Intelligent Skin Cancer Screening (ISCS) - Final Project

Class: Deep Learning

Gabriela Caetano de Jesus - GitHub
Aleksandra Ristic - GitHub
Joana Villanova - GitHub

Table of contents:

Introduction

Skin cancer, the most common form of cancer globally, arises from abnormal skin cell growth. It includes basal cell carcinoma, squamous cell carcinoma, and melanoma. Early detection through self-examination and professional screenings is crucial for effective treatment.

If skin cancer is detected early, it can be highly treatable, but late diagnosis can lead to more advanced stages of the disease and poorer outcomes. By developing accurate and efficient skin cancer detection systems, we can facilitate early diagnosis and prompt intervention.

In response to the challenges associated with skin cancer diagnosis, our project aims to develop and evaluate the effectiveness of convolutional neural network (CNN) models for classifying suspicious skin changes. We seek to create a reliable and efficient method for early skin cancer detection by leveraging deep learning techniques.

The effectiveness of CNN models in accurately distinguishing between different types of skin lesions, including benign and malignant. We seek to provide a reliable and efficient method for early skin cancer detection.

Dataset

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T

The dataset consists of 10015 dermatoscopic images, that includes a representative collection of all important diagnostic categories in the realm of pigmented lesions:

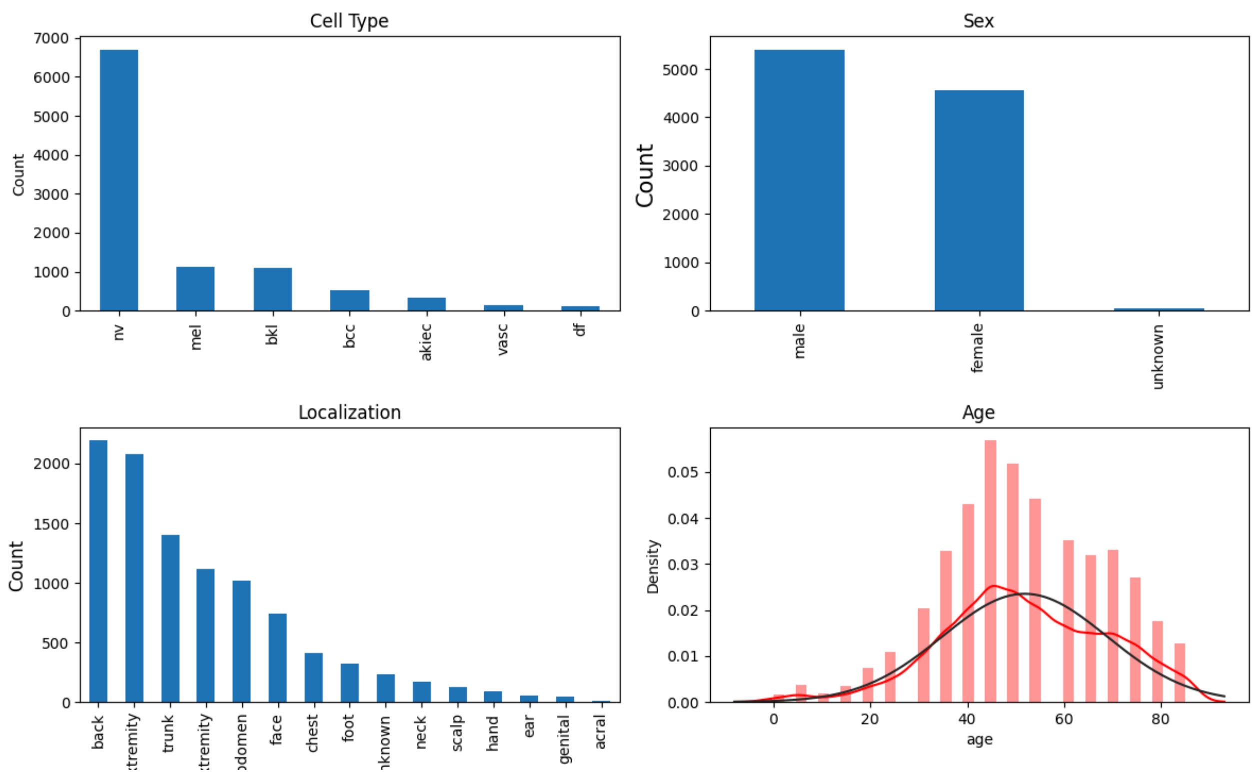
- Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec)
- Basal cell carcinoma (bcc)
- Benign keratosis-like lesions (bkl)
- Dermatofibroma (df)
- Melanoma (mel)
- Melanocytic nevi (nv)
- Vascular lesions (vasc)

Out of these 7 types of cancer, the BKL is the only benign one. We read the CSV file and matched it with our image data to understand our dataset. The names of the images are all listed in the CSV file and have been categorized into the specific groups we mentioned above. One of the results that we got is the number of images per group of folders, and this helped us to see what groups fit the best for our further analysis and usage:

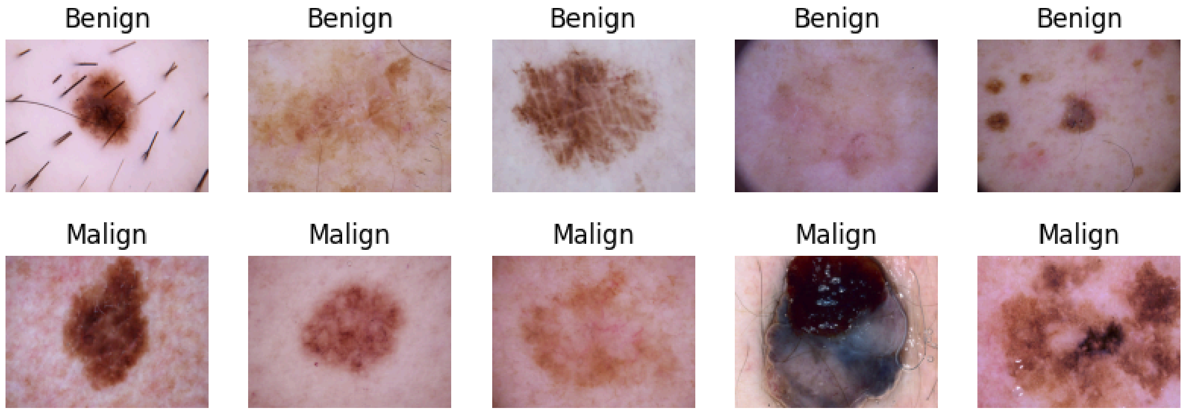
```
Counts of each dx:
dx
nv       6705
mel      1113
bkl      1099
bcc       514
akiec     327
vasc      142
df        115
Name: count, dtype: int64
```

Looking into this result, we decided to work with the two largest groups: “bkl” and “mel” and normalize the folder size to 1000 images that we are going to use for our testing and training. It is essential to ensure that our dataset is balanced, meaning that each cancer type has a similar number of samples. So, we created a Zip file containing only the images for bening (‘bkl’) and malign (‘mel’) so that we could run our models.

Our CSV file contains a lot of details about our dataset, as we showed in the graphs below. We plot the graphs to show the information and better understand the data that we are dealing with. The top left image (1) shows the number of images for each type of cancer. Top right image (2) displays the division of hows many cancer cases belog to which gender - male, female, unknown. On the bottom left (3) we can see the body location where the cancer is mostly located. The last, bottom right image (4) shows the density of cancer by age.



In the next step, we wanted to visually compare images that we are planning to use - ‘bkl’ and ‘mel’ and it is obvious how different these two types of cancer are. The figure below shows 5 random images from the bkl (Bening) and mel (malign) datasets.



Methods - Models

- ## Residual Neural Network (ResNet50)

ResNet50 is a deep convolutional neural network architecture which is known for its effectiveness in training very deep neural networks by addressing the vanishing gradient problem through the use of residual connections.

Convolutional Layers: ResNet50 consists of 50 layers, primarily composed of convolutional layers, which are the building blocks for feature extraction in convolutional neural networks (CNNs). These layers learn to detect features at different levels of abstraction from the input data.

Residual Blocks: The key innovation of ResNet50 lies in its use of residual blocks. In a typical neural network, each layer is responsible for learning a transformation of the input data. However, as the network gets deeper, it becomes harder to optimize due to the vanishing gradient problem. Residual blocks address this issue by introducing *skip connections*, which allow the network to learn residual mappings instead of directly learning the desired underlying mapping. This helps in mitigating the degradation problem, allowing for the training of very deep networks.

Global Average Pooling and Fully Connected Layers: Towards the end of the network, global average pooling is applied to reduce the spatial dimensions of the feature maps to a vector. This vector is then fed into fully connected layers, followed by a softmax layer for classification.

Using ResNet50 for skin abnormality classification involves collecting a diverse dataset of skin abnormality images labeled with malignant or benign cancer, preprocessing the images by resizing and normalizing them, and then leveraging transfer learning by fine-tuning the pre-trained ResNet50 model on the dataset. This entails freezing the initial layers and retraining the later ones. Subsequently, the model is trained on the dataset to learn to extract relevant features and classify images accurately. After evaluation on a separate dataset to measure performance metrics, such as accuracy and loss, the model can be deployed in real-world applications for automatic classification of skin abnormalities based on their likelihood of being malignant or benign cancer.

- Densely Connected Convolutional Networks (DenseNet121)

Is a flexible architecture applicable to a variety of computer vision applications including picture classification, object identification, and semantic segmentation.DenseNet introduces a novel connectivity pattern between layers, where each layer is connected to every other layer in a feed-forward fashion. The architecture is characterized by dense connectivity between layers, where feature maps from all preceding layers are concatenated and passed as input to subsequent layers.

The densenet121 function is a convenience function that creates a DenseNet-121 network. It takes two arguments: num_class and pre-trained. The num_class argument specifies the number of output classes, and the pre-trained argument specifies whether to use a pre-trained network version. If pre-trained is set to None, the function returns a new instance of the DenseNet-121 network. If pre-trained is set to a path to a pre-trained model file, the function loads the pre-trained model weights and returns the model.

Dense Blocks and Transition Layers: A dense block consists of multiple bottleneck layers connected to each other in a feedforward fashion. Each bottleneck layer takes the output of the previous layer as input and produces a fixed number of output feature maps. A transition layer consists of a batch normalization layer, a 1x1 convolutional layer, and a 2x2 average pooling layer.

Offers a compelling combination of improved accuracy, parameter efficiency, gradient flow, and interpretability, making it a popular choice for various computer vision tasks where deep learning models are deployed.

- GoogLeNet - InceptionV3

Inception (GoogLeNet) is a groundbreaking deep convolutional neural network introduced by Google researchers in 2014. It is primarily used for large-scale visual recognition tasks like ImageNet. Its architecture revolutionized deep learning with its efficient design and pioneering concepts.

Inception (GoogLeNet) is a testament to innovation in deep learning architecture, showcasing the power of parallel convolutional pathways and efficient feature extraction.

At the heart of Inception lies its distinctive architecture, characterized by inception modules. These modules are pivotal components employing parallel convolutional pathways to extract features across various scales concurrently. This approach enables the network to capture intricate details while maintaining computational efficiency.

Below are the key components of Inception:

1. *Inception Modules:* These are the building blocks of the Inception network, integrating parallel convolutional layers to capture features at multiple resolutions. This design choice facilitates the network's ability to extract diverse and rich feature representations.

2. *1x1 Convolutions:* Utilized within the inception modules, 1x1 convolutions play a crucial role in dimensionality reduction and introducing non-linearity. This process enhances computational efficiency by reducing the computational burden while preserving essential features.

3. *Pooling Operations:* Inception incorporates max-pooling operations for downsampling, effectively reducing spatial dimensions while retaining significant features. This step is vital for managing computational complexity and improving the network's scalability.

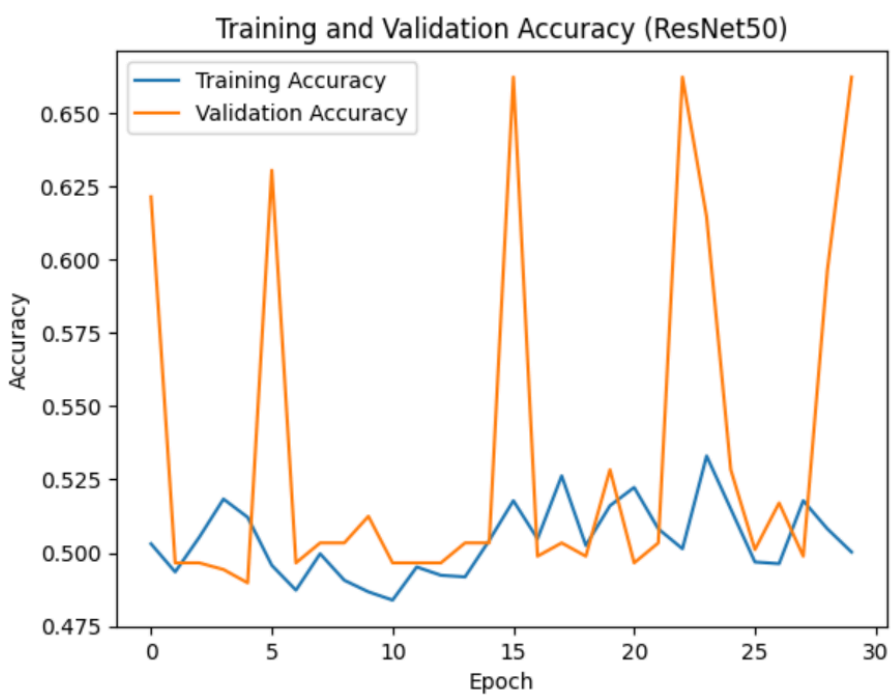
4. *Auxiliary Classifiers:* Inception integrates auxiliary classifiers at intermediate layers to combat the vanishing gradient problem and aid in regularization. These auxiliary classifiers provide additional supervision during training, thereby promoting better gradient flow and enhancing the network's generalization ability.

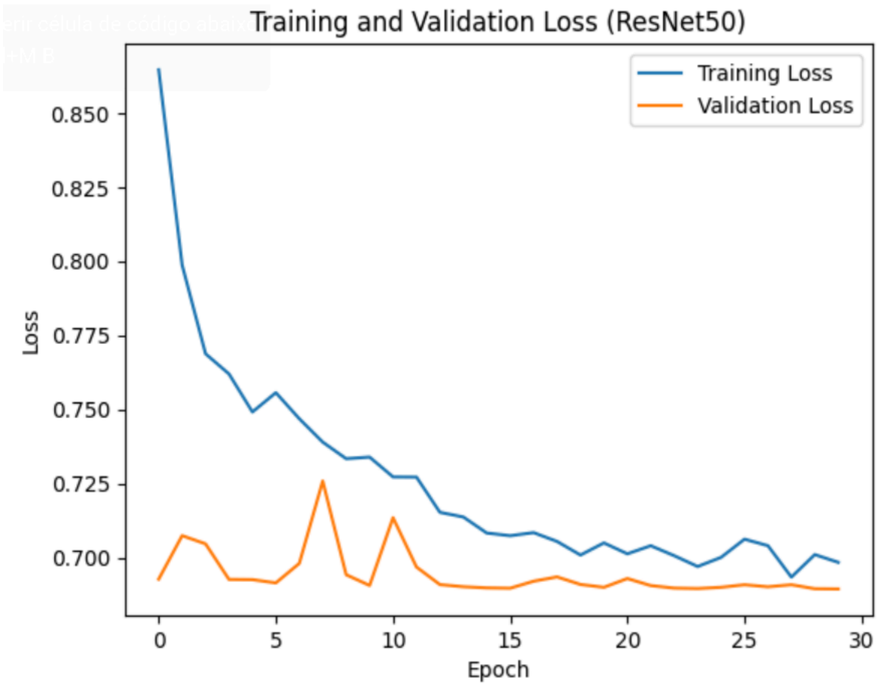
5. *Global Average Pooling:* In place of traditional fully connected layers, Inception adopts global average pooling, which computes the average of each feature map. This approach significantly reduces overfitting and parameter count while preserving spatial information, leading to more robust and efficient models.

Model Results

- ResNet50

Looking at the results of the ResNet50 model below, it's evident that there is fluctuation in both training and validation accuracy and loss throughout the epochs. Initially, the model achieves moderate accuracy on both training and validation sets, but as training progresses, validation accuracy tends to stagnate or even decrease slightly. This suggests potential overfitting, where the model is memorizing the training data rather than generalizing well to unseen data. However, towards the later epochs, both training and validation accuracy start to improve again, indicating that the model might be learning more robust features. Overall, while the model achieves relatively high accuracy on the validation set towards the end of training, further investigation into regularization techniques or adjustments to the model architecture may be warranted to address overfitting and improve generalization performance.






```
Validation Loss: 0.6893512010574341
Validation Accuracy: 0.6621315479278564
```

- DenseNet121

We use the DenseNet model for a deep learning task, such as image classification, validation loss, and accuracy metrics, which provide insights into how well the model performs on unseen data (validation set). Here's a brief interpretation of my results:

In this case, we use some techniques to improve:

Dropout: Randomly drop a proportion of neurons during training to prevent co-adaptation of feature detectors and reduce overfitting.

Early Stopping: Monitor validation loss during training and stop training when validation loss stops improving or starts to degrade to prevent overfitting.

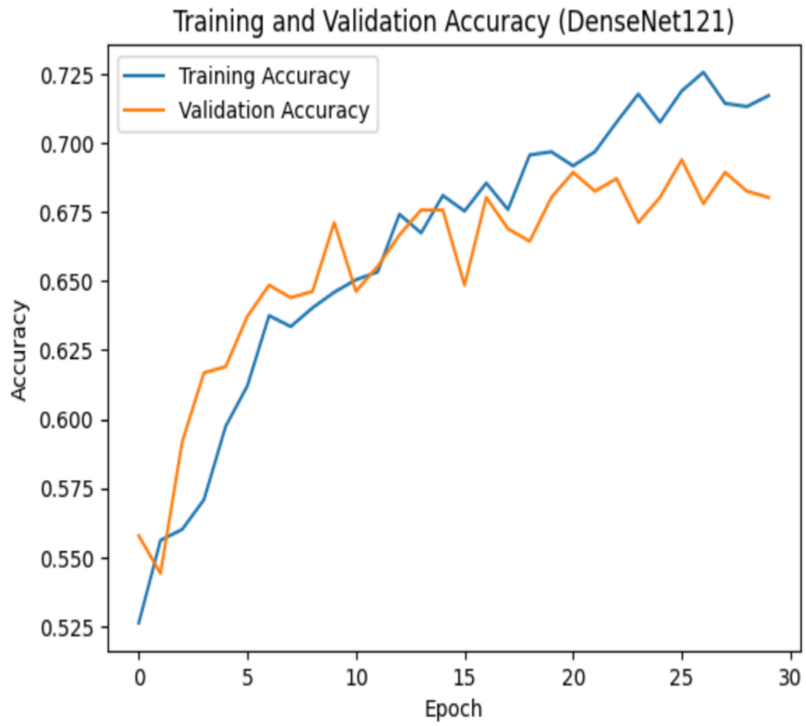
Data Augmentation: Apply transformations to the training data, such as rotation, scaling, flipping, and cropping, to increase the diversity of the training dataset and improve generalization.

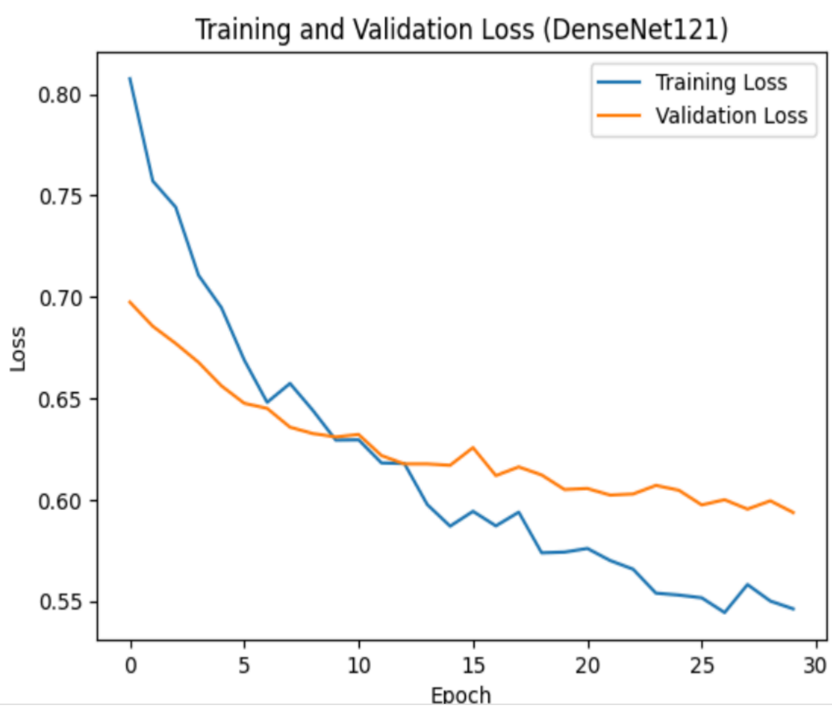
We assume the validation loss measures how well the model's predictions match the true labels on the validation set. A lower validation loss indicates that the model's predictions are closer to the true labels, while a higher loss suggests the opposite. The graphic below shows the results of training accuracy and validation. The training on the blue line shows very good accuracy, a little higher than validation accuracy, which means the training in this model is working. Still, the performance on validation accuracy needs selecting the best model or tuning hyperparameters to improve performance.

The validation accuracy measures the proportion of correctly predicted labels from all validation samples. A higher validation accuracy indicates better model performance on the validation set. With a validation accuracy of 0.68, the model correctly predicts around 68% of the labels in the validation set. This means it performs reasonably well, but there is still room for improvement, as the accuracy could be higher.

The validation loss of 0.59 indicates that, on average, the model's predictions are somewhat close to the true labels. The training loss was close to 0.55, which means models are given good results. However, there is still room for improvement, as the loss could be lower.

In summary, training accuracy provides insight into how well the model is learning from the training data. Still, it should be interpreted alongside validation accuracy and loss to assess the model's generalization ability and avoid overfitting.


Training and Validation Accuracy (DenseNet121)
Training Accuracy
Validation Accuracy
Accuracy
Epoch
0.725
0.700
0.675
0.650
0.625
0.600
0.575
0.550
0.525
0
5
10
15
20
25
30

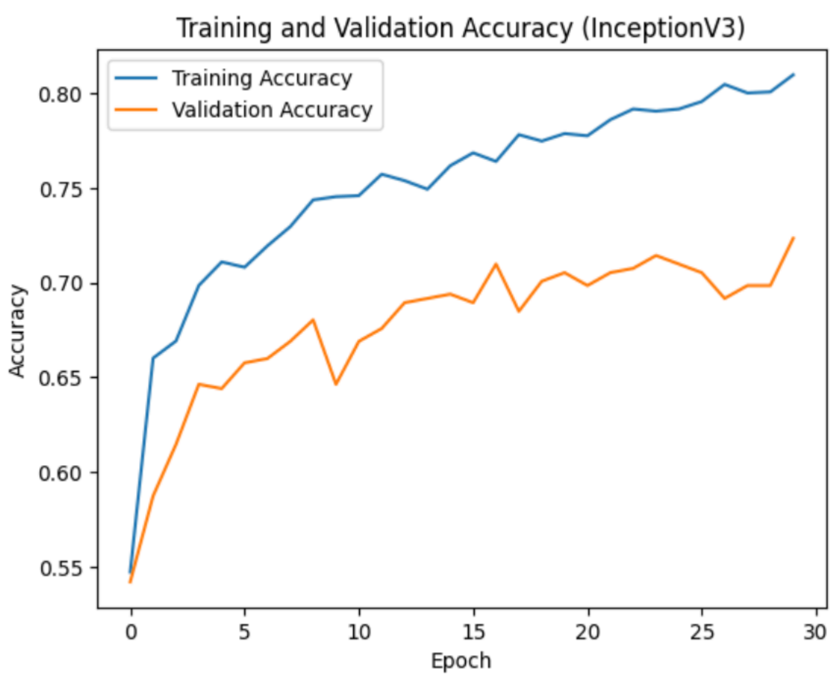

Training and Validation Loss (DenseNet121)
Training Loss
Validation Loss
Loss
Epoch
0.80
0.75
0.70
0.65
0.60
0.55
0
5
10
15
20
25
30

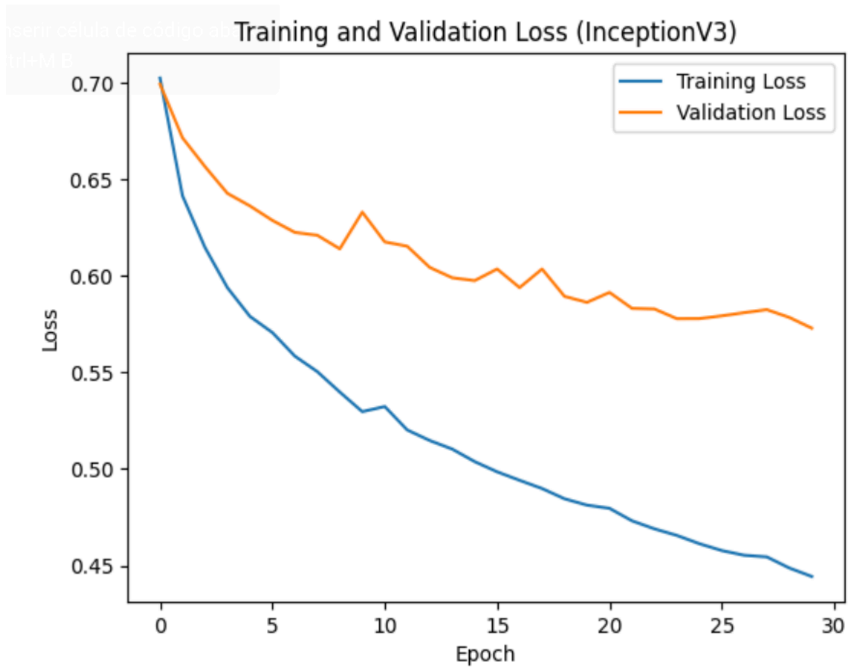
```
Validation Loss: 0.5936651825904846
Validation Accuracy: 0.680272102355957
```

- InceptionV3

We achieved promising results after training and testing our skin cancer classification model using the InceptionV3 architecture. The model exhibited an overall accuracy of approximately 78.84% on the validation dataset, demonstrating its effectiveness in distinguishing between benign and malignant skin lesions.

Throughout the 30 epochs of training, we observed a steady improvement in accuracy, with the model achieving a peak validation accuracy of 72.34% at the final epoch. These results indicate the model's capability to generalize well to unseen data and its potential utility in clinical settings for assisting dermatologists in early skin cancer detection, which were our objectives when we started this project.





- Model Comparison

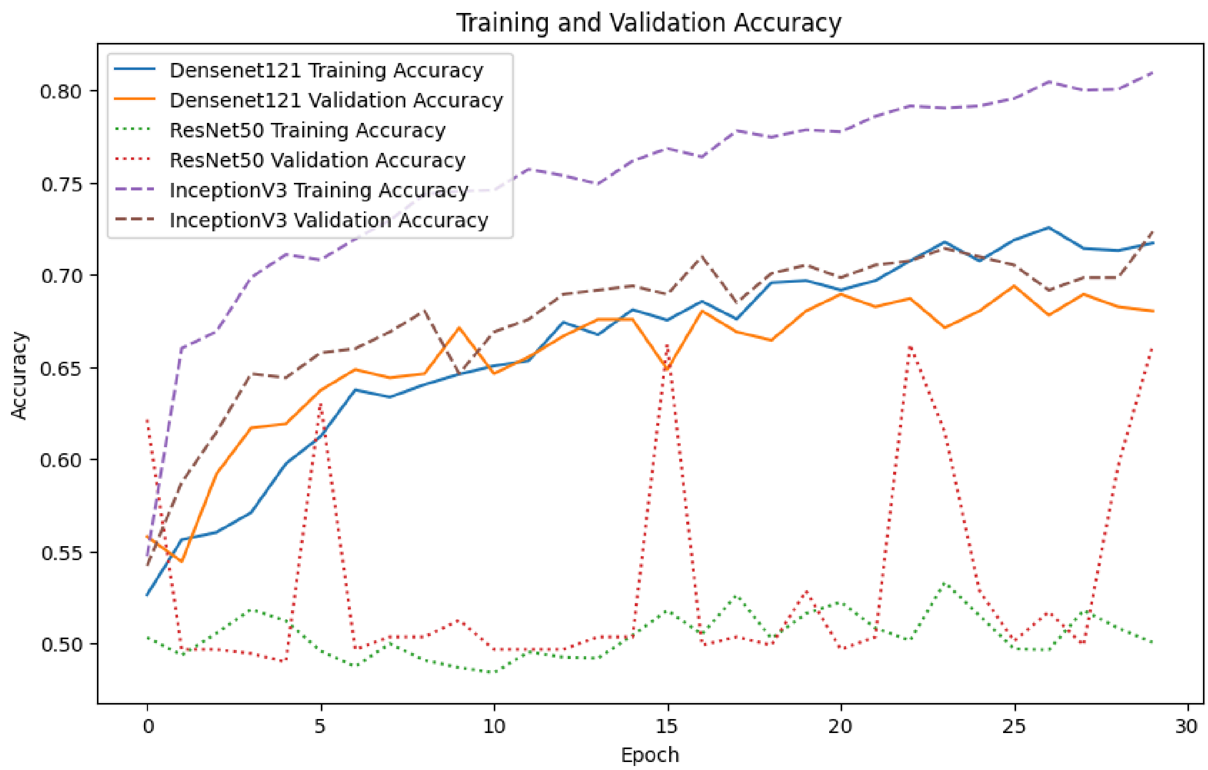
These are the results for the models mentioned above:

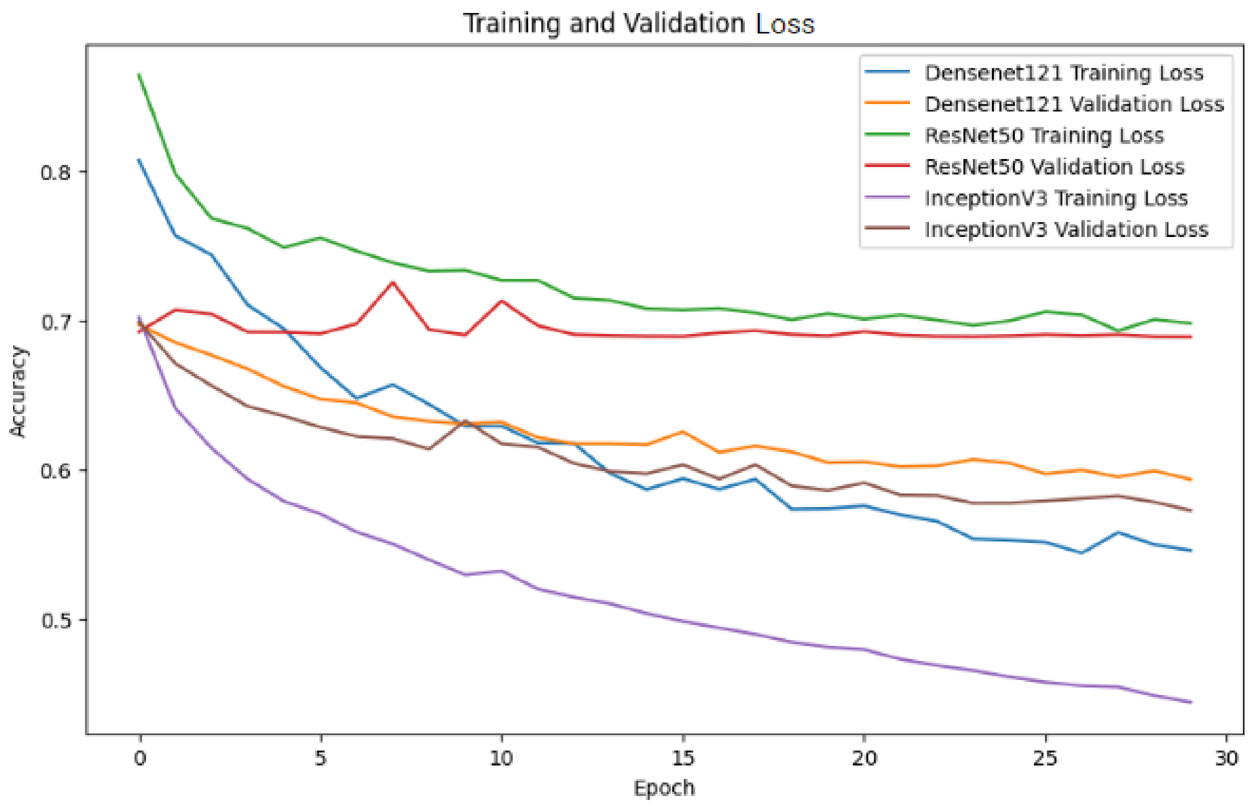
InceptionV3:
Validation Loss: 0.5729
Validation Accuracy: 0.7234

ResNet50:
Validation Loss: 0.6894
Validation Accuracy: 0.6621

DenseNet121:
Validation Loss: 0.5937
Validation Accuracy: 0.6803

InceptionV3 outperforms ResNet50 and DenseNet121 regarding both validation accuracy and loss for skin image recognition. With a validation accuracy of approximately 72.34% and a validation loss of about 0.5729, InceptionV3 demonstrates superior performance compared to ResNet50 and DenseNet121. This suggests that InceptionV3's architecture, which utilizes a more complex and efficient deep learning network, is better suited for capturing the intricate features present in skin images, leading to more accurate classification results. Additionally, the relatively lower validation loss indicates that InceptionV3 achieves better generalization on unseen data, further solidifying its effectiveness for skin image recognition tasks. Therefore, based on these findings, InceptionV3 emerges as the preferred model for skin image recognition among the three models evaluated. However, it's worth noting that model selection should also consider other factors such as computational resources, deployment requirements, and specific characteristics of the dataset. Graphs below show the comparison of both training and validation accuracy as well as training and validation between these three models:





Training and Validation Loss
Densenet121 Training Loss
Densenet121 Validation Loss
ResNet50 Training Loss
ResNet50 Validation Loss
InceptionV3 Training Loss
InceptionV3 Validation Loss
Accuracy
0.8
0.7
0.6
0.5
0
5
10
15
20
25
30
Epoch

Conclusion

In this project, we explored the effectiveness of three popular deep learning models, ResNet50, InceptionV3, and DenseNet121, for skin image recognition. The objective was to classify skin images into malignant or benign cancer categories, aiding in early detection and diagnosis. Each model underwent training and evaluation using a dataset of skin images, with performance metrics including validation accuracy and loss tracked across multiple epochs.

Upon analysis of the results, it was found that InceptionV3 emerged as the top-performing model for skin image recognition. Its architecture, which incorporates sophisticated convolutional neural networks, proved effective in capturing intricate features present in skin images, leading to more accurate classification results.

While InceptionV3 showed promising performance, there are opportunities for further improvement. Future adjustments could involve fine-tuning the model hyperparameters, such as learning rate, batch size, and optimizer choice, to enhance accuracy and convergence speed potentially. Additionally, data augmentation techniques could be employed to increase the diversity and quantity of training data, thereby improving the model's ability to generalize to unseen skin images.

In conclusion, this project underscores the importance of selecting appropriate deep-learning models for specific image recognition tasks. While InceptionV3 emerged as the best-performing model in this context, ongoing research and experimentation are essential to continually refine and optimize skin image recognition systems, ultimately contributing to advancements in medical diagnostics and patient care.

Resources

- https://www.kaggle.com/code/raniaioan/starter-skin-cancer-mnist-ham10000-6a5a3b01-0/notebook

- https://www.kaggle.com/code/fanconic/cnn-for-skin-cancer-detection

- https://www.kaggle.com/code/maheshmani13/hmnist-28-28-rgb-cnn-skin-cancer-detection/notebook

- https://github.com/ashishpatel26/Skin-Lesions-Detection-Deep-learning/tree/main/Notebooks