جامعة عجمان
AJMAN UNIVERSITY

PROJECT REPORT – Skin Cancer Classification with Neural Networks

Course: Programming for data analytics 2

prepared by:

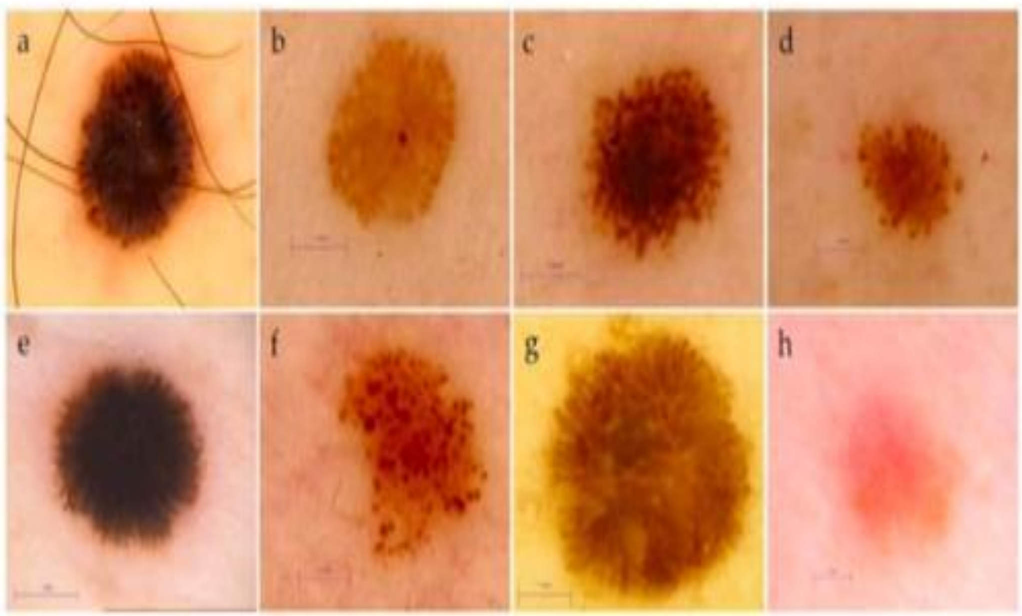
1. Bahaa Aldin Muhammed (202320258)
2. Kamal Thaer Al Najjar (202310289)
3. Ahmed Abdullah Alsaadi (202320281)
4. Muhammed Muntaser (202111209)

Under the supervision of:

Dr. Salam Fraihat

1. INTRODUCTION:

Skin cancer is among the most widely spread cancers, and early diagnosis is a key factor of recovery. Neural network-based image classification has great potential to help clinicians diagnose skin lesions with higher accuracy and efficiency.



The goal of this project is to create and analyze several deep learning models that classify skin cancer from Dermatoscopic images. Specifically,

we aim to:

1- Contrast classical CNNs with the ones proposed using transfer-learning.
2- Choose the model that does best on unseen test data.
3- Learn how model size, number of parameters, and training schedules impact classification accuracy.

2. DATA AND PREPROCESSING

2.1 DATASET DESCRIPTION

We will be using the HAM10000 dataset, which is labeled dermatoscopic images of skin lesions.

The Meta data has 7 classes / types of skin cancer:

Abbreviation	Full Name	Description
nv	Melanocytic Nevus	Common moles caused by a benign growth of melanocytes. Usually uniform in color and shape; non-cancerous.
mel	Melanoma	The most dangerous type of skin cancer; originates in melanocytes. Early detection is critical.
bkl	Benign Keratosis-like Lesion	Includes seborrheic keratoses, solar lentigines, and lichen-planus-like keratoses. All are non-cancerous but may resemble melanoma.
bcc	Basal Cell Carcinoma	A common skin cancer arising from basal cells. Grows slowly and rarely spreads but can damage nearby tissue.
akiec	Actinic Keratosis / Intraepithelial Carcinoma (Bowen’s Disease)	Precancerous or early-stage squamous cell carcinoma caused by sun damage. Needs monitoring or removal.
vasc	Vascular Lesion	Includes angiomas, hemangiomas, and pyogenic granulomas. Typically, benign and related to blood vessels.
df	Dermatofibroma	A benign skin growth, often firm and slightly pigmented, resulting from minor trauma or insect bites.

2.2 PRE-PROCESSING STEPS

To save time and make the models more inclusive we implemented most of our pre-processing and image transformation inside the layers of the models, which includes:

1- Maxpooling2D (It reduces the image size by taking the maximum value inside each block.
2- GlobalAveragePooling2D (It takes each feature map (channel) from a CNN and reduces it to 1 number by taking the average of all its values).
3- Flatten (turns any matrix into 1D)
4- Normalization (It normalizes each pixel channel (R, G, B) separately by subtracting the mean and dividing by the standard deviation)
5- Batch-Normalization: keeps the internal values of each layer stable by normalizing them for every batch (during training)
6- Data augmentation (Not in layers).
7- Train/Validation/Test split (Not in layers).

2.3 CLASS IMBALANCE

After we examined the distribution of the classes, we found out that the classes were extremely imbalanced with most of the samples belonging to the NV class.

Of course, presenting the data to the models in this form will create a heavy bias towards the NV class effect predictions for the other classes.

So, to solve this issue, we will be implementing an **Augmented image generator** which takes our data, makes small changes to it and presents it over and over again to the models in different ways.

The following transformations were applied to the images inside the generator

- Rescale (Every pixel value in your image is divided by 255)
- horizontal flip
- rotation
- zoom
- width shift
- height shift

- fill (fills empty pixels)

We also made an **Unscaled version of that generator** for the pre-trained models, as they have a specific internal way of preprocessing images so we don't want to mess with that.

Why these steps matter:

1- Data augmentation reduces overfitting and helps with the class imbalance issue.
2- Neural networks train better when RGB values are small, usually between 0 and 1
3- after those augmentations are done some parts of the image are left empty (blank pixels) fill_mode tells Keras how to fill those empty pixels.
4- Appropriate normalization enhances training stability.
5- Max pooling helps Reduces computation, extracts strongest features, prevents overfitting, and Helps model focus on important patterns.

2.3 Evaluation Metrics

The following metrics are used for evaluation:

1- Accuracy over epochs plot
2- Loss over epochs plot
3- Confusion Matrix
4- Classification Report

These metrics enable comparison between different architectures on a fair ground.

3. METHODOLOGY

We trained six different models, combining custom CNN architectures with transfer learning models:

3.1 Custom CNNs

1- First CNN – basic combination of Dense and Conv2D layers, unoptimized architecture, no early stopping or monitoring mechanisms

2- Second CNN – improved design with batch normalization, weight initializers, L2 regularization, and Bias initializers.

3.2 Pre-Trained models:

1- VGG 16
2- VGG 19
3- ResNet50V2
4- MobileNetV2

4. EXPERIMENTS

1. Experiment 1: First CNN (Baseline Model)

The first experiment aimed to give a baseline using a lightweight CNN built from scratch. This then helps us establish the performance minimum that is achievable without transfer learning, therefore allowing us to compare more advanced models.

Interpretation:

The baseline model has struggled with class imbalance and had a very large validation loss. Rather, it served as a point of reference for later improvements.

2. Experiment 2: Second CNN (Improved Custom CNN)

The second experiment enhances the baseline CNN by adding more complexity, increasing the number of convolutional layers, adding batch normalization, and incorporating stronger regularization. This tests whether a deeper custom CNN can enhance the performance or make it worse.

- Added extra convolutional layers and filters.
- Included batch normalization for stability
- Introduced dropout and L2 regularization.

- **Early stopping**: in the early stopping mechanism **we valued validation loss over accuracy** because its More sensitive than accuracy, better signal for generalization. Early stopping is about stopping before overfitting, not about squeezing the last noisy bump in accuracy. val_loss tends to peak in a smoother, more reliable way than val_accuracy.

Interpretation:

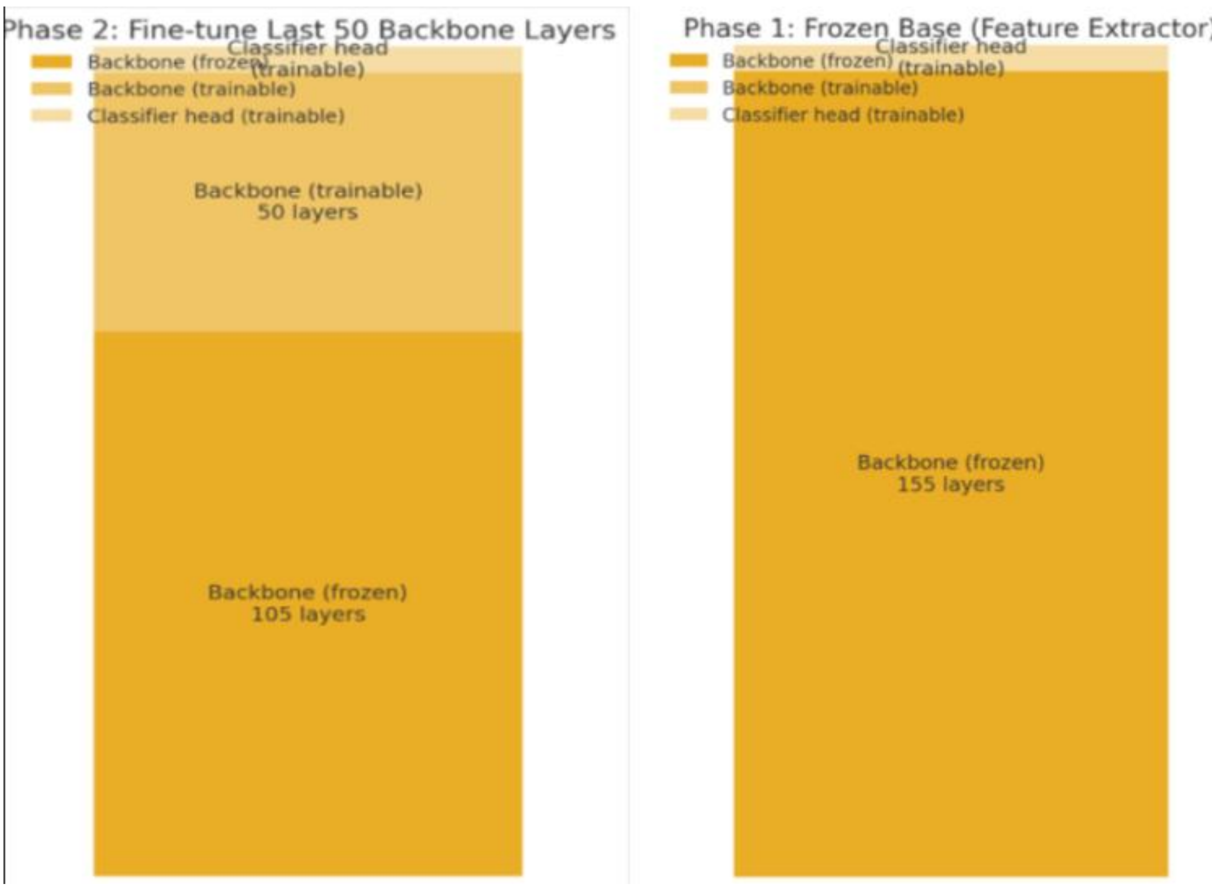
Performance seems to be slightly better compared to the baseline, proving that a deeper CNN doesn’t exactly mean a vastly improved performance, the validation accuracy was dipping while the model accuracy rose which indicates signs of overfitting.

Pretrained models

These models already have learned rich, generic image features from a large dataset such as ImageNet. We don't want to lose this knowledge so we freeze the Base layers to stop them from learning (no weight updates), then we train the top layers, after that is done, we unfreeze some of the base layers for fine tuning.

when we recompile after unfreezing, we make sure to have a slower learning rate (e.g., 1e-5) to make very gentle updates.

the model fine-tunes its higher-level features — adapting them from generic objects (edges, textures, shapes) to domain-specific features (skin lesions, pigmentation patterns, borders, etc.).

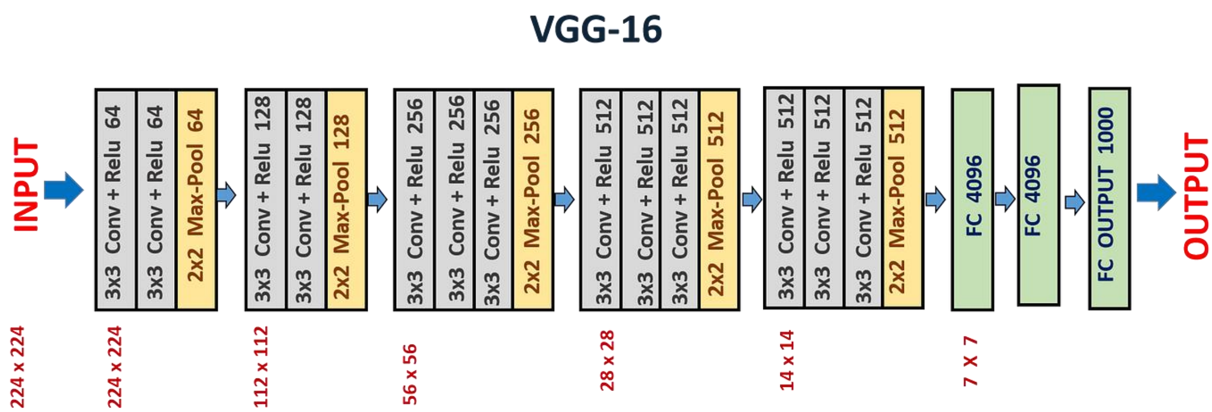


3. Experiment 3: VGG16

VGG16 is a well-known architecture for its powerful performance in visual classification tasks. This experiment investigates the gains from transfer learning over custom-designed CNNs.

Interpretation:

the VGG16 model produced a very good performance and a huge noticeable improvement compared to the previous two models making it a very good candidate to our data

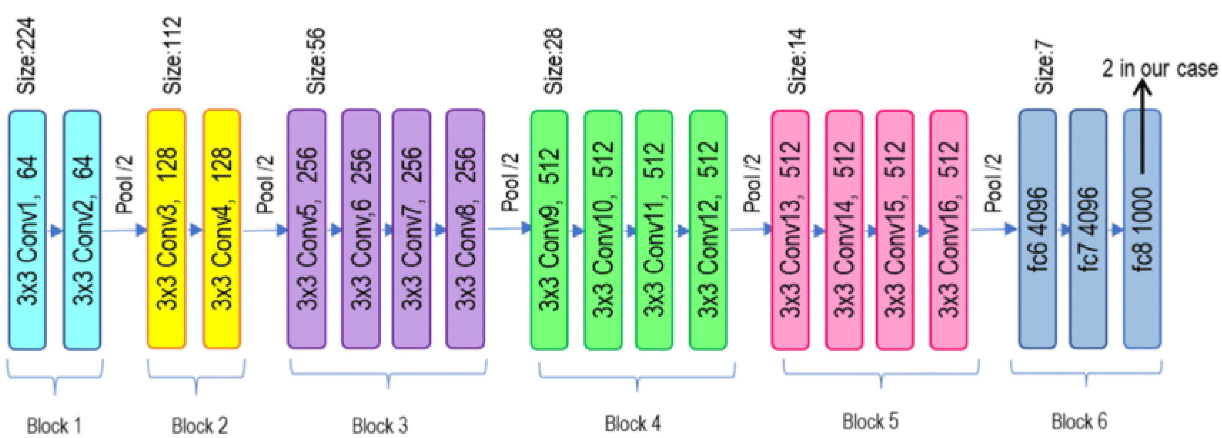


4. EXPERIMENT 4: VGG19

VGG19 is an extended version of VGG16, adding more layers on top. The goal was to see if the further layers helped or caused overfitting on this dataset.

Interpretation:

VGG19 outperformed all other models, including VGG16, by providing even stronger accuracy and lower validation loss. Its deeper architecture helped it capture more detailed features, leading to the most precise and stable predictions. Overall, VGG19 stood out as the best-performing model in the entire evaluation.

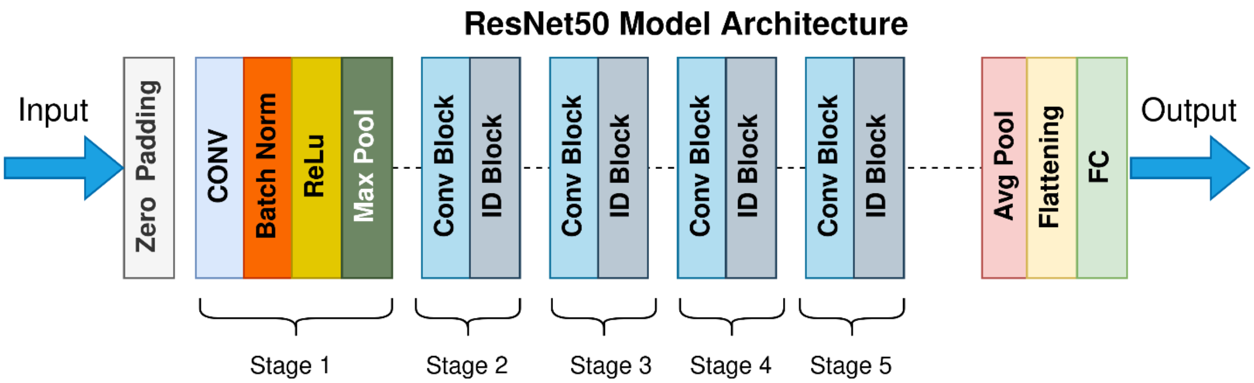


5. Experiment 5: ResNet50V2

Residual connections are applied in ResNet architectures to avoid vanishing gradients. This experiment investigates whether deeper residual learning improves the classification performance compared with VGG models

Interpretation:

ResNet50V2 performed strongly thanks to its skip-connection architecture, which helps the model learn deeper and more complex features without losing important information. Although it didn't outperform the VGG models, it still demonstrated reliable accuracy and stable learning, showing that its residual design is effective for recognizing subtle patterns in skin-lesion images.



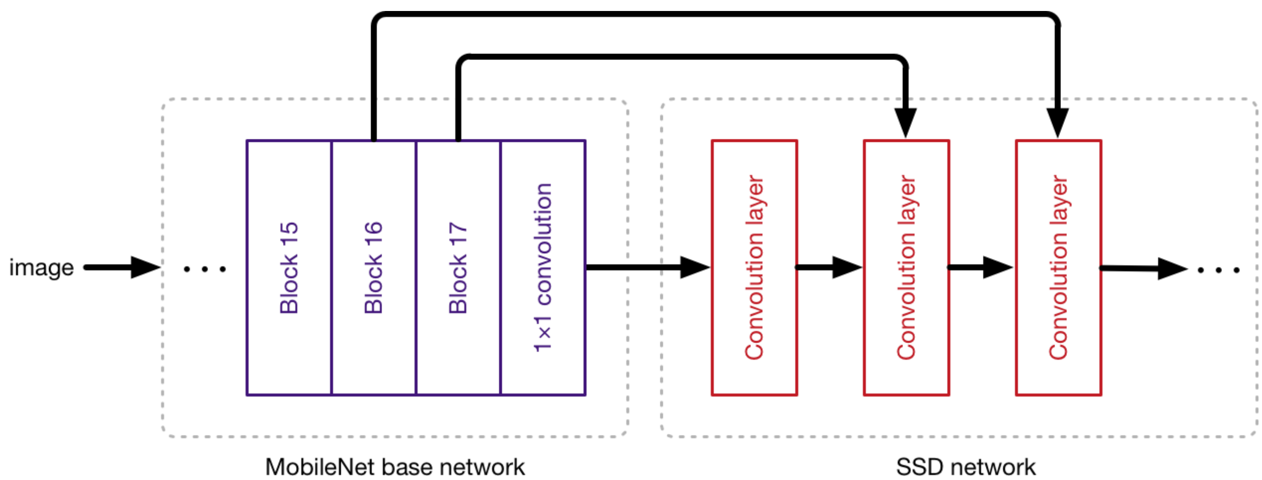
6. Experiment 6: MobileNetV2

MobileNetV2 is lightweight and intended for mobile/edge devices. This experiment serves to validate whether competitive performance can be achieved on a smaller model with fewer parameters.

52 layers

Interpretation:

For a light-weight model the performance of MobileNetV2 wasn't that bad. Even though it didn't match the accuracy of VGG or ResNet, it remains a good choice when speed and computational efficiency are more important than squeezing out the highest accuracy.



7. Summary of models and their respective Metrics

Rank	Model	Accuracy	loss	Val_loss
1	**VGG19**	**0.9202**	**0.2160**	**0.4491**
2	VGG16	0.9116	0.2220	0.4520
3	ResNet50V2	0.8826	0.3099	0.5019
4	MobileNetV2	0.8209	0.4782	0.5371
5	Second CNN	0.7894	0.5676	0.6365
6	First CNN	0.7221	0.7375	0.7715

5. Conclusions

From the experiments conducted:

1- VGG19 turned out to be the most accurate and reliable model, giving an accuracy of 92.02% and the lowest validation loss.

2- Transfer learning models performed better in every case than custom-built CNNs.

3- Proper preprocessing of data along with augmentation is required for reducing overfitting.

4- The project has shown the feasibility of deep learning for skin cancer classification automation.