

MSE 491 Final Project

Early Self-Diagnostic for Skin Lesion Disorders using Convolutional Neural Network

Submission Date: April 23, 2021

Group 15

301354473	Khoi Nguyen	Contribution = 50%
301347885	Teresa Lau	Contribution = 50%

Table of Contents

Introduction	2
Methods	3
Getting Started	3
CNN	5
Results and Discussions	6
Conclusions	9
References	10

Introduction

The most dangerous form of skin cancer is Melanoma. Melanoma can start as a new, brown or black, flat, freckle-like spot on the skin's outer surface. It can also begin as a change in the shape or colour of an existing mole or coloured spot. If melanoma is recognized and treated early, it is almost always curable. Otherwise, the cancer cells can advance and spread to other parts of the body, making it harder to be treated and turn fatal. A new spot or a new spot on the skin is an important warning sign of melanoma, and its detection largely increases the patients' abilities to be treated in early stages.

The study of the melanoma's color changes has been done, as the color tone difference of a melanoma can be detected in percentage. The study of the size of an is changing in size, shape, or color. So, early detection of any of these symptoms of any kind of melanoma will help detect a malignant melanoma by considering the changes in size and color of the mole. To facilitate the application of skin images, the International Skin Imaging Collaboration (ISIC) has created an ISIC archive, a largest publicly-collected set of classified skin cancer images as a benchmark for education and research.

For the MSE 491 final project, the objective is to train a classification model for skin monitoring at an early stage using the ISIC 2019 dataset. By using Convolutional Neural Network (CNN) and Python for image processing, this implementation is repeatable and can perform as an easily-accessible, early self-diagnostic or monitoring option for the general population.

Convolutional Neural Network is an algorithm as part of the artificial neural network (ANN) especially efficient in image processing [1]. The CNN algorithm works via implementation of relevant filters that allows it to capture spatial and temporal dependencies in images. As such, a CNN model will be trained for testing and identifying the presence or absence of skin disorders using an image dataset containing eight sets of diagnosed skin disorders and one set of healthy skin.

Methods

The ISIC 2019 dataset contains 25331 images, which are unequally distributed as we will investigate further in the project. Each image has inconsistencies ranging from 600x450 to 1024x1000 pixels .

The skin lesion disorder classification project is implemented using Keras with TensorFlow and convolutional neural networks (CNN). For the CNN, inputs of processed images with diagnostic images are fed to it for training. To prevent overfitting and improve the generalization of the model, image augmentation is also applied during the training process.

1. Getting Started

For this project, the ISIC 2019 dataset obtained from Kaggle is used to provide training data for distinguishing between the presence and absence of skin lesion disorders [2]. Setting up the Google Colab workspace environment, Kaggle is installed as per dataset requirements and necessary libraries, such as TensorFlow and NumPy, are imported.

Working with Kaggle datasets, a Kaggle JSON file is required and manually uploaded to the workspace to establish the Kaggle directory for connecting and downloading the training dataset. Once the dataset is downloaded in .zip format, the files are extracted for use in the Colab environment, and the ground truth .csv file is imported. Ground truth in machine learning is used to determine an ideal expected result [3] and in this project's application, it defines the labelling condition of each image. Counting up the number of images per each ground truth category, "unknown (UNK)" images were included but as none of the images matched that criteria, it was dropped from the ground truth for simplicity. The total number of images found for each condition, or feature, is found as follows:

- MEL (melanoma) - 4522
- NV (melanocytic nevus) - 12875
- BCC (basal cell carcinoma) - 3323
- AK (actinic keratosis) - 867
- BKL (benign keratosis) - 2624
- DF (dermato fibroma) - 239
- VASC (vascular lesion) - 253
- SCC (squamous cell carcinoma) - 628

For analysis, a random image is selected from the dataset to determine its image size and shape in Figure 1.

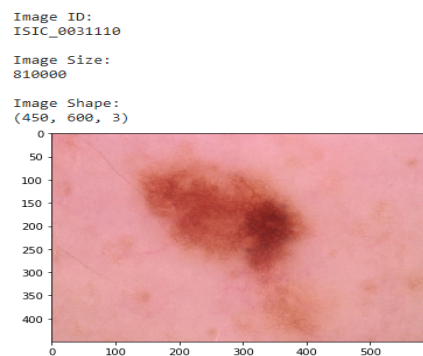


Fig 1: Randomly selected image from dataset with printout of associated information

Then the weights of each class based on the number of samples are computed. We can see that the data are imbalanced.

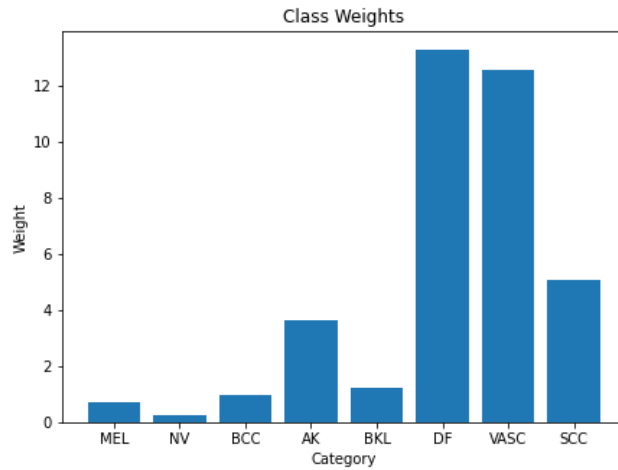


Fig2: Class weight for different weight

As the training and testing files are found as separate datasets, they are saved respectively. After the files are respectively saved and converted to numpy arrays, the files are resized to be of equal sizing. Lastly, the images are normalized per channel to improve the image scaling.

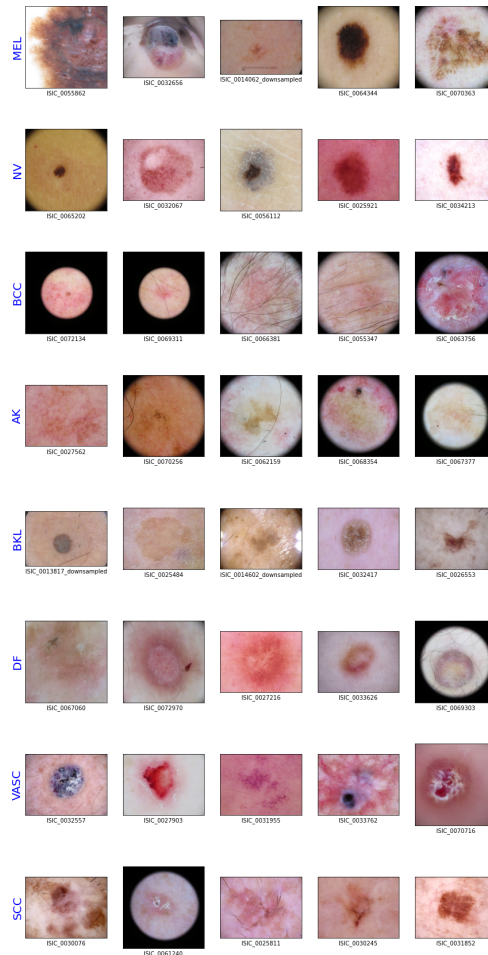


Fig 3: Sample images in each class

2. CNN

After pre-processing the images, the CNN model is built by flattening the data and adding layers to improve the model accuracy. Given that the data is not normally distributed, we

In all layers, the activation function was chosen as ReLu with a kernel size of 3. Although Sigmoid and Tanh activation functions are also highly popular for nonlinear activation functions, ReLu was chosen for the application of this project for its improved convergence speed on large datasets. In addition to being less computationally expensive, ReLu does not have the same vanishing gradient problem as the other two activation function options [4], contributing to improved effectiveness for this project's learning.

For the filter size of the layers, the starting filter was selected as 32 with increasing filter sizes for each additional layer. By starting with a lower number for the initial layer with increasing filter sizes, the model will extract and detect more shape features with each additional layer of increased filter size.

For the first the following layer command is used:

```
keras.layers.Conv2D(filters=32, kernel_size=3, padding='same',
                    activation='relu', input_shape=(input_size[0], input_size[1], 3)),
                    keras.layers.MaxPool2D(pool_size=2)
```

For the flattening layer, the following is applied:

```
keras.layers.Conv2D(filters=64, kernel_size=3, padding='same',
                    activation='relu', keras.layers.MaxPool2D(pool_size=2)
```

The third, hidden layer is defined as follows:

```
keras.layers.Conv2D(filters=128, kernel_size=3, padding='same',
                    activation='relu', keras.layers.MaxPool2D(pool_size=2)
```

The pooling layer has the following configuration:

```
keras.layers.Conv2D(filters=256, kernel_size=3, padding='same',
                    keras.layers.Dropout(0.3),
                    keras.layers.GlobalAveragePooling2D(),
                    keras.layers.Dense(8, name = 'dense_pred'),
                    keras.layers.Activation('softmax', name='probs')
```

For compiling the model, the learning rate is reduced from the 0.01 default to 0.001. This change is implemented to reduce overfitting in the model. Additionally, a callback was added as a preventative measure in case validation loss has not improved. To fit and train the model, only 10 epochs are used as neural network models require large computational resources and this model is restricted by the physical limitations of the implementation environment.

Results and Discussions

Using 10 epochs, the skin lesion disorder detection model required more than an hour of training the network to be completed. The resultant accuracies from the training model is found as:

```
Epoch 1/10
633/633 [=====] - 425s 619ms/step - loss:
1.5041 - accuracy: 0.4557 - val_loss: 1.0587 - val_accuracy: 0.6392
Epoch 2/10
633/633 [=====] - 393s 621ms/step - loss:
1.3689 - accuracy: 0.4989 - val_loss: 1.1284 - val_accuracy: 0.5963
Epoch 3/10
633/633 [=====] - 396s 625ms/step - loss:
1.2900 - accuracy: 0.5292 - val_loss: 1.0621 - val_accuracy: 0.6266
Epoch 4/10
633/633 [=====] - 389s 614ms/step - loss:
1.2421 - accuracy: 0.5509 - val_loss: 1.1034 - val_accuracy: 0.6234
Epoch 5/10
633/633 [=====] - 385s 609ms/step - loss:
1.1485 - accuracy: 0.5887 - val_loss: 1.0108 - val_accuracy: 0.6220
Epoch 6/10
633/633 [=====] - 383s 605ms/step - loss:
1.1251 - accuracy: 0.5937 - val_loss: 0.9697 - val_accuracy: 0.6485
Epoch 7/10
633/633 [=====] - 383s 606ms/step - loss:
1.1387 - accuracy: 0.5811 - val_loss: 1.0408 - val_accuracy: 0.5967
Epoch 8/10
633/633 [=====] - 383s 605ms/step - loss:
1.1241 - accuracy: 0.5941 - val_loss: 1.0274 - val_accuracy: 0.5973
Epoch 9/10
633/633 [=====] - 387s 611ms/step - loss:
1.1318 - accuracy: 0.5858 - val_loss: 0.9668 - val_accuracy: 0.6460
Epoch 10/10
633/633 [=====] - 387s 612ms/step - loss:
1.0842 - accuracy: 0.6106 - val_loss: 1.0363 - val_accuracy: 0.6011
```

To ensure the classification model is satisfactory, it is evaluated on the test data and its accuracy is reported as:

```
Test loss: 1.0246827602386475
Test accuracy: 0.6142913699150085
```

Although a reported test accuracy of 0.6143 does not seem satisfactory, the average accuracy from the training epochs is found as 0.61971, which is relatively similar to the test accuracy. With each epoch, there is increased consistency found within the model accuracy, indicating a satisfactory training process leading to the consistent accuracies. For the skin lesion disorder detection model, the model accuracy's consistency trend coincides with the test accuracy and thus, the evaluation of the model is accepted as satisfactory.

To visualize and quantify the results, the confusion matrix and classification metrics are plotted to determine the model's evaluation predictions.

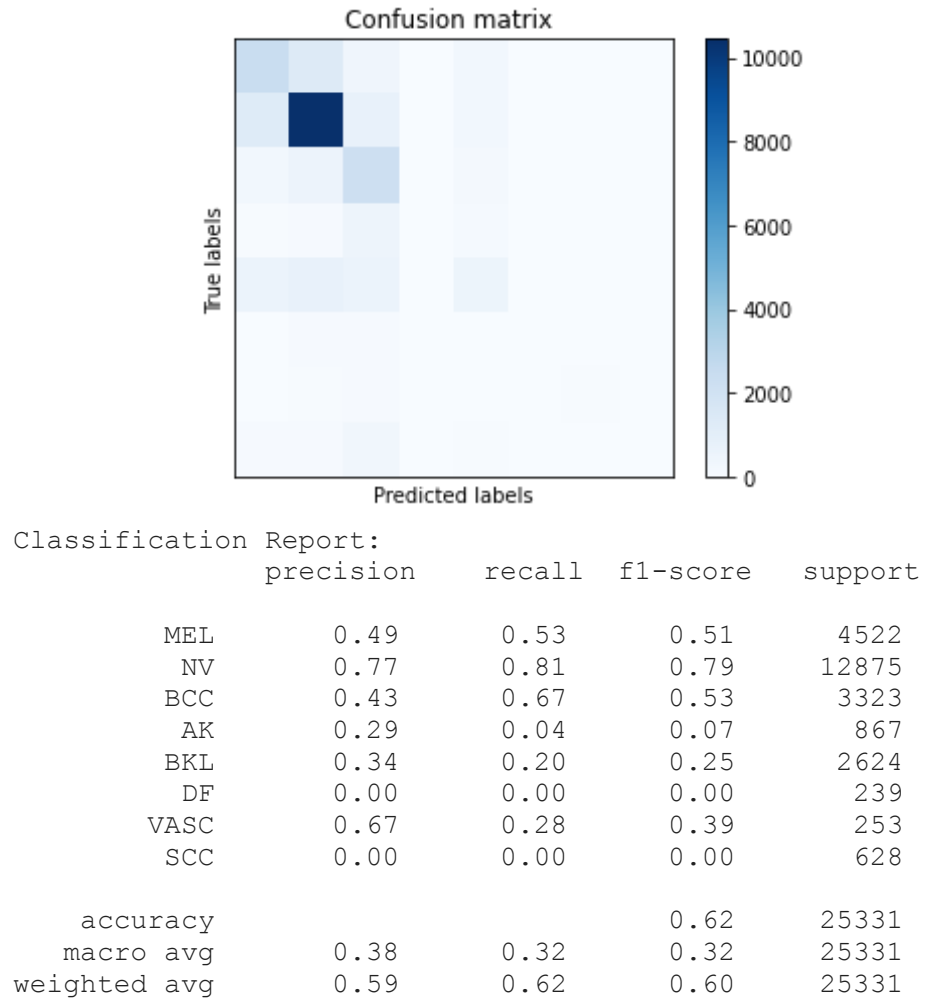


Fig 4: Confusion matrix and classification metrics for skin lesion detection model predictions

As the training dataset contained more images of MEL, NV, and BCC skin disorder types, the larger sample set led to the classifier having improved accuracy and performing better in the detection of these types of skin disorders. Per model evaluation, an image is randomly selected

from the dataset and its probability among the eight skin disorders are plotted.

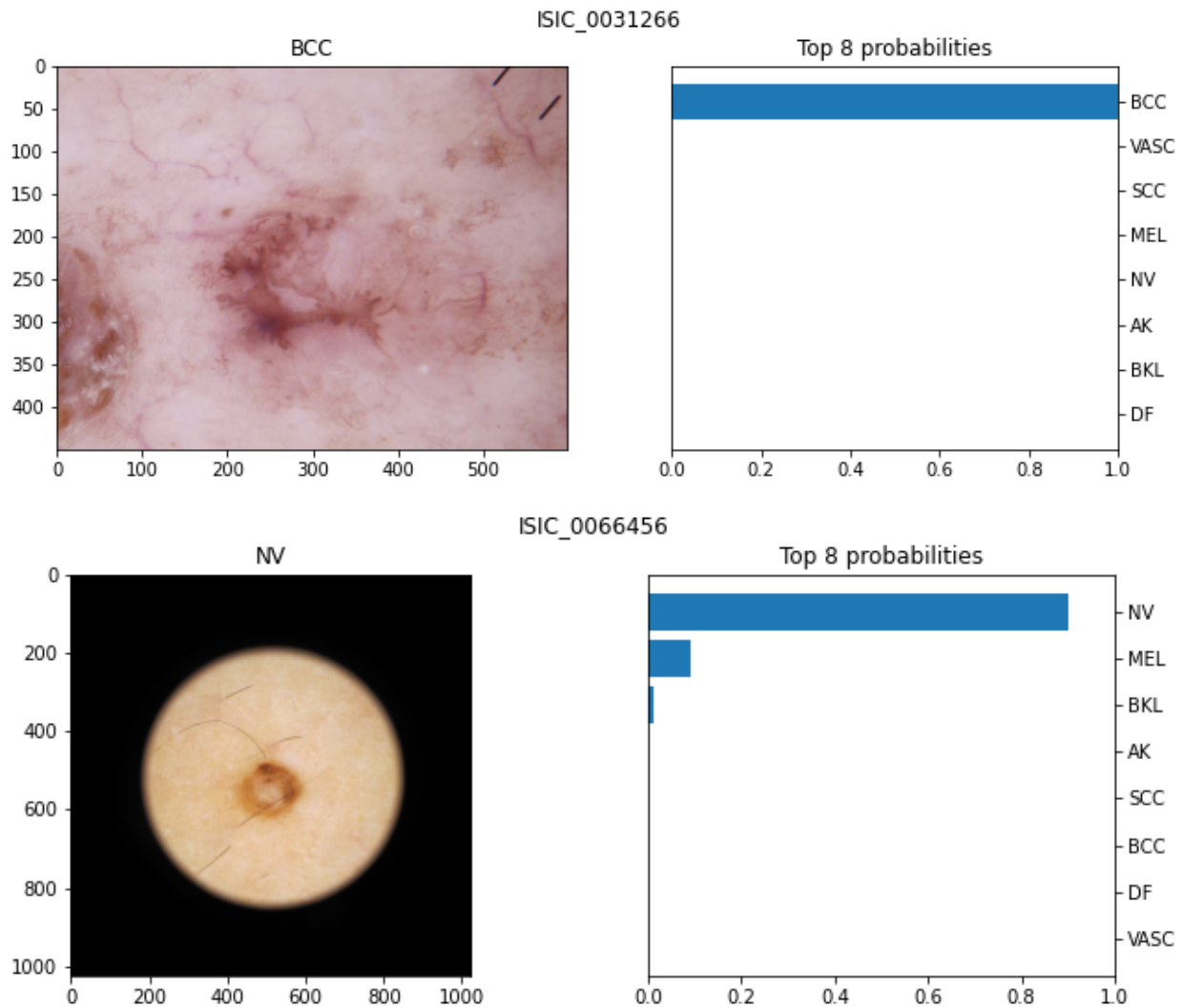
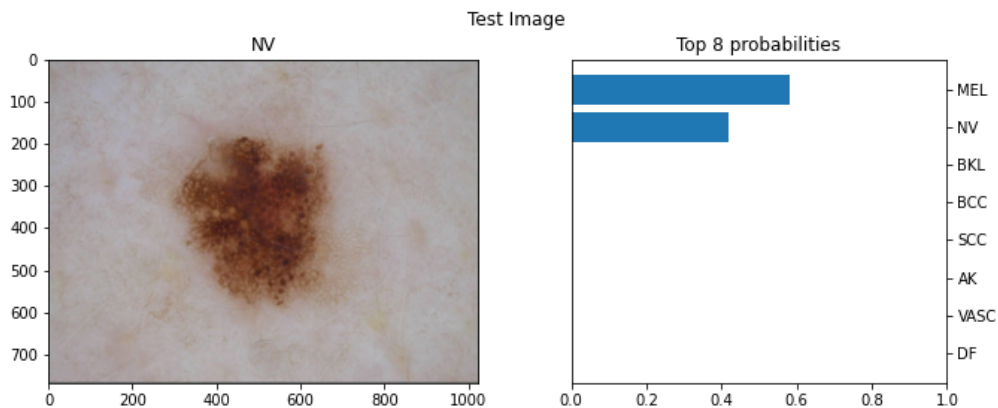


Fig 4: Testing probability in sample images from our dataset

Additionally, images from the internet of known skin disorders are downloaded and tested on the classifier model, and each instance shows consistent results in accuracy where the true disorder has the highest plotted probability.



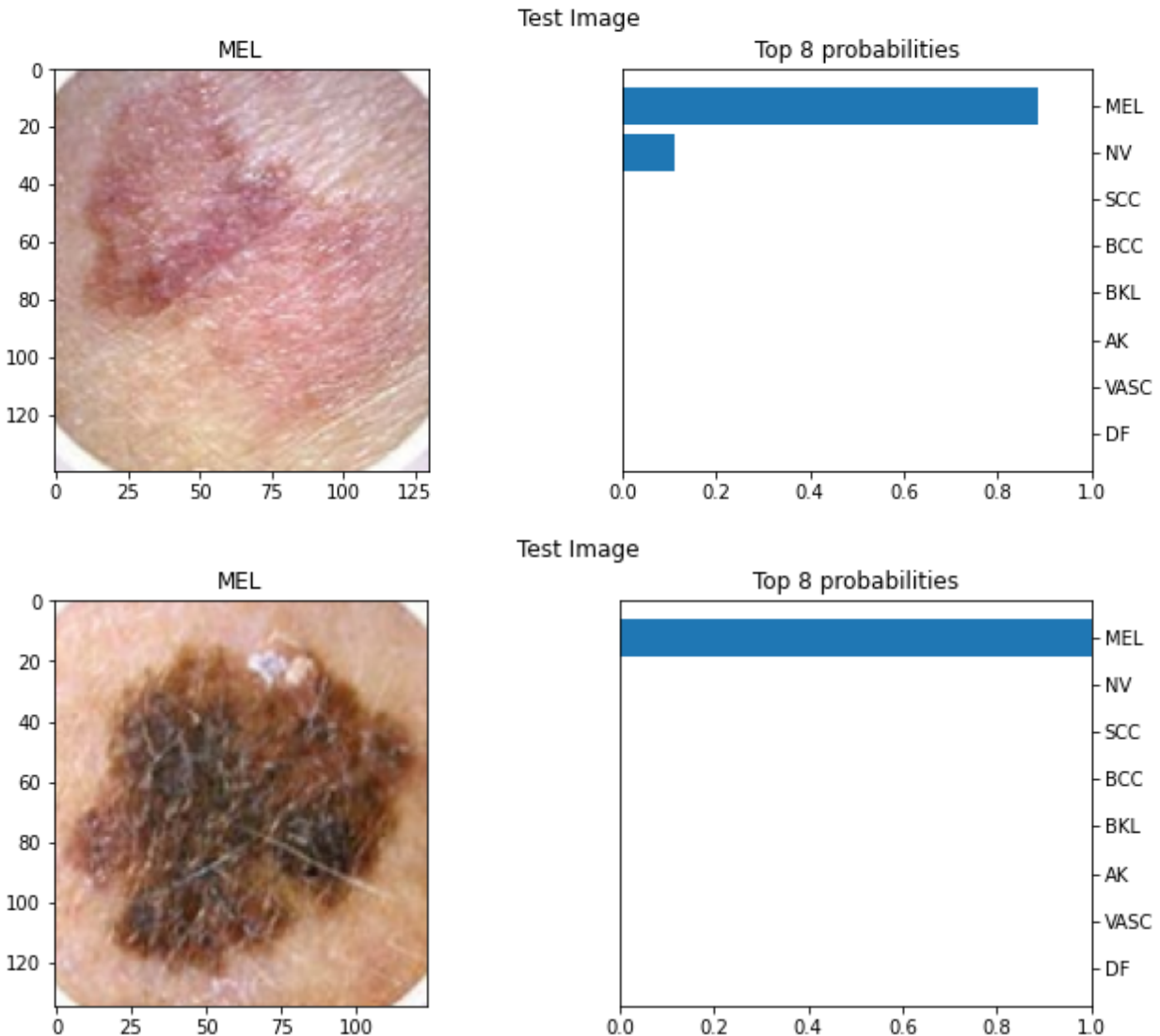


Fig 5: Testing melanoma images downloaded from internet on classifier

From the results, the neural network has a mediocre classification performance as it could show dominant classification for one class but that also increases its probability for failing on classifications of other categories.

Conclusions

From this project, we understand the ISIC challenge and the background of the competition. We also got to research different methods to come up with a possible solution in a given time. We also got to implement and refine a CNN model for different parameters to improve its accuracy.

Beside novelty detection, there are a lot of improvements for the model. For example, from Kaggle, colour constancy and segmentation could be helpful for dermoscopic image classification. Plus, a lighter model would be deployed so we could implement it with the Raspberry Pi.

In conclusion, the implementation of the convolution neural network model showed promising results for detecting skin lesion disorders. This is valuable as the classifier provides the general population with an option for early self-diagnostic for possible skin disorders simply by capturing a photograph of the affected area.

References

1. "Simon Fraser University," *Lab 3 - Neural Networks*. [Online]. Available: https://canvas.sfu.ca/files/15358216/download?download_frd=1. [Accessed: 13-Apr-2021].
2. B. Chen, "Why Rectified Linear Unit (ReLU) in Deep Learning and the best practice to use it with TensorFlow," *Medium*, 20-Jan-2021. [Online]. Available: <https://towardsdatascience.com/why-rectified-linear-unit-relu-in-deep-learning-and-the-best-practice-to-use-it-with-tensorflow-e9880933b7ef>. [Accessed: 02-Apr-2021].
3. J. Brownlee, "How Do Convolutional Layers Work in Deep Learning Neural Networks?," *Machine Learning Mastery*, 16-Apr-2020. [Online]. Available: <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>. [Accessed: 03-Apr-2021].
4. C. Kozyrkov, "What is 'Ground Truth' in AI? (A warning)," *Medium*, 02-Mar-2020. [Online]. Available: <https://towardsdatascience.com/in-ai-the-objective-is-subjective-4614795d179b>. [Accessed: 17-Apr-2021].
5. N. S. A. L. K. AlEnezi, "A Method Of Skin Disease Detection Using Image Processing And Machine Learning," *Procedia Computer Science*, 22-Jan-2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050919321295>. [Accessed: 01-Apr-2021].
6. Wanghsinwei, "wanghsinwei/isic-2019," *GitHub*. [Online]. Available: <https://github.com/wanghsinwei/isic-2019>. [Accessed: 01-Apr-2021].