[redacted]

Skin Cancer Detection: A Comparative Analysis of Support Vector Machines and Convolutional Neural Network on Melanoma Image Classification

[redacted]

[redacted]

Date: 17/05/2024
Pages: 15
Characters: 37.486

Contents

Abstract

Machine Learning is the most important branch of Artificial Intelligence and it's the capacity of the machine to constantly improve its performance autonomously without the need for a human to explain and guide it through the tasks (Brynjolfsson&McAfee, 2017).
Since the discovery of ML, several fields decided to implement it to improve efficiency, productivity, forecasting, and predictions. One of the sectors that is currently benefitting from technological innovation, and in particular machine learning, is healthcare, from research to personalized medicine to the diagnostic process.
The innovation has in fact proven extremely effective in predicting different types of cancer, such as breast, brain, lung, liver, and prostate cancer (Zhang, Shi&Wang,2023).
In this paper, we will focus on a dataset of pictures of melanoma, a type of skin tumor, and we will build different models of machine learning to achieve great reliability in identifying if the tumor is benign or malignant.
The No Free Lunch (NFL) Theorem explains that if no assumption is about the data, then there is no reason to prefer a specific model, they all need to be tested. In practice, this is not possible, so some assumptions have to be made about the data and only reasonable models are tested (Géron, 2019). In this paper, the dataset will be manipulated with Random Forest, Principal Component Analysis, Naive Bayes, Support Vector Machines, and Convolutional Neural Networks and the results of the different models will be compared in an analysis for efficiency and accuracy.

Keywords: Medical Image Analysis, Naive Bayes, Random Forest, Principal Component Analysis, Convolutional Neural Network, Support Vector Machine, Skin Cancer Detection, Data Augmentation.

Introduction

Cancer is a disease that causes the body's cells located in an area to grow uncontrollably and spread also to other parts. Usually, human cells grow and multiply in an orderly fashion and when old or damaged they die, but if an individual gets sick they keep on growing even when damaged and they multiply causing lumps. These masses can be benign or malignant, then called cancer. When a tumor is cancerous it will attack other organs or tissues forming new tumors (the process of metastasis).

In the opinion of experts, cancer is the second leading cause of death in the world, following heart disease. Melanoma skin cancer is the fifth most common type (Morgan&Khatri, 2022). Melanoma is a skin cancer of the melanocytes, which are the cells engaged in the production of the pigment melanin for skin colour.

To improve prediction and reduce invasive diagnostic procedures, machine learning (ML) has been increasingly implemented in medicine. With decreasing storage and computation costs, ML models have been applied to achieve more accurate results in oncology, cardiology, and other fields (Burton, Fathieh, Nemati et al., 2024). Previous studies have demonstrated the potential of ML algorithms in various medical fields, including the classification of skin lesions (Esteva et al., 2017; Haenssle et al., 2018). These studies have shown that ML models, particularly those based on deep learning, can achieve diagnostic performance comparable to or exceeding that of dermatologists.

Image analysis will be the particular focus of our project, implemented on a dataset of images of melanoma both benign and malignant. Different models will be trained to recognized the type of melanoma through its image, with efficiency, accuracy and other parameters taken under consideration and compared.

Motivation

The risk of melanoma has been on the rise recently, so it's important to invest in better diagnosis practices. This is because when melanoma is diagnosed early, it can be treated. However if not, it can reach a metastatic state and treatment becomes less effective (Lapides, Saravi, Mueller&al, 2023). According to the American Cancer Society, melanoma accounts for only about 1% of skin cancers but causes a significant majority of skin cancer deaths, highlighting the need for early and accurate detection (American Cancer Society, 2021).

Diagnosis is fundamental for patient care, research, and guidelines. Unfortunately, there are risks associated with diagnostic testing, both with underuse as said above, but also with overuse. Aggressive testing, which is often linked to physicians' fear of missing some signals about the possible disease, is putting patients at risk, and not improving certainty (Committee on Diagnostic Error in Health Care; Board on Health Care Services; Institute of Medicine; The National Academies of Sciences, Engineering, and Medicine, 2015).

The motivation behind this study is to leverage the advancements in ML to develop a reliable model, to predict whether a melanoma is benign or malignant. With our dataset composed of images of nevis and bruises, we aim to train a model that can assist dermatologists in making faster and more accurate diagnoses. This approach has the potential to reduce the workload on healthcare professionals but also to provide diagnostic support in regions with limited access to preventive medicine.

Research Question

" What is the effectiveness of Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) in accurately detecting and classifying skin cancer, and how do these models perform in terms of balancing false positives and false negatives in clinical diagnostics?"

This question aims to explore the effectiveness of the two machine learning techniques in identifying the type of melanoma through image analysis. The goal is to develop a reliable model that can assist dermatologists in making faster and more accurate diagnoses, ultimately improving patient care and reducing the risks associated with both underuse and overuse of diagnostic testing.

Related Work

Already in 2015, Shivangi, Vandana and Nitin were presenting "Computer Aided Melanoma Skin Cancer Detection Using Image Processing", applying automatic thresholding and border detection method to skin lesion segmentations. Many researchers in those years were focusing on shape, colour, texture and luminance using different techniques.

In 2021 with "Diagnosis of skin cancer using machine learning techniques" by Murugan and colleagues, median filter and mean shift segmentation are used to filter the skin and then Support Vector Machine, Probabilistic Neural Networks and Random Forest are the techniques adopted. SVM and CNN methods are used also in 2023 by Viknesh, Kumar, Seetharaman and Anitha in “Detection and Classification of Melanoma Skin Cancer Using Image Processing Technique”.

Therefore, factoring in previous relevant works, we decided to use as main models for our analysis SVM and CNN.

Conceptual Framework

In order to fully understand the research work performed, it is necessary to be familiar with several key concepts and techniques in the realm of machine learning. These include supervised learning, feature engineering and model evaluation metrics.

Supervised Learning: Supervised learning involves training a model on a labelled dataset, where the model learns to map input features to a target variable. Common algorithms include Naïve Bayes, decision trees, and neural networks. Our classification research is a typical supervised learning task.

Feature Engineering: Feature engineering transforms raw data into useful features. This process comprises feature selection and dimensionality reduction. In our research, We use Random Forest for feature selection to identify and rank the most important features and Principal Component Analysis for dimensionality reduction to reduce the number of features or dimensions in our dataset while retaining as much of the original information as possible.

Model Evaluation Metrics: Evaluating the performance of a machine learning model is crucial to understanding its efficacy and comparing it with the other models. Metrics such as accuracy, precision, recall, and F1-score provide a comprehensive view of the model's performance in terms of correctly predicting classes, the balance between precision and recall, and overall predictive accuracy. (Géron, 2019).

Methodology

Data Preparation

Our dataset is a collection of pictures of melanomas and bruises, divided into three categories: Benign, Malignant, and Undetected. Before proceeding with the development of the models, we need to ensure that the data we are working with is valuable, meaning it must be of high quality and balanced in terms of the number of samples in each category. To achieve this, we start by filtering the data by checking for duplicates, ensuring that the size of the images is appropriate for future resizing, and confirming that they all fall within a colour gradient range. The next step is normalizing the data to guarantee a consistent scale, and lastly, augmenting the data to aim for a similar number of pictures in each of the three categories.

Training Strategy

In this project, we used a combination of deep learning techniques and conventional machine learning techniques to classify melanoma from picture data through a multifaceted training strategy.

We initially started with a Naive Bayes model as a baseline due to its simplicity and efficiency. However, we quickly encountered underfitting, as the model was too simplistic to capture the complex patterns in the data. Recognizing the need for more robust models, we transitioned to using SVM models without any feature engineering. During this phase, we encountered significant computational constraints that necessitated dataset sampling, as kernel restarts occurred frequently even when utilizing uCloud resources.

Next, we incorporated feature selection using a Random Forest classifier, which reduced the dimensionality based on feature importances. Finally, we applied PCA to the features selected by the Random Forest to further reduce dimensionality.

Moving on to deep learning, we used TensorFlow and Keras to create Convolutional Neural Networks (CNNs). The CNN architecture consisted of standard convolutional and pooling layers, followed by batch normalization to stabilize and accelerate training. The final model was further refined by adding a dense layer and a softmax output layer for classification.

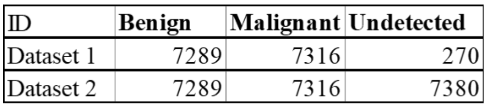
We intended to provide a detailed study of our dataset by comparing and contrasting these various methodologies, which allowed us to properly evaluate the efficacy of new deep learning techniques versus classic machine learning methods. This training strategy ensured that each model was optimized for performance and could be reliably assessed in classifying melanoma.

Data description

All images varied in resolution and quality, this means that they necessitate of preprocessing steps to ensure consistency. Each image was resized to a uniform size, we experimented using both 128x128 pixels and 64x64 pixels, and pixel values were normalized to a range of [0, 1]. These preprocessing steps were crucial in preparing the data for input into the CNN models.

To improve our training data, we also applied data augmentation techniques, including operations such as rotation, flipping, shifting, and zooming, which increased the diversity of the training dataset. These augmentations helped to balance the three categories, prevent overfitting and improved the model's ability to generalize to new, unseen data. Details of the data augmentation techniques will be discussed in a future section of this paper.

Here is the dataset image distribution:



ID	Benign	Malignant	Undetected
Dataset 1	7289	7316	270
Dataset 2	7289	7316	7380

Table 1: Overview of the amount of data

Data preprocessing

- <u>Data augmentation</u>

To tackle the challenge of expanding our dataset with diverse types of skin injuries, we collected numerous images of non-melanoma skin conditions, such as bruises, pimples, and scars, from public image repositories. Some of the publicly available pictures depicted faces with acne, or whole arms with bruises. As the benign and malignant parts of the dataset were mainly populated with close-up images of skin, the pictures selected to populate the undetected part of the dataset were cropped so they only showed a portion of the skin and not the entire arm or face. This was done to maintain a certain consistency in the dataset in order to avoid accidentally introducing biases in the model, e.g., all the images of a whole arm contain being automatically classified as undetected. Recognizing the substantial time commitment required to manually gather, crop and label a dataset of comparable size to our existing collections, which contain around 7200 images for each benign and malignant melanomas, we opted for image augmentation as a strategic solution.

Image augmentation involves artificially increasing the diversity of the dataset by applying random, yet realistic, transformations to existing images. Techniques such as rotation, zoom, shear, and horizontal flipping are used to generate new images, thus enhancing the robustness of our models without the need for additional original images. This approach not only saves significant time and resources but also improves the model's ability to generalize from the training data to real-world scenarios, making it more effective in diagnosing diverse skin conditions.

After augmentation was performed, we decided to leave the code commented. This choice is due to the fact that running the code again would result in a modification of the augmented dataset, which may cause the performance of the models to slightly vary from the ones presented in the report.

- Data filtering

To guarantee the optimal training and testing of the model we need to filter the data by checking for duplicates, aspect ratio anomaly check, and RGB color anomaly.

We detect the duplicates using three methods: Average Hash (aHash), Difference Hash (dHash), and Wavelet Hash (wHash).

Average Hash (aHash) works by converting an image to grayscale and resizing it to a small, fixed size. It then calculates the average pixel value and compares each pixel to this average. Difference Hash (dHash) also starts by converting the image to grayscale and resizing it, but to a slightly larger size to compare adjacent pixels. It checks whether the left pixel is brighter than the right pixel for each pair and assigns 1 or 0 based on this comparison. The resulting binary string captures the pattern of intensity differences. Lastly, Wavelet Hash (wHash), that uses the discrete wavelet transform (DWT) to break down an image into different frequency components after converting it to grayscale and resizing it. It focuses on the low-frequency components to generate a binary hash that captures the image's overall structure and finer details (Zauner, 2010).

We decided to implement all three methods because each method had low accuracy, as shown in Figure 1. This is because our images are all quite similar, as we are working with melanomas and bruises on small portions of skin. In the end, we decided to manually check the duplicates and remove the images that were actually duplicates.

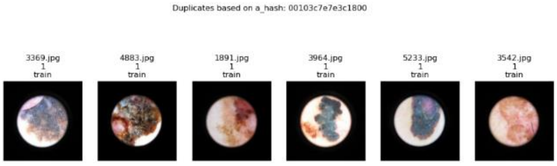


Figure 1: Instance of false duplicates

We proceed to ensure that all the images fit within the aspect ratio range of 0.25 to 4. This process is necessary because, to implement the CNN model, fixed-size input tensors are required. For SVM, resizing to a uniform size of 128x128 pixels simplifies the preprocessing step, allowing each image to be converted into a consistent feature vector. More generally, resizing ensures consistency, reduces computational load and memory usage, and enables faster processing and training.

Our results show that all the images are within the specified range, so it is not necessary to remove any of them. We can proceed with the final step of filtering: checking the average RGB values.
It is important to consider that our dataset has a significant limitation: it includes only a limited range of skin colours, primarily white skin tones. Even within this limited range, there is still a broad spectrum of undertones, from more yellow-dark to more pink-light. For this reason, detecting anomalies in RGB values is challenging and can lead to errors, as illustrated in Figure 2. Additionally, with the goal of further developing the model to include a diverse range of skin colours in the future, filtering out images based on RGB values could compromise the integrity of the full dataset. For this reason, we decide to not take into consideration the outputs and proceed with our code.

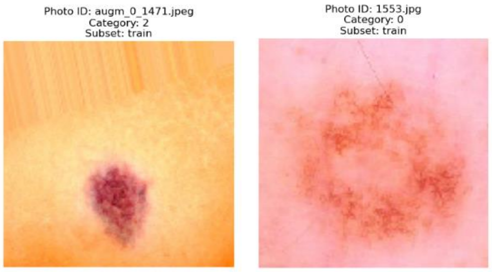


Figure 2: Instances of images out of the RGB range

- <u>Data Normalization</u>

After the necessary feature engineering, the last step before proceeding with modelling was data normalization. In the case of the data presented, we had to check if the image arrays were in 2 dimensions. This is in fact a matrix requirement and flattening the image is a fundamental step. The arrays were 3 dimensional, therefore we proceeded to flatten them and do standard scaling, fitting the scaler to the data and transforming it.

Modelling framework

Naïve Bayes – Baseline Model

Naive Bayes classifiers are a group of supervised learning algorithms that rely on Bayes' theorem. They operate under the "naive" assumption that all features are conditionally independent of each other given the class label. Despite this seemingly simplistic assumption, Naive Bayes models have proven to be effective in various real-world applications, such as document classification and spam filtering. One of the key advantages of Naive Bayes is its efficiency; it requires only a small amount of training data to estimate the necessary parameters and is computationally faster than more complex models. This is partly because the model independently estimates the distribution of each feature, which also helps to mitigate issues related to the curse of dimensionality (scikit-learn, n.d.).

As a baseline model, we trained a Naive Bayes classifier on the raw data. We split the dataset into training and testing subsets based on predefined indices. The features were scaled, and the Naive Bayes model was trained on the training data.

While this baseline provided a useful benchmark, the Naive Bayes model proved to be inadequate for our task, as it struggled to capture the complex patterns in the data, leading to underfitting. The model's simplicity, which relies on the assumption of feature independence, was insufficient to achieve satisfactory accuracy, highlighting the need to explore more sophisticated models.

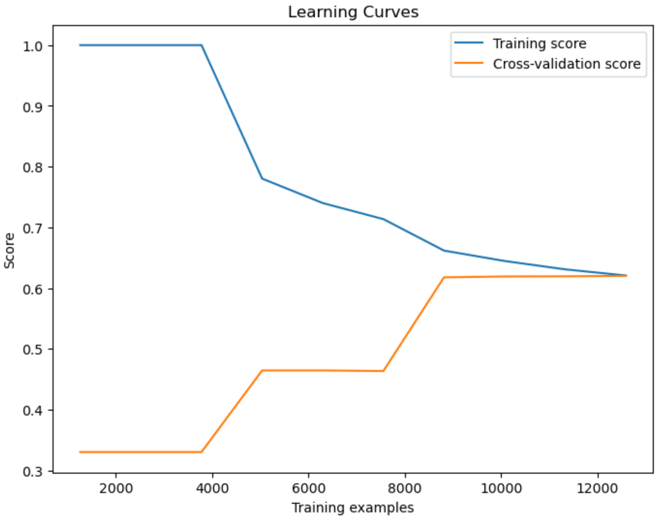


Figure 3 : Naïve Bayes Learning Curve

Support Vector Machines (SVM)

Support Vector Machines (SVM) is a supervised learning method used for classification, regression, and outlier detection. To formulate predictions, it utilizes a set of possible linear functions as models within a high-dimensional feature space. The efficiency of SVM lies in its ability to handle non-linear data by using kernel functions to transform the data into a higher-dimensional space where it becomes linearly separable.

The main applications of this model include text classification, image recognition, bioinformatics, and handwriting recognition. We decided to implement SVM for our dataset due to its efficiency in image classification, as it handles high-dimensional data effectively. Additionally, SVMs are less prone to overfitting compared to other algorithms such as neural networks. Furthermore, its application in handwriting analysis and face recognition (pattern classification and regression-based applications) has shown better results than more complex neural network models (Scikit-learn, n.d.). For these reasons, SVM is a valid option to classify melanomas and bruises.

In the implementation of our model, we decided to test SVM with and without Feature Engineering. It is known that using Random Forest, Principal Component Analysis (PCA) and Support Vector Machine (SVM) together in a pipeline smooths the process of preparing the data. This approach is particularly useful when the data is complex and large, as in our dataset, and can lead to better accuracy. For research purposes, we decided to implement our code with both options to test the efficiency of using feature selection and dimensionality reduction (Jolliffe&Cadima, 2016).

To achieve the best accuracy in our model, we tested the images with different pixel sizes and cross-validation subsets, for the GridSearchCV, during the training and evaluation phases, looking for the best compromise between accuracy, complexity, and running time.

We start with data preprocessed with lower quality images (64 x 64 pixels), GridSearchCV with a 3-fold cross-validation to test our model, with the intention of increasing these parameters in subsequent tests to achieve better accuracy. From the results, we observed that 3-fold cross-validation and 5-fold cross-validation did not significantly influence the accuracy, so we decided to keep cv=3 for complexity and running time efficiency. We proceeded by augmenting the quality of the images to 128 x 128 pixels to achieve better accuracy. We also tested this model with both 3-fold cross-validation and 5-fold cross-validation to see if there were any differences. As in the previous results, the number of subsets in the cross-validation did not significantly change the model's results. Therefore, we chose 3-fold cross-validation for complexity and running time efficiency. Notably, the results show an increase in accuracy from 64 x 64 pixel images to 128 x 128 pixel images. This improved performance demonstrates a good trade-off between accuracy, complexity, and running time, making it the best option for our model.

During our experimentation, we attempted to run the SVM model without feature selection. However, this approach caused the system to crash repeatedly due to the high computational demands of processing the entire dataset at once. To mitigate this issue and ensure the practicability of our experiments, we divided the dataset into smaller batches. This batch processing approach allowed us to manage memory usage more efficiently and complete the training without overwhelming the system resources, maintaining the integrity of our experimental results.

We evaluated then the performance of two models using different feature selection and dimensionality reduction techniques before applying a Support Vector Machine (SVM) classifier. The first model utilized Random Forest (RF) for feature selection followed by SVM, while the second model incorporated RF for feature selection combined with Principal Component Analysis (PCA) for dimensionality reduction before applying SVM. The rationale behind this approach was to assess whether the additional step of PCA would improve classification performance by reducing the feature space.

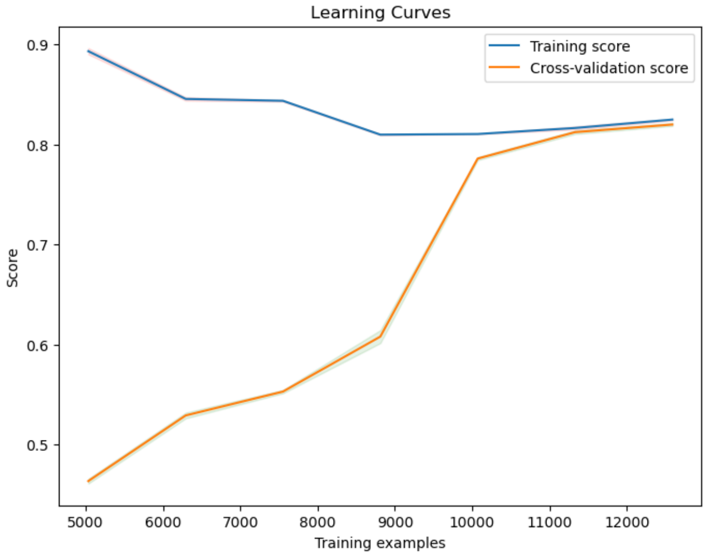


Figure 4: SVM with RF and PCA Learning Curve

Convolutional Neural Networks (CNN)

Emerged from the study of the brain’s visual cortex, convolutional neural networks (CNNs) have been used in image recognition since the 1980s, and in the last few years have managed to achieve incredible performance on some complex visual tasks., thanks to the increase in computational power and the amount of available training data (Géron, 2019). Their applications include healthcare, social network analysis, audio and speech processing (like recognition and enhancement), visual data processing methods (such as multimedia data analysis and computer vision), and NLP (translation and sentence classification), among others (Alzubaidi L et al., 2021).

We decided to employ CNNs because they are highly effective for extracting features from images and are standard for image classification tasks. The building part of CNN is in fact the convolutional layer: neurons in the first convolutional layer are not connected to every single pixel in the input image, but only to pixels in their receptive fields (Géron, 2019).

As the CNN we built uses TensorFlow, the first thing was to transform the dataframe into a Tensorflow dataset object. This time we did not apply Random Forest or PCA, but instead we used the original, but preprocessed images as CNNs are capable of feature extraction. Preprocessing consisted of standardizing pixel values to a range of [0, 1] and scaling each picture was to 128x128 pixels.

CNN Architecture

Our CNN model consists of three convolutional layers with 32, 64, and 128 filters, respectively. Each convolutional layer is followed by a max-pooling layer and batch normalization. Max-pooling is used to retain the most significant features while reducing the spatial dimensions of the data, thus enhancing the model's efficiency and mitigating the risk of overfitting. Batch normalization, as suggested by Ioffe and Szegedy (2015), is employed to normalize the inputs of each layer, which accelerates training and reduces the model's sensitivity to initialization.

The architecture is crafted to capture progressively higher-level features and patterns in the input images. After the convolutional layers, the feature maps are flattened into a 1D vector, which is then passed to a dense layer with 512 units. This dense layer helps in capturing complex patterns within the data. To further prevent overfitting, a dropout layer with a rate of 0.7 is included. The final layer is a dense layer with a softmax activation function, used for multi-class classification across 3 classes.

The model was compiled using the Adam optimizer, which adapts the learning rate for each parameter, and the categorical cross-entropy loss function, which is well-suited for multi-class classification problems. To optimize the training process, early stopping was implemented to terminate the training if the validation loss did not improve for 10 consecutive epochs. Additionally, a learning rate reduction mechanism was applied, which reduces the learning rate by a factor of 0.2 if the validation loss plateaued for 5 epochs, with a minimum learning rate of 0.0001. These strategies were employed to minimize the loss function and prevent overfitting.

The model was trained for 30 epochs, which is a typical starting point to balance training time and performance. Following the training, the model was evaluated on the test dataset.

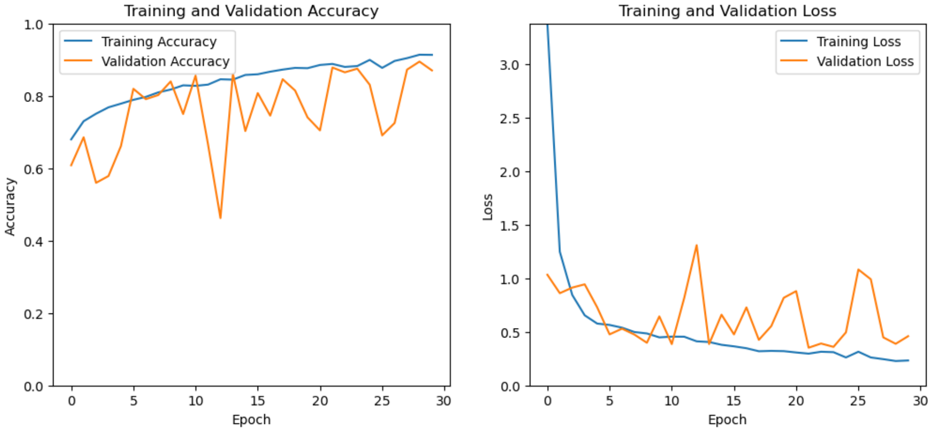


Figure 5: CNN Learning Curve

Complexity

Exploring model complexity at the end of our analysis is crucial because it helps us understand the trade-offs between accuracy and computational efficiency and assess the risk of overfitting.

Starting with Support Vector Machine is a powerful tool but inherently complex due to their reliance on finding the optimal hyperplane that separates data into classes. This involves solving a quadratic optimization problem, which can be computationally intensive, especially with large datasets. The complexity further increases when using kernel functions to handle non-linear separations, as the kernel trick transforms data into higher dimensions.

With regard to the complexity of our Convolutional Neural Network (CNN), it arises from its multi-layered architecture. This structure allows the model to capture and learn hierarchical patterns in the data, enhancing its ability to recognize complex features. However, this also means the model requires substantial computational resources, especially during training, as it involves numerous operations for each convolution and pooling step. The inclusion of a dense layer and a dropout layer adds to the computational load, but these components are crucial for improving model performance and preventing overfitting. The overall complexity, while necessary for achieving high accuracy, results in longer training times and greater memory usage, highlighting the trade-offs between model sophistication and computational efficiency.

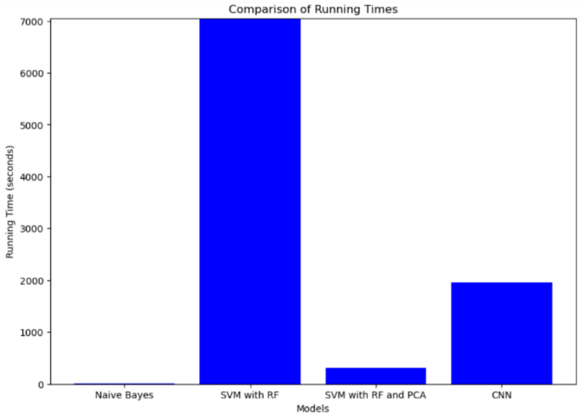


Figure 6: Histogram of the running times

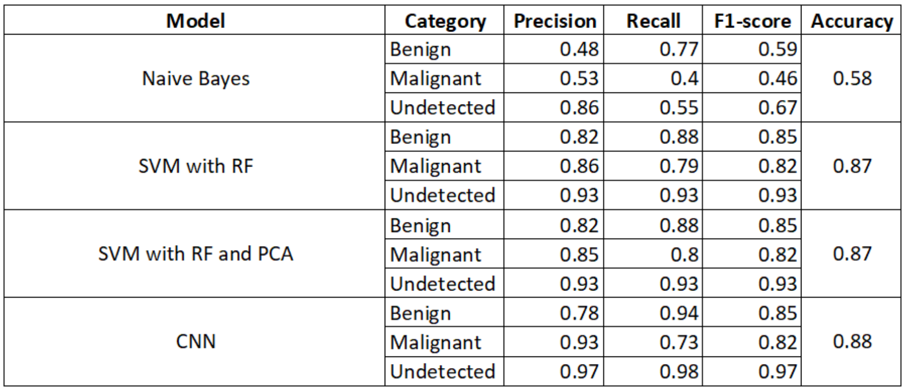
A further analysis on the computational requirements of the different models can be done by comparing the running times of each model.

In the comparison of running times among the various models, Naive Bayes emerges as the fastest, demonstrating its computational efficiency, particularly with simpler operations. The SVM with Random Forest (RF) stands out as the slowest model, taking significantly longer than the others. This is likely due to the complex ensemble nature of combining SVM with a Random Forest, which involves substantial computations during both the training and prediction phases.

The SVM with RF and PCA achieves a substantial reduction in running time compared to the SVM with RF alone. The inclusion of PCA (Principal Component Analysis) reduces the feature space, leading to quicker computations while still retaining most of the data's variance.

The CNN, while more complex due to its multi-layered architecture involving convolutional layers, pooling, and batch normalization, manages to maintain a relatively reasonable running time. Despite being more computationally intensive than the SVM and Naive Bayes models, the CNN's running time is much shorter compared to the SVM with RF. It's important to note that the CNN was trained for 30 epochs in this instance. However, with further research and extended training to 100 epochs, the running time could be increased, potentially leading to improvements in model accuracy and performance.

Results



Model	Category	Precision	Recall	F1-score	Accuracy
Naive Bayes	Benign	0.48	0.77	0.59	0.58
	Malignant	0.53	0.4	0.46	
	Undetected	0.86	0.55	0.67	
SVM with RF	Benign	0.82	0.88	0.85	0.87
	Malignant	0.86	0.79	0.82	
	Undetected	0.93	0.93	0.93	
SVM with RF and PCA	Benign	0.82	0.88	0.85	0.87
	Malignant	0.85	0.8	0.82	
	Undetected	0.93	0.93	0.93	
CNN	Benign	0.78	0.94	0.85	0.88
	Malignant	0.93	0.73	0.82	
	Undetected	0.97	0.98	0.97	

Table 2: Result of each model

The results from the first model, Naive Bayes, yielded an accuracy of 58%, with precision, recall, and F1-scores of 0.48, 0.53, and 0.59 for the benign category, and 0.86, 0.55, and 0.67 for the undetected category. While the model shows some effectiveness in identifying undetected cases, its performance on benign and malignant categories is notably weaker.

The second model, SVM with Random Forest (RF), achieved a substantial improvement with an accuracy of 87%. Precision, recall, and F1-scores for the benign, malignant, and undetected categories were 0.82, 0.86, 0.93, and 0.88, 0.79, 0.93, respectively. The learning curves show a high training score that slightly decreases as more training examples are added, indicating the model's strong performance on the training data. The cross-validation score starts lower but improves with more examples, suggesting that the model benefits from additional data, reducing overfitting and enhancing generalization.

The third model, SVM with RF and PCA, yielded similar results to the second model, with an accuracy of 87%. The precision, recall, and F1-scores are slightly improved or consistent across categories compared to the SVM with RF alone. The inclusion of PCA helped maintain a similar level of performance meanwhile improving computational efficiency. The learning curves for this model demonstrate a consistent training score with a gradually increasing cross-validation score, highlighting that the model's ability to generalize improves with more data, similar to the previous model.

The fourth model, CNN, achieved the highest accuracy of 88%. Precision, recall, and F1-scores were 0.78, 0.93, and 0.97 for the benign, malignant, and undetected categories, respectively. The CNN's multi-layered architecture allowed it to capture more complex patterns in the data, leading to a better overall performance compared to the other models. The learning curves for CNN show the training accuracy steadily increasing, approaching 0.91, indicating effective learning from the training data. The validation accuracy fluctuates but generally trends upwards, reaching around 0.87, suggesting reasonable generalization despite some variability. The loss curves exhibit consistent training loss reduction, with validation loss showing spikes that indicate occasional overfitting, though the overall trend is downward, reflecting an improvement in performance with some epochs showing better generalization than others.

Ethical Considerations

Ethical considerations in machine learning for healthcare are important to guarantee that the use of these technologies benefits society without causing damage.
In the case of our dataset and models, two are the main concerns for fairness: privacy and bias.

Data privacy and patient confidentiality are fundamental. The images used in our dataset must be handled with care to ensure that patient identities are protected, so not to identify any patient. Additionally, all images must be used only with the patients' explicit consent. Any breach of patient confidentiality could destroy public trust in using machine learning models in healthcare.

The potential for algorithmic bias must be carefully considered. Since the majority of the pictures in our dataset are of people with lighter skin tones, the models may perform poorly on darker skin tones. This lack of diversity raises the possibility of misdiagnosis among underrepresented groups, resulting in biased predictions, and contributing to healthcare inequalities. Research has shown that many computer vision models are biased against minority groups, highlighting the need for tools to assess and quantify these biases (Buolamwini & Gebru, 2018). Data augmentation, adversarial debiasing, and contrastive learning can help mitigate biases by enhancing model fairness (Thong et al., 2023).

Limitations

In the course of our research, some limitations were identified that may impact the generalizability and robustness of our findings.

A notable constraint is the range of skin tones included in our sample, since the majority of the images used in our study were of individuals with white skin. This lack of diversity can lead to a model that performs well on lighter skin tones but may not generalize effectively to individuals with darker skin tones. The limited representation of diverse skin colors in the training data may result in biased predictions, which is a critical consideration in medical applications where equitable performance across different demographic groups is essential (Adamson & Smith, 2018).

An additional constraint belongs to the composition of the 'Undetected' category in our dataset. This category predominantly includes images of bruises, which does not encompass the full range of possible skin conditions that could be mistaken for melanoma. So, our model may not perform as well when faced with additional sorts of skin lesions or disorders not included in the training data due to the limited scope of 'Undetected' pictures.

Moreover, the computational resources available for this study posed constraints on the extent of hyperparameter tuning and model complexity that could be explored. Despite utilizing UCloud resources, frequent kernel restarts and computational limitations required us to sample the dataset for certain models, potentially impacting their performance and comparison.

Conclusions

Our results show that the CNN model achieved the highest accuracy rate of 88%, making it the best-performing model in this study. The SVM models, both with Random Forest and with the addition of PCA, also performed well, achieving an accuracy of 87%. The learning curves provide additional insights into the models' behaviour. For SVM models, the training score remains consistently high, indicating good performance on training data, while the cross-validation score improves with more training examples, suggesting better generalization.

Similarly, CNN models exhibit steady improvements in training and validation accuracy, although occasional overfitting is observed in the validation loss graph. Further analysis of the results reveals that each model performs exceptionally well in predicting "Undetected" data, with the CNN model achieving a precision of 97% for this category. Despite these promising results, there is room for further work to optimize the CNN model. By continuing to refine the CNN model, it is possible to achieve even higher accuracy and better generalization to unseen data, ultimately improving its utility in clinical settings.

In cancer diagnostics, the preponderance of false positives over false negatives or vice versa makes a great difference. False positives, i.e., where the model incorrectly identifies a benign tumour as malignant, may lead to unnecessary medical interventions or treatments for patients who are not sick. While this can result in additional stress and healthcare costs, it is generally considered less harmful than false negatives. On the other hand, those occur when the model incorrectly identifies a malignant melanoma as benign. This could delay the diagnosis, allowing the cancer to progress to a more advanced stage. Early detection can greatly increase the chances of survival of a skin tumour patient, making false negatives particularly concerning.

It's important to recognize that machine learning models can assist but never replace the expertise of trained healthcare professionals. Our suggestion is thus to train physicians to interpret model results and use them to make more informed decisions about patient care.

Therefore, our study emphasizes the importance of using machine learning as a supportive tool in healthcare, rather than a standalone diagnostic solution. By integrating machine learning capabilities with healthcare professionals' expertise, we can enhance diagnostic accuracy, improve patient outcomes, and ultimately, save lives.

These results highlight the potential of machine learning as a valuable tool in aiding medical professionals in early and accurate cancer detection.

References

Adamson, A. S., & Smith, A. (2018). Machine learning and health care disparities in dermatology. *JAMA Dermatology*, 154(11), 1247-1248. doi:10.1001/jamadermatol.2018.2348

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., A. Fadhel M., Al-Amidie M. & Farhan L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data*, 53. Retrieved https://doi.org/10.1186/s40537-021-00444-8 11 May 2024

Brynjolfsson, E., McAfee, A. (2017). The Business of Artificial Intelligence. *Harvard Business Review*.

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. *In Proceedings of the 1st Conference on Fairness, Accountability, and Transparency* (Vol. 81, pp. 77-91).

Burton, T., Fathieh, F., Nemati, N., Gillins, H.R., Shadforth, I.P., Ramchandani, S., Bridges, C.R. (2024). Development of a Non-Invasive Machine-Learned Point-of-Care Rule-Out Test for Coronary Artery Disease. *Diagnostics*. 14(7):719. https://doi.org/10.3390/diagnostics14070719

Committee on Diagnostic Error in Health Care; Board on Health Care Services; Institute of Medicine; The National Academies of Sciences, Engineering, and Medicine.(2015). Improving Diagnosis in Health Care. *National Academies Press*. Available from: https://www.ncbi.nlm.nih.gov/books/NBK338593/

Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. *O'Reilly Media*, Inc.

IBM. (2023). What is principal component analysis (PCA)? Principal Component Analysis. Retrieved: https://www.ibm.com/topics/principal-component-analysis 14 May 2023
Jolliffe, I. T., Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. *Royal Society*, 374(2065). doi: 10.1098/rsta.2015.0202.

Lapides, R., Saravi, B., Mueller, A., Wang-Evers, M., Maul, L.V., Németh, I., Navarini, A., Manstein, D., Roider, E. (2023). Possible Explanations for Rising Melanoma Rates Despite Increased Sunscreen Use over the Past Several Decades. *Cancers (Basel)*. 15(24):5868. doi: 10.3390/cancers15245868.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2020). Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(1), 6765-6816.

Morgan, K., (Medically Reviewed) Khatri, M. (2022). How many people die of cancer a year?. *WebMD*. Retrieved: https://www.webmd.com/cancer/how-many-cancer-deaths-per-year 11 May 2024

Murugan, A., Anu H Nair, S., Angelin Peace Preethi, A., Sanal Kumar, K. P. (2021). Diagnosis of skin cancer using machine learning techniques. *Microprocessors and Microsystems*. ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2020.103727.

Scikit-learn. (n.d.). GridSearchCV. In Scikit-learn: Machine Learning in Python. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Scikit-learn. (n.d.). SGDClassifier. In Scikit-learn: Machine Learning in Python. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Scikit-learn. (n.d.-a). Support vector machines. In Scikit-learn: Machine Learning in Python. Retrieved from https://scikit-learn.org/stable/modules/svm.html

Shivangi, J., Vandana, J., Nitin, P. (2015). Computer Aided Melanoma Skin Cancer Detection Using Image Processing. Procedia Computer Science, pages 735-740, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.209

Thong, W., Joniak, P., & Xiang, A. (2023). Beyond Skin Tone: A Multidimensional Measure of Apparent Skin Color. *Cornell University*. arXiv preprint arXiv:2309.05148v2.

Viknesh, C.K., Kumar, P.N., Seetharaman, R., Anitha, D. (2023). Detection and Classification of Melanoma Skin Cancer Using Image Processing Technique. *Diagnostics (Basel)*. 13(21):3313. doi: 10.3390/diagnostics13213313. PMID: 37958209; PMCID: PMC10649387.

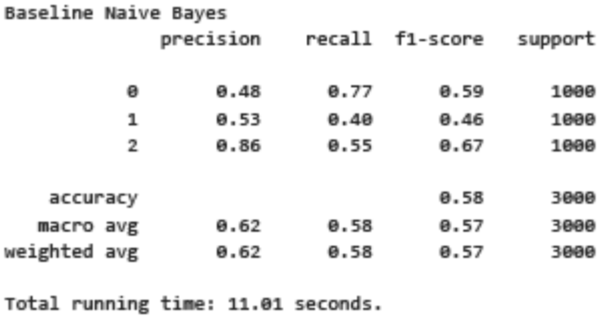
Lin, Z., He, J., Tang, X., & Tang, C.-K. (2007). Limits of learning-based superresolution algorithms. *In IEEE 11th International Conference on Computer Vision* (pp. 1-8). Rio de Janeiro, Brazil.

Zauner, C. (2010). Implementation and Benchmarking of Perceptual Image Hash Functions. *Upper Austria University of Applied Sciences, Hagenberg Campus*. Retrieved from https://wavelab.at/papers/Zauner2010b.pdf

Zhang, B., Shi, H., Wang, H. (2023). Machine Learning and AI in Cancer Prognosis, Prediction, and Treatment Selection: A Critical Approach. *J Multidiscip Healthc*. 16:1779-1791. doi: 10.2147/JMDH.S410301. PMID: 37398894; PMCID: PMC10312208.

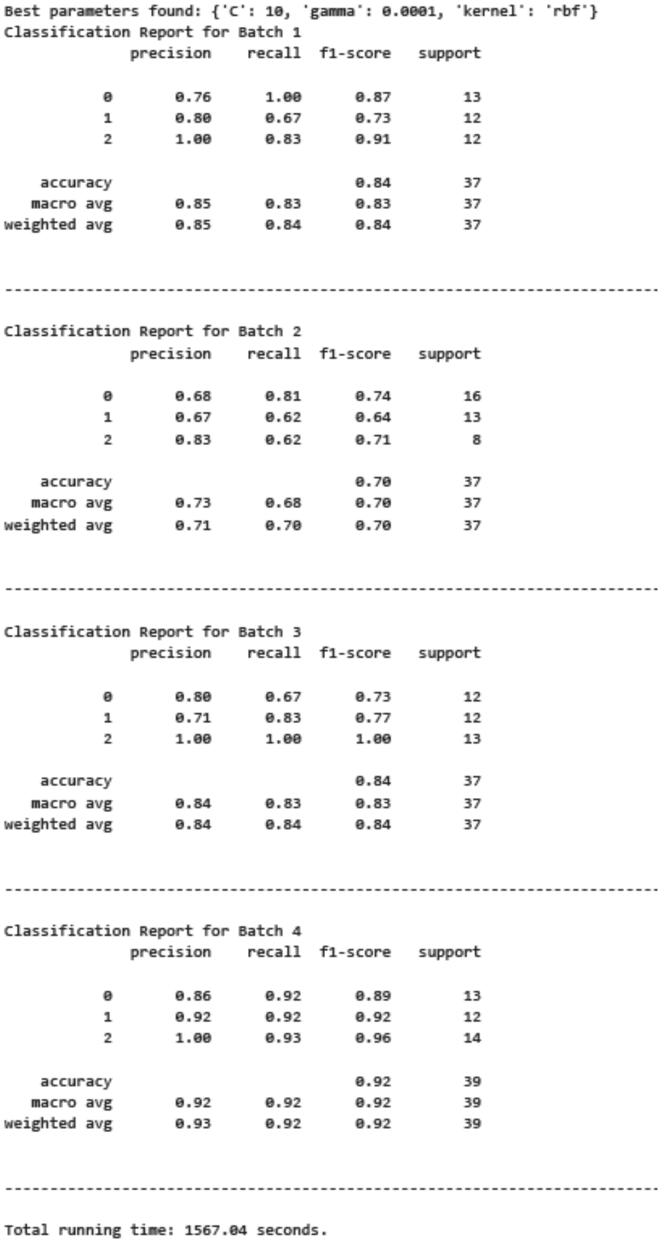
Appendix

Naïve Bayes Results:



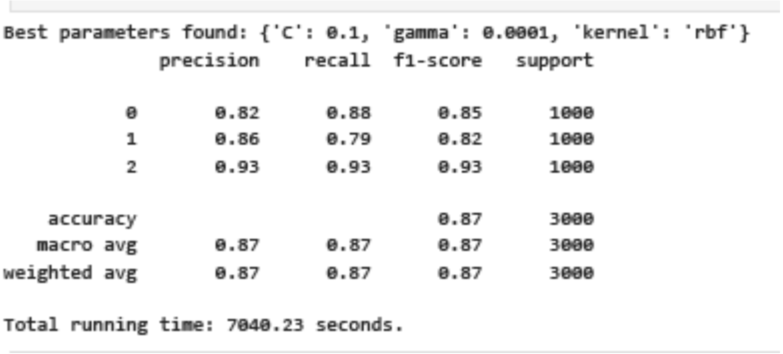
```
Baseline Naive Bayes
              precision    recall  f1-score   support

           0       0.48      0.77      0.59      1000
           1       0.53      0.40      0.46      1000
           2       0.86      0.55      0.67      1000

    accuracy                           0.58      3000
   macro avg       0.62      0.58      0.57      3000
weighted avg       0.62      0.58      0.57      3000

Total running time: 11.01 seconds.
```

SVM Results:



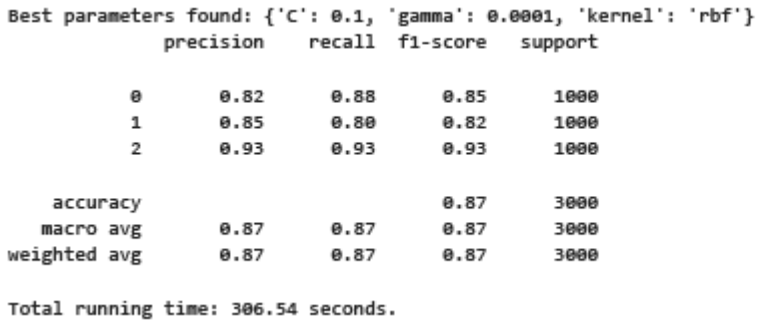
```
Best parameters found: {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
Classification Report for Batch 1
              precision    recall  f1-score   support

           0       0.76      1.00      0.87        13
           1       0.80      0.67      0.73        12
           2       1.00      0.83      0.91        12

    accuracy                           0.84        37
   macro avg       0.85      0.83      0.83        37
weighted avg       0.85      0.84      0.84        37


--------------------------------------------------------------------------

Classification Report for Batch 2
              precision    recall  f1-score   support

           0       0.68      0.81      0.74        16
           1       0.67      0.62      0.64        13
           2       0.83      0.62      0.71         8

    accuracy                           0.70        37
   macro avg       0.73      0.68      0.70        37
weighted avg       0.71      0.70      0.70        37


--------------------------------------------------------------------------

Classification Report for Batch 3
              precision    recall  f1-score   support

           0       0.80      0.67      0.73        12
           1       0.71      0.83      0.77        12
           2       1.00      1.00      1.00        13

    accuracy                           0.84        37
   macro avg       0.84      0.83      0.83        37
weighted avg       0.84      0.84      0.84        37


--------------------------------------------------------------------------

Classification Report for Batch 4
              precision    recall  f1-score   support

           0       0.86      0.92      0.89        13
           1       0.92      0.92      0.92        12
           2       1.00      0.93      0.96        14

    accuracy                           0.92        39
   macro avg       0.92      0.92      0.92        39
weighted avg       0.93      0.92      0.92        39


--------------------------------------------------------------------------

Total running time: 1567.04 seconds.
```

SVM with Random Forest:



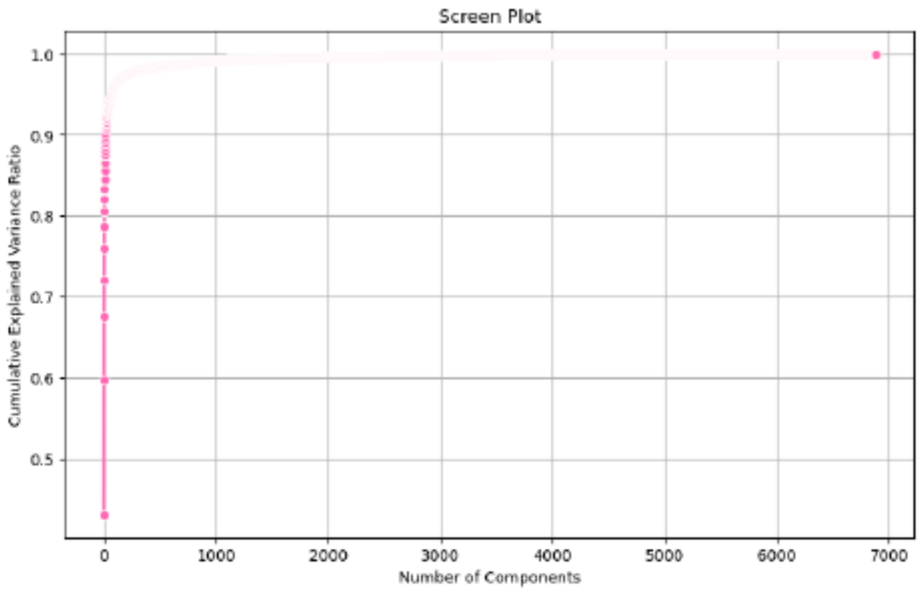
```
Best parameters found: {'C': 0.1, 'gamma': 0.0001, 'kernel': 'rbf'}
              precision    recall  f1-score   support

           0       0.82      0.88      0.85      1000
           1       0.86      0.79      0.82      1000
           2       0.93      0.93      0.93      1000

    accuracy                           0.87      3000
   macro avg       0.87      0.87      0.87      3000
weighted avg       0.87      0.87      0.87      3000

Total running time: 7040.23 seconds.
```

SVM with Random Forest and PCA:

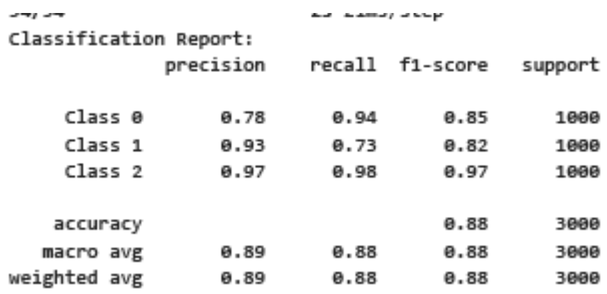


```
Best parameters found: {'C': 0.1, 'gamma': 0.0001, 'kernel': 'rbf'}
              precision    recall  f1-score   support

           0       0.82      0.88      0.85      1000
           1       0.85      0.80      0.82      1000
           2       0.93      0.93      0.93      1000

    accuracy                           0.87      3000
   macro avg       0.87      0.87      0.87      3000
weighted avg       0.87      0.87      0.87      3000

Total running time: 306.54 seconds.
```

PCA:

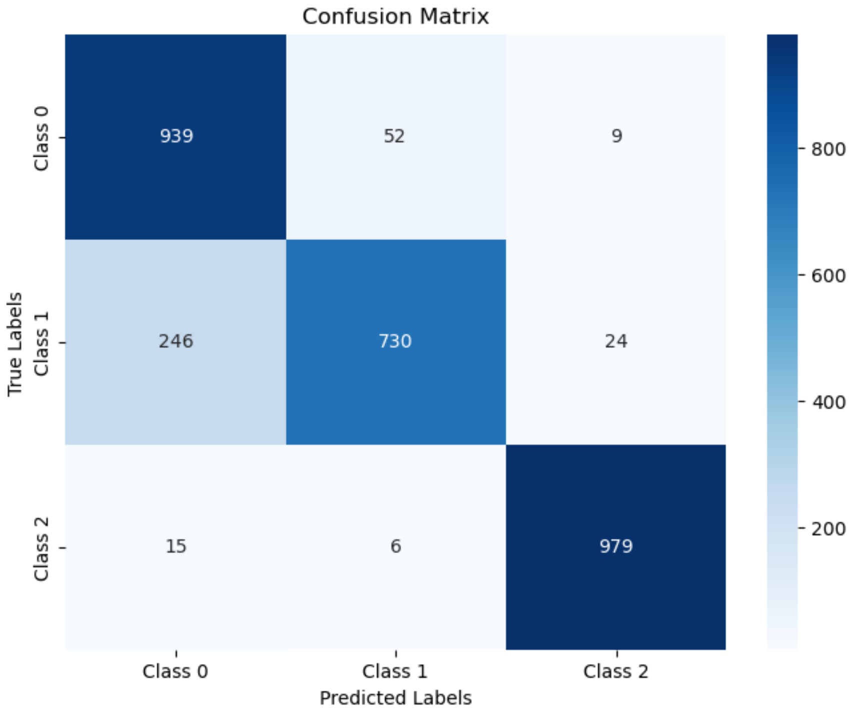


CNN results:

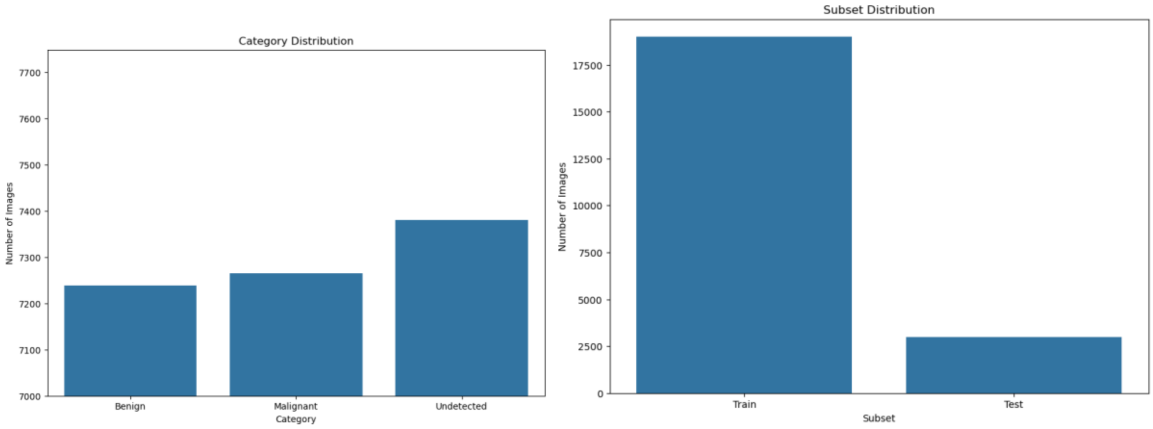


```
Classification Report:
              precision    recall  f1-score   support

     Class 0       0.78      0.94      0.85      1000
     Class 1       0.93      0.73      0.82      1000
     Class 2       0.97      0.98      0.97      1000

    accuracy                           0.88      3000
   macro avg       0.89      0.88      0.88      3000
weighted avg       0.89      0.88      0.88      3000
```

CNN Confusion Matrix:



Category and Subset Distribution after filtering


Category Distribution
Number of Images
Benign
Malignant
Undetected
Category
Subset Distribution
Number of Images
Train
Test
Subset