# Skin Lesion Detection using Convolutional Neural Network

<sup>1</sup>C.Rekha, <sup>2</sup>R.Jegatha, <sup>3</sup>P.Sharmila, <sup>4</sup>A.Cibi

1,2,3,4 Assistant Professor

<sup>1,2,3</sup>Sri Sai Ram Institute of Technology, <sup>4</sup>Rajalakshmi Engineering College

*Abstract*: Melanoma is the most destructive form of skin cancer. Early diagnosis of melanoma can be curable. At the same time accurate diagnosis is very essential because of the similarities of melanoma and benign lesions. Hence computerized recognition approaches are highly demanded for dermoscopic images. The main purpose of this project is to develop an automatic system to improve the classification performance of melanoma. The effectiveness of this framework is evaluated on ISBI 2016 Skin Lesion Analysis towards Melanoma Detection Challenge dataset. Initially the extraction of discriminate features is done with the help of Convolution Neural Network. ResNet-50 algorithm is used to classify every lesion in a dermoscopic image as a Benign or Melanoma. Results are found from classification with and without segmented images. This work performs a comparative evaluation of classification alone (using the entire image) against a combination of the two approaches (segmentation followed by classification) in order to assess which of them achieves better classification results. The proposed method would constitute a valuable support for physicians in every day clinical practice.

Key Terms: CNN - Convolution Neural Network, AUC - Area under the curve, ISIC - International Skin Imaging Collaboration,

## **INTRODUCTION**

Skin cancer, the most predominant type of cancer, is reduced when skin cells begin to grow out of control. Skin cancers that are not melanomas are commonly grouped as non-melanoma skin cancers. Melanoma is one of the most harmful type of skin cancer that begins in the pigment cells (melanocytic) of the skin. Due to the excess revelation of ultraviolet radiation from the sun, the skin cells are damaged and can affect the resistant capacity. Quick diagnosis and treatment can result in a very high possibility of melanoma survival. However, the manual assessment made by dermatologists from dermoscopic images is a lengthy process and error-prone. Hence automated algorithms have become a necessity to classify melanoma which is assist for early diagnosis and improve accurate diagnosing performance. This project focuses on the problem of automatic skin lesion detection, particularly on melanoma detection, by applying semantic segmentation and classification from dermoscopic images using a deep learning based approach. For the first problem, U-Net convolution neural network architecture is applied for an accurate extraction of the lesion region. For the second problem, the current model performs a binary classification (benign versus malignant) that can be used for early melanoma detection. The model is general enough to be extended to multi-class skin lesion classification. The proposed solution is built around the ResNet-50 architecture. Finally, this work performs a comparative evaluation of classification alone (using the entire image) against a combination of the two approaches (segmentation followed by classification) in order to assess which of them achieves better classification results.

## EXISTING SYSTEM

First, the lesion is segmented using an automatic segmentation method. Then, a set of features from the ABCD rule (color and texture features) is extracted and used to train a classifier to perform binary classification as melanoma or benign. The extracted features aim to reproduce each one of the accounted scores. The most common features used in these studies include shape features (e.g., compactness, aspect ratio, and maximum diameter), which represent both asymmetry and border; color features in several color spaces (e.g., mean and standard deviation); and texture features (e.g., gray-level co-occurrence matrix) Interesting results have been obtained by different authors with different sets of features and classifiers. Ganster extracted shape and color features and used a k-Nearest Neighbor classifier to distinguish between melanoma and benign. The aforementioned works consider that the three commonly used types of features (shape, color, and texture) are equally relevant for the automatic detection of melanomas. Other works use a single class of features as a descriptor for melanoma detection. Color distribution in the RGB color space (mean RGB distance, variance, and maximum distance) was used to distinguish between melanomas, atypical nevi, and benign nevi. The gray-level co-occurrence matrix and other texture descriptors, such as Laws energy masks and Gabor like filters, were applied to classify skin lesions using texture information only.

## **PROPOSED SYSTEM:**

The primary objective is to perform an automatic prediction of lesion segmentation from dermoscopic images taking the form of binary masks. In order to achieve the project goal, a successful and well-known Convolution Neural Networks architecture have been adopted. The impact and trade-offs of removing skin image background by applying a semantic segmentation for a subsequent classification of the disease is studied. Dermatogist-level classification of skin cancer with deep neural networks

## LITERATURE SURVEY

[1] Study presented a major breakthrough in the classification of skin lesions. The result of the learning model was compared with 21 board-certified dermatologists and proven to be more accurate in this task. It was performed to classify clinical images, indicating

whether a lesion is a benign or malignant one. The classification was done in such a way that the model was trained to classify between 757 fine-grained classes, and then as the probabilities were predicted it was fed into an algorithm that selected the two different classes (malignant or benign). Using this approach, this work achieved a new state of the art result.

[2] This study showed a technique that uses imaging processing as a previous step before training. Furthermore, the author used a technique called data augmentation to increase the dataset, using three transformations (cropping, scaling and rotation) and multiplied the dataset by a factor of 36 times. Finally, a pre-trained convolution neural network (CNN) is used to classify between melanoma and melanocytic nevus for 200 epochs (20,000 iterations, using a batch size of 64 and a dataset with 6,120 examples).

[3] It was also reported the use of data-augmentation with at least 3 different transformations (cropping, flipping, and zooming). Also, it is reported that the points that were critical to the success of the project were mainly due to the volume of data gathered, normalization of the input images and utilizing meta learning. The latter is elucidated as an SVM layer in the final output of the deep-learning models, that map the outputs to the three classes that were proposed in the challenge.

[4]In this work transfer learning is applied, using two different learning models, VGG19 and ResNet-50, both pre-trained on Image Net 1,000 classes dataset. These were used to classify between malignant and benign lesions, using 10,000 dermoscopic images. For the correct learning process, it was also used the up- sampling of the underrepresented class. This process was done using a random number of transformations, chosen between rotation, shifting, zooming, and flipping. Furthermore, in this paper, it was presented 3 experiments, first with the VGG19 architecture with the addition of two extra convolution layers, two fully connected layers, and one neuron with a sigmoid function. Second it experimented with the ResNet-50 model, and finally a implementation of VGG-19 with an SVM classifier as the fully-connected layer. As a final result, the modified implementation of the VGG-19 had the best results.

[5] For the architecture, they chose the ResNet-50 implementation on the framework Keras, with personal modifications. This model was pre-trained with the weights for a generic object recognition model and finally used two optimizers AdaGrad and RMSProp.

[6]The skin lesions were classified as unique classes, not composing meta-classes such as benign and malignant. It used the ResNet-152 pre-trained on the Image Net model to classify 12 lesions. However, for training was used other 248 additional classes, that were added to decrease the false positive and improve the analysis of the middle layers of the model. Furthermore, this was done in such a way that the train sampling for the 248 diseases did not outgrow the main 12, thus when used for inference the model predicted one of the 12 illness, even when the lesion does not belong to one of them. For training was used 855,370 images, augmented approximately 20 to 40 times, using zooming and rotation

## **ARCHITECTURE DIAGRAM:**

The diagram describes the system architecture.



Figure 4.1 Architecture diagram

### MODULE DESCRIPTION

### **Preprocessing Module**

Preprocessing is done in order to objective is to avoid issues caused due to poor contrast images. The process involves cropping the image to the same aspect ratio as needed and resizing the original image to 64 x 80 pixels. The secondary objective is to convert the image into numpy format (.NPY) for the easy access in the segmentation process. Mean subtraction is performed in order to centre the cloud of RGB values from input data around zero along every dimension of the image, a mean subtraction is applied across the image features. Image normalization is performed by dividing each RGB dimension of input images by its standard deviation, a normalization is obtained from its original 0 and 255 pixel values to 1 and 0 normalized values. The image is further converted into greyscale in order to reduce the number dimension of the image used. This pre-processing technique will avoid further issues caused by poor contrast images. In image cropping & resizing the input images are pre- processed to be accepted by the architecture though cropping the image to the same aspect ratio as needed and resizing the original image to 64 x 80 pixels for the U-Net.

### **Mean Subtraction**

The reason we do both of those things is because in the process of training our network, we're going to be multiplying (weights)

and adding to (biases) these initial inputs in order to cause activations that we then backpropogate with the gradients to train the model. We'd like in this process for each feature to have a similar range so that our gradients don't go out of control (and that we only need one global learning rate multiplier). Another way you can think about it is deep learning networks traditionally share many parameters - if you didn't scale your inputs in a way that resulted in similarly-ranged feature values (ie: over the whole dataset by subtracting mean) sharing wouldn't happen very easily because to one part of the image weight w is a lot and to another it's too small.

## **Image Normalization**

There are two definitions for normalization. The first one is to "cut" values too high or too low. i.e. if the image matrix has negative values one set them to zero and if the image matrix has values higher than max value one set them to max values. The second one is to linear stretch all the values in order to fit them into the interval [0, max value]. The above preprocessed images are stored in a numpy format in order to make it easily accessible for the segmentation part of the project. These images are stored in three different files namely, imgs\_mask\_train (which contains preprocessed image of the binary mask of the training images in numpy array format), imgs\_test (it contains the preprocessed image of the testing images in numpy array format), and imgs\_train (it contains the preprocessed image of the images after performing all the preprocessing steps is 64 x 80.

### **Segmentation Module:**

It is identifying parts of the image and understanding what object they belong to. Segmentation lays the basis for performing object detection and classification.

## U-Net

Each block takes an input applies two 3X3 convolution layers followed by a 2X2 max pooling. The number of kernels or feature maps after each block doubles so that architecture can learn the complex structures effectively. The bottommost layer mediates between the contraction layer and the expansion layer. It uses two 3X3 CNN layers followed by 2X2 up convolution layer. But the heart of this architecture lies in the expansion section. Similar to contraction layer, it also consists of several expansion blocks. Each block passes the input to two 3X3 CNN layers followed by a 2X2 up sampling layer. Also after each block number of feature maps used by convolution layer get half to maintain symmetry. However, every time the input is also get appended by feature maps of the corresponding contraction layer. This action would ensure that the features that are learned while contracting the image will be used to reconstruct it. The number of expansion blocks is as same as the number of contraction block. After that, the resultant mapping passes through another 3X3 CNN layer with the number of feature maps equal to the number of segments desired.

## Loss calculation in U-Net

U-Net uses a rather novel loss weighting scheme for each pixel such that there is a higher weight at the border of segmented objects. This loss weighting scheme helped the U-Net model segment cells in biomedical images in discontinuous fashion such that individual cells may be easily identified within the binary segmentation map. First of all pixel-wise softmax applied on the resultant image which is followed by cross-entropy loss function. So we are classifying each pixel into one of the classes. The idea is that even in segmentation every pixel have to lie in some category and we just need to make sure that they do. So we just converted a segmentation problem into a multiclass classification one and it performed very well as compared to the traditional loss functions.

## **Data Augmentation Module**

Furthermore, it is expected that data augmentation should also help prevent over fitting (a common problem in machine learning related to small datasets, when the model, exposed to too few examples, learns patterns that do not generalize to new data) and, for this reason, improving the model's ability to generalize.

### **Classification Module**

The main issue of the classification task is to avoiding over fitting caused by the small number of images of skin lesion in most dermatology datasets. In order to solve this problem, the objective of the proposed model is to firstly extract features from images and secondly load those extracted representations on a fine-tuned ResNet architecture. Due to the reduced size of the ISIC dataset, the suggested approach initializes the model with weights from the ResNet trained on a larger dataset (such as Image Net), a process known as transfer learning. The underlying assumption behind transfer learning is that the pre-trained model has already learned features that might be useful for the classification task at hand. This corresponds, in practice, to using selected layer(s) of the pre-trained Convent as a fixed feature extractor, which can be achieved by freezing all the convolution blocks and only training the fully connected layers with the new dataset.

### **ResNet:**

The portion within the dotted-line box in the left image must directly fit the mapping f(x). This can be tricky if we do not need that particular layer and we would much rather retain the input x. The portion within the dotted-line box in the right image now only needs to parameterize the deviation from the identity, since we return x+f(x). In practice, the residual mapping is often easier to optimize. We only need to set f(x)=0. The right image in fig 2 illustrates the basic Residual Block of ResNet. ResNet follows VGG's full  $3\times33\times3$  convolutional layer design. The residual block has two  $3\times33\times3$  convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function. Then, we skip these two convolution operations and add the input directly before the final ReLU activation functions. This kind of design requires that

the output of the two convolutional layers be of the same shape as the input, so that they can be added together. If we want to change the number of channels or the stride, we need to introduce an additional  $1 \times 11 \times 1$  convolutional layer to transform the input into the desired shape for the addition operation.

In the diagram, conv stands for Convolutional layer, Pool stands for MaxPool layer, batch norm stands for batch normalization, Relu stands for rectified linear unit activation layer, Sum stands for the addition in ResNet, and FC stand for fully connected hidden layers. In this architecture, we have eight ResNet modules which are modified by adding a dropout layer after the second convolutional layers.

## **Densenet:**

The DenseNet architecture was proposed in the seminal paper, "Densely Connected Convolutional Networks". This architecture resulted from the desire to improve higher layer architectures that were being developed. Specifically, improving the problem that many of the layers in high-layer networks were in a sense redundant. The DenseNet architecture attempts to solve this problem by densely connecting all the layers. This means that each layer receives inputs from all the preceding layers and passes its own information to all subsequent layers, which means that the final output layer has direct information from every single layer including the very first layer. This right here is supposed to improve the problem of redundant layers.

DenseNet architecture has several significant advantages over other architectures. This architecture beats the results and benchmarks of the other architectures in Imagenet. Also, the improved parameter efficiency makes the network easier to train. When compared to other architecture, this holds true. The training time is also comparatively significant to that of other architectures. Counter-intuitive effect of this dense connectivity pattern is that it requires fewer parameters than traditional convolutional networks, as there is no need to relearn redundant feature maps. Traditional feed-forward architectures can be viewed as algorithms with a state, which is passed on from layer to layer. Each layer reads the state from its preceding layer and writes to the subsequent layer. It changes the state but also passes on information that needs to be preserved. One of the biggest downside of the ResNets is the large number of parameters which is harder to be trained. Besides better parameter efficiency, one big advantage of DenseNets is their improved flow of information and gradients throughout the network, which makes them easy to train. Each layer has direct access to the gradients from the loss function and the original input signal, leading to an implicit deep supervision. This helps training of deeper network architectures. Further, we also observe that dense connections have a regularizing effect, which reduces over- fitting on tasks with smaller training set sizes.



The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling. Concatenating feature maps learned by different layers increases variation in the input of subsequent layers and improves efficiency.

This constitutes a major difference between DenseNets and ResNets. DenseNets are built from dense blocks and pooling operations, where each dense block is an iterative concatenation of previous feature maps. This architecture can be seen as an extension of ResNets , which performs iterative summation of previous feature maps. However, this small modification has some interesting implications: (1) parameter efficiency, DenseNets[10] are more efficient in the parameter usage; (2) implicit deep supervision, DenseNets[10] perform deep supervision ; and (3) feature reuse, all layers can easily access their preceding layers making it easy to reuse the information from previously computed feature maps. The above given model is tested across various parameters, and the model with best accuracy is found. The model is tested across different number of convolutional networks used for compression and decompression, training and testing, epoch and batch size s.

### Variations in the model:

As, it can been seen that there is no significant increase in dice co-efficient loss with the increase in the number of convolutional networks for compression and decompression stages, only 2 convolutional layers are used in order to reduce the time required to compute the entire model. For the following variations of the model, the U-Net with 2 convolutional layers are used for both compression and decompression stages. The model is tested for various number of training and testing images. As the number of training image is very low, the model seems to under-fit the testing images and as the number of training images increases the model's accuracy increases (dice co-efficient loss decrease). After a reaching a saturation in the accuracy, the dice co-efficient loss starts to increase again due to the presence of over fitting phenomenon i.e., the model has trained to fit the training images perfectly but the testing images falls out to be inaccurate. The highest recorded accuracy after performing 5 epoch on the unsegmented images is about 86.31% and further increase in number of epochs there isn't a significant change over the accuracy.

105



The highest recorded accuracy after performing 5 epoch on the segmented images is about 68.421% and it can be seen that the accuracy of the classifier is more with the unsegmented images than the segmented images. This classifier must be studied for further metrics in order to come to an conclusive solution. The sensitivity of any classifier is important than the accuracy, in medical applications.

## Number of Test samples vs Accuracy:

For decideding, the number of the training images for the model is taken as 1100 as that brings the maximum accuracy to this model. Increasing the number of training images over 1100 makes the classifier to over fit and hence reducing the overall accuracy of this type of classifier.

## RESULTS



Figure Examples of poor segmentation results

а

b



a, the original images and b are corresponding segmented images

Fig Results of the classifier when it is tested against the segmented data and unsegmented data set



b





106

	Batch Size	Accuracy	
1	5	72.8	
<u>.</u>	10	78.7	
 (	15	80.4	
2	20	82.45	
<u>.</u>	25	84.77	- i
	30	86.15	
	35	86.31	

Table: Batch Size vs Accuracy

Number of Test samples	Accuracy
100	20
200	26.45
300	34.1
400	40.95
500	48
600	55.05
700	62.1
800	69.15
900	76.2
1000	83.25
1100	86.31
1200	83.1
1300	80.1

Table Number of Test samples vs Accuracy

#### REFERENCES

[1] ISIC Archive. International skin imaging collaboration: Melanoma project website. [online]. https://isic-archive.com/, 2016.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105), 2012.

[3] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1520-1528), 2015.

[4] Skin Lesion Detection FromDermoscopic Images Using Convolutional Neural Networks, A Degree Thesis Submitted to the Faculty of the Escola T'ecnica d'Enginyeria.

[5] E. Mohedano. Deep learning for computer vision barcelona - image classification.

[online].http://www.slideshare.net/xavigiro/image-classification-dlcv-d112, 2016.

[6] Python Software Foundation. Python programming language. [online]. https://www.python.org/, 2016.

[7] Tensor Flow Development Team. Tensorflow library formachine intelligence. [online]. https://www.tensorflow.org/, 2016

[8] F. Chollet. Introducing keras 1.0. [online]. https://blog.keras.io/introducing- keras-10.html, March 2015.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc.* — 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 2261–2269, 2017.

### **AUTHOR PROFILE**

C.REKHA Working as an Assistant Professor in Sri Sai ram Institute of Technology.

**R.JEGATHA** Working as an Assistant Professor in Sri Sai ram Institute of Technology.

P.SHARMILA Working as an Assistant Professor in Sri Sai ram Institute of Technology.