Skin Cancer Classification Using Deep Convolutional Neural Networks

I. Overview

This project focuses on skin cancer classification using Deep Convolutional Neural Networks (DCNNs) and transfer learning.
Inspired by Esteva et al.'s work on dermatologist-level skin cancer classification, this study fine-tunes several powerful pretrained models on the HAM10000 dermatoscopic image dataset.

The goal was to evaluate:

- How well different DCNN architectures transfer from ImageNet to medical imaging.
- Whether fine-tuning only top layers or fine-tuning the entire model works better.
- Which architecture performs best on this dataset.

Models tested:

- Baseline CNN
- VGG16
- InceptionV3
- Inception-ResNet-V2
- DenseNet201
- Ensemble (InceptionV3 + DenseNet201)

The highest accuracy achieved is 88.52% using an ensemble model.

II. Objectives

The project aims to:

1. Fine-tune deep CNNs (VGG16, InceptionV3, Inception-ResNet-V2, DenseNet201) for skin cancer classification.
2. Compare their performances by testing both:
   - Fine-tuning only the top layers
   - Fine-tuning the entire pretrained model

3. Build a baseline CNN model to understand dataset difficulty.
4. Create an ensemble model combining InceptionV3 and DenseNet201 to improve accuracy.

III. Dataset Used

The project uses the HAM10000 (“Human Against Machine”) dataset.

Key details:

- Total images: 10,015 dermatoscopic images
- Time span: Images collected over 20 years
- Image type: Dermoscopic
- 7 classes of skin lesions:
    - Melanocytic nevi
    - Melanoma
    - Benign keratosis-like lesions
    - Basal cell carcinoma
    - Actinic keratoses / Bowen's disease
    - Vascular lesions
    - Dermatofibroma

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

IV. Technologies and Libraries used

Programming Language: Python

Deep Learning Frameworks:

- TensorFlow
- Keras

Supporting Libraries:

- NumPy, Pandas
- Matplotlib, Seaborn
- OpenCV

Computer Vision Techniques:

- Data Augmentation
- Transfer Learning
- Batch Normalization
- Ensemble Learning

V. Methodology

A. Dataset Exploration

- Examined distribution of 7 lesion types
- Noted strong class imbalance
- Visualized sample images per class
- Resized and normalized images

B. Baseline Model

A small custom CNN was built to understand dataset difficulty.

Architecture:

- 3 convolution blocks (16 → 32 → 64 filters)
- Each followed by max-pooling
- Fully connected layers + softmax

Training:

- Adam optimizer (LR=0.01)
- Learning rate decay
- Data augmentation
- Trained for 35 epochs

C. Transfer Learning Models

1. VGG16

- Removed ImageNet top layers

- Added:
    - Global Max Pooling
    - Dense(512)
    - Dropout
    - Softmax(7)
- Steps:
    - Freeze base → train new top layers
    - Unfreeze last block → fine-tune 20 epochs

2. InceptionV3

Two experiments:

1. Fine-tune last 2 inception blocks
2. Fine-tune entire model

Important detail:
Batch Normalization layers must be left trainable to avoid mismatch in statistics between training and inference.

3. Inception-ResNet-V2

- Uses residual connections + inception modules
- Fine-tuned top layers for 30 epochs

4. DenseNet201

Two experiments:

1. Fine-tune only last dense block
2. Fine-tune the entire model (20 epochs)

DenseNet worked best because:

- Each layer connects to all previous layers
- Very efficient feature reuse
- Fewer parameters compared to similar deep CNNs

D. Ensemble Model

Combined predictions of fully fine-tuned:

- InceptionV3
- DenseNet201

Result: highest improvement in accuracy.

VI. Results

A. Fine-Tuning Only Top Layers

Model	Validation Accuracy	Test Accuracy
Baseline CNN	77.48%	76.54%
VGG16	79.82%	79.64%
InceptionV3	79.94%	79.94%
Inception-ResNet-V2	80.82%	82.53%
DenseNet201	85.80%	83.90%

B. Fine-Tuning Whole Model

Model	Validation Accuracy	Test Accuracy
InceptionV3	86.92%	86.83%
DenseNet201	86.70%	87.72%

C. Ensemble Model

Model	Validation Accuracy	Test Accuracy
InceptionV3 + DenseNet201	88.80%	88.52%

Key Observations

- Fine-tuning the whole model performs better than tuning only top layers.
- Full fine-tuning also trains faster (20 epochs vs 30).
- DenseNet201 consistently outperforms all other single models.
- The ensemble achieved the highest accuracy overall.