UCCD3074 Deep Learning for Data Science
Group Assignment
Deep Learning-Based Classification of Skin Diseases and Cancer

Research-based ☐ Application-based ■

Name	Loh Chia Heung (Leader)	Tan Yi Xin	Bester Loo Man Ting	Cornelius Wong Qin Jun
Programme	CS	CS	CS	CS
ID	2301684	2101990	2207066	2104603
Contribution	1/4	1/4	1/4	1/4

1. INTRODUCTION

The accurate classification of pigmented skin lesions from dermatoscopic images is a critical task in dermatology for the early detection of skin cancer, particularly melanoma. The diagnostic process, which relies on this visual analysis, is often subjective and dependent on the extensive experience of a dermatologist. This project aims to develop and evaluate an automated system for classifying skin lesions into seven distinct categories using deep learning, with a primary focus on achieving high diagnostic accuracy for malignant types. Melanoma is one of the most aggressive forms of skin cancer, and its prognosis is strongly tied to early detection. An automated diagnostic support tool can serve as a valuable "second opinion" for clinicians, reducing the rate of misdiagnosis and potentially decreasing the time to treatment.

This project utilizes the HAM10000 dataset [1], which presents a significant real-world challenge: severe class imbalance. Over 60% of the images belong to the benign "Melanocytic nevi" class, while critical classes like "Melanoma" are severely underrepresented. The scope of this project is to implement, train, and rigorously compare four modern deep learning architectures to address this problem: **EfficientNet-B0**, **DenseNet-121**, **Inception V3**, and the **Swin Transformer**. The core objectives are to build a complete machine learning pipeline that mitigates class imbalance, conduct a comprehensive evaluation of each model, analyze their comparative performance, and save the final trained models in a production-ready format suitable for an application.

2. RELATED WORK / LITERATURE REVIEW

The application of deep learning in dermatoscopic image analysis has become a prominent area of research, with initiation breakthroughs with Convolutional Neural Networks (CNNs) could achieve performance on par with board-certified dermatologists [2].

InceptionV3 is a deep convolutional neural network (CNN) architecture developed as its primary design goal was to achieve high accuracy in image classification while remaining computationally efficient. This allows it to be used in environments where computational resources may be limited. While larger models offered higher accuracy, InceptionV3 was designed to solve two key challenges they created. First, their high computational cost, caused by a massive number of parameters, made them slow to train and deploy resource-limited devices. Second, these large networks were prone to overfitting, becoming overly confident in their predictions and failing to generalize to new, unseen images [3].

To improve computational efficiency, InceptionV3 applied method called Spatial factorizing to create Asymmetric convolutions. The main idea is to be decomposing an large n×n convolution into a sequence of a 1×n convolution followed by an n×1 convolution. For example, replacing a 3×3 convolution with a 3×1 convolution followed by a 1×3 convolution. This can be 33% cheaper for the same number of output filters. The number of calculations required

reduces without sacrificing its ability to learn complex spatial features s. This method performs well in medium grid sizes which feature map range 12-20 but not well in early layer [3]. InceptionV3 also implements two main strategies to combat overfitting and improve the model's ability to generalize. It uses Auxiliary Classifiers, which are special side branches attached to intermediate layers. These classifiers predict the image class, and their loss is added to the main loss function, acting as a regularize, especially near the end of training, to help reduce overfitting of the main classifier [3]. Additionally, it uses Label Smoothing Regularization (LSR), a mechanism that regularizes the classifier by smoothing the target distribution. For a ground-truth label y, the 100% probability is replaced by spreading a little bit of probability across all other labels. This encourages the model to be less confident and reduces overfitting [3].

A weakness of InceptionV3 is its input image size requirement. It was designed to work with an input image size of 299x299 pixels, which is larger than the more common 224x224 input used by many other models. Consequently, images often need to be resized during preprocessing, which can add computational overhead [4].

DenseNet, originally proposed by [7], introduced the “dense block” concept on CNN architecture allowing feature reuse through the concatenation of feature maps from preceding layers. The author proposed that the design mitigates the vanishing-gradient problem, encourages feature propagation, and reduces the number of parameters in contrast with traditional architectures like Resnet. With DenseNet-121 as their baseline, variants were introduced based on the model’s number of convolutional layers in its “dense blocks”.

In [5], pretrained DenseNet121 framework was adapted, and U-Net was combined for the image analysis task, demonstrating segmentation performance on ISIC-2018 and classification on HAM10000 with particularly strong results in differentiating between cancerous and non-cancerous lesions—79.49 % accuracy for cancerous and 93.11 % for non-cancerous categories. In similar research by [6], the authors applied a two-phase learning approach comprising transfer learning and fine-tuning on various models including DenseNet121. It was found that although the model showed slower training convergence, it ultimately generalized better in the fine-tuning phase, achieving a test accuracy of 96.95%.

As deep learning models grew larger for marginal gains, the focus shifted towards optimizing the balance between performance and computational resources. **EfficientNet**, introduced by Tan and Le [8], represents a landmark in this effort. It moves beyond brute-force scaling by introducing "compound scaling," a principled method that uniformly scales network depth, width, and resolution. This core insight allows for the creation of models that achieve state-of-the-art accuracy with significantly fewer parameters and floating-point operations (FLOPs) than their predecessors.

The efficacy of this design has been repeatedly validated in dermatological applications. For instance, a comparative analysis by Abayomi-Alli et al. [9] confirmed that EfficientNet provides an excellent balance of high accuracy and low computational overhead for skin lesion classification. Further demonstrating its utility, Kassem et al. [10] successfully employed EfficientNet as a feature extractor on the related ISIC dataset, highlighting its ability to produce powerful, discriminative features from dermatoscopic images. Therefore, its inclusion in our study is motivated by its exceptional efficiency-to-accuracy ratio, making it a highly compelling candidate for practical, real-world clinical applications where diagnostic tools may need to run on hardware with limited computational power.

In addition, one of the models used is the **Swim Transformer**, introduced by Liu et.al in 2021 [11]. Different from the original Vision Transformer (ViT), which applies global self-attention across the entire image, Swin divided the images into local windows and applies attention

within each window. The windows are shifted across layers to allow cross-region connections, making the model more efficient while still capturing both local details and global context [11]. Swin also adopts a hierarchical structure like CNNs, enabling it to process features at multiple scales. This property is particularly useful for medical images, such as skin lesion photos, where lesions vary in size, shape, and texture.

Recent works have shown the potential of Swin Transformers in skin cancer detection. Pacal et al. [12] enhanced a Swin-based model and achieved 89.36% accuracy on an eight-class skin cancer dataset, outperforming CNN and ViT models. Another study emphasized that the shifted window mechanism is effective in handling both fine lesion structures and global image context, making Swin suitable for dermatoscopic images [13]. These findings indicate that Swin Transformer is a strong candidate for our task of classifying the seven skin lesion classes in the HAM10000 dataset [1].

3. SYSTEM DESIGN

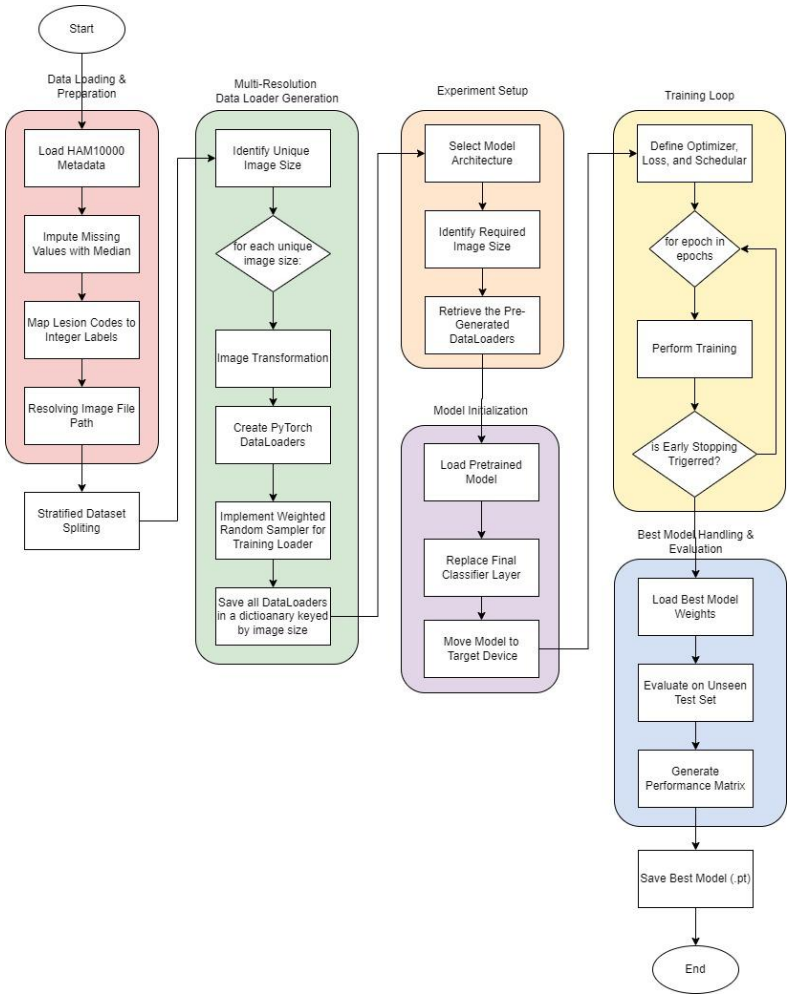

Figure 3.1: Flowchart of the Project Pipeline

3.1 Data Preparation and Loading

The initial stage of this project begins with preparing and loading HAM10000 metadata. This step analyzes and explores the data while iteratively handling missing value. The description of the dataset conveys the different skin lesions classes in the dataset. The distribution and characteristics of these lesions are first understood for identifying issues such as missing data. The issue stated is then resolved through median imputation, to prevent outliers while ensuring the completeness of the dataset.

3.2 Data Preprocessing

This stage focused on creating balanced and efficient data loaders to improve model generalization. The pipeline generated **multi-resolution data loaders** and performed **image augmentations** (resizing, flips, rotations) to enhance model robustness and generalizability across varying image scales and orientations. To address class imbalance, the dataset was split using **stratification**, ensuring all classes were adequately represented in the training and validation sets. Finally, a **weighted random sampler** was employed to assign higher sampling probabilities to underrepresented classes, enabling the model to learn more effectively from rare but critical lesions. These preprocessing steps prepared the data for effective model training and classification.

3.3 Environment setup

This stage acts as the control center for configuring parameters for launching a training run for model architecture. It ensures that each model is paired with the correct data format and configuration before training begins. The system is designed flexible for the training and evaluation of multiple models. The model selection includes efficientnet_b0, densenet121, inception_v3 and swin_t.

The next step is to identify the required image size. Image size is a critical parameter for each model for its expected input image dimension. Except for inception_v3 model which required image size of 299 * 299 as input, the rest models require 224 * 224. This image size configuration will be used for applying transformation and creating the correct data loader to the respective model. Once complete, the appropriate data loader can be called to align with the correct model.

3.4 Model Initialization

This stage will prepare the selected model architecture for training. The model configuration is created and passed into run_experiment function. The models were initially trained using weights from the IMAGENET1K_V1 dataset for transfer learning. This method uses strong, general features learned from a large amount of data. The feature extractor is set to false for fine tune all layers by setting requires_grad for all layers in the model. For the inception_v3 case, the feature extractor is set to true and “last3” for the unfreeze strategy. The inception_v3 is partially fine-tuned to prevent overfit and computation is too heavy since it has a 299-input size which is way larger than other models. The last 2 inception blocks and the full connected layer are unfrozen by setting required_grad to true.

Next, the fully connected layer of the four models is replaced with a new classification layer, designed to output predictions for the 7 distinct skin lesion classes in our project. For the inception_v3 case, the auxiliary classifier, a unique feature of the Inception architecture, is similarly modified. This is because it acts as a side branch to predict the image and add the loss into the main classifier [3]. After that, the model will transfer to GPU to accelerate the training process.

3.5 The Training Loop

Before the training, Optimizer, Loss, and Scheduler are defined. The AdamW optimizer is used to update the model's weights with learning rate of 1e-4 for stable and fast learning. The weight_decay, 1e-5 is set for regularization. For the InceptionV3 model, a specialized technique of differential learning rates was employed: the pretrained convolutional layers were fine-tuned with a small learning rate (1e-4), while the newly added classification layers were trained with a higher learning rate (1e-3). This allows for stable, minor adjustments to the robust pretrained features while enabling the new layers to learn the specifics of our dataset more quickly for handling 299 size which is much heavier than others. The CrossEntropyLoss function is used, which is standard for multi-class classification to fit our case. Finally, a

ReduceLROnPlateau learning rate scheduler is implemented to monitor the validation loss and decrease the learning rate if the model's performance plateaus, helping to achieve better convergence.

The model is trained for a set number of epochs. The run one epoch for training and validate is called per epoch. In training per epoch, the models and set to training mode. The inputs and label dataset batches are moved to GPU for forward propagation, compute loss, backpropagate, and update model weights. For the inception_v3 case, the use_amp flag can be set to true. This is to activate Automatic Mixed Precision (AMP) to speed up calculations and reduce GPU memory usage for Inception_v3's computational demand. AMP utilizes lower-precision floating-point operations whenever feasible. Next, additional loss handling is needed for auxiliary classifiers for stabilizing training. The loss from the auxiliary classifier is added to the main loss with 40% which total loss=main loss+0.4×aux loss. This can act as a regularizer by giving less weight. After training one epoch, validating one epoch is run by settings model to evaluation mode to check modal performance.

Instead of running to complete all epoch's, Early Stopping mechanism is active throughout this process, monitoring the validation loss. If the loss does not improve for 10 consecutive epochs called "patience", the training is halted automatically and then records the best model.

3.6 Best Model Handling & Evaluation

This final stage is to evaluate the best model and save it as. pth for application development.

After the training loop terminates, the weights from the epoch that achieved the lowest validation loss saved by the Early Stopping mechanism are loaded back into the as best model.The best model is tested with the unseen test data for generating performance matrix and confusion matrix. The best model among the 4 selected models (efficientnet_b0, densenet121, inception_v3 and swin_t) is used for application development.

4. EXPERIMENT & EVALUTION

4.1 Experimental Setup

The HAM10000 dataset was used for training and evaluation. It consists of 7 classes of skin lesions with the significant class imbalance. All images were resized to 224x224 pixels (except InceptionV3, which used 299x299) and normalized before training. To address class imbalance, a **WeightedRandomSampler** was applied on the training set using inverse-frequency weights, ensuring that minority classes were more likely to be sampled.

All models were implemented in PyTorch and trained on a GPU environment (GPU T4 x2). The **AdamW** optimizer was used with an initial **learning rate** of **0.0001** and **weight decay** of **1e-5**. The loss function was **CrossEntropyLoss** without class weights. Training incorporated the **ReduceLROnPlateau** scheduling, **early stopping** was applied based on the validation loss, and **mixed-precision training** for efficiency. For InceptionV3, both the main and auxiliary classifiers were considered in the loss calculation.

Evaluation was based on accuracy, macro F1-score, and weighted F1-score, as these metrics provide a fair view of model performance for class imbalance. Confusion matrices were also generated to highlight the common misclassifications.

4.2 Hyperparameter Tuning

Basic hyperparameter tuning was carried out to improve model stability and performance. For EfficientNet-B0, DenseNet121, and Swin Transformer, the final configuration used a **learning rate of 0.0001**, **batch size of 32**, and **patience of 10 epochs** for early stopping. For InceptionV3, due to higher computational cost, the batch size was reduced to 16, and a two-tier learning rate

was applied (0.001 for the classifier layers and 0.0001 for the backbone) with a slightly larger weight decay of 1e-4.

The ReduceLROnPlateau scheduler was used to automatically reduce the learning rate when the validation loss plateaued, which helped prevent stagnation during training. Each model was trained for a maximum of 100 epochs, and the checkpoint with the lowest validation loss was selected for testing.

4.3 Results and Analysis

Table 4.1 summarizes the performance of the four models on the HAM10000 test set. The metrics reported are overall accuracy, macro F1-score, and weighted F1-score, together with each model's best-performing class and weakest-performing class.

Table 4.1 Performance Comparison of Models

Model	Accuracy (%)	Macro F1	Weighted F1	Best Class (F1)	Weakest Class (F1)
EfficientNet-B0	84.5	0.780	0.850	df (0.84)	mel (0.61)
DenseNet121	86.5	0.812	0.867	df (0.90)	mel (0.64)
Swin Transformer	**88.9**	**0.834**	**0.890**	**vasc (0.95)**	**mel (0.67)**
InceptionV3	77.3	0.675	0.782	nv (0.88)	mel (0.54)

From Table 4.1, **Swin Transformer** achieved **the best overall performance** with an accuracy of 88.9%, and a weighted F1-score of 0.890. It also achieved the strongest per-class performance on vascular lesions (F1:0.95). DenseNet121 achieved an F1 of 0.90 on dermatofibroma, maintaining balanced results across classes and ranked second. EfficientNet-B0 performed well as a baseline, but its accuracy and macro F1 were slightly lower. On the other hand, InceptionV3 recorded the weakest overall performance, although it performed well on the dominant nevus class (F1:0.88).

Across all models, melanoma consistently remained the weakest class (F1 between 0.54 and 0.67). This reflects the clinical difficulty of distinguishing melanoma from visually similar classes such as nevus. The result of confusion matrices confirmed that many misclassifications occurred between these two categories. In conclusion, the Swin Transformer reduced such errors more effectively than the CNN models, demonstrating the benefit of transformer-based global attention in this medical imaging context.

4.4 Comparative Discussion

The experimental results demonstrate clear differences between CNN-based models and the transformer-based architecture. Among the CNNs, DenseNet121 outperformed EfficientNet-B0 by achieving higher accuracy and F1-scores. Its dense connectivity pattern allowed features to be reused more effectively across layers, leading to stronger representation learning. On the other hand, EfficientNet-B0 served as a strong baseline, offering competitive performance while being computationally lighter and faster to train, which is an advantage in resource-constrained environments. In contrast, InceptionV3 showed the weakest performance overall. While it performed adequately on the dominant nevus class, it struggled with rarer lesion types, showing that this is the limitations of older architectures compared to modern, deeper designs. The Swin Transformer consistently achieved the best overall results. Its self-attention mechanism enabled the model to capture long-range dependencies and global contextual

information, which proved especially useful in reducing the misclassification between visually similar lesions such as melanoma(mel) and nevus (nv). This finding aligns with the recent research in medical imaging that highlights the strength of transformer-based models in handling fine-grained classification tasks [14], [15]. However, these gains bring the cost of greater computational requirements, which shows that it is a trade-off between the accuracy and efficiency. In practice, DenseNet121 or EfficientNet-B0 might still be preferable when hardware resources are limited, while Swin-T would be more suitable in clinical settings where higher diagnostic accuracy is critical.

Across all the models, the most challenging class was melanoma (mel), which consistently recorded the lowest F1-score. This is concerning because mel is the most clinically significant class due to its high mortality risk if it is undetected. The difficulty arises from its visual similarity to benign classes such as nevus and from its relatively small representation in the dataset, an issue highlighted in previous studies [1].

To address these challenges, researchers have proposed various strategies. Data-level approaches such as advanced augmentation, synthetic lesion generation using GANs, and targeted oversampling have been shown to improve classification performance for minority classes [16]. On the algorithmic side, cost-sensitive learning and focal loss functions have been successfully applied to increase sensitivity for melanoma detection [17]. Improving recall for melanoma should be prioritized, as false negatives in this class could have severe clinical consequences.

5. CONCLUSION

In conclusion, this project investigated skin lesion classification on the HAM10000 dataset using 4 deep learning architectures. They are EfficientNet-B0, DenseNet121, Swin Transformer, and InceptionV3. The results showed Swin Transformer achieved the best overall performance, followed by DenseNet121, EfficientNet-B0, and lastly InceptionV3. These findings highlight the balance between accuracy and computational efficiency when selecting models for medical imaging tasks.

Across all models, melanoma remained the most difficult class to classify, reflecting its clinical importance and the dataset's imbalance issue. To solve these challenges in future, it may require strategies such as data augmentation, GAN-based synthesis, or cost-sensitive learning, which can improve recall and move closer to this project's objective, which is providing a reliable AI-assisted diagnostic tool for skin cancer detection.

REFERENCES

[1] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," Sci. Data, vol. 5, no. 1, p. 180161, 2018.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, pp. 115–118, 2017.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," arXiv.org, https://arxiv.org/abs/1512.00567

[4] PyTorch, "Inception v3," Torchvision Models Documentation. [Online]. Available: https://pytorch.org/vision/main/models/generated/torchvision.models.inception_v3.html.

[5] A. Zarea and O. Pourkazemi, "A combined U-Net and DenseNet-121 framework for the segmentation and classification of skin lesions in dermatoscopic images," arXiv preprint arXiv:2110.04632, 2021.

[6] E. H. I. Eliwa, “Enhancing Skin Cancer Diagnosis Through Fine-Tuning of Pretrained Models: A Two-Phase Transfer Learning Approach,” Int. J. Breast Cancer, vol. 2025, no. 1, p. 4362941, Feb. 2025.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in International Conference on Machine Learning (ICML), 2019, pp. 6105–6114.

[9] Abayomi-Alli, O., Damaševičius, R., Maskeliūnas, R., & Abayomi-Alli, A. (2021). "An Ensemble of Deep Learning-Based Models for Automatic Skin-Lesion Classification." Electronics, 10(23), 3020.

[10] Kassem, M. A., Hosny, K. M., & Fouad, M. M. (2021). "Skin Lesions Classification into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning." IEEE Access, 9, 132694-132704.

[11] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986.

[12] Pacal, I., Alaftekin, M. & Zengul, F.D. Enhancing Skin Cancer Diagnosis Using Swin Transformer with Hybrid Shifted Window-Based Multi-head Self-attention and SwiGLU-Based MLP. J Digit Imaging. Inform. med. 37, 3174–3192 (2024). https://doi.org/10.1007/s10278-024-01140-8

[13] K. Ranjeet, S. Saha, and M. Ratnakumari, “Skin Cancer Detection using Swin Transformer Model,” 2025, doi: https://doi.org/10.2139/ssrn.5292024.

[14] I. Pacal, M. Alafatekin, and F. D. Zengul, “Enhancing Skin Cancer Diagnosis Using Swin Transformer with Hybrid Shifted Window-Based Multi-head Self-Attention and SwiGLU-Based MLP,” J. Imaging Inform. Med., vol. 37, pp. 3174–3192, 2024.

[15] J. H. L. Goh et al., “Comparative Analysis of Vision Transformers and Conventional Convolutional Neural Networks in Detecting Referable Diabetic Retinopathy,” Ophthalmology Science, vol. 4, no. 6, pp. 100552–100552, Nov. 2024, doi: https://doi.org/10.1016/j.xops.2024.100552.

[16] Z. Qin, Z. Liu, P. Zhu, and Y. Xue, “A GAN-based image synthesis method for skin lesion classification,” Computer Methods and Programs in Biomedicine, vol. 195, p. 105568, Oct. 2020, doi: https://doi.org/10.1016/j.cmpb.2020.105568.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, doi: https://doi.org/10.1109/iccv.2017.324.