Convolutional Neural Networks (CNNs) to Identify Malignant Moles

Noah Keogh *
Department of Computer Science
Rice University
njk6@rice.edu

Abstract

Convolutional Neural Networks (CNNs) are widely used in computer vision applications for classifying objects within images. Their ability to classify objects within images allow them to be applied to a range of applications and fields including healthcare fields such as dermatology, ophthalmology or radiology where image classification tasks are routinely performed by trained physicians. This project aims to compare the performance of three different CNN architectures in classifying dermoscopy images of moles as benign or malignant. A dataset composed of over 10,000 dermoscopy images was used to assess each model's performance on evaluation metrics such as accuracy, precision, recall and F1 scores. Performance was also assessed through visual explanation using Gradient-weighted Class Activation Mapping (Grad-CAM) to improve the interpretability of each model's classification predictions and to ensure mole features were used as the basis of each model's classification predictions. The results demonstrate three promising CNN architectures with high accuracies (>87%) that could be used to form the basis of a model to be used in healthcare settings.

1 Introduction

1.1 Background

Skin cancer is a significant health problem in the United States with 1 in 5 Americans at risk of developing it before the age of 70 [1]. This problem only appears to be getting worse as the number of cases observed from 1994 to 2014 has increased by over 77% [1]. This increase poses a significant risk to the American population, as skin cancers can become life-threatening if left untreated. In fact, more than two people die of skin cancer every hour in the United States [1].

Though skin cancers can be life-threatening, many skin cancers are treatable when detected early. For example, the 5-year survival rate of melanoma skin cancers is nearly 99% when they are detected and treated early [1]. Many of these skin cancers (20-40%) arise from preexisting nevi (birthmarks or moles) and the risk of developing skin cancers is correlated with the number of nevi that one possesses [2]. Thus, many skin cancers can be detected and treated early by getting annual mole inspections to identify and remove the malignant nevi that could develop into melanoma skin cancer.

*Therapeutic Innovation Center (THINC), Baylor College of Medicine, Houston, TX 77004, USA; Alternative Email: noah.keogh@bcm.edu; Github: https://github.com/noahkeogh

Though many skin cancers can easily be prevented by regular visits to a dermatologist, for many Americans regular visits can be a financial burden or challenge. Around 9.2% of the American population (around 30 million people) possess no form of health insurance [3]. Thus, since dermatologist visits can be expensive without insurance, it means that almost 9% of the American population may not have the financial means to see a dermatologist regularly. Without regular visits, early detection and treatment of malignant skin cancers is unlikely. One could attempt to perform a mole check at home, however, this can be challenging for an untrained professional that does not know the physical properties that would distinguish a malignant mole from a benign mole. Thus, this means that a significant portion of the American population may be at risk of developing life-threatening skin cancers simply because they do not have the financial means to see a dermatologist regularly.

Artificial intelligence (AI) may provide a solution to those at risk of developing life-threatening skin cancers. Artificial intelligence is a multidisciplinary field that aims to automate tasks that have traditionally required human intelligence [4]. AI is an umbrella term that encompasses several fields such as machine learning and deep learning [4]. Machine learning encompasses programs utilizing algorithms that allow computers to improve their performance based on past experiences without explicitly having to be reprogrammed [4]. In machine learning, extraction of the most important features are required by a person. The data contained in these features is then given to a machine learning model to learn and form predictions. Deep learning is a sub field of machine learning that is based on the use of artificial neural networks to learn important features in the data to make its predictions [5]. Convolutional Neural Networks (CNNs) are one model in the field of deep learning that have the ability to extract information from the input dataset without the need for human intervention in the feature extraction step [5]. It is a model that has been shown to be highly capable in extracting features from 2-dimensional shapes, such as images. Thus, because of their ability to extract important features from images, they have been shown to be useful in visual recognition and medical image analysis tasks [5].

Since skin cancers are easily preventable through early detection, the creation of a CNN capable of identifying malignant moles may prove beneficial in allowing Americans, who may not have the financial means to visit the dermatologist, the ability to detect malignant moles from home. This will, therefore, allow a given individual the ability to determine if they need to see a dermatologist. Thus, successful adoption of the model may hopefully decrease the number of mortalities caused by skin cancers by giving people the ability to detect and treat their skin cancers early.

1.2 Objectives

Objective 1: To examine the feasibility of training a CNN to identify malignant moles, and to assess the model's accuracy to determine if it could be used in a healthcare setting.

Objective 2: To identify the features within the images of moles the CNN uses to determine whether a given mole is benign or malignant.

2 Related Works

The application of deep learning models to the healthcare field has been revolutionizing the care and treatment patients receive [4]. One reason this change has occurred is because some recently developed deep learning models have been shown to achieve the same level of accuracy as trained physicians and, in some cases, the models are able to achieve a higher level of accuracy than trained physicians [6]. One example of where deep learning is revolutionizing healthcare treatment is in the field of ophthalmology.

Diabetic retinopathy (DR) is the leading cause of blindness in adults, which, when detected early, has treatment options [6]. However, in some parts of the world, there are too few ophthalmologists to read and interpret the fundus photographs (pictures of the retina and optic nerve) taken of each patient [6]. Researchers at Google, therefore, created an AI system to detect the extent of DR in fundus images [6]. This AI system was shown to achieve accuracies that were comparable to the performance of ophthalmologists [6]. The model performed so well, in fact, that it has now been implemented in clinical practice at eye hospitals in India [6].

Another reason for the adoption of deep learning models in the field of healthcare is that trained models can be easily deployed on smartphones [6]. This means that highly accurate models, such as the one created for DR, can be easily deployed on small devices such as smartphones. This means that the general public could access and use these models on their smartphones to receive accurate medical advice and diagnostics from the comfort of their home. The general public could benefit from this in many ways such as in cost-savings from not having to visit the doctor's office as often, and for the early detection of certain conditions – such as skin cancers – which are easily treatable when detected early. Thus, it is clear that there are many benefits to applying deep learning models to the healthcare field.

2.1 CNNs for the Identification of Malignant Moles

Malignant and benign moles have clear differentiating physical properties. Physicians use the asymmetry, border, color, and diameter of moles in order to determine whether they are malignant [6]. For example, malignant moles tend to be composed of a multitude of colors, whereas benign moles tend to be composed of only a single color. Details of the differentiating characteristics of benign and malignant moles can be visualized in a graphic made by the Yashoda hospital and research center shown in Figure 1. Since there a differing physical properties between malignant and benign moles, it would be reasonable to assume that CNNs – which are capable of extracting features from 2-dimensional inputs – are capable of extracting features from images of moles to classify them.

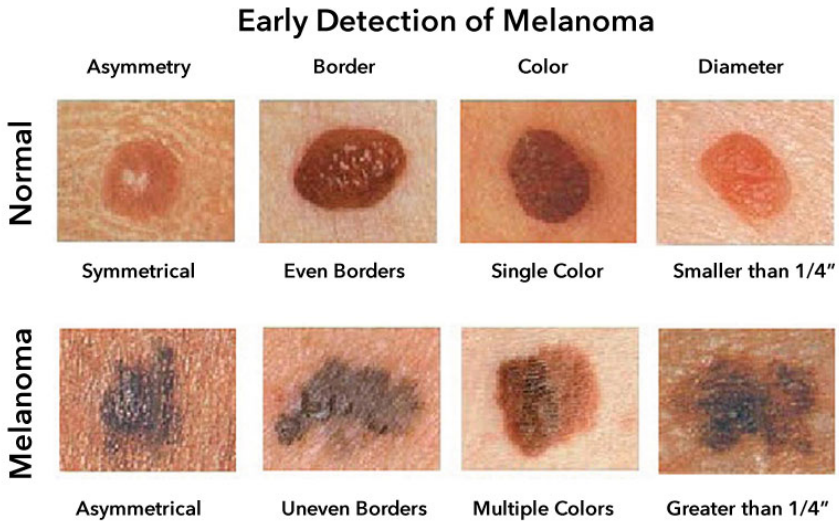

Figure 1: Defining characteristics of benign and malignant moles presented by the Yashoda hospital and research center [7].

Recent projects in the field of dermatology focused on classifying skin lesions using CNNs have shown great promise. A group of researchers from Rzeszow University in Poland published a paper showing the results of four different image classification models trained on a dataset of 8,123 skin lesion dermoscopy images [8]. The group trained four different classification models to classify images of skin lesions into one of seven categories: malignant melanoma, melanocytic nevus, basal cell carcinoma, bowen's disease, benign keratosis, dermatofibroma, and vascular lesion. The best performing model achieved an average of 0.88 precision, 0.83 sensitivity, 0.99 specificity and an F1 score of 0.85 in classifying each of the skin lesion types [8]. The best model achieved an F1 score of 0.85 which means that the classification model is performing relatively well as an F1 score of 1 is considered to be a perfect model.

Each of the four models the researchers tested were CNNs composed of 100 convolutional layers and a single fully connected layer. The convolutional layers of a neural network extract features from a given 2-dimensional input (such as an image) by sliding a kernel of a specified size (such as 2x2) over the input data. This operation produces a feature map of size $W_{out}xH_{out}$ which can be calculated using the following equations where the input dimensions are defined as $W_{in}xH_{in}$, the kernel is defined as size KxK, the stride is defined as S, and the padding is defined as P:

$$W_{out} = \frac{W_{in} - K + 2P}{S} + 1 \tag{1}$$

$$H_{out} = \frac{H_{in} - K + 2P}{S} + 1 \tag{2}$$

The values in the feature map of size $(W_{out}xH_{out})$ are generated by performing an element-wise multiplication of the input values with the weights of the kernel and performing a summation of those values. The value at each position (i, j) of the feature map can be described as the following equation where x is the 2-dimensional input and k is the kernel function:

$$y(i,j) = [x \otimes k](i,j) = \Sigma_m \Sigma_n k(m,n)x(i-m, j-n) \tag{3}$$

The architecture of the CNN used by the researchers was based on ResNet-101. ResNet-101 is a Residual Network (ResNet), which is a neural network architecture that was originally introduced in 2016 to combat the problems of the gradient vanishing problem that can occur in a network with many convolutional layers [9]. The ResNet-101 architecture is quite complicated composed of 1 convolutional layer followed by a max pooling layer, followed by 33 stacked residual blocks - where each block is composed of three convolutional layers - which is followed by an average pool layer and a fully connected layer. The exact specifications of each of these residual blocks can be read in the original paper published by Kwiatkiwska et al. in 2016 [9]. The residual blocks that make up the ResNet architecture possess skip connections that connect the activation of one layer to a layer deeper in the network, thereby skipping some layers in between. This allows the gradient information to pass through the layers. The output of one residual block can be described by the equation:

$$H(X) = ReLU(F(x) + x) \tag{4}$$

Here x represents the input of the residual block and $F(x)$ is the underlying function the residual block tries to learn. The output $H(X)$ of one residual block is then passed to the next residual block. An example residual block can be visualized in Figure 2.

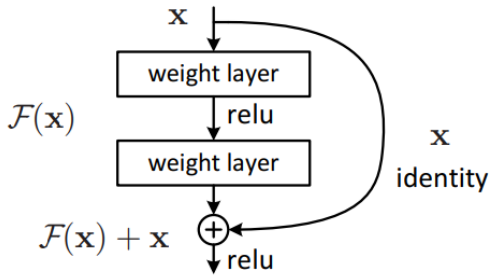

Figure 2: One residual block of a ResNet CNN [9].

Though the four ResNet-101 based models trained by the researchers had relatively good performance, no other architectures besides those based on ResNet-101 were explored. This means that different CNN architectures could possibly perform a better job of classifying skin lesions.

In contrast to the research performed by the group in Poland that used a CNN to classify moles into one of seven types of skin lesions, this project aims to design a CNN that is able to perform binary classification of moles into the categories of benign or malignant. Alternative structures to the ResNet-101 model used by the Poland group will be explored to determine if a better performing and more accurate model can be made. The CNN architectures that will be explored include VGG11, Tiny VGG, and AlexNet.

The VGG11 architecture was first introduced in 2014 and was designed to perform the classification of an image into one of 1,000 different classification groups [10]. The input of VGG11 is a 224x224 image with three color channels. VGG11 contains a total of eleven layers that are comprised of eight convolutional layers and three fully connected layers. The architecture of the network can be observed in Figure 3. The kernel size of each convolutional layer is of size 3x3 and uses a stride of 1. After each convolutional layer a rectified linear unit (ReLU) activation is applied to introduce non-linearity into the network. In addition, ReLU is applied to each layer of the fully connected network except for the last layer which uses softmax activation to form the final predictions. The ReLU activation function takes the max value of the input x and 0. If the input is negative then x is converted to 0, otherwise the input x remains the same. ReLU is defined mathematically below:

$$f(x) = max(x, 0) \tag{5}$$

The VGG11 architecture uses a consistent pattern of convolution followed by max pooling as can be seen in Figure 3. The two first two convolutional layers are directly followed by max pooling, while the last six convolutional layers are grouped into pairs such that two consecutive convolutional layers are stacked before max pooling is applied. The network finishes with three fully connected layers ending in an output layer of size 1,000.

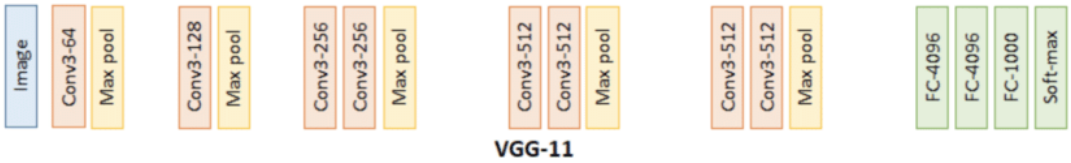

Figure 3: VGG11 Architecture [11]

The Tiny VGG model follows a similar layout as the VGG11 model except that it has a fewer number of layers. The Tiny VGG structure was published in 2020 for the purpose of demonstrating a simple model that could classify images into one of ten classes [12]. The Tiny VGG architecture is much simpler than the VGG11 architecture as it is composed of four convolutional layers, as compared to 11 convolutional layers present in the VGG11 model. Like in VGG11, ReLU activation is applied after each of the convolutional layers. Each convolutional layer is significantly smaller than VGG11 with each convolutional layer being comprised of 10 kernels. Like in VGG11, the Tiny VGG uses a kernel size of 3x3 in all convolutional layers and uses a stride of 1. There is only one fully connected layer in this network which connects the flattened layer to the output layer of ten neurons. Thus, it is clear that this network is much simpler than the VGG11 network and will be beneficial in determining how well a simple CNN can perform the classification task of identifying whether a given mole is malignant.

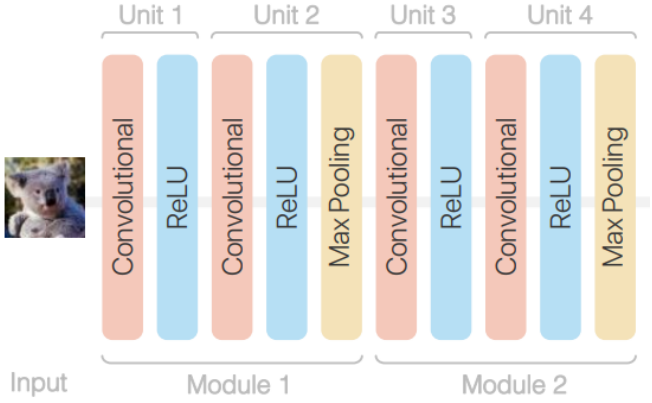

Figure 4: Tiny VGG Architecture [12]

The last CNN architecture that will be studied is AlexNet which was first introduced in 2014 [13]. AlexNet was originally designed to perform the classification of an image into one of 1,000 different classification groups [13]. The input of AlexNet is a 224x224 image with three color channels. The architecture of AlexNet is shown in Figure 5. AlexNet is composed of five convolutional layers. The first convolutional layer uses 96 kernels of size 11x11 with a stride of 4. The second layer uses 256 kernels of size 5x5 and stride of 1. The third and fourth layers use 384 kernels of size 3x3 with a stride of 1. The fifth convolutional layer consists of 256 kernels of size 3x3. The network then has a fully connected three layer network ending in 1,000 neurons to form the final predictions. Thus, AlexNet is smaller than the VGG11 architecture as it uses five convolutional layers as compared to the eleven convolutional layers used in VGG11. AlexNet also differs from VGG11 in that each convolutional layer has differing kernel sizes and stride sizes. In VGG11, all the convolutional layers have the same kernel size of 3x3 with a stride of 1.

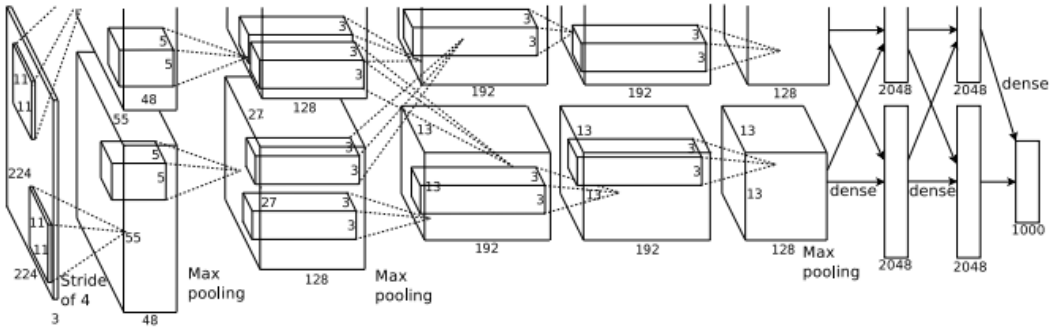

Figure 5: AlexNet Architecture [13]

The models described above differ greatly in their structure as compared to the ResNet-101 model used by the researchers in Poland. Unlike the ResNet-101 model, these models do not use residual blocks. Instead they use the traditional approach of stacking convolutional and max pooling layers. These networks are also much smaller in size as compared to the ResNet-101 model. In addition, to being different from the ResNet-101 model, the three network architectures to be tested differ from one another. Tiny VGG is a very simple model architecture having a small number of convolutional layers. AlexNet is slightly larger than Tiny VGG containing five convolutional layers. VGG11 is the largest of the three networks, containing eleven convolutional layers. Since the three network architectures vary in size a determination may be made that could demonstrate if a fewer number or larger number of convolutional layers is beneficial to completing the classification of malignant and benign moles.

Though CNNs are a powerful model capable of extracting features from 2-dimensional inputs, these models suffer from low interpretability. Since CNNs will be used in predicting whether a given mole is benign or malignant, it is essential that it is known how the model is forming its predictions. The interpretability of the CNN models must, therefore, be improved so it can be identified that the CNNs is extracting information about the features of a mole to form its predictions and not miscellaneous features present in the image that may not pertain to a mole. In order to improve the interpretability of the model, gradient-weighted class activation mapping (Grad-CAM) will be used. Grad-CAM is a technique that was published in 2017 and aims to improve the interpretability of CNNs by creating a visualization that identifies the most important regions in the image that a given model is using to form its classification prediction [14]. Grad-CAM works by generating a localization map that highlights the most relevant parts of the image for predicting a certain class by using the gradient of that class going into the last convolutional layer of the network [14].

An example heat activation map that was presented in the original Grad-CAM paper can be visualized in Figure 6 [14]. These heat activation maps demonstrate how Grad-CAM can be used to identify the regions in the image that the model determines to be important for making its classification predictions. It is clear that the interpretability of the trained model can been improved with the use of Grad-CAM as the example heat activation maps demonstrate that the model has learned the distinguishing features between dogs and cats based on the highlighted regions of the image for each target class. Thus, Grad-CAM will be used in this project to improve the interpretability of the trained CNNs. This will allow for the identification of whether the final trained models appear to have learned the features that distinguish malignant moles from benign moles.

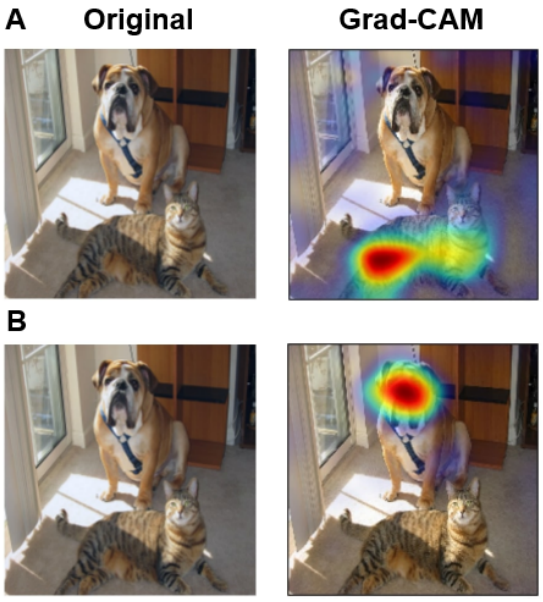

Figure 6: **A.** Demonstrates a Grad-CAM heat activation map for the most important region in predicting the target class "Cat." **B.** Demonstrates a Grad-CAM heat activation map for the most important region in predicting the target class "Dog." [14]

3 Data Description

The dataset that will be used to train the models is publicly available from Kaggle [15]. The dataset was published by Muhammad Hasnain Javid. There are a total of 10,605 images in the dataset that contain one mole in the center of each image. The 10,605 images are comprised of 5,500 images of benign moles and 5,105 images of malignant moles. There is a good balance between the labels in the dataset, therefore, class imbalance should not be a problem. Two examples images (one benign and one malignant mole) can be visualized in Figure 7.

(a) Benign Mole

(b) Malignant Mole

Figure 7: Side-by-side comparison of an example image of a (A) benign mole and a (B) malignant mole from the dataset that will be to train the CNN models.

All of the images in the dataset are of size 300x300 and are saved in JPEG format. The total compressed size of the dataset is around 104 MB. The position of the mole in each image is relatively consistent with the majority of the moles being centered in the frame. There is, however, some variation in the area that surrounds each of the moles. In some cases, the mole appears to be encircled by a black circle. This black circle is most likely out of the field-of-view of the dermatoscope (instrument used to inspect moles) and the camera has captured the surrounding edges of the instrument. There are also a few cases where hair may be obscuring portions of the mole. An example image from each of the cases can be seen in Figure 8.

(a) Mole surrounded by black circle

(b) Mole obscured by hair

Figure 8: Demonstrates a few example images that differ from the optimal conditions shown in Figure 7.

The dataset is appropriate for training a model for the binary classification of moles into malignant and benign moles. The first reason the data is appropriate is because the dataset is large (over 10,000 images) and the number of examples from each class (benign and malignant) are almost equal. The second reason is that all the images have already been labeled so a supervised model can be used to perform the classification task. The labeled dataset will allow for the training of a CNN to perform the binary classification task. The third reason is that the dataset has good variety. The images are all the same size which is ideal for training a model, but there are many varying and suboptimal conditions contained in the dataset. Since suboptimal conditions, such as black circles or hair in the images are contained in the dataset, in addition to the optimal conditions shown in Figure 7, the models trained on this dataset may be well generalized. This means the trained model may be able to better classify new images as compared to a model trained on a dataset only containing images of moles in optimal conditions.

Though the dataset does appear to be good for a variety of reasons, it still has some limitations. One of these limitations is that it appears that all of the moles captured in the dataset are surrounded by skin that is more lightly pigmented. This means that a model trained on this dataset may perform poorly if it encounters an image of a mole that is surrounded by darker pigmented skin. The reason the model may have poor performance is because in this case, the sharp boundaries between the mole and the surrounding skin may no longer exist since the skin color may be close in color to the mole. This may make it hard for a model trained on lighter pigmented skin to distinguish between the mole and surrounding skin, leading to inaccurate predictions. Thus, since the dataset only contains images of moles surrounded by lighter pigmented skin, the model will most likely need to be retrained with additional data if it is to be used in all the conditions discussed above. This, therefore, is a limitation of the dataset.

4 Data Science Pipeline

4.1 Data Wrangling

The dataset consists of 10,605 color images of malignant and benign moles. In order to train an effective model, the dataset was broken down into three sets: training, testing, and validation. The test dataset contained 1,000 images of benign moles and 1,000 images of malignant moles. This dataset was used to assess the final accuracy and performance of the trained model. After the test dataset images were removed from original dataset 8,605 images remained. The validation dataset was created by randomly sampling 10% of the remaining 8,605 images to produce a dataset of size 860 images composed of 468 images of benign moles, and 392 images of malignant moles. The validation dataset was used to assess the accuracy of the trained model on new data at each training epoch. The training dataset was composed of the remaining 7,745 images that was composed of 4,032 images of benign moles and 3,713 images of malignant moles. The training dataset was used to train each of the CNNs.

The images in the dataset are all the same size with dimensions of 300x300 and composed of three color channels: red, green, and blue (RGB). The pixel values in each of the color channels range from 0 to 255. In order to improve the generalization performance of each of the models, data augmentation was used to artificially introduce more variety into the dataset. Three data augmentation techniques were randomly applied to the images in the dataset. The first technique called horizontal flipping was used to randomly flip the image across the vertical axis, creating a mirrored version of the original image. The second technique called random jitter was used randomly change the brightness, saturation and hue of the original image by transforming each of the features by a randomly selected factor ranging from 0.8 to 1.2 times the original feature value. The third technique called random rotation was used to randomly rotate the image by -10 to 10 degrees. Each of these techniques were randomly applied with a probability of 0.05 to all the images in the training dataset. Example images demonstrating these data augmentation transformations can be visualized in Figure 9.

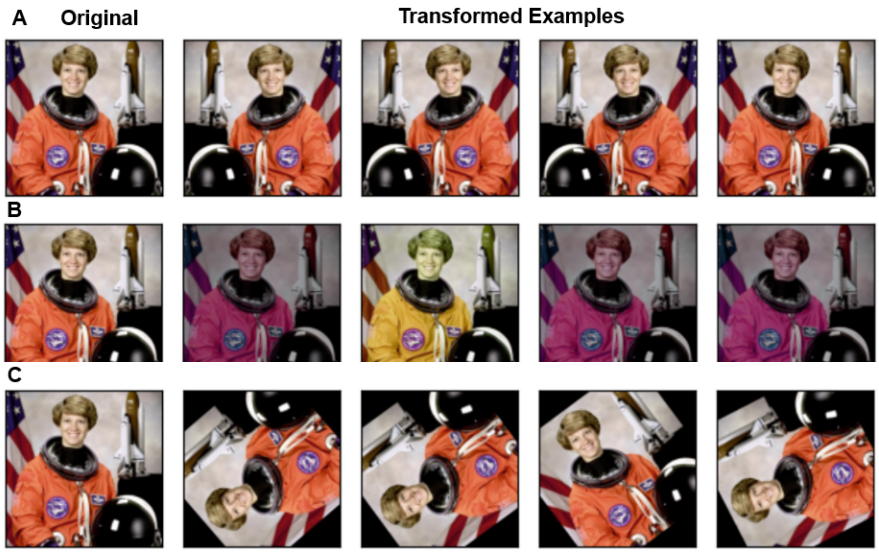

Figure 9: Demonstrates the results of three data augmentation transformations applied four times to an example image. **A.** Demonstrates the random horizontal flip transformation. **B.** Demonstrates the color jitter transformation. **C.** Demonstrates the random rotation transformation. [16]

After data augmentation transformations were applied to the training dataset the pixel values of the images were normalized from a range of 0 to 255 to a range of 0 to 1. Performing this normalization reduces the computation time required to train the model as computations with smaller values are less complex than computations with large values. After normalizing the pixel values of the image the images were reshaped for the requirements of each model.

The Tiny VGG model requires images of size 300x300 with three color channels. Thus, after the processing described above was performed the images were ready to be used to train the Tiny VGG model.

The VGG11 and AlexNet CNN models required additional processing to prepare the images for training. Both model architectures require that the images be of size 224x224 with three color channels. Thus, to reduce the image in size the images were resized to be a size of 256x256. The images were then center cropped to a size of 224x224. The pixel values of the images were then normalized using means of 0.485, 0.456, 0.406, and a standard deviation of 0.229, 0.224, 0.225 for the red, green and blue color channels respectively. These transformations were performed according to PyTorch documentation which indicated the proper technique for preparing images for use in the VGG11 and AlexNet CNN architectures [17, 18]. After these pre-processing operations were applied, the images were ready to be used in training the VGG11 and AlexNet models.

4.2 Data Exploration

There are only two classes present in the dataset: benign and malignant. All of the images are properly classified and there are no images that do not contain a label. All of the images contain an image of either a malignant or benign mole. The images are fairly consistent with some variations observed in the background of the image that surrounds the mole. As discussed in section 3, the images of the mole may sometimes be surrounded by a black circle as shown in Figure 8. These images are observed in both the benign and malignant classes, therefore, this feature is unlikely to affect the classification predictions made by the models. Some images of the mole are occluded by objects in the image such as hairs, as shown in Figure 8. Since these occlusions of the mole are observed in both the malignant and benign classes, it is unlikely that this would affect the classification predictions made by the models. In addition, it is likely that these features may be able to help the models be more generalized. Since there are examples of sub-optimal images of malignant and benign moles, it means that the models may be able to generalize well when they encounter new images of moles that may be obscured or the field of view has strange features such as a black circle. Thus, it is not expected that these features would affect the prediction capabilities of the models, and may, in fact, actually allow the models to be more generalized and robust.

4.3 Data Modeling

As discussed in section 4.1, the variety of the dataset was enhanced through the use of data augmentation techniques. After the data augmentation techniques were applied, each image would need to be properly processed and resized for each CNN architecture to be used.

The first CNN architecture to be trained is Tiny VGG (introduced in section 2.1). This model takes a 2-dimensional image of size 300x300 and three color channels. The output size of the Tiny VGG was modified from the original ten output neurons to an output layer of one neuron. This one output neuron will allow for the binary classification of malignant and benign moles. A sigmoid activation function (as described in the equation below) will be applied to the raw logit value produced by the output layer of

the network.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

After applying the sigmoid activation function, if the value obtained is greater than or equal to 0.5 the mole will be classified as malignant, and if it is less than 0.5 the mole will be classified as benign. The architecture of the model is the same as described in Figure 4 and in section 2.1. The only difference between the two architectures is the number of neurons in the output layer has been reduced from ten neurons to one neuron. The architecture to be used for the Tiny VGG model can be seen in Figure 10. The input and output sizes of each layer are clearly displayed along with the kernel size, stride, and padding used.

```
TinyVGG(
  (conv_block_1): Sequential(
    (0): Conv2d(3, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (conv_block_2): Sequential(
    (0): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Flatten(start_dim=1, end_dim=-1)
    (1): Linear(in_features=56250, out_features=1, bias=True)
  )
)
```

Figure 10: Tiny VGG architecture to be used for the classification of malignant and benign moles.

The second CNN architecture to be trained is VGG11 (introduced in section 2.1). This model takes a 2-dimensional image of size 224x224 and three color channels. Therefore, additional processing of the input images was required before the network was trained, as described in section 4.1. The output of the VGG11 CNN was reduced from the original 1,000 output neurons to a size of 1 output neuron. Like in Tiny VGG, a sigmoid activation will be applied to the raw logit value produced by the network and values greater than 0.5 means the mole is malignant and values less than 0.5 means a given mole is benign. The architecture of the network is the same as described in Figure 3 and described in section 2.1 except for the modified output layer which reduces the number of output neurons from 1,000 to 1. The architecture to be used for the VGG11 model can be seen in Figure 11. The input and output sizes of each layer are clearly displayed along with the kernel size, stride and padding used.

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (11): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (12): ReLU(inplace=True)
    (13): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (14): ReLU(inplace=True)
    (15): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (16): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (17): ReLU(inplace=True)
    (18): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (19): ReLU(inplace=True)
    (20): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=1, bias=True)
  )
)
```

Figure 11: VGG11 architecture to be used for the classification of malignant and benign moles.

The third CNN architecture to be trained is AlexNet (introduced in section 2.1). Like VGG11, this model takes a 2-dimensional image of size 224x224 and three color channels. The pre-processing of the image such as reshaping and center cropping will be required as described in section 4.1. The output layer of the AlexNet model will be reduced from 1,000 neurons to 1 neuron to perform the binary classification task. The overall architecture of the model is the same as described in section 2.1 and shown in Figure 5. The implementation will vary slightly from the original architecture such as the reduction in output neurons from 1,000 to 1, and some of the layers may have a different number of kernels. The reason that this number may vary is because the original AlexNet model was trained on two graphics processing units (GPUs) [13]. Since this implementation performs all computations on a single GPU, there are some variations that exist in the number of kernels. Though this is the case, the number of layers and layer architecture remain the same. The architecture of the AlexNet model to be used in this project can be seen in Figure 12. The input and output sizes of each layer are clearly displayed along with the kernel size, stride and padding used.

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=4096, out_features=1, bias=True)
  )
)
```

Figure 12: AlexNet architecture to be used for the classification of malignant and benign moles.

5 Experiments

5.1 Setup

Dataset

As described previously, the dataset of malignant and benign moles used in training the CNN models was obtained from a publicly dataset available on Kaggle consisting of over 10,605 images of [15]. This dataset was broken down into three partitions as outlined in section 4.1: training, validation, and testing. The training set consisted of 7,745 images composed of 4,032 images of benign moles and 3,713 images of malignant moles. The validation set consisted of 860 images made up of 468 images of benign moles and 392 images of malignant moles. The testing dataset was 2,000 images in size made up of 1,000 images of benign moles and 1,000 images of malignant moles.

Data augmentation techniques, as described in section 4.1, were used to artificially increase the variety of the dataset. This was performed through the random application of transformations such as horizontal flip, color jitter, or random rotations to images of the training dataset. The effects of these transformations included random rotations, and transformations of the color, saturation, and hue of the image. The goal of the data augmentation transformations was to improve the generalization behavior of each of the CNN architectures trained on the dataset.

Each of the images in the dataset were transformed to be the expected input size of the particular CNN architecture used. For example, images were reshaped to sizes of 224x224, and the pixel values normal-

ized according to the specifications required for the VGG11 and AlexNet models. These procedures are described in detail in section 4.1.

Models

All models were trained in 10 training epochs with binary cross entropy being used as the loss function. The models were trained in batches of 32 images. The Adam algorithm was used to optimize the trainable parameters in each of the networks using a learning rate of 0.0001, β_1 of 0.9, and β_2 of 0.999. The Adam algorithm is an optimization algorithm that is a combination of exponential weighted moving average and momentum. The equations that compose an Adam optimizer are described below:

$$g^k = \frac{\delta \mathcal{L}(\theta^k)}{\delta \theta^k} \tag{7}$$

$$m^k = \beta_1 m^{k-1} + (1-\beta_1) g^k \tag{8}$$

$$s_k = \beta_2 S_{k-1} + (1-\beta_2) \cdot [g^k]^2 \tag{9}$$

$$\theta^{k+1} = \theta^k - \alpha \cdot \frac{m^k}{\sqrt{S_k}} \tag{10}$$

In the equations above m^k represents the exponential weighted moving average of the gradients (momentum) where β_1 is a term for how strongly to weight the previous computed term. The s_k term represents the exponential weighted moving average of the square of the gradients where β_2 is a term for how strongly to weight the previous computed value. The g^k term represents the gradient of the loss function with respect to the model parameters at the k^{th} iteration. The parameters (θ) are then updated according to a combination of the m^k and s_k terms and a learning rate α.

The first model trained was the Tiny VGG model that was described in section 4.3 & 2.1. The architecture of this model can be visualized in Figure 10. The input size of the network was a 300x300 image with three color channels. The output size of the network was a single neuron which was used to form the prediction of whether a given mole is benign or malignant. Sigmoid activation was applied to the raw logit value produced by the network and values greater than or equal to 0.5 indicated a malignant mole prediction and values less than 0.5 indicated a benign mole prediction.

The second model trained was the VGG11 model described in section 4.3 & 2.1. The architecture of the model can be visualized in Figure 11. The input size of the network was a 224x224 image with three color channels. The original 300x300 images were transformed by the approaches described in section 4.1. Like Tiny VGG, the output size of the network was a single neuron. Sigmoid activation was applied to the raw logit value produced by the network and values greater than or equal to 0.5 indicated a malignant mole prediction and values less than 0.5 indicated a benign mole prediction.

The third model trained was the AlexNet model described in section 4.3 & 2.1. The architecture of the model can be visualized in Figure 12. Like VGG11, the input size of the network was a 224x244 image with three color channels. The original 300x300 images were transformed by the approaches described in section 4.1. This network also has an output size of one neuron. Sigmoid activation was applied to the raw logit value produced by the network and values greater than or equal to 0.5 indicated a malignant mole prediction and values less than 0.5 indicated a benign mole prediction.

Evaluation

To measure the performance of each model, the accuracy, precision, recall, F1, specificity, and area under the receiver operating curve (AUC) will be calculated using the predictions made by each model on the test dataset. The equations for calculating the metrics are shown below. In this case, TP = true

positives, TN = true negatives, FP = false positives, FN = false negatives.

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1} = \frac{2 * precision * recall}{precision + recall} \tag{14}$$

$$\text{specificity} = \frac{TN}{TN + FP} \tag{15}$$

Visual Explanation

In order to explain the classifications of each CNN model, Grad-CAM was used. Grad-CAM allows for the identification of the most important regions of the image that were used for the classification prediction. Grad-CAM produces an activation heat map that overlays on top of the original image, highlighting the regions in red that were most important for the classification prediction. Regions highlighted in blue were considered to be the least important regions for the classification prediction. Thus, Grad-CAM allows for an additional measurement of performance on top of the quantitative evaluation metrics. For example, if regions relating to the features of a mole are highlighted by a given model, then it means the model is most likely learning the features that distinguish a benign mole from a malignant mole. If a model, however, is using regions that do not pertain to the mole, the given model may not be learning the features that distinguish a benign mole from a malignant mole. Thus, Grad-CAM will act as a qualitative measure for how well a given model is performing and learning the features that distinguish a malignant mole from a benign mole.

5.2 Experimental Results

Evaluation of CNNs

After training the models, the prediction results of each CNN architectures were evaluated using the methods described above. Trained models were tested on the test dataset consisting on 2,000 total images comprised of 1,000 benign and 1,000 malignant moles. The metrics were calculated based on the test dataset predictions made by each model. These values were compared and used to assess the performance of each model.

Based on accuracy, AlexNet appeared to be superior with an accuracy of 0.908 followed by VGG11 with an accuracy of 0.906, and lastly Tiny VGG model with an accuracy of 0.879. Thus, all models appeared to have performed well having high accuracies that differed by 2.9 points. To further quantify the performance of each model, precision, recall, F1, and the specificity of each model was assessed. This comparison can be seen in Figure 13.

Based on the performance metrics, AlexNet and VGG11 had the highest F1 scores with a value of 0.90. AlexNet appeared to perform better overall, as compared to VGG11, as AlexNet had the highest precision and specificity scores. Tiny VGG had the lowest F1 score of 0.89, but it did, however, contain the highest recall value of all the models with a value of 0.907. All models appeared to perform quite well as indicated by their high F1 scores.

Plotting the receiver operating characteristic (ROC) curves for each model and calculating the corresponding area under the curve (AUC) yields the same conclusion as the F1 score metric. The ROC curves and AUC values are shown in Figure 14. All models achieved very good performance as demonstrated by their high AUC values, which were all within 5 points of a perfect classifier model. AlexNet had the highest performance with an AUC of 0.971, followed by the VGG11 model which has an AUC of 0.966, and lastly the Tiny VGG model which had an AUC value of 0.951. Thus, the ordering of the models in their AUC values is consistent with the ordering determined by the F1 scores and other performance metrics.

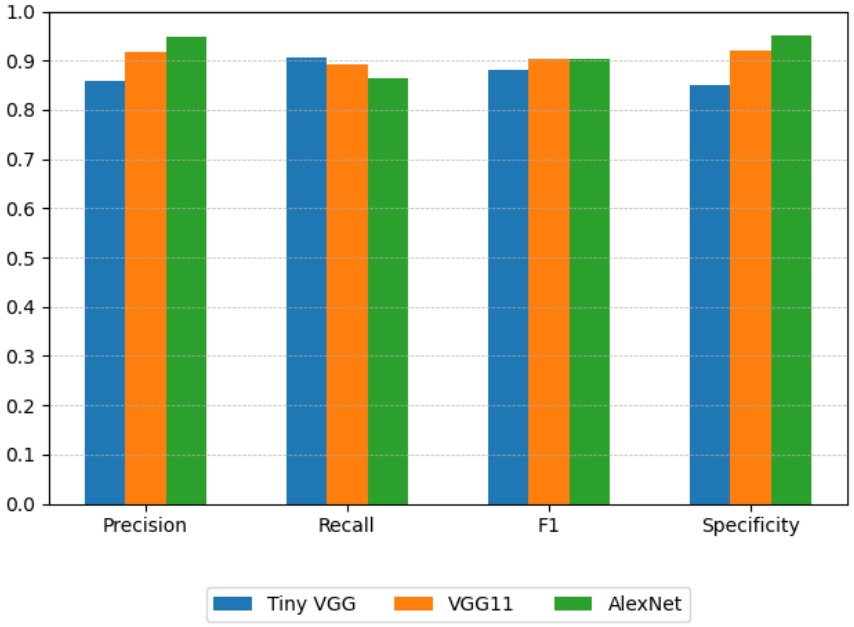

Figure 13: Comparison of score metrics on test dataset predictions made by Tiny VGG, AlexNet, and VGG11 models.

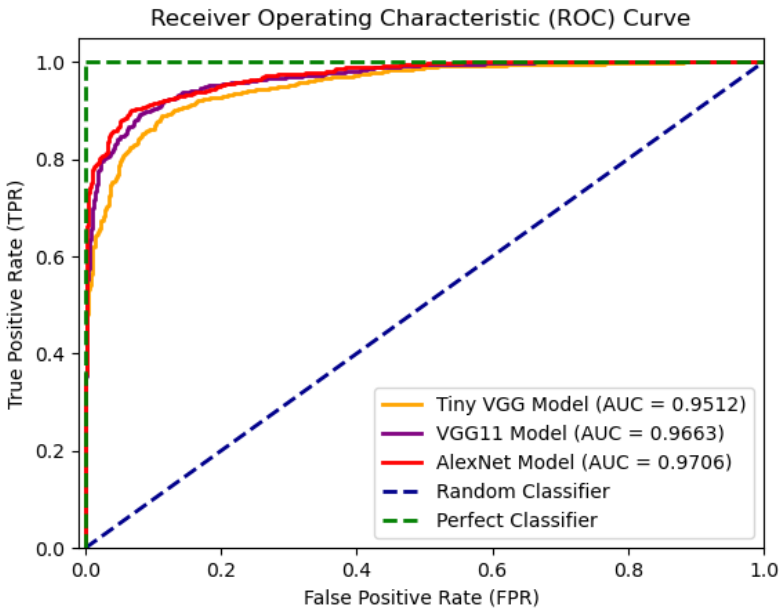

Figure 14: Receiver Operating Characteristic (ROC) curve for each model compared to random and perfect classifier models.

Visual Explanation

Based on the Grad-CAM activation heat maps of several example skin lesions in the test dataset, the Tiny VGG model appeared to outperform the AlexNet and VGG11 models in identifying the important characteristics of malignant moles that differentiate them from benign moles. Figure 15 shows the activation heat maps of several example images that were generated using the values generated by each model at the last layer of the CNN before flattening and the fully connected layers. The activation heat maps were, therefore, generated from the outputs of the last max pooling layer of each model architecture. All the example images shown here were correctly identified by each model as malignant.

The Tiny VGG and AlexNet models appeared to be better at identifying the important features within the image while ignoring the non-important features as compared to the VGG11 model. This is demonstrated by the Grad-CAM maps of the images in the third row of Figure 15. The image in the third row demonstrates that the VGG11 model does a poor job of identifying the important features of the mole to form its classification predictions. In this image, the VGG11 model identifies regions very far from the mole as most important to its classification prediction. This suggests the VGG11 model may be learning and using features not related to moles to form the basis of its classification predictions. The AlexNet model appears to perform a much better job of identifying the important regions on the image in the third row of Figure 15. This is shown from its Grad-CAM map which highlights a portion of the mole. It must be noted, however, that the AlexNet model highlights regions surrounding the mole in addition to the mole itself. This suggests that the AlexNet model, like the VGG11 model, may also be influenced by non-important features in the image, but to a lesser extent. Unlike the AlexNet and VGG11 models, the Tiny VGG model does a good job of ignoring the non-important features in the image in the third row of Figure 15. This was demonstrated by the Grad-CAM map of Tiny VGG which only highlighted the mole as being important for the Tiny VGG's classification prediction. This suggests that Tiny VGG may

be better at identifying the important features in the image and the moles as compared to the AlexNet and VGG11 models.

It is important to note, however, that the Tiny VGG model also appears to be using features in the image not directly related to the mole to form its classification predictions. This is demonstrated by the Grad-CAM activation heat map of the image in the second row. Tiny VGG appears to highlight the black portion of the image surrounding the mole, and the mole itself as the important portions of the image for classification. Since the black circular portion does not relate to whether the mole is benign or malignant, this Grad-CAM map suggests that the Tiny VGG may be using features of the image not related to the mole to form its classification predictions.

Overall, it appeared that the Tiny VGG outperformed the AlexNet and VGG11 models in identifying the important characteristics of the moles. This was demonstrated in the example images when it was demonstrated that the VGG11 model may use no features related mole to form its predictions (as shown by the image in the third row). The Tiny VGG and AlexNet do also suffer from using features not related to the mole to form predictions, however, the mole was always a part of the most significant region determined for their classification prediction. The Tiny VGG model appeared to excel in identifying the mole, and highlighting significant regions within it, which was something the other models struggled to do. Thus, the Tiny VGG11 model appeared to perform a better job of identifying the important features of a mole as compared to the AlexNet and VGG11 models.

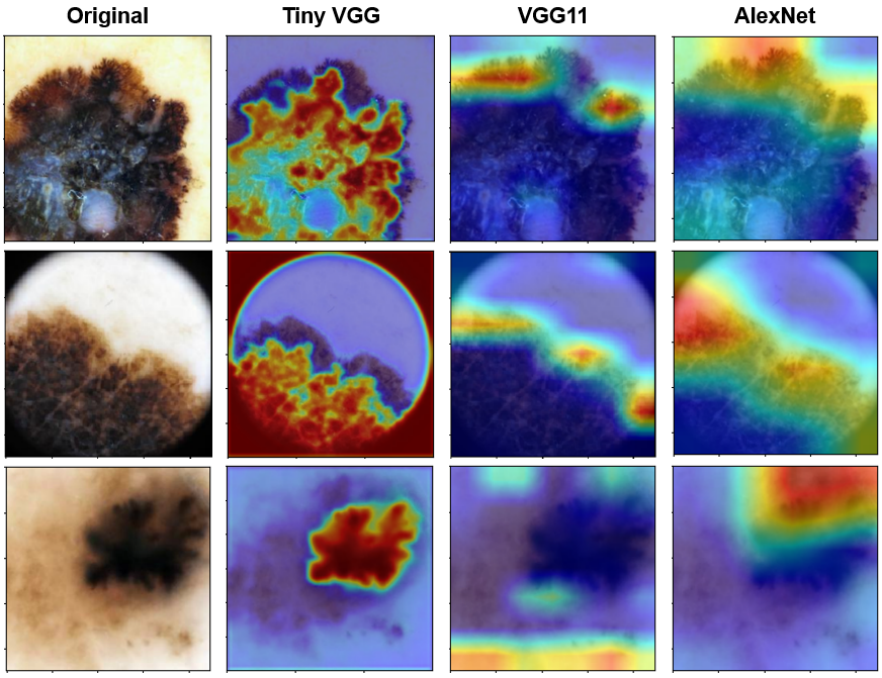

Figure 15: Comparison of Grad-CAM activation heat maps generated using Tiny VGG, AlexNet, and VGG11 models with images of malignant moles from the test dataset.

Discussion

In the field of dermatology, it is essential that physicians be able to identify the important features of a mole to identify it as benign or malignant. Thus, it is essential that a CNN trained to do the same task is also able to explain its classification results, such as which features it discovered as most important for determining whether a given mole is malignant or benign. For this reason, the results of the Grad-CAM will be weighted more heavily than the performance metrics used to evaluate the CNNs. This is because it is more important that the model derives it classification predictions using features present in the mole, instead of using parts of the image that do not relate to the features of a mole.

The Tiny VGG model appeared to perform the best when weighing the results of the Grad-CAM more heavily than the performance metrics used, such as the F1 score or AUC value. Though the Tiny VGG model had the lowest F1 score and AUC value, it greatly outperformed the other models in terms of forming its classification predictions based on features of the mole. These results were clearly shown in Figure 15. In all example images, the Tiny VGG model highlighted different features of the mole as most important in forming its classification decision. It also was able to ignore extraneous features present in the image, solely, focusing on the mole itself. The VGG11 and AlexNet models were less consistent in their use of features present in the moles to form their classification predictions. This is demonstrated in the example images where both models used portions of the image that did not include the mole to form their classification predictions. Thus, despite VGG11 and AlexNet models having slightly better performance in terms of accuracy, F1 and AUC values, as compared to the Tiny VGG model, the models perform a much poorer job in identifying the features of a mole that determine malignancy. Hence, the Tiny VGG performed a superior job in learning the features of a mole that determine malignancy, despite having slightly lower accuracy, F1 scores, and AUC values compared to the other models. It is, therefore, the best model to choose to achieve the project objective.

6 Conclusions

6.1 Impact

From the analysis performed above, it was determined that the Tiny VGG model architecture may have outperformed the other model architectures in terms of identifying the most important features for predicting malignancy with high accuracy. It is important to note that this conclusion was reached based on the analysis of Grad-CAM images. This analysis was performed by an untrained professional, however, so input from a trained physician may be required to best determine which model may have identified the features of malignancy the best. From the perspective of an untrained professional, the Grad-CAM images of the Tiny VGG model appeared to most clearly and consistently highlight regions of the mole as important for its classification prediction, suggesting that it had better learned the features of malignancy as compared to the AlexNet and VGG11 models. The Tiny VGG model achieved both project objectives as it is highly accurate (0.879 accuracy and 0.95 AUC value), and its classification predictions are interpretable through the use of Grad-CAM, showing that the model is able to identify features within moles to classify them as benign or malignant.

Since the Tiny VGG model is able to identify malignant moles with high accuracy and in a manner that is interpretable through the use of Grad-CAM, the model could feasibly be implemented in a personal healthcare setting. The reason that this accuracy could be determined to be of an acceptable level at this setting is because an untrained person is unlikely to achieve accuracies higher than a random classifier. An untrained person would be unlikely to know or identify the features that can be used to identify malignant moles. Furthermore, even if one knew the characteristics of malignant moles, one's

classifications are unlikely to be accurate without proper experience or training. Since this model is able to achieve accuracies of 0.879, it is likely that the model could outperform an untrained person in identifying whether a given mole is malignant. This means that the model could be used to aid people who do not have the financial means to see a doctor regularly. The model could be used by people at home to get a preliminary assessment of whether they need to see a doctor for treatment of a mole based on the results of the Tiny VGG classifier model. With further improvements and possible increases in accuracy, it may further support the use of such a model at the personal healthcare level.

The Tiny VGG model may also be able to match or outperform some of the published models that were trained to identify malignant melanoma. Researchers from the Institute of Medical Science in Rzesow, Poland published the results of several models they trained to classify seven different types of skin lesions based on the ResNet architecture [8]. The four models they developed had F1 scores ranging from 0.68 to 0.73 and AUC values ranging from 0.95 to 0.96 in identifying malignant melanoma [8]. Thus, since Tiny VGG model developed here obtained a higher F1 score of 0.882 and similar AUC value of 0.95, the model developed may be able to match or outperform the models developed by the group in Poland. To further test this hypothesis much more testing would need to be performed with the two models forming predictions on the same test data for the fairest comparison. The models' similar scores suggest that they may be able to achieve similar performance, and, thus, may warrant further testing.

The Tiny VGG model may also be beneficial to physicians in identifying new traits or characteristics of malignant moles that may have previously been unknown. Since the Tiny VGG model appears to be basing its predictions on features present in each mole, it is possible that the model could be identifying features that were previously unknown to be characteristic of malignant moles. A trained physician could analyze the Grad-CAM activation heat maps for many moles and analyze whether there are any features the model identifies as important that are inconsistent with known characteristics of malignant moles. Analysis from a trained physician may be able to determine whether there is merit in these consistencies, and if these inconsistencies could be classified as newly identified characteristics of malignant moles.

6.2 Future Work

Before any of the purposed models can be implemented in any type of healthcare setting, further testing of each of the models will be required. It is essential that the final model be verified to work when it is introduced to new data. One such case is to test the model using images generated by phones (such as would be the case in a personal at-home healthcare setting). If the model has poor performance then retraining the model on new data generated by smartphones may be beneficial to ensuring the model is well generalized.

In addition, to testing whether the final model could detect malignant moles from images generated from smartphones, the model will also need to be analyzed by a trained physician to determine if the model is using the correct characteristics within each mole to identify it as benign or malignant. This would require a physician to look at the Grad-CAM activation heat maps generated by the model and identifying whether the model appears to be recognizing the correct characteristics in the mole to form its classification predictions. Deciding upon a final model may be best performed by a trained physician as only they would be able to identify which model is best highlighting the features that determine malignancy. Thus, more research needs to be performed to verify the models are working properly before any could be introduced to the public in a personal healthcare capacity.

Another aspect that needs to be analyzed is whether each of the models may be able to generalize well in all settings. For example, the image dataset used consisted of only lighter pigmented skin in the background of the the images of the moles. This means that the model may not be able to generalize

well and form accurate predictions on moles that are surrounded by darker pigmented skin. Thus, more training data would need to be collected to determine if the model can generalize well to this case or if the model needs to be retrained using images consisting of moles surrounded by darker pigmented skin.

It should be further investigated if the Tiny VGG model could be optimized by adjusting the number of convolutional layers. The Tiny VGG used only two convolutional blocks, therefore, increasing the model complexity through the number of convolutional blocks may allow the model to better learn the characteristics of malignant moles. In addition, the size and number of parameters of the AlexNet and VGG11 models could be adjusted to further optimize their performance. Thus, the parameters of all models studied could be adjusted to optimize their prediction accuracies and enhance their predictive capabilities.

This project has proposed several models that were able to learn the features of malignant moles and achieved accuracies ranging from 0.879 to 0.908. Thus, the project objectives to build models with high accuracy that based their predictions on the features of malignancy has been achieved. There are many improvements that could be made to increase the generalization and accuracy of each of the models studied; however, these models are a good start for identifying a model that can be utilized in an at-home healthcare setting.

References

[1] Skin cancer facts statistics. https://www.skincancer.org/skin-cancer-information/skin-cancer-facts. [Accessed 25-Jun-2023].

[2] Mehul Bhatt, Adam Nabatian, David Kriegel, and Hooman Khorasani. Does an increased number of moles correlate to a higher risk of melanoma? *Future Medicine*, 2016. [Accessed 25-Jun-2023].

[3] Nathan Paulus. How many americans are uninsured?, 2023. [Accessed 25-Jun-2023].

[4] Anant Manish Singh and Wasif Bilal Haju. Research paper on artificial intelligence. *International Journal for Research in Applied Science and Engineering Technology*, 10:1–5, 2022.

[5] Iqbal H. Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2, 2021.

[6] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nature Biomedical Engineering*, (2):719–731, 2018. [Accessed 25-Jun-2023].

[7] Yashoda Hospital. Melanoma skin cancer, types, stages, grades, signs, symptoms, risk factors, doctors specialist, 2021.

[8] Dominika Kwiatkowska, Piotr Kluska, and Adam Reich. Convolutional neural networks for the detection of malignant melanoma in dermoscopy images. *Advances in Dermatology and Allergology*, 38(3):412–420, Jul 2021.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[11] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport, 2023.

[12] Zijie J. Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Polo Chau. CNN explainer: Learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1396–1406, feb 2021.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[14] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.

[15] Hasnain Javid. Melanoma skin cancer dataset of 10000 images. https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images?resource=download, 2020. Accessed on August 3, 2023.

[16] PyTorch. Image transformations — torchvision 0.11.2 documentation, 2021. Accessed on August 2, 2021.

[17] Vgg - pytorch. https://pytorch.org/hub/pytorch_vision_vgg/. Accessed: 2023-07-31.

[18] Alexnet - pytorch. https://pytorch.org/hub/pytorch_vision_alexnet/. Accessed: 2023-07-31.