Skin Cancer Detection Project Report

BY: Linoy Croitoru 326191269, Yair Yaakov 207723198, Linoy Godali(arami) 208833632

January, 2025

Abstract

This report presents our work on skin cancer detection using photographic data, with a focus on classifying images into two categories: Benign and Malignant. The project explores four different models: a baseline approach, logistic regression, a neural network (NN), and a convolutional neural network (CNN). Each model is evaluated based on its classification performance, with results demonstrated through statistical metrics and visual analysis.

1 Introduction

1.1 Project Goal

The goal of this project is to classify skin lesions as Benign or Malignant using image data. The dataset is split into training and testing sets with labeled subfolders. Four models are developed—baseline, logistic regression, neural network, and CNN—each improving on the last.

1.2 Dataset

The dataset for this project consists of images of skin lesions labeled as either Benign or Malignant. Each image is resized to a uniform resolution of 112×11 pixels, ensuring consistency for model training and evaluation. The data is divided into training and testing sets, with 2,000 images reserved for testing and the remaining images used for training. The dataset is organized into two main directories: one for training and one for testing, each containing subfolders for Benign and Malignant classes. For classification, Benign lesions are labeled as 0, and Malignant lesions are labeled as 1. This structure enables effective model development and evaluation.

Total images is 11879: Total malignant images is 5590 (47.06) Total benign images is 6289 (52.94)

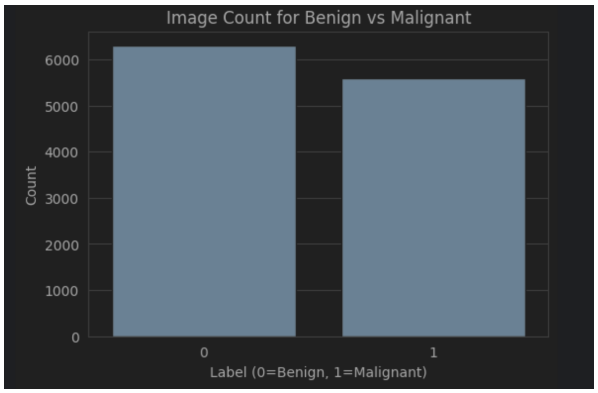

Figure 1: graph of the data count.

1.3 About The Models

This project explores different models for classifying skin lesions as Benign or Malignant, starting with simple methods and moving to more advanced ones. The baseline model is the easiest to understand,

predicting the majority class (Benign or Malignant) for all images. Logistic regression builds on this by using pixel values as features to make better predictions. A neural network, with multiple layers, is then introduced to learn complex patterns in the image data. Finally, a convolutional neural network (CNN) is implemented, utilizing specialized layers to focus on important parts of the images and achieve better classification results.

2 Models

In this section the models will be described more detailed, including the results for each model, which allowing a better understanding of the improvements from on model to the next.

2.1 Baseline Model

The baseline model predicts the majority class (either Benign or Malignant) for all images in the test dataset. It determines this majority class by analyzing the distribution of classes in the training data. This model does not rely on any features or patterns from the images, it simply assumes that predicting the majority class will yield the highest accuracy in the absence of additional insights.

2.1.1 Results

The baseline model predicts all samples as belonging to the majority class, which is "Benign" in this dataset. As expected, this simplistic approach leads to poor performance metrics, as the model does not consider any features or patterns in the data. Instead, it relies solely on the distribution of the dataset classes.

Metric	Value
Accuracy	50.00%
Precision	0.00%
Recall	0.00%
MSE	0.50

Table 1: Model Performance Metrics

True Positives (TP): 0 (Correctly identified Malignant cases)
True Negatives (TN): 1000 (Correctly identified Benign cases)
False Positives (FP): 0 (Benign cases incorrectly identified as Malignant)
False Negatives (FN): 1000 (Malignant cases incorrectly identified as Benign)
Observations:

- The model achieves an accuracy of 50%, which is equivalent to random guessing in a binary classification problem where the classes are balanced.
- Precision and recall are both 0 for the "Malignant" class, indicating that the model fails to identify any Malignant cases.
- The Mean Squared Error (MSE) of 0.50 reflects the model's inability to differentiate between the two classes effectively.
- The confusion matrix confirms that the model classifies all samples as "Benign," leading to 1000 False Negatives and no True Positives for the "Malignant" class.

Conclusion: As expected, the baseline model performs poorly due to its simplistic strategy of predicting only the majority class. This underscores the need for more sophisticated models that can learn and generalize from the dataset features to improve diagnostic reliability.

2.2 logistic regression Model

In a logistic regression model, the result obtained is the probability that a sample belongs to class 1. The calculation is done by multiplying each feature by its corresponding weight, followed by summing all the products. This operation is performed in the Dense layer, where each neuron computes the sum of weights multiplied by the features. The resulting value is then passed through the sigmoid activation function, which maps the output to a range between 0 and 1. If the probability is low, the model classifies the sample as belonging to the second class ("Benign"), and if the probability is high , it classifies it as belonging to the first class ("Malignant").

The model built here is a logistic regression model, suitable for binary classification tasks. It includes four main layers:

1. Flatten Layer:

The model starts with a Flatten layer, which serves to transform the 3D input data into a 1D vector. This conversion is necessary to ensure that the subsequent layers, particularly the Dense layer, can process the data effectively.

2. Dense Layer:

Following the Flatten layer, the model includes a Dense layer with a single neuron. This layer performs a linear calculation on the input it receives, summing the weighted inputs to generate the model's prediction. The goal of this layer is to produce the model's classification output.

3. Activation Layer:

The final layer in the model is the Activation layer, which uses the Sigmoid activation function. This function maps the output of the Dense layer to a value between 0 and 1. This value represents the probability of the input belonging to class 1 ("Malignant"). If the probability is close to 0, the sample is classified as "Benign"; if it's close to 1, the sample is classified as "Malignant."

The model has a total of 37,633 parameters. The Flatten layer has no parameters, as it only converts the input into a one-dimensional vector. The Dense layer contains 37,633 parameters, which are the weights of the neurons in the layer. The Activation layer has no parameters, as it simply applies the sigmoid function to the output of the Dense layer.

If images are 112×112 with 3 channels, then input dimension will be:

$$112 \times 112 \times 3 = 37,632$$

$$1 + (37,632 \times 1) = 37,633$$

2.2.1 Results

Training accuracy steadily improves from 63.95% to approximately 80.31% over 15 epochs, stabilizing in the later epochs. Test accuracy fluctuates significantly, ranging from 62.7% to 80%, with sharp drops observed in certain epochs (e.g., epoch 3 and epoch 8). Training loss decreases consistently from 0.6341 to 0.4235, while test loss demonstrates high variability, spiking at epochs like 3 and 8 (above 0.6) and stabilizing around 0.47 towards the end of training.

Metric	Value
Accuracy	67.85%
Precision	89.40%
Recall	40.50%
MSE	0.32

Table 2: Model Performance Metrics

True Positives (TP): 405 malignant cases correctly identified.
True Negatives (TN): 952 benign cases correctly identified.
False Positives (FP): 48 cases where benign was incorrectly classified as malignant.
False Negatives (FN): 595 cases where malignant was incorrectly classified as benign.
Observations:

- The **Mean Squared Error (MSE)** of **0.32** reflects a moderate level of prediction error, indicating that the model struggles with accurately predicting malignant cases compared to benign ones.
- Training accuracy improved steadily, demonstrating the model's ability to learn from the training data.
- Test accuracy showed notable fluctuations, suggesting potential issues with generalization or sensitivity to specific data subsets.
- While the **precision** of **89.40%** is high, indicating a strong ability to identify malignant cases, the **recall** of **40.50%** highlights the model's struggle to avoid missing true malignant cases, which could be critical in a medical diagnostic context.
- The high number of **False Negatives (FN)** underscores the need for further optimization to improve diagnostic reliability, as missing malignant cases poses significant risks.

Our goal minimizing false negatives (595 cases), which are critical for medical applications, as they involve misclassifying malignant lesions as benign.

Graphs:

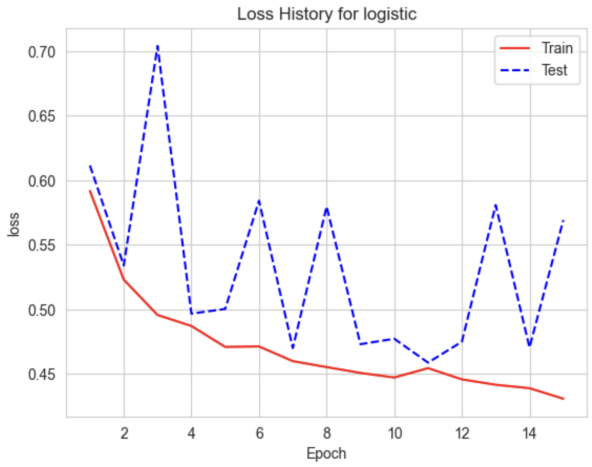

Figure 2: graph of the loss.

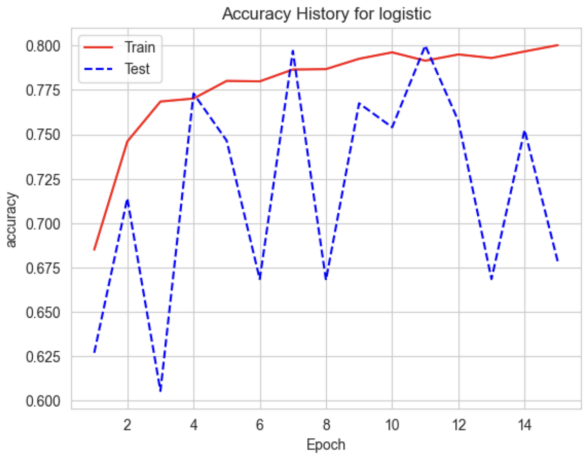

Figure 3: graph of the accuracy.

2.3 NN Model

The model we defined consists of several layers, here is an explanation of each one:

1. Flatten Layer:

The model starts with a Flatten layer, which aims to convert the 3D input into a 1D vector. This operation allows the following Dense layers to process the data more effectively.

2. Hidden Dense Layers:

After the Flatten layer, the model includes three Dense layers. Each layer contains a variable number of neurons, according to the array [128, 64, 64]. Each Dense layer uses the ReLU activation function, which is a non-linear function that allows the model to learn complex patterns from the data. These layers enable the model to identify more complex and higher-level relationships between the features of the input.

3. Output Layer:

At the final stage, the model includes an output layer with a single neuron. This layer uses the Sigmoid activation function, which is suitable for binary classification tasks. The output is a value between 0 and 1, representing the probability that the input belongs to one of the categories ("malignant" or "benign").

4. Optimization and Compilation:

The model is compiled using the Adam optimizer with a low learning rate of 0.0001. The chosen loss function is binary_crossentropy, which is appropriate for binary classification tasks. The goal of the loss function is to minimize the gap between the model's predictions and the actual values. Additionally, the model tracks the accuracy of its predictions during training using the accuracy metric.

The model consists of several layers with varying numbers of parameters, which contribute to the overall complexity of the model. The `Flatten` layer, which is responsible for reshaping the input data, has 0 parameters since it only transforms the data without learning any weights. The `Dense` layer has the largest number of parameters, totaling 4,817,024. This is due to the large number of neurons (128) and the size of the input data, requiring a substantial number of weights to connect each neuron to the previous layer. The following `Dense` layer, `dense_1`, contains 8,256 parameters, reflecting a smaller number of neurons (64) compared to the previous layer. The next `Dense` layer, `dense_2`, has 4,160 parameters, corresponding to another reduction in the number of neurons (64). Finally, the `dense_3` layer has only 65 parameters, as it consists of a single output neuron responsible for binary classification.

2.3.1 Results

The graphs show consistent improvement in the model's performance throughout the training process. The accuracy graph indicates steady improvement in training performance, with approximately **83.90% accuracy** on the test set by the end of training. The loss graph shows a gradual decrease in loss during the epochs, with test loss stabilizing at lower values toward the end. Occasional fluctuations in test performance (e.g., a spike in test loss at epoch 2) are observed, but the overall trend indicates stable and effective learning.

Metric	Value
Accuracy	83.90%
Precision	91.24%
Recall	75.00%
MSE	0.16

Table 3: Model Performance Metrics

True Positives (TP): 750 cases of malignant tumors correctly identified.
True Negatives (TN): 928 cases of benign tumors correctly identified.
False Positives (FP): 72 cases where benign tumors were incorrectly identified as malignant.
False Negatives (FN): 250 cases where malignant tumors were incorrectly identified as benign

Observations:

- Training accuracy increased consistently, reaching **84%** in the final epoch, while test accuracy fluctuated slightly but remained stable between **82% and 86%** in the later stages of training.
- The loss graph shows steady improvement, with training loss decreasing consistently and test loss stabilizing around **0.35** by the end of training.
- The balance between precision and recall indicates the model is effective at distinguishing between malignant and benign categories. However, there is room to further reduce FP and FN to improve diagnostic reliability.
- The mean squared error (MSE) provides an additional perspective on the model's prediction quality, with a value of **0.16**, indicating a low overall prediction error.

The model consists of multiple layers with a total of **4,829,505 parameters**, primarily allocated to the dense layers. This reflects the model's complexity and capacity to learn intricate patterns from the input data.

Graphs:

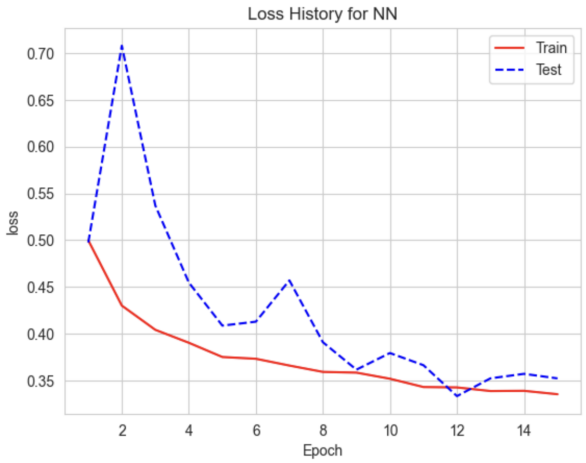

Figure 4: graph of the loss.

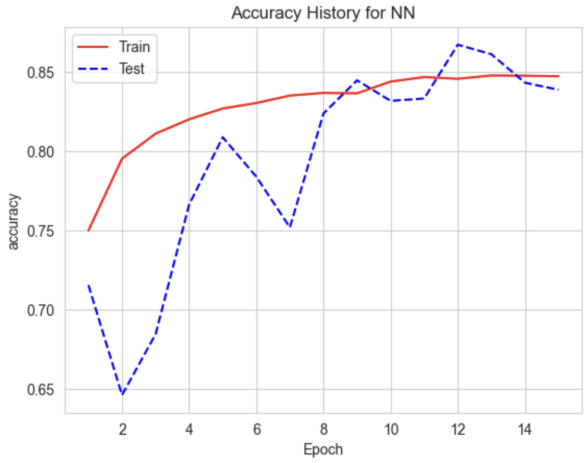

Figure 5: graph of the accuracy.

2.4 CNN Model

The model we defined is a Convolutional Neural Network (CNN) made up of several layers, each with a unique role in improving the model's learning ability and performance. Here's a detailed description of each layer:

1. First Convolutional Layer:

This layer consists of 64 filters of size 3x3, which scan the image and look for basic features like edges and geometric shapes. The `relu` activation function is used to activate the neurons and helps the model learn non-linear patterns. The `"same"` padding ensures that the output size is the same as the input size, preserving information without losing details from the image's edges. `BatchNormalization` is applied to help stabilize the model and accelerate the learning process.

2. Max-Pooling Layer:

This layer performs down-sampling by selecting the maximum value from a 2x2 region, reducing the data size and helping prevent overfitting.

3. Additional Convolutional Layers:

The model includes two more convolutional layers with 128 and 256 filters of size 3x3. These layers allow the model to learn more complex patterns such as textures and higher-level shapes. Each

convolutional layer is followed by `BatchNormalization` and `MaxPooling`, which aid in data reduction and improve the model's general performance.

4. Flatten Layer:
After the convolutional layers, the output is passed to the Fully Connected (Dense) layers. Since the data is still in a 2D format , the Flatten layer converts it into a 1D vector, making it ready for the Dense layers.

5. Fully Connected (Dense) Layer:
This layer has 256 neurons, which receive the output from the previous layer and connect to all neurons in the Dense layer. Here, the model learns the most complex relationships between the features identified in the previous layers. `Dropout` is also applied in this layer to help prevent overfitting by randomly deactivating neurons during training.

6. Output Layer:
The final layer has one neuron with a `Sigmoid` activation function, suitable for binary classification tasks. The output is between 0 and 1, representing the probability that the input belongs to one of the two categories .

7. Optimization and Regularization:
The model is trained using the Adam optimizer with a low learning rate of 0.0001, which adapts the learning rate for each parameter. The loss function used is `binary_crossentropy`, which is appropriate for binary classification, aiming to minimize the gap between the predicted probabilities and the actual values. Additionally, `EarlyStopping` is implemented to halt training if no improvement is observed after a few epochs, preventing overtraining.

2.4.1 Results

The graphs show significant improvement in the model's performance throughout the training process. The accuracy graph indicates steady improvement in training performance, with approximately **92.35%** accuracy on the test set by the end of training. The loss graph shows a consistent decrease in loss during the epochs, with test loss stabilizing at lower values toward the end. While there are occasional fluctuations in test performance (e.g., slight increases in test loss at epochs 8 and 13), the overall trend demonstrates a stable and effective learning process. Additionally, the model achieved a **Mean Squared Error (MSE)** of **0.08** on the test set, which indicates a low average squared difference between the predicted probabilities and the actual values, demonstrating high reliability in classification performance.

Metric	Value
Accuracy	92.35%
Precision	92.56%
Recall	92.10%
MSE	0.08

Table 4: Model Performance Metrics

True Positives (TP): 921 cases of malignant tumors correctly identified.
True Negatives (TN): 926 cases of benign tumors correctly identified.
False Positives (FP): 74 cases where benign tumors were incorrectly identified as malignant.
False Negatives (FN): 79 cases where malignant tumors were incorrectly identified as benign.
Observations:

- Training accuracy steadily improved, reaching approximately **92.64%** in the final epoch, while test accuracy remained stable between **92% and 93%** in the later stages of training.
- The loss graph shows steady improvement, with training loss decreasing consistently and test loss stabilizing around **0.3242** by the end of training.
- The **MSE** value of **0.08** further supports the model's high accuracy and low error rate, as it quantifies the model's ability to make accurate predictions.
- The balance between precision and recall highlights the model's ability to distinguish effectively between the two categories (malignant and benign). However, there is still room to further reduce **FP** and **FN** to improve diagnostic reliability.

The model consists of several layers with varying numbers of parameters, contributing to its overall complexity. The total number of parameters is **13,219,201**, most of which are allocated to the dense layers.

Graphs:

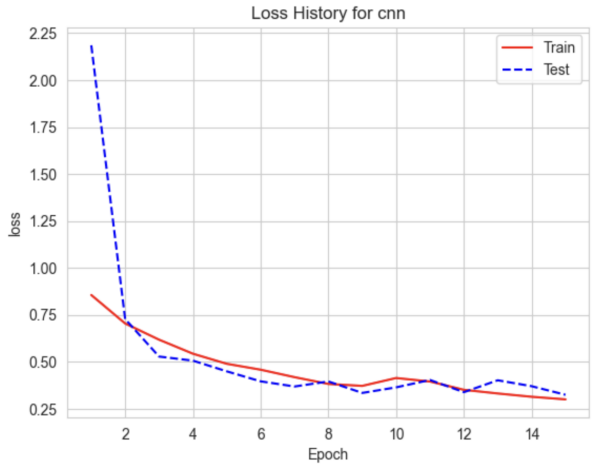

Figure 6: graph of the loss.

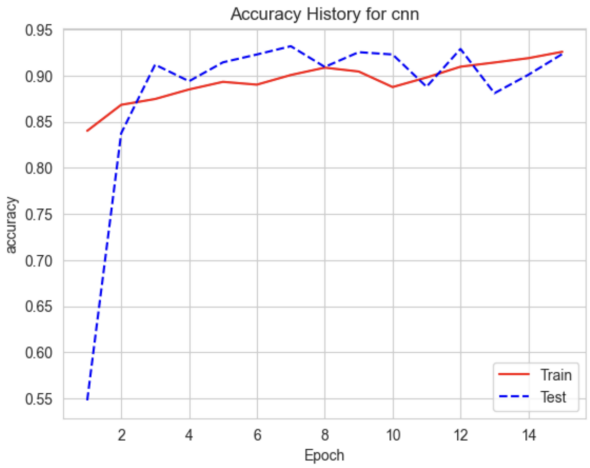

Figure 7: graph of the accuracy.

3 Conclusions

Model	Accuracy	Precision	Recall	MSE
Baseline	50%	0%	0%	0.5
Logistic Regression	67.85%	89.40%	40.50%	0.32
NN	83.90%	91.24%	75.00%	0.16
CNN	92.35%	92.56%	92.10%	0.08

Table 5: Model performance metrics

During the process, we evaluated four different models to identify the most effective one for the task. The Convolutional Neural Network (CNN) achieved the best results, demonstrating superior performance across key metrics such as accuracy, recall, and precision. This made it the most suitable model for solving our problem. We were particularly interested in the **Recall** metric due to its importance in scenarios where missing a positive case (False Negative) can have serious consequences. Recall measures the proportion of actual positive cases that the model correctly identifies, making it crucial in situations like disease diagnosis. By maximizing Recall, we ensure that as many positive cases as possible are detected, even if it means accepting a higher number of false positives. This focus aligns with our goal of prioritizing safety and reducing the likelihood of overlooking critical cases.

Dataset - Bhavesh Mittal. Melanoma Cancer Dataset, Kaggle.

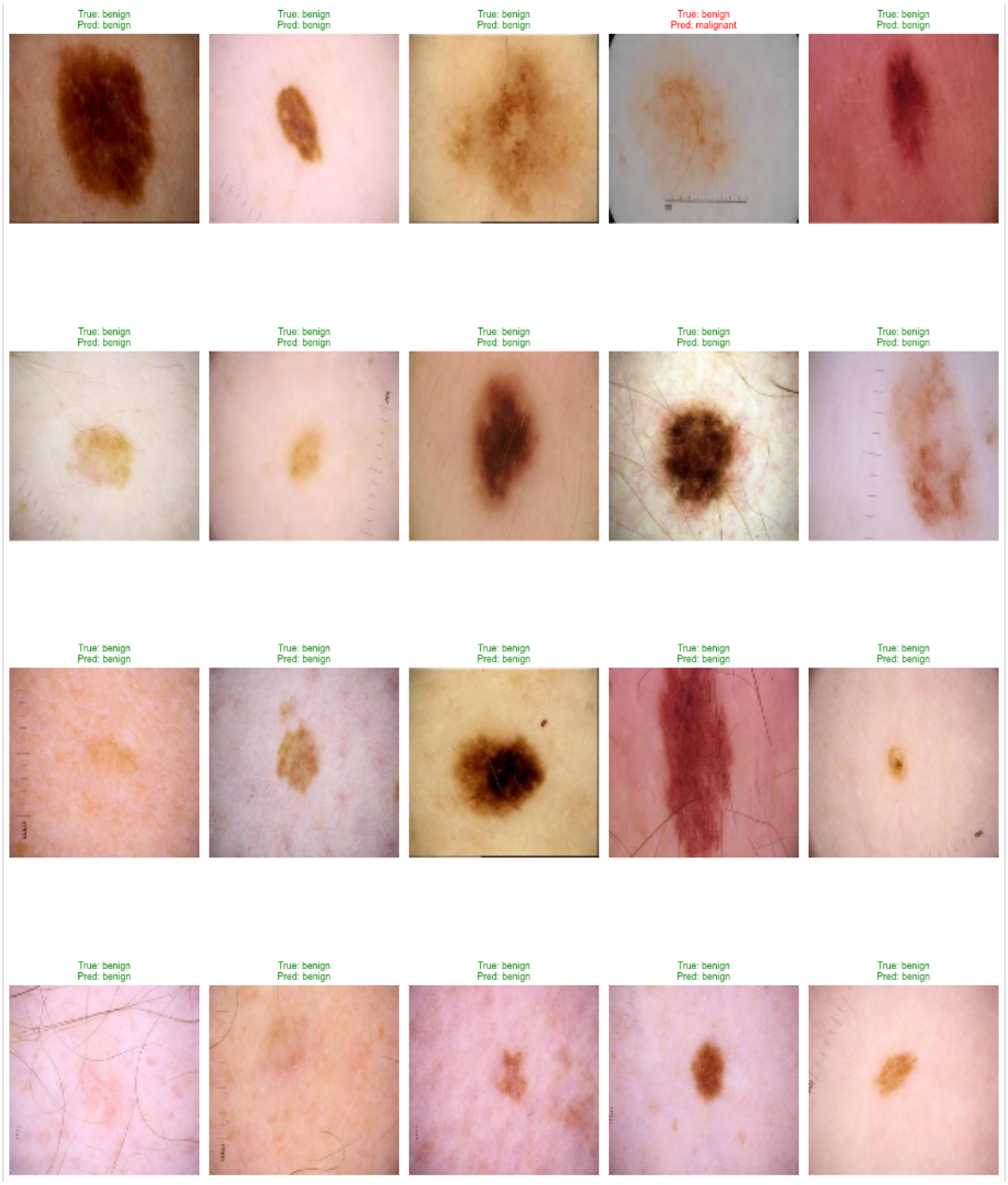

Figure 8: A collage of results from the CNN model.