Skin Lesion Classification using Neural Networks

Connor Pelkey
College of Computer and Information Science
Northeastern University
Boston, MA

Abstract

Malignant skin lesions, such as melanomas, are easily curable with a high likelihood of survival if treated early enough. An accurate, readily available, and easy-to-use method of identifying problematic skin lesions, in conjunction with trained experts, could prevent deaths due to undiagnosed or misdiagnosed lesions. This paper explores the design of convolutional neural network models to classify images of skin lesions as either benign or malignant. The resulting model achieved 78% accuracy on a held-out test set.

Introduction

In the United States, skin cancer is the most common form of cancer, affecting nearly 80,000 people per year, a number which continues to climb. When caught early enough, there is a high likelihood of survival with a 5 year relative survival rate of 98%. This survival rate rapidly drops off to 20% once the disease has spread to other bodily organs. Therefore, the early detection of malignant skin lesions is critical.

Diagnosis of skin cancers includes a physical exam, where a medical professional applies the **ABCD**'s:

Asymmetry – benign lesions are typically symmetrical.
Border – benign lesions have smooth and even borders.
Color - benign moles typically only exhibit one color.
Diameter – benign moles are usually less than 6mm.

Suspicious or alarming lesions may then be biopsied or excised for further examination. This method depends upon the vigilance and experience of a trained medical professional. Limited access to experts, lack of funds, or expert misdiagnosis are all failure modes that can result in easily preventable deaths.

A system able to classify images of skin lesions as either malignant of benign could be used by medical professionals to aid in diagnosis as a secondary opinion, potentially increasing accuracy rates for diagnosis and reducing the need for unnecessary biopsies. With the prevalence of hand-held devices, such a system could additionally be a valuable tool in areas without access to expert opinion.

This paper explores the design of convolutional neural network models for the binary classification of skin lesion images as either benign or malignant.

Background

The experiments discussed utilize different convolutional neural network (CNN) architectures for the purposes of image classification. CNNs have been extensively used for the purposes of image recognition. They are typically comprised of convolution layers and pooling/subsampling layers followed by fully connecting layers.

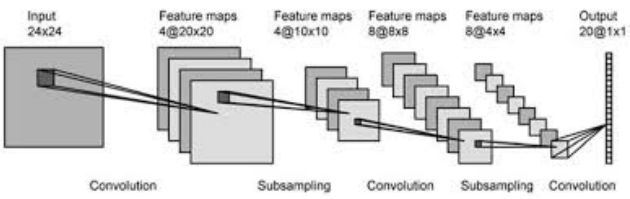

Figure 1: Structure of a CNN

Each layer is composed of neurons, which take as input a weighted sum of a subset of the outputs of the previous layer's neurons plus an additional bias term:

$$WeightedInput(n) = Weights(n) * Input(n) + b$$

The output of a neuron is the result of applying an activation function to the weighted input:

$$Output(n) = Activation(WeightedInput(n))$$

The two activation functions used the experiments discussed are the hyperbolic tangent:

$$tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

and the rectified linear unit:

$$ReLU(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$

Convolution Layer

Convolution layers are used to extract features from inputs while preserving the spacial relationship between them. Each neuron in a convolution layer receives inputs from a localized set of neurons in the previous layer. This allows the neurons to extract features from the input layer.

The structure of a convolution layer depends on its hyperparameters:

Depth: the number of neurons that connect to the same localized input region.
Filter Size: size of the localized input area.
Stride: the number of input neurons between the start of adjacent filters.

Increasing depth increases the number of feature maps that can be applied, allowing the convolution layer to extract more information from the input. For example, one feature map in the layer may extract all edges formed in its input while another may extract bright spots. Applying subsequent convolution layers may extract higher level features.

Pooling Layer

Pooling layers reduce the dimensionality of the input layer. They are typically applied after one or more convolution layers. A pooling layer with a window size of *d* will sweep through the input layer with *d*x*d* windows, applying some down-sampling method to the *d*x*d* inputs resulting a single output value. Pooling reduces the complexity of inputs, improving the tractability of the network as well as controlling overfitting. In the described experiments, down-sampling is performed by selecting the maximum input within each window, a method often used in CNNs.

Fully Connecting Layer

In a fully connecting layer, each neuron in the layer receives input from every neuron in the previous layer. These layers serve to "connect the dots" and use the output of the convolution steps to classify the image.

Dropout Layer

Dropout is a regularization method for preventing overfitting during training. It prevents a neuron from becoming too dependent on a singular input neuron by randomly removing connections with some specified probability *p* between layers and reinstating them after some length of time has passed. This ensures that a neuron produces output values based on multiple input values, allowing the network to generalize to unseen images. Dropout layers are often applied between fully connected layers.

Network Optimization

The most common form of optimization used in neural networks is stochastic gradient descent (SGD). Given the model's parameters θ and a differentiable cost function $C(\theta, x, y)$, SGD updates the parameters in the opposite direction of the gradient of the cost function for each training example x_i with label y_i as follows:

$$\theta = \theta - \alpha \nabla_\theta C(\theta, x_i, y_i)$$

where α is a learning rate between 0 and 1 that decreases over time in order for SGD to converge.

Another method used in the described experiments is adaptive moment estimation (Adam), which has been observed to work well in practice. Adam stores the exponentially decaying average of previously seen gradients, m_t, and squared gradients, v_t, which are updated as follows:

$$m_t = \beta_1 v_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 m_{t-1} + (1 - \beta_2) g_t^2$$

where g_t is the gradient at time *t*. The mean gradient and mean squared gradient are then bias corrected:

$$\hat{m}_t = \frac{m_t}{(1 - \beta_{1,t})}$$

$$\hat{v}_t = \frac{v_t}{(1 - \beta_{2,t})}$$

and used to update the parameters for the next timestep:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Project Description

In order to develop a neural network architecture to classify skin lesions, we first obtained lesion images previously classified by experts, and performed preprocessing. The second step was an iterative model construction sequence, in which architecture and parameters were fine-tuned and compared based on performance in classifying a held-out validation set. The final step was selection of the best performing model and evaluation on a final held-out test set.

Data Preprocessing

The data used in the following experiments were the HAM10000 set of images obtained from the International Skin Imaging Collaboration (ISIC) Archive. The dataset consisted of 1,113 images of malignant lesions, and 6,705 images of benign lesions. Images were 600x450 pixels in size.

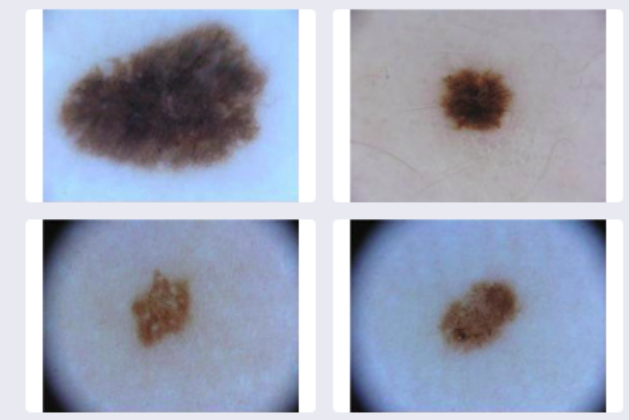

Figure 2: HAM10000 images of benign lesions.

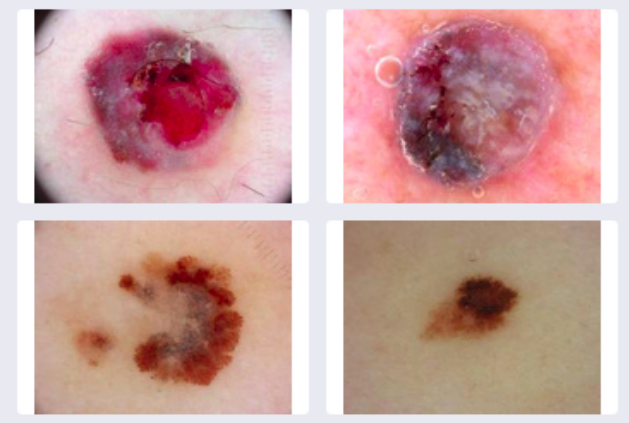

Figure 3: HAM10000 images of malignant lesions.

Images were segmented based on their expert diagnosis. A series of transformations were applied to each segment using Augmentor, a Python image augmentation library, in order to artificially expand the dataset. The stochastic transformation pipeline applied consisted of:

1) rotation
2) distortion
3) shear
4) skew
5) zoom

where each step i was applied in that order with some probability p_i . This step expanded the set of malignant images to 7,146 items and benign images to 11,620 items.

Images were resized to 60x45, a 90% reduction, to improve computational tractability and model training times.

Model Construction

Convolutional neural network models were constructed using the TFLearn deep learning high-level TensorFlow API and trained on an Intel i5 CPU with 4GB of RAM. Given the limited computational resources, all models were designed such that they could be trained on a consumer grade machine in under 2 hours, and limited to 20 epochs of training. For all models, 2x2 max pooling was applied immediately after each convolution layer and dropout with probability 0.5 was applied between each fully-connected layer. A two-step approach was used in construction.

The first step was to determine the architecture of the network. Models were constructed by first varying the number of convolution layers (between 2 and 4), and then varying the number of fully connected layers (between 1 and 2).

The second step was to modify hyper parameters of the best performing architecture. This included comparing ReLU and tanh activation functions, and comparing SGD and Adam optimizers. The model exhibiting highest validation accuracy was then chosen as the final model.

Experiments

The first models evaluated were 2, 3, and 4 convolution layer models using ReLU activation and SGD as the optimizer with 2x2 max pooling applied after each convolution. These were run with both 1 and 2 fully connected layers with 50% dropout between each. No discernible difference was observed between using 1 and 2 fully connected layers, so all future models were constructed with 1 layer to reduce training time and memory footprint. Validation accuracy of these runs are shown in Figure 4.

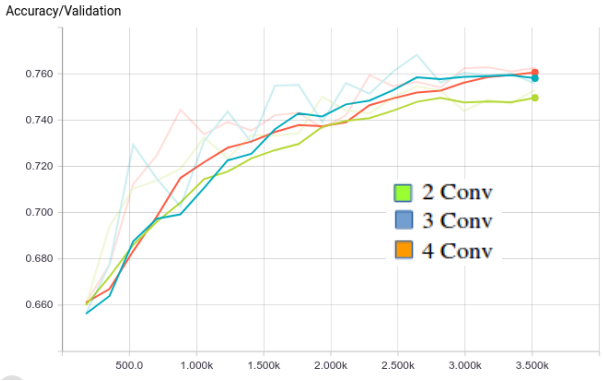

Figure 4: Smoothed validation accuracies for 2 Convolutions (green), 3 Convolutions (blue), and 4 Convolutions (orange).

The 2-layer model plateaued before both the 3-layer and 4-layer models. The 3-layer model was selected to continue experimentation as there was little discernible difference between the validation accuracies of the 3-layer and 4-layer models and the 3-layer model was quicker to train.

The next experiment was to use tanh activation in place of ReLU activation in the convolution layers. The results of this are shown in Figure 5.

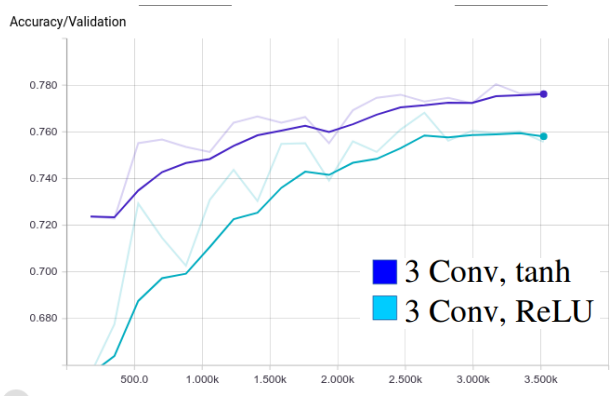

Figure 5: Smoothed validation accuracies for tanh activation (dark blue) and ReLU activation (light blue)

Applying the tanh activation function instead of ReLU resulted in a 2.2% higher validation accuracy. Therefore the model using the tanh activation was selected for further experimentation.

The final experiment was to replace the SGD optimizer with an Adam optimizer. The results of this experiment are shown in Figure 6.

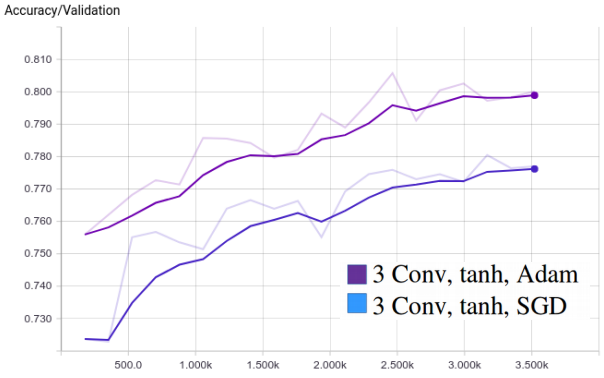

Figure 6: Smoothed validation accuracies for SGD optimizer (blue) and Adam optimizer (purple)

The Adam optimizer achieved a validation accuracy of 80%, 2.3% higher than the SGD optimizer. This model was selected as the final model to test. Running the final model of a held-out testing set consisting of 3,753 images, 62% of which were benign, resulted in an accuracy of 78%. The confusion matrix for the final model is depicted in Table 1.

	Predicted Malignant	Predicted Benign
Malignant	1138 (30.3%)	281 (7.5%)
Benign	541 (14.4%)	1793 (47.7%)

Table 1: Confusion matrix for the final model.

Conclusion

Through experimentation, a convolutional neural network model architecture was developed and refined which achieved 78% accuracy on a held-out test set, with a false negative rate of 7.5%. While this model alone is not sufficient to handle medical diagnosis, it could be used in conjunction with trained medical professionals to provide a second opinion and reduce the likelihood of undiagnosed or misdiagnosed malignant skin lesions.

These experiments were performed using limited computational resources. Using more powerful machines, it is likely that a more robust model can be developed using more complex convolutional neural networks and/or using images that have not been reduced to 10% of their original size. Larger datasets of skin lesions would additionally improve the potential of such models. Further work must be conducted in order to develop more accurate models.

References

Marcus D. Bloice, Christof Stocker, and Andreas Holzinger: 2017. An Image Augmentation Library for Machine Learning

Pathiranage, A. 2017 Convolutional Neural Networks for Predicting Skin Lesions of Melanoma, Regis University, Denver, CO

American Cancer Society. 2018. Cancer Facts & Figures 2018. American Cancer Society, Atlanta

LeCun et al. 2001. Gradient-Based Learning Applied to Document Recognition. *Intelligent Signal Processing 306-351*, IEEE Press

Diederik, K., Ba, J. 2014. Adam: A method for stochastic optimization.

Tschandl, P., Rosendahl, C. & Kittler, H. 2018. The HAM10000 dataset, a large collection of mult-source dermatoscopic images of common pigmented skin lesions.