Skin Cancer Detection

Andrew Plum
Department of Computer Science
University of Idaho
Moscow, ID, USA
plum0598@vandals.uidaho.edu

***Abstract*—This paper covers the implementation of a deep learning model which determines if skin moles are cancerous.**

***Index Terms*—deep learning, neural networks, convolution, image detection, skin cancer**

I. Introduction

Cancer is one of the leading causes of death in the US and world wide. Detection of cancer must first be done before it can be treated. The focus of this project was to design a deep learning model which can detect skin cancer to a high degree of accuracy.

A. Dataset

The dataset used for model training contains 3298 images of skin moles which can be classified as either benign or malignant where the labels for each are 0 or 1 respectively. The images are in jpg file format and are 224 by 224 pixels. Figures 1 and 2 shown below are examples of benign and malignant samples, respectively, from the dataset.

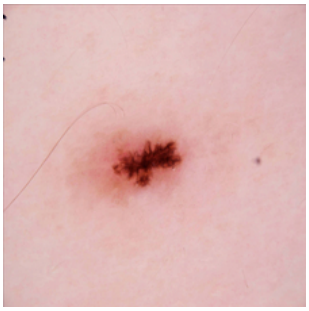

Fig. 1. Benign sample.

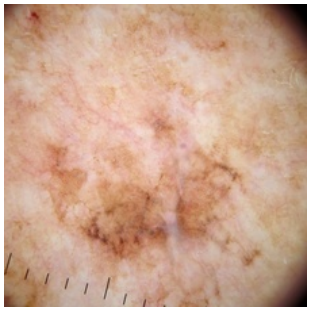

Fig. 2. Malignant sample.

The dataset is from a Kaggle competition and an initial code template used was provided by University of Idaho professor, Professor Xian. The initial code template I started with and the dataset for the project were downloaded from the links listed in the references below. The initial code template prepared the image data to be used in a model. The data was initially split 80:20 into training and test sets respectively, but in the code implementation, after the initial loading and processing of the data, the data was further split the data 60:20:20 into training, validation, and test sets respectively.

II. The Proposed Method

Because the project entails classifying images into two categories (benign or malignant), the deep learning model that was used is a convolutional neural network with three convolutional layers and two max pooling layers with one max pooling layer after each of the first two convolutional layers. After the convolutional and max pooling layers of the model, the output thus far is then flattened using a flattening layer. The output was then sent as input into a dense layer with 32 nodes and ReLU as its activation. The output afterwards was fed into a final dense layer with a single node which used sigmoid as its activation function. From there, thresholding with a 0.5 cutoff was used to classify the predictions as a 0 or 1. Binary cross-entropy was used as the loss function of the model. To evaluate the model's performance, the test accuracy of the model will be assessed. The model when compiled has 5,594,177 parameters all of which were trainable parameters. The summary of the model's architecture is displayed below in figure 3 as well as a diagram of the model's architecture is shown with figure 4.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	36,928
flatten (Flatten)	(None, 173056)	0
dense (Dense)	(None, 32)	5,537,824
dense_1 (Dense)	(None, 1)	33

Total params: 5,594,177 (21.34 MB)
Trainable params: 5,594,177 (21.34 MB)
Non-trainable params: 0 (0.00 B)

Fig. 3. CNN model summary.

III. Experimental Results

A. My CNN Model

The model achieved a test accuracy of 79.39 percent while its training accuracy was 81.24 percent. This was after training the model for 10 epochs and restoring the best set of weights associated with the lowest loss calculated. The loss curves of the model are presented as figure 5 shown below.

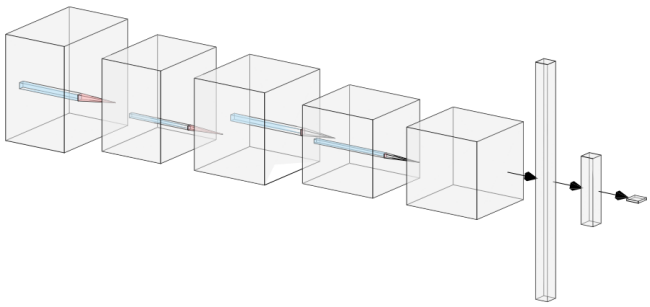

Fig. 4. CNN model architecture.

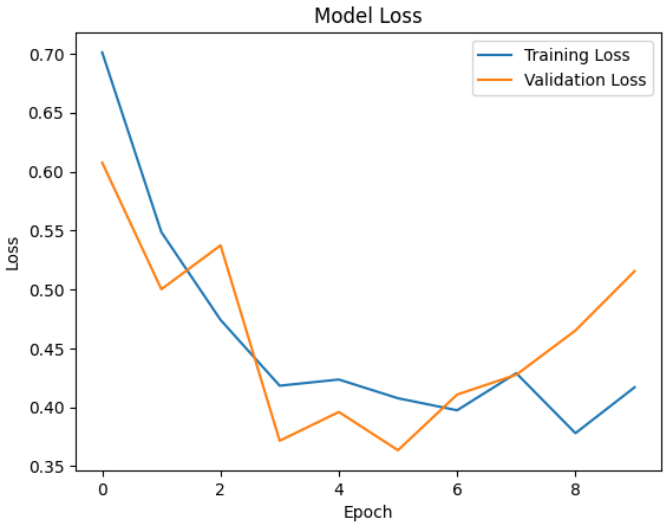

Fig. 5. CNN model loss.

Towards the latter half of training, where the 10th epoch was nearing, the model began to start overfitting. To address the model overfitting, I attempted to implement various regularization techniques, although they failed to different degrees to improve the model's test performance. L2 regularization and dropout were the two techniques closest to improving the model's test performance with L2 regularization doing better than dropout. Because they were closest to improving the model, future work could entail finetuning these hyperparameters to determine if they are viable methods to reducing the models overfitting. Another method to improve the model's test performance which could be explored further would be to reduce the learning rate using a scheduler as the training goes further along.

B. VGG16 Transfer Learning Model

In comparison, a modified VGG16 transfer learning model achieved a training accuracy of 82.96 percent and test accuracy of 83.64 percent after 5 epochs of training. The loss curves of the modified VGG16 model are presented as figure 6 shown below.

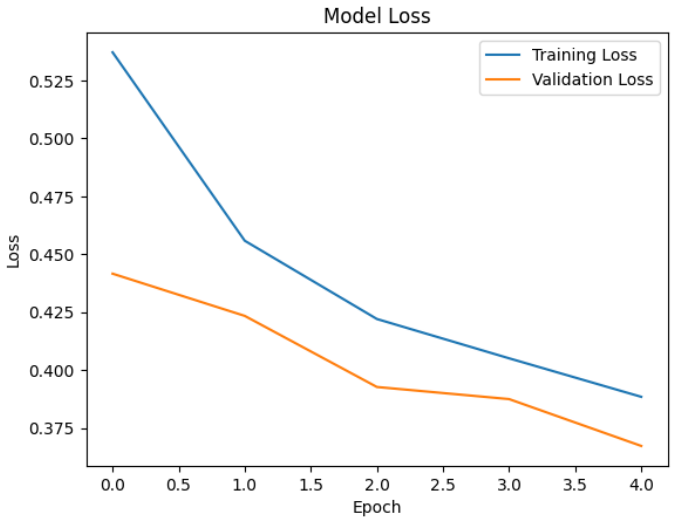

Fig. 6. VGG16 transfer learning model loss.

The model was trained for 5 instead of 10 epochs like the model designed from scratch because the VGG16 transfer learning model had significantly more training parameters and therefore required significantly more training time. The model performance likely would have improved if it had kept training, although the marginal improvement over each subsequent epoch seemed to be declining.

IV. Conclusion

Overall, the model designed from scratch is effective at classifying skin cancer as benign or malignant as its 79.39 percent test accuracy is better than the baseline model with a test accuracy of 75 percent. It also reached a test performance which was near the 83.64 percent test accuracy of the VGG16 model which was impressive when taking into account that the model's complexity is considerably less than the complexity of the VGG16 model. Future work like finetuning the model hyperparameters especially in regards to the regularization techniques that were attempted can be done to potentially improve on this model's performance.

Resources

1) Skin Cancer: Malignant vs Benign Dataset
https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign
2) Code Template
https://canvas.uidaho.edu/courses/30734/files/3673113?wrap=1