INDIAN STATISTICAL INSTITUTE

Severity Gradation of Psoriatic Plaques using Ensemble of Deep Convolutional Neural Networks

by

Sayan Chatterjee

Under the Guidance of

Prof. Utpal Garain
CVPR Unit,ISI Kolkata

A thesis submitted in partial fulfillment for the degree of M.Tech in Computer Science

July 2018

Declaration of Authorship

I, **Sayan Chatterjee**,registered as a student of **M.Tech CS Program, ISI Kolkata**, declare that this thesis titled, **Severity Gradation of Psoriatic Plaques using Ensemble of Deep Convolutional Neural Networks** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an M.Tech degree at this institute.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

Severity gradation of psoriatic plaque is important for the estimation of Psoriasis Area Severity Index (abbreviated as PASI) that facilitates the diagnosis as well as the treatment of the disease. Severity assessment by manual examination of the plaques of the diseased person or by observation of images of the affected skin area suffers from inter and intra-observer variability.Therefore, automated techniques can not only lead to reduced effort but also can reduce inaccuracy, provided a sufficient amount of correctly annotated data is available. Recent advancements of deep learning in computer vision and medical imaging domain has led to a substantial improvement of performance over traditional image processing techniques.This work has proposed five new methods for severity scoring in order to bring about improvement in accuracy from the baseline. The first method uses a small-sized CNN with very less number of parameters for classification; the second method uses a two-stage classification approach based on CNN; the third method is a pair wise CNN based classification method; the fourth one is a majority voting based CNN ensemble and fifth one is a stacking or super-learning based CNN ensemble. Other approaches like Texture CNN based classification have also been tried and depicted in the thesis as well.

Acknowledgements

Firstly, I would like to convey my deepest gratitude to my guide **Prof. Utpal Garain**, CVPR Unit, ISI Kolkata. I thank him for giving me ideas, criticizing and appreciating my efforts when necessary. I would also like to thank my senior and colleague **Anabik Pal**, SRF, CVPR Unit, ISI, Kolkata for bearing with me and helping me by giving important suggestions throughout the year. I would also like to thank ISI Kolkata, for providing me the opportunity to be acquainted with so many great minds, my teachers and my friends.I thank all of them for making me a part of the family. I thank my parents, whose constant support made me the person I am now.

Certification

This is to certify that this thesis titled **Severity Gradation of Psoriatic Plaques using Ensemble of Deep Convolutional Neural Networks** submitted by **Sayan Chatterjee**, embodies the work done under my supervision.

Prof. Utpal Garain,
CVPR Unit,
ISI Kolkata

Contents

List of Figures

List of Tables

Abbreviations

Acronym	**W**hat (it) **S**tands **F**or
CNN	**C**onvolutional **N**eural **N**etwork
CLAHE	**C**ontrast **L**imited **A**daptive **H**istogram **E**qualization
GLCM	**G**ray **L**evel **C**occurrence **M**atrix
KNN	**K** **N**earest **N**eighbour
LBP	**L**ocal **B**inary **P**attern
MLP	**M**ulti **L**ayer **P**erceptron
MTL	**M**ulti **T**ask **L**earning
PASI	**P**soriasis **A**rea **S**everity **I**ndex
SGD	**S**tochastic **G**radient **D**escent
STL	**S**ingle **T**ask **L**earning
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

1.1 Psoriasis

Psoriasis is a chronic, autoimmune, inflammatory skin disease that causes elevated, reddish patches with scale formation on human skin surface.The skin lesions may vary in severity from minor localized patches to patches that cover the entire body. Psoriasis is a non-contagious disease, i.e it cannot transmit from one person to another.The causes of psoriasis are not yet completely known.Experts do not consider it to be purely a skin disorder and according to them it can also have an impact on other organ systems. It is generally considered to be a genetic disease influenced by environmental factors.Psoriasis develops when the immune system mistakes a normal skin cell for a pathogen, and sends out faulty signals that cause overproduction of new skin cells. The difficulty faced in the treatment of psoriasis arises from the fact that it's nature varies among patients by a large extent.So one treatment methodology will not work for all sorts of patients affected by the same disease.

1.2 Problem Statement

Due to the inherent variability of nature of the skin lesions, Psoriasis needs a quantitative assessment of severity.There exists an index named Psoriasis Area Severity Index(abbreviated as PASI) [2] for this purpose.PASI considers two quantities for severity assessment: 1)percentage of the body surface area affected by the disease and 2)severity of the plaques formed on the skin.The four body parts:(i) head(h), (ii) trunk(t), (iii) upper extremity(u) and (iv) lower extremity(l) are considered.The extent of area involved in each part $(A_h, A_u$ and $A_l)$ is given a score between 0-6.The severity of the plaques is measured based on three parameters: degree of redness or erythema$(E_h, E_t, E_u$ and $E_l)$,thickness or the induration$(I_h, I_t, I_u$ and $I_l)$and scaling$(S_h, S_t, S_u$ and $S_l)$and given a value between 0-4 according to the level of severity as shown in the table 1.1.The severity scores according to the area affected is given in table 1.2.

Finally the complete severity score is given by the formula: $0.1(E_h + I_h + S_h)A_h + 0.2(E_u + I_u + S_u)A_u + 0.3(E_t + I_t + S_t)A_t + 0.4(E_l + I_l + S_l)A_l$. The PASI score varies in gradation of 0.1 units from 0 to 72.0. Patients having a PASI index > 10 are considered to be suffering from a severe form of this disease.

Severity(%)	Score
Absent	0
Mild	1
Moderate	2
Severe	3
Very Severe	4

TABLE 1.1: Scoring of each Severity Parameter

Area Involved(%)	Score
0	0
1-10	1
10-29	2
30-49	3
50-69	4
70-89	5
90-100	6

TABLE 1.2: Severity Scoring on the Basis of the Percentage of Body Surface Area Affected

The objective of the work is to grade the severity of the plaques according to the three parameters of the skin lesions (erythema, induration and scaling).

1.3 Related Works

1.3.1 Severity Grading of Psoriatic Plaques using Deep CNN based Multi-task Learning (Pal et al, ICPR 2016)

Complete severity assessment on the basis of all three skin parameters using deep learning based approach is quite a new topic explored first by Pal et al in 2016. The work was published in International Conference of Pattern Recognition, 2016. They used the dataset collected by themselves for assessment purpose.

The dataset used for this purpose are RGB images collected by layman photographers in an uncontrolled environment with different viewing angle,distance and varying background.The images were cropped to a size of (227 $\times$ 227) for feeding them into the CNN.The data set had been annotated manually by a domain expert. Each image has 3 annotations corresponding to three parameters (erythema, induration and scaling) as mentioned. A total of 707 images had been collected.

For classification, i.e severity grading purpose, a deep convolutional neural network based multi-task learning approach has been explored. In MTL approaches, the network models learn multiple tasks but share common layers. Moreover, it has less memory requirement leading to reduced testing time.These models also have the flexibility of handling different error functions for different tasks.The network architecture used in this approach is as shown in the Figure 1.1.

The network uses a shared sub-net and three different sub-nets for classification according to the three different parameters as shown in Figure1.1. Images of size (227$\times$227) is given as input to the shared sub-net of the MTL framework and the produced feature maps by the shared sub-net are given as input to the three different sub-nets predicting three different scores for three different parameters.

The shared sub-net in the architecture proposed in this work uses five convolutional+ReLU layers, three max-pooling and two normalization layers with number of parameters (feature maps and kernel sizes) as mentioned in the figure. The loss function used by this approach was the cross-entropy loss function.

For evaluation of the model, a 7-fold cross validation had been done and for each fold the validation accuracy had been measured. The average of the accuracy values over all the folds is the accuracy of the proposed model.

Several experiments had been performed to compare results with the proposed approach.For obtaining the results using an STL framework only one sub-net had been used at a time. Moreover, local binary pattern feature extraction followed by a traditional machine learning based classifier like support vector machines and KNN (K-Nearest Neighbour) had been performed. Deep CNN based approach has certainly outperformed all other methods by a large margin as shown in the table.

Method	Erythema	Scaling	Induration
LBP+KNN	24.2857	28.1429	34.5714
LBP+SVM	39.6040	38.8967	45.6860
FC6+KNN	25.8571	25.0000	35.8571
FC6+SVM	53.6068	51.2023	57.2843
FC7+KNN	26.8571	27.7143	32.0000
FC7+SVM	57.5672	49.7878	58.9816
STL	59.6888	58.9816	60.6789
MTL	60.6789	54.8797	61.1032

Table 1.3: Experimental Results as Reported in the Work [1]

The model was reported to be trained using a batch size of 256 images with SGD optimizer having momentum value 0.9, weight decay 0.0005 with learning rate 0.001. The learning process was said to be stopped when the network parameters got saturated (when the training error and the objective function did not change significantly for more than 10 epochs).In case of the MTL framework, it was said to be found that 200 epochs are sufficient for the parameters to get saturated.

An accuracy measure allowing (± 1) deviation from the predicted score had been analyzed as the severity scoring suffers from inter and intra observer variability. It reported an accuracy of 93.64% for erythema,93.78% for scaling and 93.78% for induration for the MTL approach proposed in this work.

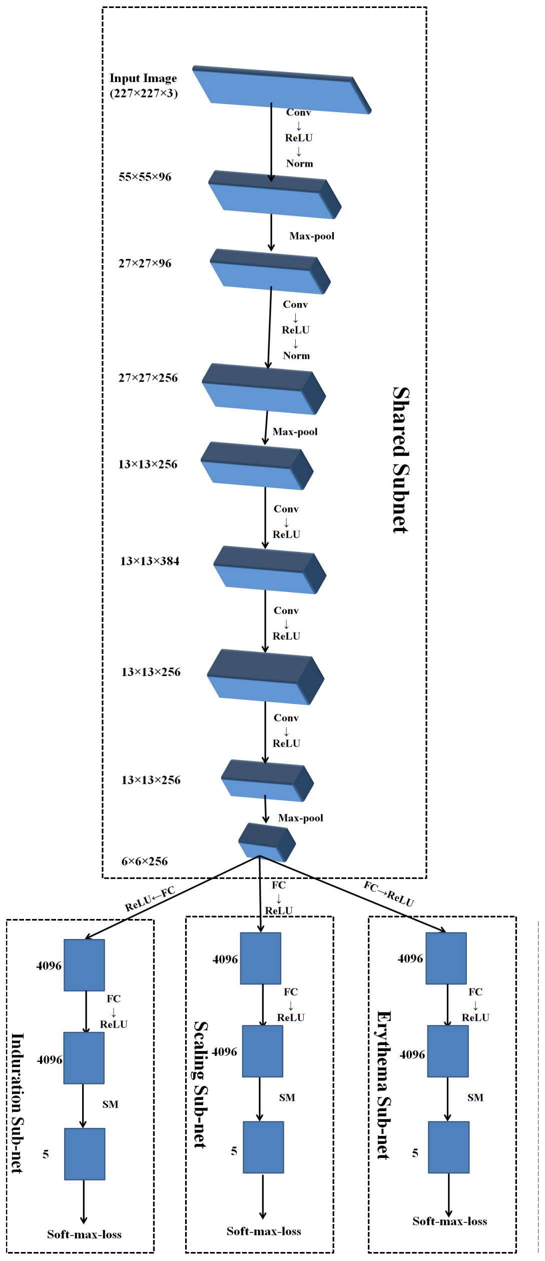


FIGURE 1.1: DCNN structure used for the MTL framework

1.3.2 Other Works

As already mentioned, automated severity assessment of psoriatic plaques is not a well explored topic. A few approaches for segmentation of psoriasis images had been explored ([3], [4], [5], [6]). Taur et al [3] have proposed a multiresolution based orthogonal subspace technique for segmentation of images, Lu et al have proposed Markov Random Field and SVM based approaches for segmentation. Bogo et al[5] had tried to locate lesional plaques based on chromatic information and then expanding these zones to achieve accurate segmentation through Geometric Active Contours method.Pal et al [6] had applied a mixture model based color clustering approach for segmentation.Ring, Jacques, and Kontinen [7] have provided an interactive method for segmenting psoriasis lesions from normal skin using color thresholding. Automatic erythema gradation had also been attempted but there had been no effort for grading erythema, scaling and induration altogether before the work done by Pal et al [1].

Chapter 2

Proposed Methods and Results

2.1 Dataset

The dataset used is the same as the dataset used in the work [1] done by Pal et al,i.e it has 707 number of images of psoriasis plaques having 3 labels corresponding to erythema,scaling and induration respectively.The images are RGB images having dimension (224×224×3).A few samples of the images for each parameter are shown in table 2.1

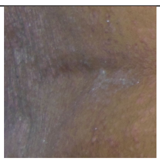
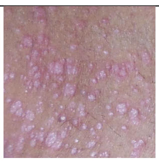
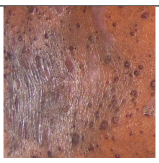
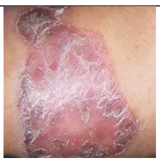
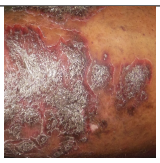
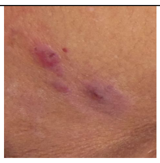
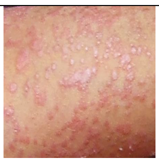
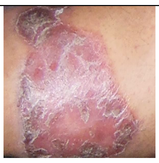
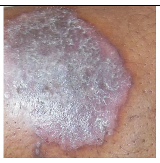
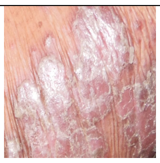
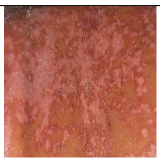
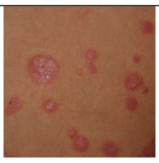
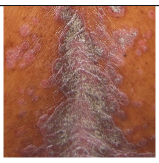
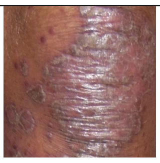
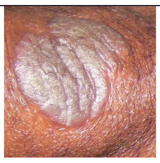
Severity Factor	0	1	2	3	4
Erythema					
Scaling					
Induartion					

Table 2.1: A few sample of the images with the corresponding severity scores. The erythema score increases as the skin becomes redder.The scaling score increases as the skin becomes more silvery. The induration score increases if the patches become more elevated.

2.2 The Primary Challenges to Solve this Problem

Data Insufficiency The number of image samples in the data is substantially low (707) for a deep neural network to be trained from scratch.It is very difficult to avoid overfitting.

Data Imbalance The number of data samples corresponding to each severity score is not the same.The number of data samples per class for each parameter is shown in the table 2.2.There is a chance for the model to get biased towards the majority classes.

Parameter	class0	class1	class2	class3	class4
Erythema	94	212	231	144	26
Scaling	86	239	237	121	24
Induration	116	297	251	43	0

Table 2.2: Number of Data Samples per Class

Noise and Unwanted Components As the data have been captured in uncontrolled environment, the illumination condition is not uniform for all the images.Some images have glares due to light falling on it and in some cases the presence of skin hair makes it difficult to locate the lesion.Existing image processing techniques cannot remove the glare and skin-hair totally from psoriasis plaque images.Even though we try extracting handcrafted features from the images, chances are less that some meaningful information indeed comes out.

Here it is to be noted that data augmentation techniques do not help much here. Conventional image augmentation techniques include flipping (vertical and horizontal), rotation with different angles, adding Gaussian noise, power law transforms etc. Among the techniques mentioned, only flipping and rotation can be applied here. The severity grading depends on the contrast of color between the affected area of the skin and the healthy part of the skin. With approaches like power law transforms, color properties can get destroyed. The data itself is not free from noise. So adding Gaussian noise makes no sense.

There have been efforts in the literature for processing psoriasis images so that some meaningful feature extraction can be carried out like removal of skin hair from 2D psoriasis images by George et al [8].

2.3 Design of a Smaller Sized CNN for Classification

As the number of data samples is too low for a very deep CNN to be trained from scratch, a different architecture of CNN with much less number of parameters has been designed for classification purpose.In the previous work [1] the network used was inspired from AlexNet which has a very high number of trainable parameters, the number of feature maps per layer was also very high. The CNN designed, although has same number of convolutional layers but the number of feature maps per layer and number of units in last two fully connected layers is much less. So the total number of parameters has been reduced to 1,572,261 from that of AlexNet which has 58,301,829 number of parameters.The detailed design of the network is as shown in the table 2.3.

Layer	Output Shape	No of Parameters
Input	(224,224,3)	0
Conv2D	(224,224,32)	896
BatchNorm	(224,224,32)	128
ReLU	(224,224,32)	0
Maxpool	(112,112,32)	0
Conv2D	(112,112,32)	25632
BatchNorm	(112,112,32)	128
ReLU	(112,112,32)	0
Maxpool	(56,56,32)	0
Conv2D	(56,56,64)	51264
BatchNorm	(56,56,64)	128
ReLU	(56,56,64)	0
Maxpool	(28,28,64)	0
Conv2D	(28,28,64)	102464
BatchNorm	(28,28,64)	256
ReLU	(28,28,64)	0
Maxpool	(7,7,64)	0
Conv2D	(7,7,128)	204928
BatchNorm	(7,7,128)	512
ReLU	(7,7,128)	0
Maxpool	(3,3,128)	0
Flatten	(1152,)	0
Dropout(0.5)	(1152,)	0
FC	(1024,)	1180672
Dropout(0.5)	(1024,)	0
Softmax	(5,)	5125

TABLE 2.3: CNN1 Architecture with Number of Parameters

Here the image of size (224×224×3) is given as input to the network.Then it is passed through a convolution layer with [3 × 3] kernel. The number of feature map for this layer is 32.A Batch Normalization layer followed by a ReLU activation layer is applied on the feature map. Later the feature map is reduced to a size (112×12×32) through the application of a Max Pooling layer of kernel size [2 × 2] with stride 2.This block of Convolution, Batch Normalization and ReLu followed by a Max Pooling has been applied 5 times to reduce the image size to (3×3×128). Then it has been flattened and a fully connected layer with number of units 1024 followed by a dropout layer with a dropout probability of 0.5 has been applied .Lastly a softmax layer has been connected to generate class probabilities.From the 2nd convolutional layer onwards, the kernel size has been increased to [5×5].As kernel size is also a hyperparameter, different settings have been tried. This setting of the kernel size worked best.

The number of layers in the network cannot be reduced much because in that case MaxPooling operation with a higher stride has to be used and due to that there may be loss of information.

The network was trained for 100 epochs with SGD optimizer with learning rate 0.001, momentum 0.09, a decay of .000001 and batch size 2. It has been observed that the models get saturated at about 80 epochs.The saturation point may vary from one fold to another; that is why 100 epochs has been used. The model has been saved epoch by epoch using model checkpoint. The plot of validation and training accuracy vs the number of epochs has been shown in figure 2.1 for one trial run for 350 epochs.The validation accuracy scores for 7 fold cross validation are tabulated in the table 2.4.

So it shows that this model performs a bit poorer than that reported in the work by Pal et al.The reason may be two; firstly the model used in the paper was a pre-trained AlexNet.The hyperparameter tuning of the proposed model does not match that of the pre-trained AlexNet and secondly, the number of parameters in the proposed model is also very less.It is important to note that the result reported for the proposed model is the best one obtained after an extensive hyperparameter tuning. Many different learning

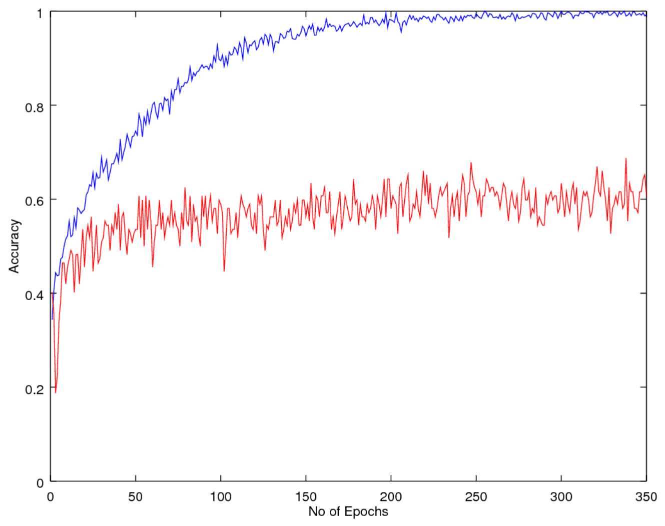


FIGURE 2.1: One Instance of a Trial Run showing Training and Validation Accuracy Variation over 350 epochs

Fold No	Accuracy(%)		
	Erythema	Scaling	Induration
1	55.45	58.42	62.38
2	49.50	54.45	60.39
3	60.39	52.47	63.37
4	57.42	54.45	61.38
5	56.43	50.49	58.42
6	61.38	56.43	59.41
7	58.41	56.44	58.42
average	57.01	54.74	60.54

TABLE 2.4: Accuracy Values across Different Folds for CNN1

rate values, regularization constants, feature map numbers etc have been tried with and the best one is reported. Moreover, other optimizers like RMSprop, AdaGrad, Adam etc have been tried with but SGD seemed to work best in this problem.

2.4 Two Stage Classification: Five Class Base CNN followed by Multiple Binary CNNs

The confusion matrices of the CNN for each fold was analyzed to guess whether the classifier is predicting randomly or they are predicting within (±1) of the actual classes and it was found that if we allow a (±1) tolerance for accuracy calculation it can be increased even up to 94%. Therefore,most of the misclassifications are within the actual

Fold No	Confusion Matrix		
	Erythema	Scaling	Induration
1	$\begin{bmatrix}9&3&2&1&0\\1&21&9&2&0\\1&10&13&13&0\\0&4&5&4&0\\0&0&1&0&2\end{bmatrix}$	$\begin{bmatrix}12&1&1&0&1\\6&17&1&5&2\\1&9&12&11&1\\1&1&1&15&3\\0&0&0&0&0\end{bmatrix}$	$\begin{bmatrix}13&3&1&0\\2&32&4&1\\0&10&18&0\\0&1&6&0\end{bmatrix}$
2	$\begin{bmatrix}8&4&1&0&0\\0&24&5&0&2\\1&15&15&1&3\\0&2&7&3&6\\0&0&2&1&1\end{bmatrix}$	$\begin{bmatrix}11&2&1&0&0\\6&9&13&2&0\\2&6&18&8&0\\0&1&3&17&1\\0&0&0&1&0\end{bmatrix}$	$\begin{bmatrix}13&4&3&0\\3&26&11&0\\0&14&22&0\\0&2&3&0\end{bmatrix}$
3	$\begin{bmatrix}13&1&1&1&0\\6&13&17&0&0\\2&3&27&3&1\\0&2&5&4&0\\0&0&1&0&1\end{bmatrix}$	$\begin{bmatrix}10&3&1&1&0\\5&16&7&4&0\\1&9&14&3&1\\0&3&5&15&0\\0&1&0&2&0\end{bmatrix}$	$\begin{bmatrix}9&3&5&0\\2&37&7&0\\2&11&8&2\\0&3&2&4\end{bmatrix}$
4	$\begin{bmatrix}7&1&0&0&0\\2&29&7&2&0\\1&12&20&2&0\\0&4&7&4&0\\0&1&0&1&1\end{bmatrix}$	$\begin{bmatrix}4&6&0&0&0\\1&26&9&0&0\\0&7&16&4&0\\0&4&16&13&0\\0&2&1&2&0\end{bmatrix}$	$\begin{bmatrix}8&5&0&1\\6&24&13&1\\5&9&27&0\\0&0&2&1\end{bmatrix}$
5	$\begin{bmatrix}6&1&0&0&0\\1&25&1&2&1\\0&6&23&4&0\\0&4&8&3&0\\0&0&2&1&0\end{bmatrix}$	$\begin{bmatrix}3&4&0&0&0\\1&20&11&3&0\\0&10&20&5&0\\0&3&8&8&1\\0&0&1&1&2\end{bmatrix}$	$\begin{bmatrix}5&4&2&0\\0&21&15&1\\0&15&29&4\\0&0&3&2\end{bmatrix}$
6	$\begin{bmatrix}10&0&0&0&1\\4&27&3&4&0\\1&9&13&1&1\\0&5&5&9&1\\0&0&1&4&2\end{bmatrix}$	$\begin{bmatrix}12&3&0&2&0\\0&13&6&4&0\\1&6&19&9&0\\0&4&4&11&0\\0&0&0&7&0\end{bmatrix}$	$\begin{bmatrix}10&7&1&0\\0&32&8&0\\0&14&15&0\\0&6&7&1\end{bmatrix}$
7	$\begin{bmatrix}11&3&2&0&0\\1&14&8&1&0\\1&9&20&4&2\\0&3&7&12&1\\0&0&1&0&1\end{bmatrix}$	$\begin{bmatrix}13&1&0&2&0\\3&16&5&1&0\\1&11&21&5&0\\0&2&6&8&0\\0&1&2&2&1\end{bmatrix}$	$\begin{bmatrix}10&7&2&0\\2&31&7&1\\0&19&16&1\\0&1&2&2\end{bmatrix}$

Table 2.5: Confusion Matrices across Different Folds after Classsification using CNN1

class,the previous one and the next one. The confusion matrices are shown in the tables.In each confusion matrix the rows are the actual classes and columns are the predicted ones.

As most of the predicted classes are within (± 1) of the actual classes,an algorithm exploiting this property has been designed to boost the performance of the classifiers.The algorithm will be effective where the number of classes are small. Otherwise,it will be computationally expensive to train a large number of binary CNNs.The algorithm goes as follows:

1. Train 7 binary classifiers for classifying class pairs (0, 1), (1, 2), (2, 3), (3, 4), (0, 2), (1, 3), (2, 4)i.e if there are total N number of classes,train classifiers to classify $(n-1, n)$, $(n, n+1)$ and $(n, n+2)$ class-pairs for each n (except 0 and $N-1$). These binary classifiers have to be at least as accurate as the base classifier for performance improvement.
2. For each data sample do the following.
   - Feed the data sample to the N class base classifier.Say the predicted class label is n.
   - From the confusion matrices, it can be said there is almost 90% probability that the actual class label is within the triplet $(n-1, n, n+1)$.Feed the data sample to all three binary classifiers $(n-1, n)$, $(n, n+1)$, $(n-1, n+1)$
   - Among the three predictions from the three binary classifiers,take majority voting. The class getting the majority vote is the predicted class for this 2nd level classifier.

It is like, the base 5 class CNN is providing the interval that the actual class may be in and the voting of the second stage binary CNNs are trying to figure out the correct class labels.

So following the strategy as mentioned above, 7 CNNs were designed to classify the class pairs (0, 1), (1, 2), (2, 3), (3, 4), (0, 2), (1, 3), (2, 4).For a particular fold, among the training data samples, those which belong to the particular class-pair that we are going to classify, have been taken as training data.Other data samples are ignored.For validation data also the same approach has been taken.The CNNs were designed by transfer learning from the actual 5 class base classifier as depicted in section 2.3 so that the performance of the binary CNNs do not get worsened. The last 5 level softmax layer from the CNN described in table 2.3 has been removed and a fully connected layer for binary classification with signmoid activation has been added. The initial weights were the weights learned previously and then it is fine tuned for the particular binary classification problem.

The binary CNNs have been trained for 80 epochs with SGD optimizer with learning rate .001, momentum 0.09, decay 1e-6 and batch size 2. Applying the above mentioned algorithm, around 1-4 percent improvement of average accuracy over 7 folds was noticed.The confusion matrices after the application of the algorithm are shown in the table 2.6.The accuracy values before and after the application of algorithms are shown in table 2.7 and 2.8.So it is evident from table 2.7 and 2.8 that we are getting $\mathbf{1.705}\%$ performance improvement for erythema, $\mathbf{3.96}\%$ improvement for scaling and $\mathbf{1.14}\%$ performance improvement for induration.The values are slightly different than that shown in the previous section because it had to be retrained on a different machine due to some issues.

Here it is to be noted that like the binary CNNs that have been designed to boost performance, ternary CNNs were also designed for the same task.Those ternary CNNs were designed just like the previous binary ones by replacing the last 5 level softmax layer by a 3 level softmax layer. But the ternary CNNs did not perform as good as the binary ones because the misclassification is highest among the (± 1) of the predicted class.That is why the aggregation at the 2nd level was not performed.

Fold No	Confusion Matrix		
	Erythema	Scaling	Induration
1	$\begin{bmatrix} 11 & 1 & 2 & 1 & 0 \\ 1 & 19 & 11 & 2 & 0 \\ 1 & 10 & 20 & 6 & 0 \\ 0 & 3 & 9 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 13 & 0 & 1 & 0 & 1 \\ 7 & 17 & 1 & 4 & 2 \\ 1 & 8 & 17 & 7 & 1 \\ 1 & 1 & 4 & 12 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 14 & 1 & 2 & 0 \\ 3 & 21 & 16 & 0 \\ 0 & 10 & 26 & 0 \\ 0 & 0 & 5 & 0 \end{bmatrix}$
2	$\begin{bmatrix} 11 & 1 & 1 & 0 & 0 \\ 0 & 21 & 8 & 0 & 2 \\ 0 & 10 & 21 & 0 & 3 \\ 0 & 3 & 5 & 4 & 6 \\ 0 & 0 & 2 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 11 & 3 & 0 & 0 & 0 \\ 7 & 12 & 9 & 2 & 0 \\ 2 & 5 & 21 & 6 & 0 \\ 0 & 1 & 1 & 19 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 13 & 4 & 3 & 0 \\ 3 & 21 & 16 & 0 \\ 0 & 10 & 26 & 0 \\ 0 & 0 & 5 & 0 \end{bmatrix}$
3	$\begin{bmatrix} 14 & 0 & 1 & 1 & 0 \\ 6 & 16 & 14 & 0 & 0 \\ 2 & 5 & 25 & 3 & 1 \\ 0 & 2 & 6 & 3 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 10 & 3 & 1 & 1 & 0 \\ 5 & 14 & 11 & 2 & 0 \\ 1 & 8 & 16 & 2 & 1 \\ 0 & 2 & 7 & 14 & 0 \\ 0 & 0 & 1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 10 & 6 & 1 & 0 \\ 2 & 36 & 8 & 0 \\ 2 & 10 & 9 & 2 \\ 0 & 2 & 3 & 0 \end{bmatrix}$
4	$\begin{bmatrix} 7 & 1 & 0 & 0 & 0 \\ 2 & 29 & 7 & 2 & 0 \\ 1 & 12 & 20 & 2 & 0 \\ 0 & 4 & 7 & 4 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 6 & 4 & 0 & 0 & 0 \\ 1 & 24 & 11 & 0 & 0 \\ 0 & 5 & 19 & 3 & 0 \\ 0 & 6 & 13 & 14 & 0 \\ 0 & 1 & 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 & 0 & 1 \\ 6 & 28 & 9 & 1 \\ 5 & 15 & 21 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$
5	$\begin{bmatrix} 7 & 0 & 0 & 0 & 0 \\ 1 & 26 & 9 & 0 & 1 \\ 1 & 10 & 21 & 2 & 0 \\ 0 & 4 & 7 & 10 & 0 \\ 0 & 0 & 1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 & 3 & 0 & 0 & 0 \\ 1 & 19 & 13 & 2 & 0 \\ 1 & 7 & 22 & 5 & 0 \\ 0 & 2 & 8 & 9 & 1 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 7 & 2 & 2 & 0 \\ 0 & 21 & 15 & 1 \\ 0 & 14 & 30 & 4 \\ 0 & 0 & 3 & 2 \end{bmatrix}$
6	$\begin{bmatrix} 10 & 0 & 0 & 0 & 1 \\ 5 & 26 & 4 & 3 & 2 \\ 1 & 10 & 13 & 0 & 1 \\ 0 & 3 & 9 & 7 & 1 \\ 0 & 0 & 1 & 4 & 2 \end{bmatrix}$	$\begin{bmatrix} 14 & 1 & 1 & 1 & 0 \\ 2 & 12 & 6 & 2 & 1 \\ 1 & 6 & 21 & 7 & 0 \\ 0 & 3 & 4 & 10 & 2 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 12 & 5 & 1 & 0 \\ 1 & 35 & 4 & 0 \\ 0 & 18 & 10 & 0 \\ 0 & 4 & 7 & 3 \end{bmatrix}$
7	$\begin{bmatrix} 11 & 3 & 2 & 0 & 0 \\ 1 & 12 & 10 & 1 & 0 \\ 1 & 6 & 24 & 3 & 2 \\ 0 & 3 & 7 & 12 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 13 & 1 & 1 & 0 & 1 \\ 3 & 17 & 5 & 0 & 0 \\ 1 & 12 & 24 & 1 & 0 \\ 0 & 3 & 7 & 6 & 0 \\ 0 & 1 & 2 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 10 & 6 & 3 & 0 \\ 3 & 31 & 5 & 2 \\ 0 & 18 & 17 & 1 \\ 0 & 1 & 2 & 2 \end{bmatrix}$

TABLE 2.6: Confusion Matrices after Two Stage Classification

Fold No	Accuracy(%)		
	Erythema	Scaling	Induration
1	48.51	55.45	62.38
2	50.50	54.45	60.39
3	57.43	54.45	63.37
4	60.39	58.41	59.41
5	62.37	52.47	56.43
6	60.39	54.45	57.42
7	57.42	58.41	58.42
average	56.715	55.44	59.68

Table 2.7: Accuracy Values across Different Folds after Classification with CNN1

Fold No	Accuracy(%)		
	Erythema	Scaling	Induration
1	52.47	58.41	66.34
2	57.42	62.37	59.41
3	58.42	53.47	64.35
4	60.39	63.36	57.43
5	63.37	55.45	59.41
6	60.39	62.37	59.40
7	59.41	60.39	59.42
average	**58.42**	**59.40**	**60.82**

Table 2.8: Accuracy Values across Different Folds after Two Stage Classification

2.5 CNN based Pairwise Classification

Pairwise classification is a conventional machine learning approach where an N class problem is converted into a series of binary class problems.Many learning algorithms can handle or work better only on two class problems.In order to make them work in an N class setting, a class binarization technique is required.One solution for this is a one-against-all approach where one classifier for each class is constructed where the positive training examples are the data samples that belong to that particular class and the negative ones are formed by combining all other classes.Another approach is round-robin or pairwise classification.The idea is to transform the original N class problems into $\binom{N}{2}$ number of binary class problems i.e building one classifier for each pair of classes and later combining their predictions by some aggregation method. The later approach had been shown to produce more accurate results than the one-against-all approach for a wide variety of algorithms like SVM [9] or rule learning algorithms [10].Furnkranz [11] has claimed in his paper that pairwise classification can be used as an ensemble technique to obtain a performance improvement comparable to bagging or boosting. As the misclassification of the designed CNNs are mainly between the neighbouring classes, this pair wise classification technique has a potential to improve performance.The idea is the following:

1. Train $\binom{5}{2}$ i.e 10 CNNs for classifying the 10 possible class-pairs among the 5 classes.
2. For every test data
    - Feed it to each binary classifier and collect the 10 predictions.From the set of 10 predictions select the most probable prediction by majority voting.

Therefore, 10 classifiers were trained like that said in the above algorithm. For the algorithm applied in section 2.4, 7 classifiers were already trained. So 3 more classifiers were to be trained.The results after combining the predictions are shown in table 2.9.

Therefore, using this algorithm we are achieving **3.675%** performance improvement from the base 5-class CNN in case of erythema, in case of scaling we are getting a performance improvement of **5.36%** and in induration we get **1.82%** improvement in accuracy.This method even outperforms the baseline MTL result in case of scaling and induration by **5.93%** and **1.5568%** respectively.

Fold No	Accuracy(%)		
	Erythema	Scaling	Induration
1	54.45	60.39	66.33
2	56.44	66.33	60.39
3	62.37	56.43	67.33
4	59.41	62.37	61.39
5	66.34	54.45	60.39
6	64.35	63.36	59.41
7	58.40	62.37	63.37
average	**60.30**	**60.81**	**62.66**

Table 2.9: Accuracy Values after Classification with Pairwise CNN

2.6 CNN Ensembles

Ensemble learning is a way of combining multiple weak classifiers to produce a stronger classifier.So in ensembles, multiple different learners are generated and their outputs are combined. Different base learners can be generated in several ways like (i)using different algorithms,(ii)using different hyperparameters in the same algorithm,(iii) different representations or (iv) different training sets (in case of the learners which have a high variance).

The basic idea of ensembles is that if n independent weak learners make independent errors, the combination of their outputs will have a lower error probability.But there is a condition that each base weak learner should have an error rate which is greater than the random.Let us consider a situation where there are n learners and each of them have an accuracy rate, say p. In an ideal situation, if the learners are independent and they all agree on an input data, the confidence of the learners on the data will be $1 - (1 - p)^n$. So if n is high, the confidence on the data will be close to 1. In practical situations, as the accuracy of the learners are not 1, it is less likely that all the learners will agree on some particular data. Therefore as a solution, we can take the majority votes of the learners.There can be multiple ways in which the output of the learners can be combined.

For convolutional neural networks also, ensemble has been proved to be a very efficient technique for performance enhancement.Krizhevsky et al [12] showed that on ImageNet 2012 classification benchmark, their model with 5 CNNs achieved a top-1 error rate of 38.1% while a single model achieved 40.7% top-1 error rate.Zeiler and Fergus [13] showed that by the ensemble of 6 CNNs, they could reduce the top-1 error from 40.5% to 36.0%.

For solving this problem, 5 CNNs were used. It is to be noted that for the ensembling to be effective, the networks have to be different so that they make errors in independent manners. So 5 different architectures had to be used.

2.6.1 CNN Model 1

The first model is the same as that described in section 2.3.

2.6.2 CNN Model 2

Architecture for CNN-2 is described in table 2.10.The difference in the architecture has been achieved by using one less number of layers than the previous one.Here 4 such convolutional, batch normalization and ReLU layer have been used with the same sized kernel and feature maps as described in case of the first CNN. Only difference is that here the feature map (28×28×64) has been MaxPooled to (14×14×64) and then it has been flattened to feed to the fully connected layers. That is why the number of trainable parameters has increased to 13,032,229 from that of the previous CNN.

2.6.3 CNN Model 3

The architecture of the 3rd CNN has been described in table 2.11.The difference in the architecture has been created by using a global average pooling layer after the last convolutional layer. The global average pooling layer pools an average value from each of the 2 dimensional feature map of the convolutional layer and combines each of the value in a single vector. Thus the number of parameters can be reduced greatly by the use of this layer and so overfitting can also be reduced substantially. Here it is to be noted that even after the feature size has been reduced to (7×7×128) another layer of MaxPooling has been used to make it (2×2×128) and then after another convolution, Batch Normalization and ReLU block, the global average pooling layer has been added. The number of parameters of this model is 933,925.The kernel dimensions and other parameters that have been used in this network are the same as the previous one. It is to be noted that these variations in the architecture have been incorporated mostly

Layer	Output Shape	No of Parameters
Input	(224,224,3)	0
Conv2D	(224,224,32)	896
BatchNorm	(224,224,32)	128
ReLU	(224,224,32)	0
Maxpool	(112,112,32)	0
Conv2D	(112,112,32)	25632
BatchNorm	(112,112,32)	128
ReLU	(112,112,32)	0
Maxpool	(56,56,32)	0
Conv2D	(56,56,64)	51264
BatchNorm	(56,56,64)	128
ReLU	(56,56,64)	0
Maxpool	(28,28,64)	0
Conv2D	(28,28,64)	102464
BatchNorm	(28,28,64)	256
ReLU	(28,28,64)	0
Maxpool	(14,14,64)	0
Flatten	(12544,)	0
Dropout(0.5)	(12544,)	0
FC	(1024,)	12846080
Dropout(0.5)	(1024,)	0
Softmax	(5,)	5125

Table 2.10: CNN2 Architecture with Number of Parameters

to make the models different and independent.That the number of parameters doesn't grow too much has also been taken care of.

Layer	Output Shape	No of Parameters
Input	(224,224,3)	0
Conv2D	(224,224,32)	896
BatchNorm	(224,224,32)	128
ReLU	(224,224,32)	0
Maxpool	(112,112,32)	0
Conv2D	(112,112,32)	25632
BatchNorm	(112,112,32)	128
ReLU	(112,112,32)	0
Maxpool	(56,56,32)	0
Conv2D	(56,56,64)	51264
BatchNorm	(56,56,64)	128
ReLU	(56,56,64)	0
Maxpool	(28,28,64)	0
Conv2D	(28,28,64)	102464
BatchNorm	(28,28,64)	256
ReLU	(28,28,64)	0
Maxpool	(7,7,64)	0
Conv2D	(7,7,128)	204928
BatchNorm	(7,7,128)	512
ReLU	(7,7,128)	0
Maxpool	(2,2,128)	0
Conv2D	(2,2,128)	409728
BatchNorm	(2,2,128)	512
GlobalAveragePooling	(128,)	0
FC	(1024,)	1180672
Softmax	(5,)	5125

Table 2.11: CNN3 Architecture with Number of Parameters

2.6.4 CNN Model 4

The fourth CNN that has been used is a pre-trained ResNet50. Residual connections had been introduced by He et al [14] in ImageNet challenge 2015. They showed that by using residual connections/skip connections, more depth can incorporated in the network without the fear of gradient being vanished. They used a 150 layer Residual Deep Network for training the ImageNet dataset. The residual connections are like that shown in figure 2.2

The baseline "plain" models in the ResNet paper, as they have claimed, are based on the VGG-net [15] .They have used all equal sized [3×3] kernels.Upon this framework they have incorporated, what they have called the "shortcut connections".The shortcut

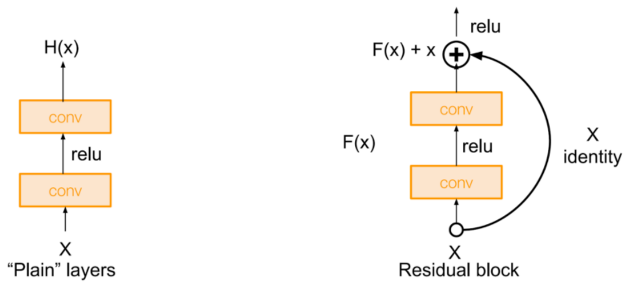


FIGURE 2.2: Skip Connections in ResNet

connections perform identity mapping and it's output is added with the main mapping $F(x)$. If the desired mapping is $H(x)$, then the stacked nonlinear layers should learn $F(x) := H(x) - x$. So the original mapping is recast into $F(x) + x$. They claimed that this residual mapping is easier to optimize than the original, unreferenced mapping.The identity mapping can be used when the input and output are of the same dimension.These identity mappings neither increase the number of parameters nor increase the computational complexity.Moreover, as the addition operation distributes the backpropagated gradient in equal values to the two branches,the gradient vanishing problem does not occur.They showed that by using a 152 layer single ResNet, they achieved a top-5 validation error of 4.49% in ImageNet dataset.

In this case, a ResNet50 model pretrained on ImageNet dataset has been used. The last softmax layer of the pretrained model has been removed and one 5 level softmax layer has been added for generation of the class probabilities. The whole model was fine tuned for the psoriasis dataset. The total number of parameters for this transfer learned model is 2,468,097 which is substantially low for a model with 50 layers. The reason for using a pre-trained model is that the number of data samples is very less. So transfer learning from a very deep network like ResNet50 is the only feasible idea.

2.6.5 CNN Model 5

The 5th CNN architecture that has been used is a pre-trained DenseNet121 model. DenseNet architecture was introduced by Huang et al [16] in CVPR 2017. Just like ResNet, DenseNet also addresses the problem of vanishing gradient with very deep architecture. Whereas ResNet uses shortcut connections from earlier layers to later layers, DenseNet ensures maximum information flow between the layers of the network by connecting all layers directly with each other. DenseNet does not combine features through summation, rather it combines features through concatenation. It's l^{th} layer has l inputs from all the feature maps of the preceding convolutional blocks.

Suppose, $x_0, x_1, x_2........x_{l-1}$ are the feature maps of all the preceding layers of the l^{th} layer. The l^{th} layer output then will be $x_l = F([x_0, x_1, x_2, ...x_{l-1}])$ where $[x_0, x_1, x_2, ...x_{l-1}]$ denotes the concatenation of the tensors $x_0, x_1, x_2....x_{l-1}$.This l^{th} layer's own feature maps are passed to all $L - l$ subsequent blocks.So in an L layer network, there are $L(L + 1)/2$ connections.They have termed this connections dense connections.These dense connections effectively reduces the number of parameters in the network, as there is no need to relearn the redundant feature maps.Also it has used narrow layers consisting of only 12 filters per layer.If each function F produces k feature maps, the number of feature maps in the input of the l^{th} layer will be $k_0 + k \times (l - 1)$, where k_0 is the number of channels in the input layer.The parameter k is called the "growth rate" of the network.They have claimed that a relatively small growth rate is sufficient to obtain state-of-the-art results in ImageNet dataset. A five layer DenseNet block with growth rate 4 has been depicted in the figure 2.3

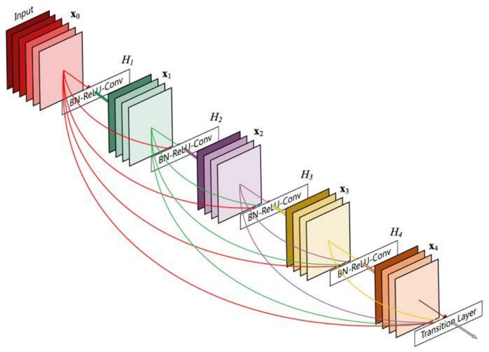


FIGURE 2.3: 5 Layer DenseNet Block with Growth Rate 4

Apart from the better parameter efficiency, other advantages of the DenseNet architecture are:

- Information and gradient flow through the network improves making it easy to train
- Dense connections has a regularizing effect which reduces overfitting on smaller training sets

To make the architecture independent from the previous four architectures, and to take the advantages of transfer learning from a pre-trained model, DenseNet 121 was chosen.The last softmax layer was removed and a 5 level softmax layer was added with the network. The total number of parameters of the used DenseNet is 7,042,629 which is noticeably less for a model with 121 layers.

All the 5 models were trained using SGD optimizer with learning rate .001, momentum 0.09, batch size 2, decay rate 1e-6 for 150 epochs. The models were saved using model checkpoints epoch by epoch at the best points obtained till 150 epochs. Thus the effect of fluctuations in validation accuracy can be avoided.

2.6.6 Ensemble by Majority Voting

The outputs from the 5 CNNs can be combined in multiple ways in case of ensembles. One way is majority voting. A sample from the test data is taken and it is fed to 5 networks for testing, they all issue a vote. The sample is assigned the majority class. The result of classification with 5 CNN ensembles combined by majority voting for erythema , scaling and induration is described in table 2.12, 2.13, and 2.14 respectively.

Fold No	Accuracy(%)					
	CNN-1	CNN-2	CNN-3	CNN-4	CNN-5	Ensemble
1	55.45	51.48	47.52	52.48	54.45	55.45
2	49.50	51.48	55.45	52.48	53.46	59.41
3	60.39	56.43	58.41	55.44	64.35	70.29
4	57.42	51.48	49.5	52.47	58.41	61.39
5	56.43	57.42	55.45	45.54	57.42	65.35
6	61.38	53.46	53.46	50.49	65.34	63.37
7	58.41	53.46	54.45	56.43	55.44	66.33
average	57.00	53.61	53.46	52.19	58.42	**63.08**

Table 2.12: Accuracy Values across Different Folds for Erythema for Ensemble with Majority Voting

Therefore,this majority voting based ensemble has brought about a noticeable improvement in the accuracy value for all the three parameters. It has outperformed the baseline AlexNet based MTL approach in case of erythema by **2.401%**, in case of scaling by **8.9103%**, and in case of induration by **4.3768**%.

Fold No	Accuracy(%)					
	CNN-1	CNN-2	CNN-3	CNN-4	CNN-5	Ensemble
1	58.41	53.46	50.41	53.46	55.45	63.36
2	54.45	56.43	55.45	60.39	66.33	66.34
3	52.47	48.51	53.46	55.44	55.44	59.41
4	54.45	58.41	55.45	57.42	66.33	70.29
5	50.49	45.54	49.51	46.53	52.47	52.47
6	56.43	55.44	61.38	57.42	67.32	68.32
7	56.43	56.44	49.50	56.43	59.41	66.33
average	56.43	53.46	53.61	55.30	60.39	**63.79**

Table 2.13: Accuracy Values across Different Folds for Scaling for Ensemble with Majority Voting

Fold No	Accuracy(%)					
	CNN-1	CNN-2	CNN-3	CNN-4	CNN-5	Ensemble
1	62.37	61.38	58.42	57.43	66.33	68.31
2	60.39	59.40	57.42	46.53	51.48	62.37
3	63.36	65.35	62.38	51.49	56.43	66.33
4	61.38	62.37	57.43	47.53	51.48	68.31
5	58.41	61.30	63.36	51.49	55.44	67.32
6	59.40	53.46	57.42	50.49	58.42	62.37
7	58.41	57.42	54.45	55.45	53.46	63.36
average	60.53	60.11	58.69	51.48	56.15	**65.48**

Table 2.14: Accuracy Values across Different Folds for Induration for Ensemble With Majority Voting

2.6.7 Ensemble by Stacking or Super-Learning

Ensemble of multiple algorithms can be performed by averaging, voting or weighted voting or weighted linear or non-linear combinations. In weighted combination, the weights can either be assigned arbitrarily or it can be proportional to accuracy or some other metric decided by the user. Stacking is a special case of weighted combination process where the weights are learned by the use of another learner called "metalearner".The idea of stacking was originally proposed by Wolpert(1992) [17]. It says that stacking works by reducing the bias of the generalizers with respect to the provided learning set.

Let there are N classifiers.The i^{th} classifier is denoted by $f_i : (x_1, x_2, x_3....x_p) \rightarrow \mathbb{R}/\mathbb{I})$, i.e it outputs either a class label or class probability distributions.The purpose of stacking is to learn a metalearner $h : (f_1(x), f_2(x), f_3(x)......., f_N(x)) \rightarrow \mathbb{R}/\mathbb{I})$ such that it optimizes the weights of the combinations of the outputs of the 1st level learners to produce the correct class labels or probabilities of the original problem. Each of the first level classifiers can be assumed to be producing some features, i.e class labels or probability distributions. These produced features can be stacked in a vector or matrix.These vectors/matrices and the corresponding class labels are the training data for the metalearner. Here a k-fold cross validation approach must be used to produce the class labels or probability values from the base learners. For each fold, the predicted class labels or the class-probabiliies from all the learners are stacked to produce training data for the metalearner.The training labels are the same as the original labels.

Stacking has been proved to work better than other ensemble techniques like bagging or boosting in various cases.It also reduces overfitting to a great extent.

In our case, as a metalearner, an MLP (Multi Layer Perceptron) has been used. Both the probability distributions and the class labels generated by the five CNNs have been combined separately to feed the metalearner.

The probability vectors generated by all the base learners have been stacked in a matrix to produce a feature corresponding to a particular image. During the 7 fold cross validation, for a particular fold, the trained model has been applied to the test data for that particular fold and the probability scores for 5 classes have been generated and these 5 probability vectors have been stacked into a matrix of size (5×5).These matrices are the inputs of the MLP model described in the table. The number of parameters for this MLP is 27129. The result across different folds for erythema, scaling and induration has been described in the table 2.16. The accuracy values are shown in table 2.17

Layer	Output Shape	No of Parameters
Input	(5,5)	0
Flatten	(25,)	0
FC	(512,)	13312
LeakyReLU	(512,)	0
Dropout	(512,)	0
FC	(512,)	13312
LeakyReLU	(512,)	0
Dropout(0.5)	(512,)	0
Softmax	(5,)	505

Table 2.15: MLP Architecture for Meta-Learning with Probability Scores

Fold No	Accuracy(%)		
	Erythema	Scaling	Induration
1	52.48	61.39	71.29
2	64.35	73.27	64.36
3	66.34	61.39	67.33
4	71.29	72.27	70.38
5	64.36	53.47	63.36
6	62.38	67.33	60.40
7	69.31	69.31	67.33
average	**64.36**	**65.49**	**66.34**

Table 2.16: Accuracy Values across Different Folds for Stacking with Probability Scores

Therefore, this approach has been able to bring in changes in the accuracy. It has outperformed the baseline AlexNet based MTL approach in case of erythema by **3.6811%**,in scaling by **10.61%** and in induration by **5.236%**.

The class labels predicted by all the base learners have been stacked in a vector to produce a feature corresponding to a particular image. During the 7 fold cross validation, for a particular fold, the trained model has been applied to the test data for that particular fold and the labels and the corresponding 5 element feature vector has been designed.These feature vectors are the inputs of the MLP model described in the table. The number of parameters for this MLP is 1105. The result across different folds for erythema, scaling and induration has been described in the table 2.18.

Layer	Output Shape	No of Parameters
Input	(5,)	0
FC	(100,)	600
LeakyReLU	(100,)	0
Softmax	(5,)	505

Table 2.17: MLP Architecture for Meta-Learning with Class Labels

Fold No	Accuracy(%)		
	Erythema	Scaling	Induration
1	64.36	63.36	73.27
2	72.27	61.39	65.35
3	64.36	72.27	71.29
4	70.29	69.31	69.31
5	59.40	71.30	65.35
6	72.27	65.35	61.39
7	66.34	62.38	66.34
average	**67.04**	**66.47**	**67.51**

Table 2.18: Accuracy Values across Different Folds for Stacking with Class Lables

Therefore, this approach has also been able to bring in a significant change in the accuracy. It has outperformed the baseline AlexNet based MTL approach in case of erythema by **6.36%**,in scaling by **11.59%** and in induration by **6.406%**. The labels for a few sample images have been shown in figure 2.6. The first set of labels (enclosed by the first parentheses) represents the ground truth labels and the second set of labels (enclosed by the second parentheses) represents the predicted class labels (in order of erythema, scaling and induration respectively).

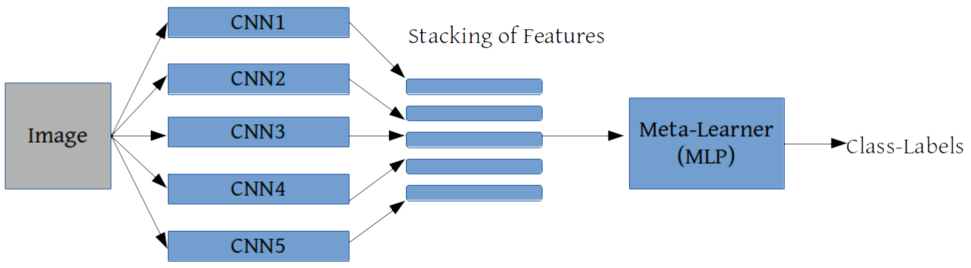
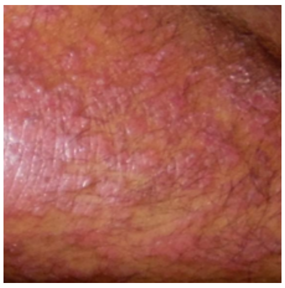
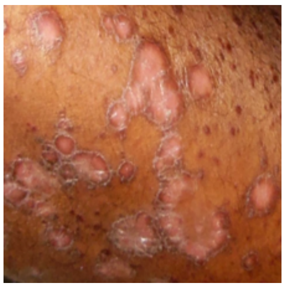
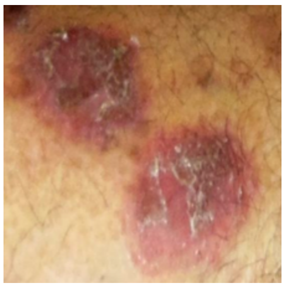
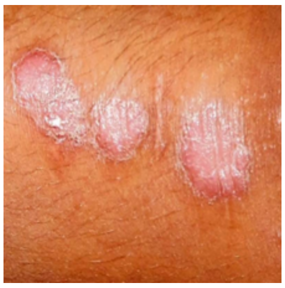


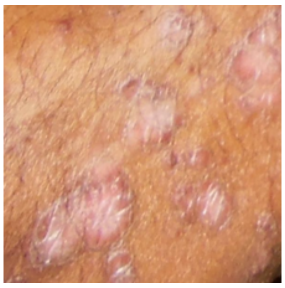
Figure 2.4: Stacked CNN Ensemble

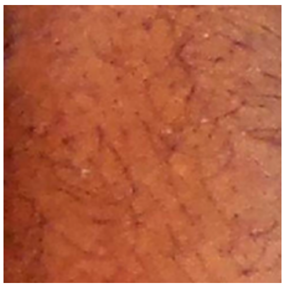

(A) (2,1,2)(2,1,2)

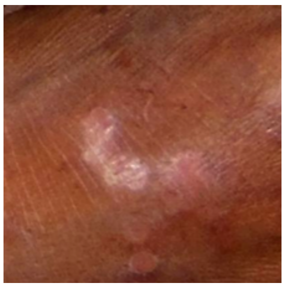

(B) (2,2,1)(2,2,1)

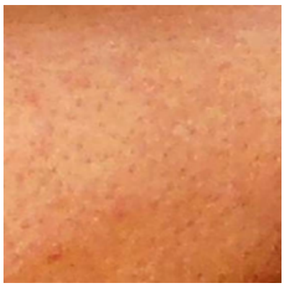

(C) (3,2,2)(3,2,2)

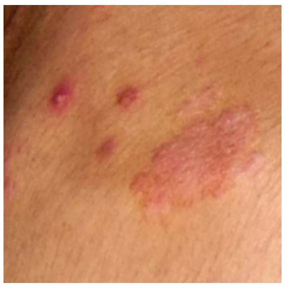

(D) (1,2,2)(1,2,2)

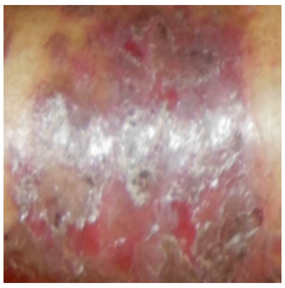

(E) (2,2,1)(2,2,1)

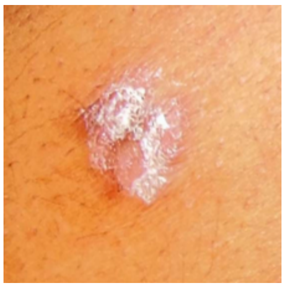

(F) (0,0,0)(0,0,0)

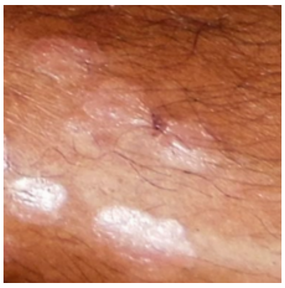

(G) (1,1,1)(1,1,1)

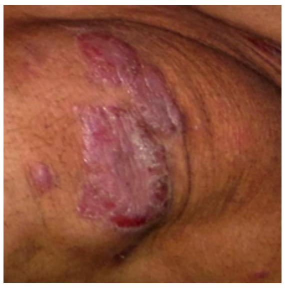

(H) (0,0,0)(0,0,0)

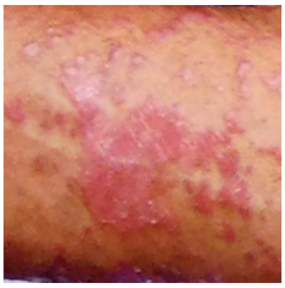

(I) (1,1,1)(1,1,1)


(J) (3,3,1)(3,3,2)


(K) (1,3,1)(1,2,1)


(L) (1,3,1)(1,2,1)


(M) (2,2,3)(2,2,2)


(N) (3,1,2)(2,1,2)

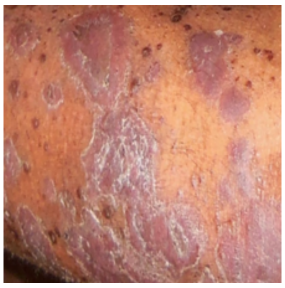

(O) (2,2,1)(2,2,2)

FIGURE 2.5: Actual and Predicted Labels for Stacking With Class Labels for a Few Sample Images

Chapter 3

Other Methods Tried and Future Work

3.1 Manual Feature Set Development Followed by Classification with MLP

As the number of data samples is very less, a manual feature extraction approach followed by a classification with MLP had been taken.In literature, there had been efforts to detect melanoma by using such techniques ([18],[19]) but for psoriasis there is not any. Images had been pre-processed by CLAHE followed by median filtering to remove the noise. From the pre-processed image several features including GLCM features had been extracted.

Let P be the gray level co-occurrence histogram.The value $P(i, j)$ is the number of times that grey-level j occurs at a distance d and at an angle theta from grey-level i. The GLCM features and other features are calculated abiding by the following definitions:

1. **Mean Pixel Intensities** For all 3 channels R, G ,B the mean values were calculated
2. **Standard Deviation** For all 3 channels
3. **Entropy** For all three channels image entropy values were calculated separately

4. **GLCM Features** From 4 Gray Level Co-Occurrence Matrices for four angles, the following features had been extracted

   - **Contrast**

     $$\sum_i \sum_j (i-j)^2 P(i,j)$$

     .

   - **Dissimilarity**:

     $$\sum_i \sum_j |i-j|\, P(i,j)$$

   - **Homogeneity**

     $$\sum_i \sum_j P(i,j)/(1+(i-j)^2)$$

     .

   - **Energy**

     $$\sqrt{\sum_i \sum_j P(i,j)^2}$$

   - **Correlation**

     $$\sum_i \sum_j P(i,j)\left[(i-\mu_i)(j-\mu_j)/\sqrt{\sigma_i \sigma_j}\right]$$

   - **Asymmetry Index**

     $$\sum_i \sum_j P(i,j)^2$$

Therefore, a total of 33 features had been developed per image. Then an MLP had been trained with the above mentioned feature set.The MLP structure is as shown in the table

Layer	Output Shape	No of Parameters
Input	(33,)	0
FC	(132,)	4488
ReLU	(132,)	0
FC	(132,)	17556
ReLU	(132,)	0
Softmax	(5,)	665

TABLE 3.1: MLP Architecture for Classification on Manually Developed Feature Set

The MLP was trained by SGD optimizer with the parameter settings used in the previously mentioned methods. Again a 7 fold cross validation approach has been taken.The average accuracy of the 7 folds were found to be 39.74% for erythema, 35.53% for scaling and 43.68% for induration. So it is working very poorly as compared to CNN.The reason for this may be two. Either the feature set that have been developed are not so important for this particular problem or due to noise, glare, presence of skin hair meaningful features could not be extracted. If the reason is the first one, then detailed domain expertise is required to extract feature pertinent to this problem and if the reason is the second, extensive image pre-processing techniques should be explored to eliminate all the unwanted parts.

As the LAB color space is perceptually uniform with respect to human color vision, feature extraction of the images in LAB color space had been attempted. As **a** channel in LAB color space is for green-red color perception, GLCM and other features only from this channel had been taken for erythema classification and the same features had been taken from **L** channel for scaling classification. These features then were used for classification using MLP, SVM and even with Random Forest algorithm. No improvement over the method described in this section was noticed.

3.2 Transfer Learning from Pre-trained Texture Network

This is not a problem of object identification from an image. Generally CNNs do perform well in case of classification where objects with particular shapes have to be identified. All the standard datasets like ImageNet, CIFAR, MNIST are examples where a particular animal, vehicle, other objects, digits have to be identified. The shallower layers of any CNN detect generic features like edges, curves, blobs etc and deeper layers combine them to extract global spatial information based on which the object is identified. In texture identification or classification, traditional CNN has not performed well. We thought at first that our problem is solely about texture, so a texture CNN based approach had been taken.Andrearczyk et al (2015) [20] have proposed a model for texture identification where a novel energy layer has been introduced. According to them the global spatial features are not at all important for texture classification. So just after the shallower convolutional layers of any CNN an energy layer had been added. The architecture of the model, as given in the paper, is shown in figure

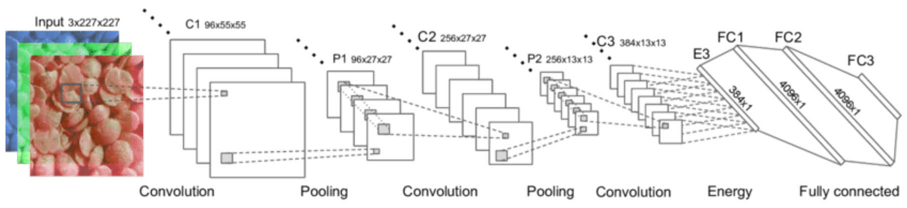


Figure 3.1: Texture CNN

Here in the paper, after first few layers of AlexNet, the energy layer had been added. The energy layer is similar to global average pooling layer. It extracts an average value from each feature map of any particular convolutional layer. Here, in this work, an architecture similar to the one proposed in section 2.3 had been used and a global average pooling layer had been introduced after only 3 convolutional layers as described in the paper. The number of parameters could be reduced to a large extent thus but no change in accuracy was noticed.In fact it deteriorated. The average accuracy over 7 folds for erythema, scaling and induration was found to be 48.5%,46.4% and 49.5% respectively.

The same model was trained on a texture dataset named CUReTCOL.This data set consists of 5600 texture images having 61 different classes. This dataset is derived from the original CUReT (Columbia-Utrecht Reflectance and Texture Database) dataset. The same model gave an accuracy around 72% in the same dataset. After training this model, this was transfer learned to our psoriasis dataset. No significant change in accuracy was noticed. The reason is that the problem is entirely not dependent on texture. Erythema and scaling score depends on the color contrast and induration score depends on the elevation of the affected area with respect to the normal skin surface.

3.3 Future Work

As the misclassifications are mostly between neighbouring classes, embedding the images in hyperbolic space can lead to the improvement in separability of the classes as in hyperbolic space, volumes grow exponentially as opposed to the Euclidean space where it grows polynomially. Recently, a paper by Google DeepMind [21] has proposed a hyperbolic attention network that has outperformed conventional attention networks.There has been efforts to incorporate hyperbolic geometry in deep learning recently like hyperbolic neural embeddings [22], Poincare embeddings for learning hierarchical representations [23] etc. Therefore, as future work proposal, we can try embed the images in hyperbolic space or explore other methods to incorporate hyperbolic geometry in our model.

Bibliography

[1] A. Pal, A. Chaturvedi, U. Garain, A. Chandra, and R. Chatterjee. Severity grading of psoriatic plaques using deep cnn based multi-task learning. *International Conference of Pattern Recognition*, December 2016.

[2] T Fredriksson and U Pettersson. Severe psoriasis–oral therapy with a new retinoid. *Dermatology*, 157(4):238–244, 1978.

[3] J. S. Taur, G. H. Lee, C. C. Chen, and C. W. Yang. Segmentation of psoriasis vulgaris images using multiresolution-based orthogonal subspace techniques,. *IEEE Transactions on Systems, Man And Cybernetics*, 36(2):390–402, April 2006.

[4] J. Lu, E Kazmierczak, J. H. Manton, and R. Sinclair. Automatic segmentation of scaling in 2-d psoriasis skin images. *IEEE Transactions on Medical Imaging*, 32(4): 719–730, April 2013.

[5] F. Bogo, M. Samory, A. B. Fortina, S.Piaserico, and E. Peserico. Psoriasis segmentation through chromatic regions and geometric active contours. *Annual International Conference of the IEEE EMBS*, August 2012.

[6] A. Pal, A. Roy, K. Sen, R. Chatterjee, U. Garain, and S. Senapati. Mixture model based color clustering for psoriatic plaque segmentation. *Asian Conference on Pattern Recognition*, 2015.

[7] J Rning, R Jacques, and J Kontinen. Area assessment of psoriatic lesions based on variable thresholding and subimage classification. *Vision Interface*, 1999.

[8] Yasmeen George, Mohammad Aldeen, and Rahil Garnavi. Skin hair removal for 2d psoriasis images. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pages 1–8. IEEE, 2015.

[9] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[10] Johannes Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2(Mar):721–747, 2002.

[11] Johannes Fürnkranz. Pairwise classification as an ensemble technique. pages 97–110, 2002.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[13] M. D Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *ArXiv e-prints*, November 2013.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, December 2015.

[15] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints*, September 2014.

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. *ArXiv e-prints*, August 2016.

[17] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[18] J Premaladha and KS Ravichandran. Novel approaches for diagnosing melanoma skin lesions through supervised and deep learning algorithms. *Journal of medical systems*, 40(4):96, 2016.

[19] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(4):115–125, February 2017.

[20] Vincent Andrearczyk and Paul F Whelan. Using filter banks in convolutional neural networks for texture classification. *Pattern Recognition Letters*, 84:63–69, 2016.

[21] C. Gulcehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. M. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro, and N. de Freitas. Hyperbolic Attention Networks. *ArXiv e-prints*, May 2018.

[22] B. P. Chamberlain, J. Clough, and M. P. Deisenroth. Neural Embeddings of Graphs in Hyperbolic Space. *ArXiv e-prints*, May 2017.

[23] M. Nickel and D. Kiela. Poincar\'e Embeddings for Learning Hierarchical Representations. *ArXiv e-prints*, May 2017.

[1] [3] [4] [5] [6] [7] [9] [10] [11] [12] [14] [16] [17] [20] [8] [18] [19] [15] [21] [22] [23] [2]