine learning solutions on edge devices or embedded systems where limited RAM capacity imposes stringent constraints. The ability to run compact models efficiently on devices with modest computational resources extends the applicability of machine learning to scenarios where real-time processing, such as in IoT applications or mobile devices, is paramount.

Furthermore, the iterative time spent in refining and fine-tuning models is markedly reduced. With faster training cycles enabled by optimized model architectures, researchers can explore more iterations and variations of their models to achieve higher performance thresholds. This iterative improvement process is essential for pushing the boundaries of model accuracy and generalization, particularly in complex tasks such as image recognition or natural language processing. The shortened iterative time not only accelerates the pace of innovation in AI research but also facilitates rapid adaptation to evolving data and business requirements, ensuring that machine learning solutions remain agile and responsive in dynamic environments.

In summary, the efficiency gains derived from reducing learnable parameters and model size have profound implications across the machine learning lifecycle. From faster training times and reduced memory consumption to accelerated iterative cycles, these optimizations empower practitioners to deliver robust and efficient machine learning solutions that are both scalable and adaptable to diverse application domains and computing environments.

6. Conclusion

In Conclusion, the Storage Optimization using Singular Value Decomposition(SVD) significantly compresses data while capturing most of the variance and discriminatory features of the images. This reduces storage requirements on any platform significantly and also be performed at scale without much loss in information. There is also another observation here, that reconstructed images now have psycho-visual and inter-pixel redundancy introduced so comparatively less number of convolution filters can identify the discriminatory patterns. The disadvantage is the increased overhead time and cost of analysis of what number of singular values needed for reconstruction so there is no compromise in image quality and essential features are preserved. Model Pruning reduces number of hidden neurons using a pruning factor for convolution layers and the next two fully connected layers. This helps in reduction of model

size, reduction in number of filters, number of learnable parameters and faster training and inference times. This also makes it easy to deploy the model on small scale devices. The disadvantage here is this is a quite simple, uninformed and naive approach on model pruning and requires lots of iterations in fine-tuning it. Thus, a combination of SVD and Model Pruning can help in optimizing storage and compute of a Deep Neural Network based Application used in Skin Cancer detection for Deep and complex architectures provided a comprehensive analysis and fine-tuning is done by scientists and researchers.

References

[1] W. C. R. F. International, "Skin cancer statistics," *WCRF International*, 2020. [Online]. Available: https://www.wcrf.org/cancer-trends/skin-cancer-statistics/

[2] W. Salehi, S. Khan, G. Gupta, B. Alabduallah, A. Almjally, H. Alsolai, T. Siddiqui, and A. Mellit, "A study of cnn and transfer learning in medical imaging: Advantages, challenges, future scope," vol. 15, p. 5930, 03 2023.

[3] B. Shetty, R. Fernandes, A. Rodrigues, R. Chengoden, S. Bhattacharya, and K. Lakshman, "Skin lesion classification of dermoscopic images using machine learning and convolutional neural network," *Scientific Reports*, vol. 12, 10 2022.

[4] K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Skin cancer classification using deep learning and transfer learning," in *2018 9th Cairo International Biomedical Engineering Conference (CIBEC)*, 2018, pp. 90–93.

[5] M. Ali, M. S. Miah, J. Haque, M. M. Rahman, and M. Islam, "An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models," *Machine Learning with Applications*, vol. 5, p. 100036, 04 2021.

[6] P. Naga Srinivasu, J. Gnana Siva Sai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm," *Sensors*, 04 2021.

[7] D. N. A. Ningrum, S.-P. Yuan, W.-M. Kung, C.-C. Wu, I.-S. Tzeng, C.-Y. Huang, J. Y.-C. Li, and Y.-C. Wang, "Deep learning classifier with patient's metadata of dermoscopic images in malignant melanoma detection," *Journal of Multidisciplinary Healthcare*, vol. 14, pp. 877 – 885, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233408343

[8] P. Tschandl, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," 2018. [Online]. Available: https://doi.org/10.7910/DVN/DBW86T

[9] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.