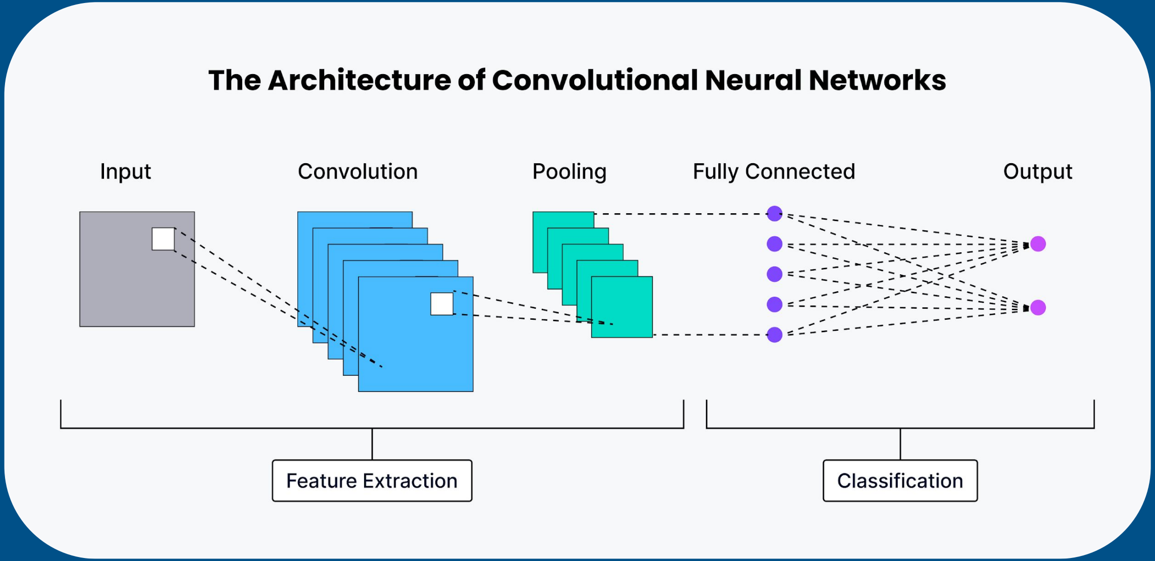
03.19.2025

Binary Classification of Dermoscopic Lesions via Bayesian-Optimized Ensemble Convolution Neural Networks

Michael Li
University of Notre Dame

Mentored by: Dr. Jianneng Li



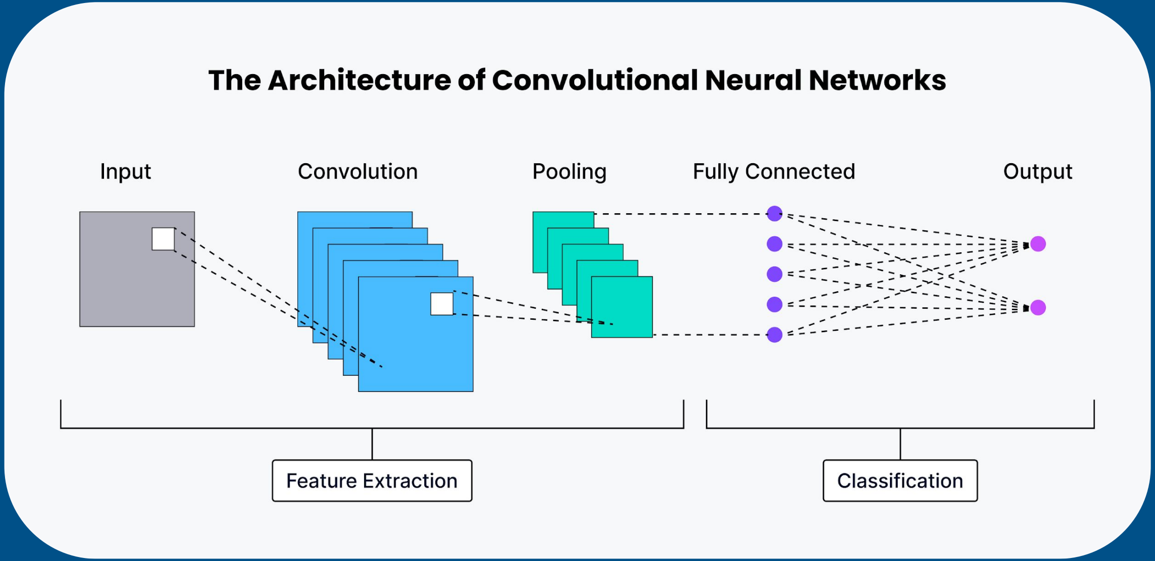
The Architecture of Convolutional Neural Networks

03.19.2025

Artificial Intelligence Powered Detection of Skin Cancer

Michael Li
University of Notre Dame

Mentored by: Dr. Jianneng Li



The Architecture of Convolutional Neural Networks

Introduction

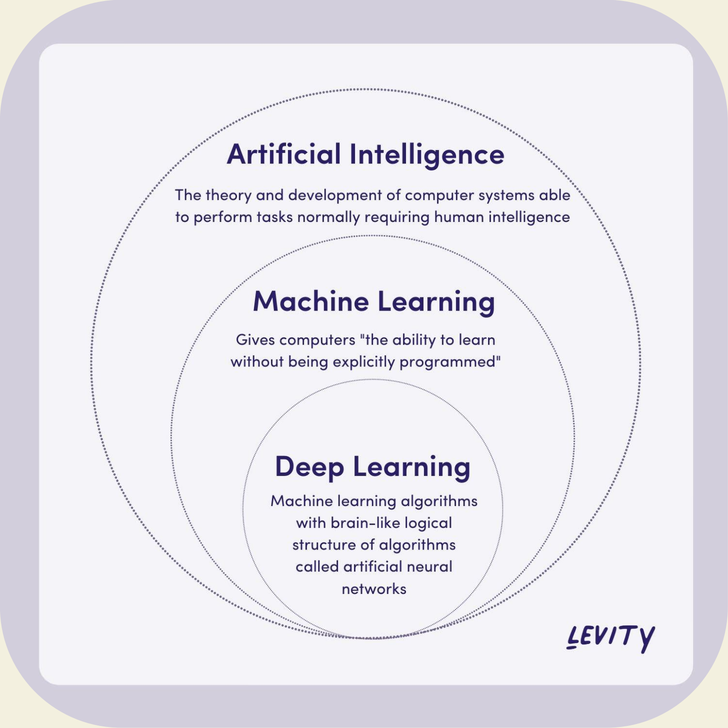
My name is Michael Li.

Class of 2026

Neuroscience Major (B.S)

Montreal, Quebec —----> Allentown, Pennsylvania

1. What is Deep Learning?



Artificial Intelligence

The theory and development of computer systems able to perform tasks normally requiring human intelligence

Machine Learning

Gives computers "the ability to learn without being explicitly programmed"

Deep Learning

Machine learning algorithms with brain-like logical structure of algorithms called artificial neural networks

- AI: Creating machines that can perform tasks that require human intelligence
- Machine Learning: Focused on algorithms, learn patterns from data
- Deep Learning: subset of machine learning using neural network with layers
  - Like a brain!

1. Deep Learning

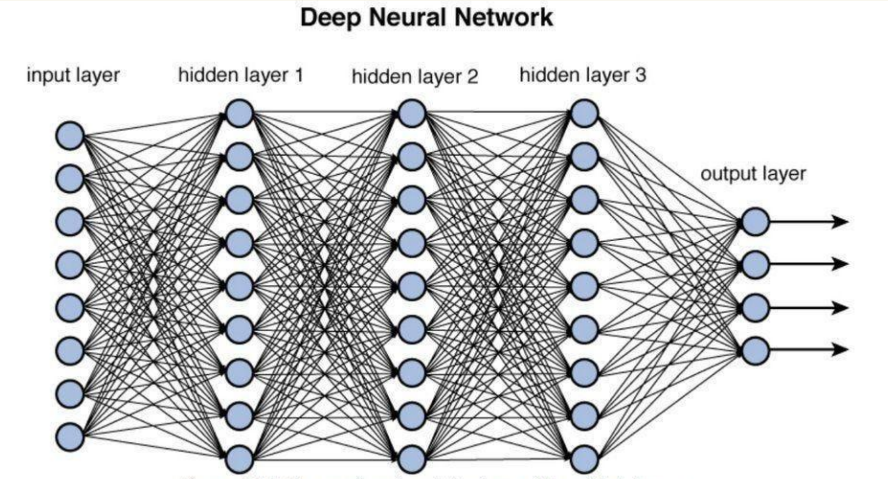
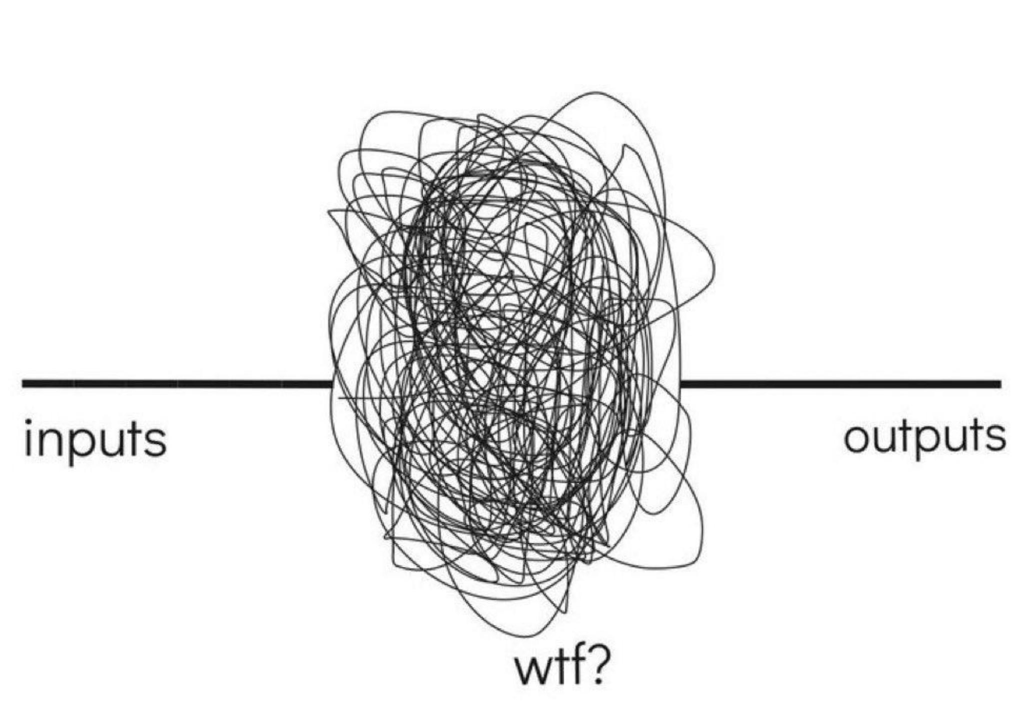


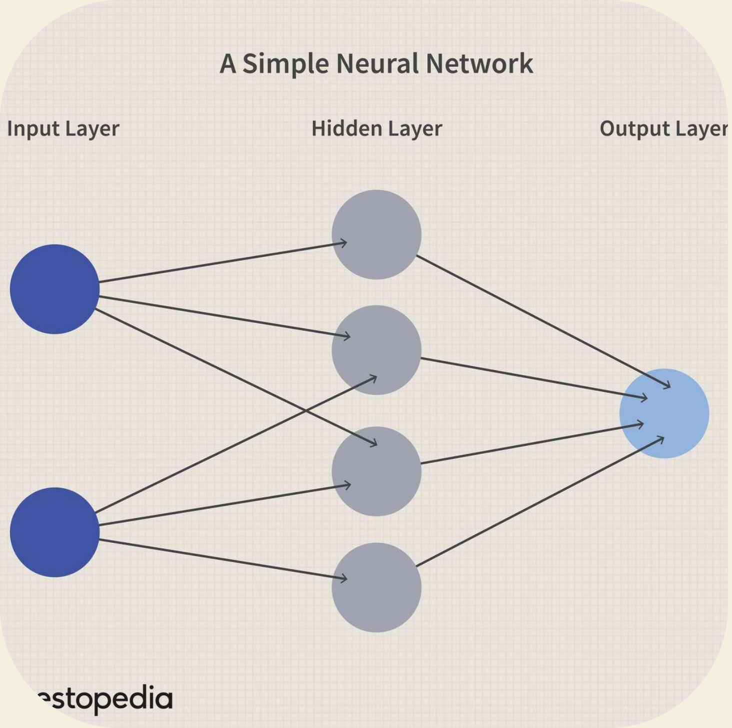
Figure 12.2 Deep network architecture with multiple layers.

1. What is a Neural Network



- Does anyone know what is going on?

1. What is a Neural Network



- Input Layer: Receives the initial data: like a pixel value in an image
- Hidden Layer: Intermediate layer where processing occurs
- Output Layer: Produce a final result/prediction

1. How does a Neural Network Work

- Each connection has a “weight”, which adjusts as the network learns
- 1. Receives a signal from previous neurons, multiplied by connection weight
- 2. Neuron applies an activation function to the sum of the weighted inputs
- 3. If the result exceeds a threshold, the neuron “fires”, sending is signal to the next layer
- 4. Backpropagation: Network adjusts its weights based on the error of its prediction

1. Neural Network: Gossiping

- Group of students (neurons) passing notes in a classroom. Each student has different relationships (weights) with other students that determine how seriously they take each other's messages.

1. When a rumor (input signal) starts at the front of the class:
2. Each student receives notes from multiple classmates, but values each note differently based on their relationship strength with each sender
3. Each student has their own "thinking process" (activation function) to decide what to believe based on all the notes they received
4. If they're convinced enough, they'll write new notes and pass them to the next row of students
5. When the rumor reaches the back of the class, the teacher compares it to the truth

1. Neural Network: Gossiping

6. Backpropagation: The teacher walks backward through the classroom, telling each student how wrong the final rumor was. Each student then adjusts how much they trust (weight) messages from different classmates based on who led them astray.

7. Over many rumors, the students learn optimal trust levels for each classmate, so accurate information flows through the classroom while minimizing distortions.

1. Types of Neural Networks

Feedforward Networks: Information flows in one direction

Convolutional Neural Networks (CNNs): Specialized for processing grid-like data such as images

Recurrent Neural Networks (RNNs): Have connections that form cycles, allowing them to maintain memory of previous inputs

Transformers: Use attention mechanisms to weigh the importance of different parts of the input data

1. What is a CNN

1. Convolutional Layers
   a. Apply “kernels” across the input (which would be an image)
   b. Each kernel detects patterns (like edges, textures, or shapes)
   c. Earlier layers detect simple features, deeper detects complex
2. Pooling Layers
   a. Reduces spatial dimensions (L and W) of the data
   b. Helps make the network less sensitive to exact positions of features
3. Feature Maps
   a. Output of applying filters to the previous layer
   b. Each feature map highlights where specific patterns appear
4. Fully Connected Layer
   a. Connect every neuron in one layer to every neuron in the next
   b. Process features and perform final classification

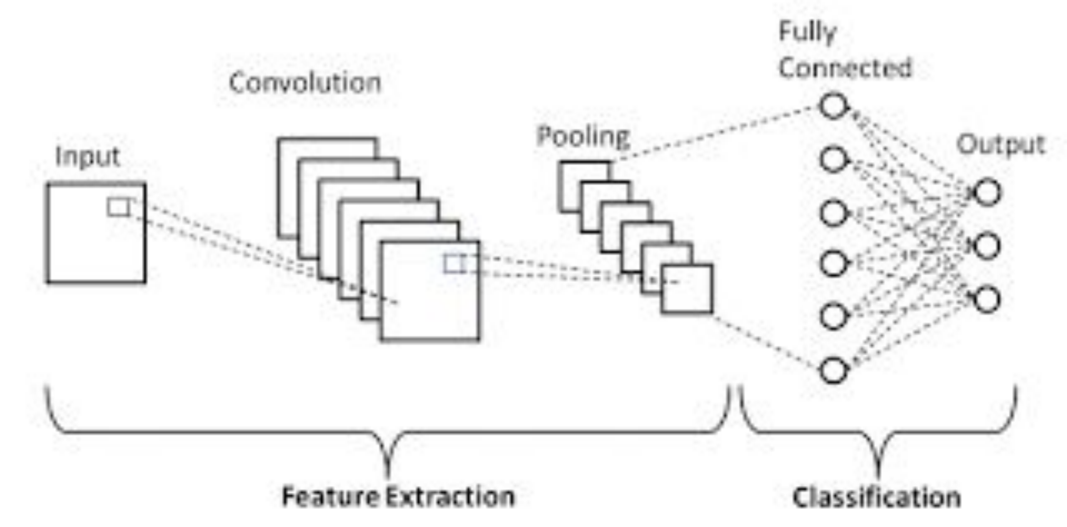
1. CNN Basic Workflow

1. An image enters as a grid of pixel values
2. Convolutional layers apply different filters to detect features (edges, corners, textures)
3. Pooling layers simplify these feature maps by reducing their size
4. More convolutional and pooling layers detect increasingly complex patterns
5. Fully connected layers interpret these high-level features for classification

1. Analogy (CNN)

Art Detectives examining a painting for authenticity

1. Junior Detectives aka, first convolutional layers work with a magnifying glass and look at basic stuff, like straight lines, curves, etc
2. Shift Supervisors (Pooling Layers): review junior detectives work, create simplified summaries keeping the important findings
3. Senior Detectives (Deeper Convolutional Layers) combine the simplified reports from supervisors and look for more complex patterns (Brush techniques)
4. Agency Director (Fully Connected Layer) receives all the processed information and makes final judgement


Input
Convolution
Pooling
Fully Connected
Output
Feature Extraction
Classification

1. What are the specific CNN's I used?

1. Resnet101
2. EfficientNet-B7
3. Densenet121

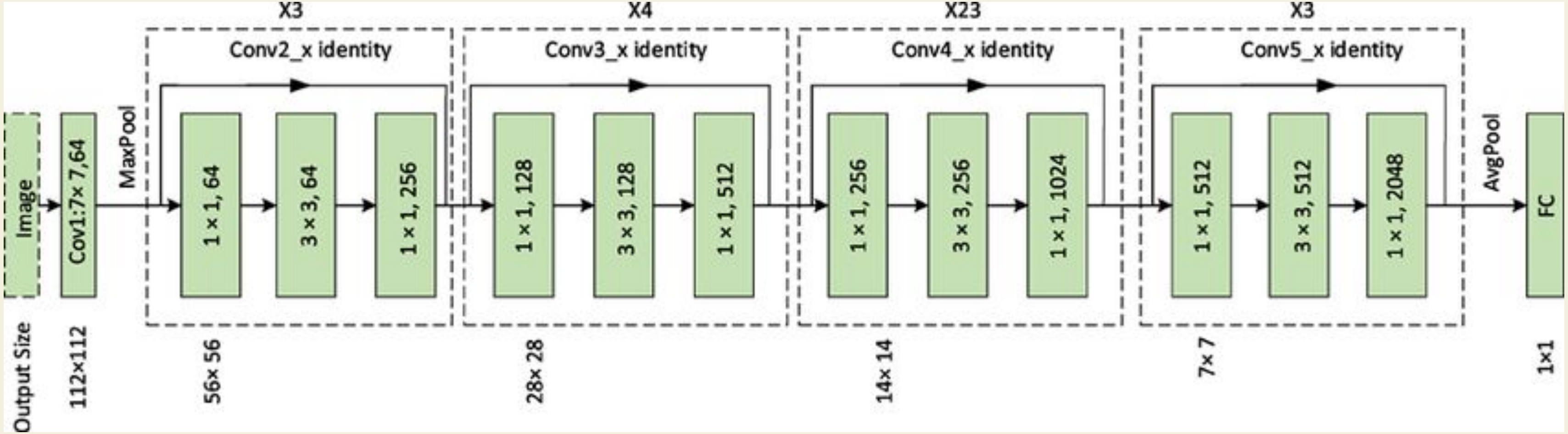
Why use multiple?

- Ensembling combines multiple models to produce better predictions than any single model alone. Different models make different mistakes, and have different strengths/weaknesses so combining their predictions reduces overall error and improves reliability?

Why do dynamic weighting?

- One model might be better than the others?

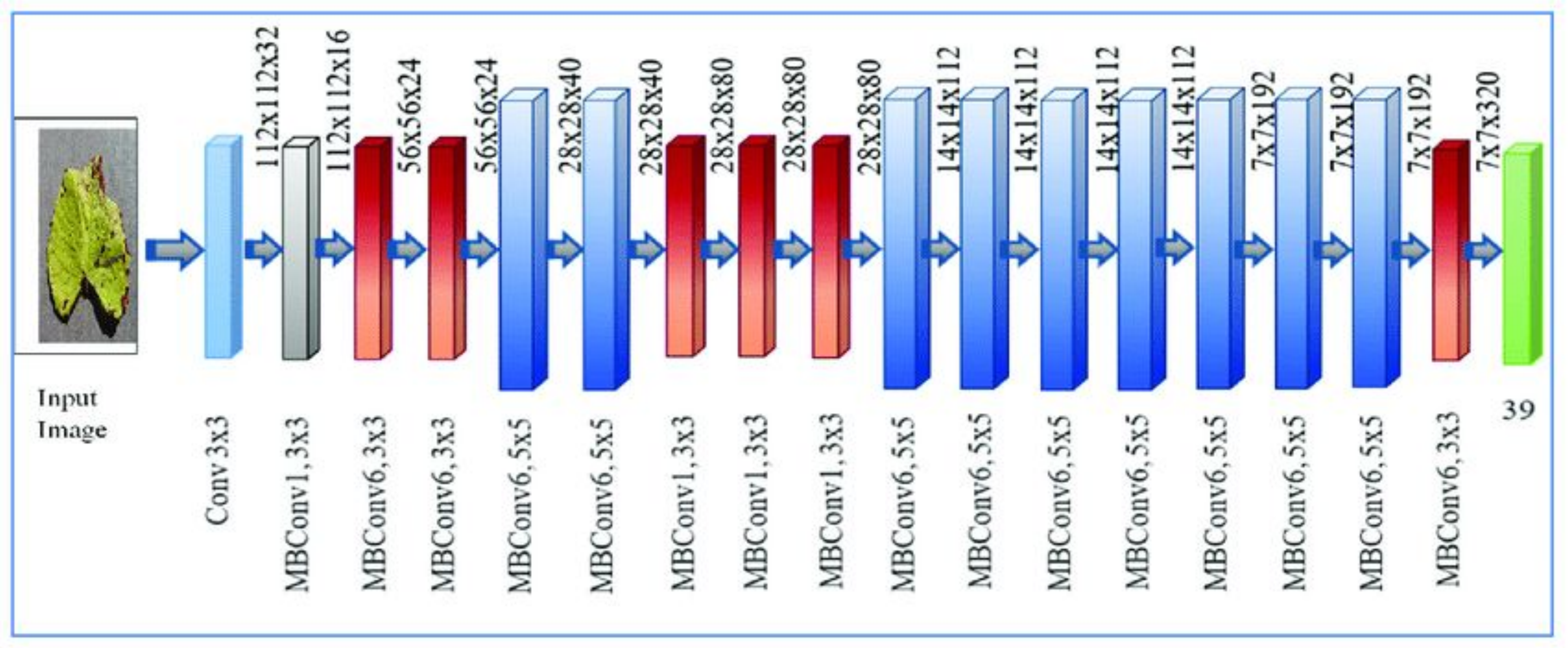
1. ResNet101


X3
Conv2_x identity
X4
Conv3_x identity
X23
Conv4_x identity
X3
Conv5_x identity
Image
Cov1:7× 7,64
MaxPool
1 × 1, 64
3 × 3, 64
1 × 1, 256
1 × 1, 128
3 × 3, 128
1 × 1, 512
1 × 1, 256
3 × 3, 256
1 × 1, 1024
1 × 1, 512
3 × 3, 512
1 × 1, 2048
AvgPool
FC
Output Size
112×112
56× 56
28× 28
14× 14
7× 7
1×1

1. Resnet101 Simplified

1. 101 layers
2. Its special feature is "skip connections" that let information jump ahead, solving the problem where very deep networks struggle to learn effectively.
3. Think of it like adding express elevators to a very tall building – information can travel directly between distant floors instead of stopping at every level, making the whole system work better.

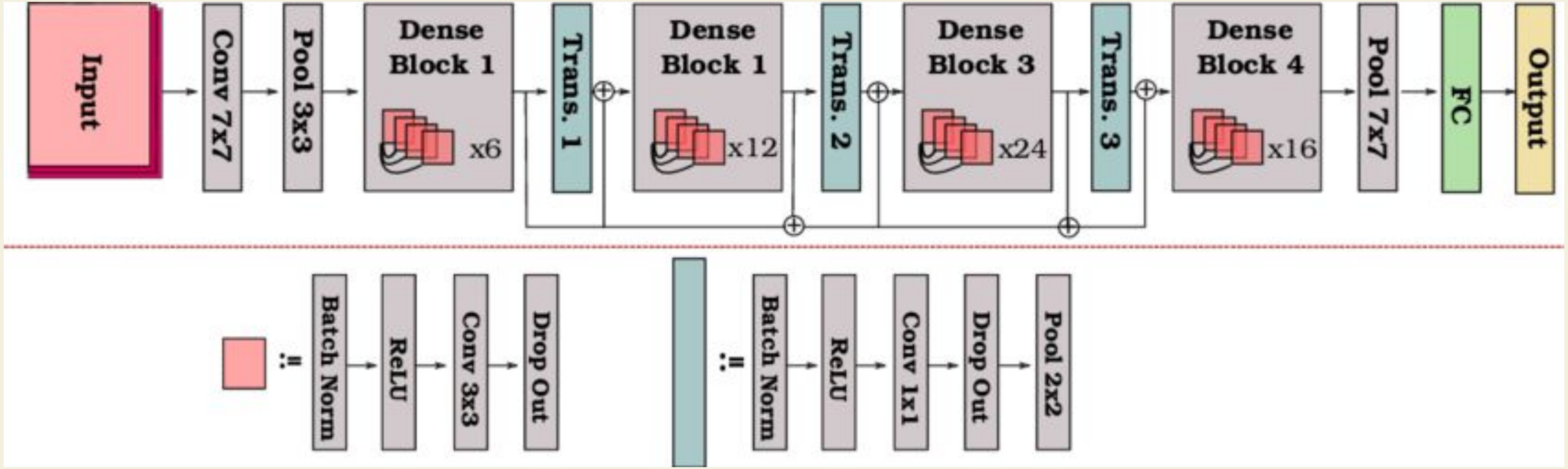
1. EfficientNet–B7



1. EfficientNet-B7 Simplified

1. Instead of just adding more layers like older models, it carefully grows in three balanced ways: it gets deeper (more layers), wider (more filters), and looks at higher resolution images.
2. Think of it like building a house - rather than just making it taller or wider, EfficientNet-B7 grows proportionally in all dimensions for the best results.
3. It's like getting a premium camera that takes excellent photos while using less battery power than other high-end cameras.

1. DenseNet-121



1. DenseNet-121 Simplified

1. 121 layers
2. DenseNet121 is like a classroom where every student can see all the notes from previous classes.
3. Instead of just building on the last lesson, each layer in the network can access all the information learned in earlier layers. This helps the network reuse features instead of relearning them, making it more efficient with fewer parameters.
4. Imagine connecting everything to everything ahead of it - that's what makes DenseNet "dense" and helps it make better predictions while requiring less computing power.

1. What did I do?

1. Is this important? Maybe

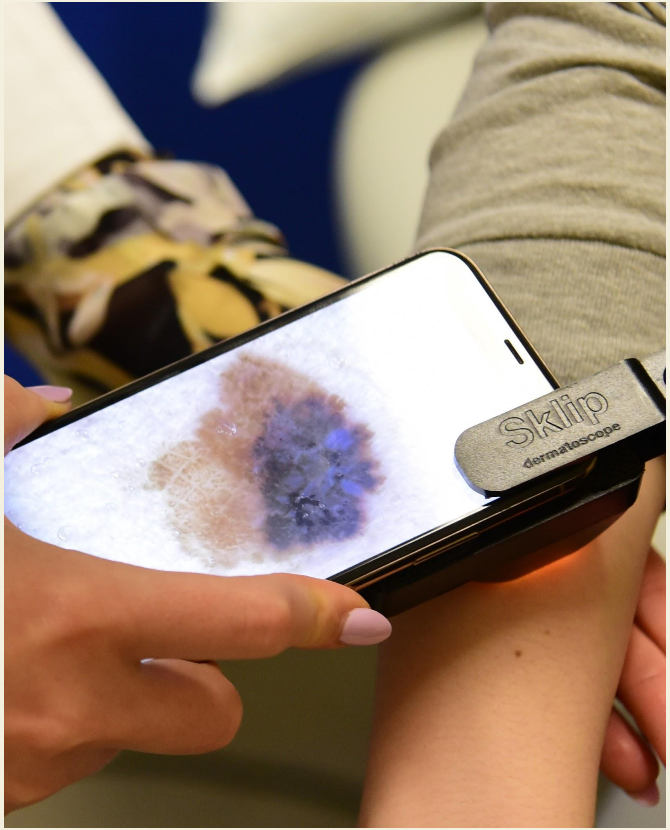
- Skin cancer is the most common cancer in the US
- 2 million Americans a year are diagnosed with it
- 1 in 5 people in America will develop skin cancer by the age of 70
- Statistics suggest that this is on the rise

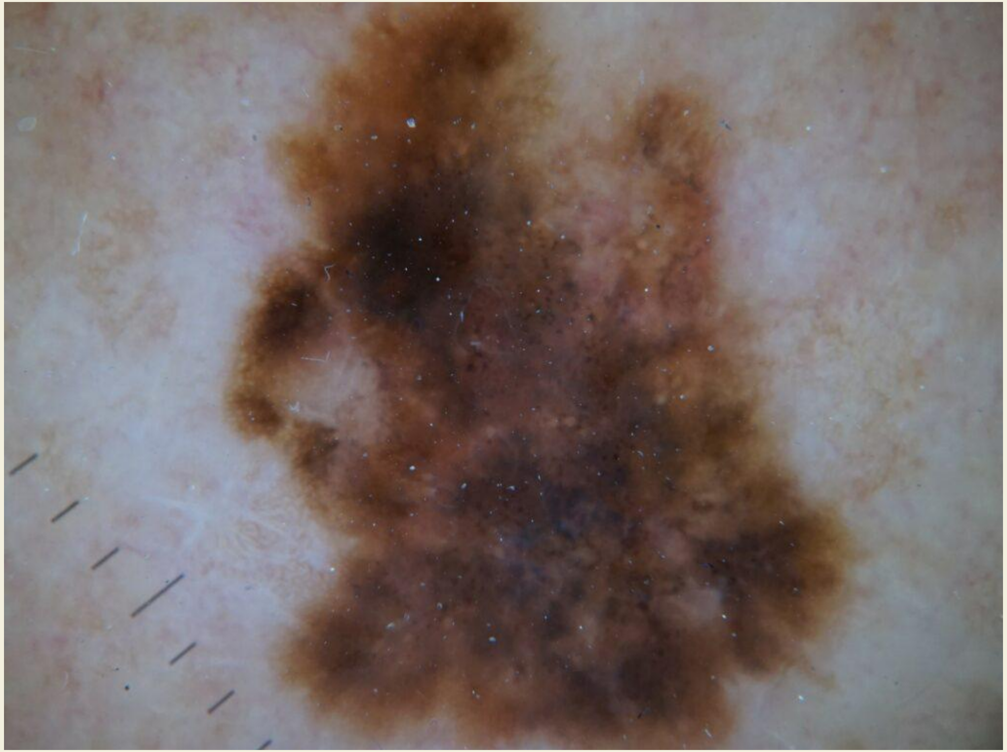
- Early detection is IMPORTANT!
- When melanoma is detected early, its 5-year survival rate is approximately 99%.
- This rate drops dramatically to 74% if the malignancy to nearby lymph nodes and to a reported 35% if it is allowed metastasizes to distant organs.

1. Is this needed? Maybe

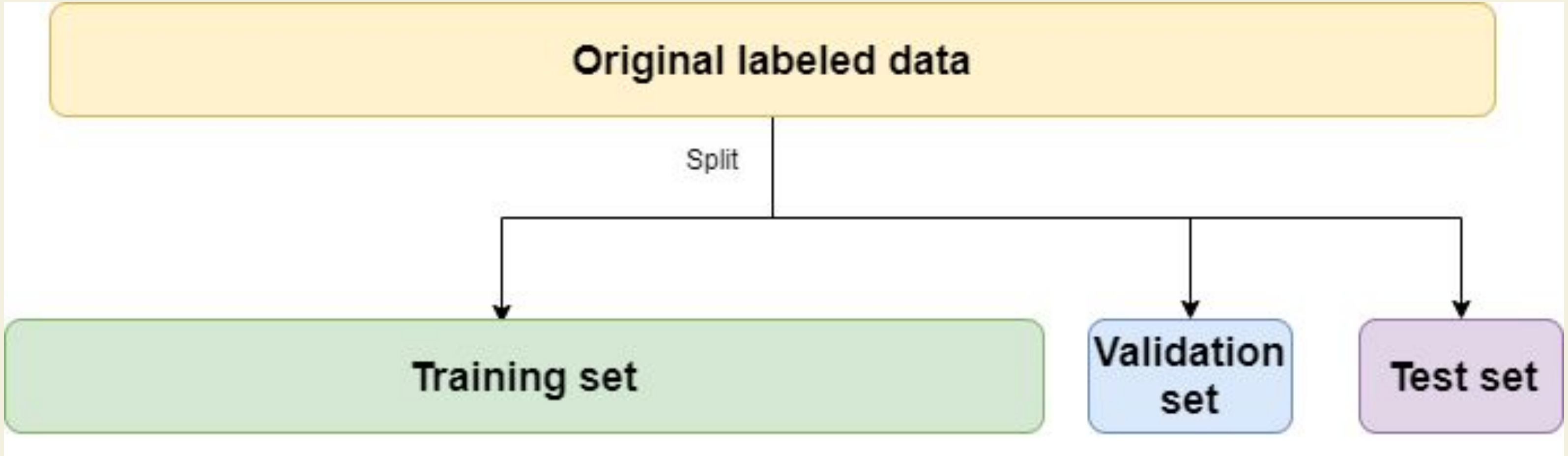
- Sensitivity and Specificity using clinical examination and images of melanoma were 76.9% and 89.1% for experienced dermatologists
- Compared with 78.3% and 66.2% for inexperienced dermatologists
- 37.5% and 84.6% for primary care physicians (PCPs)

- Using in-person dermoscopy and dermoscopic images, they were 85.7% and 81.3% for experienced dermatologists, 78.0% and 69.5% for inexperienced dermatologists, and 49.5% and an impressive 91.3% for PCPs.


Sklip
dermatoscope



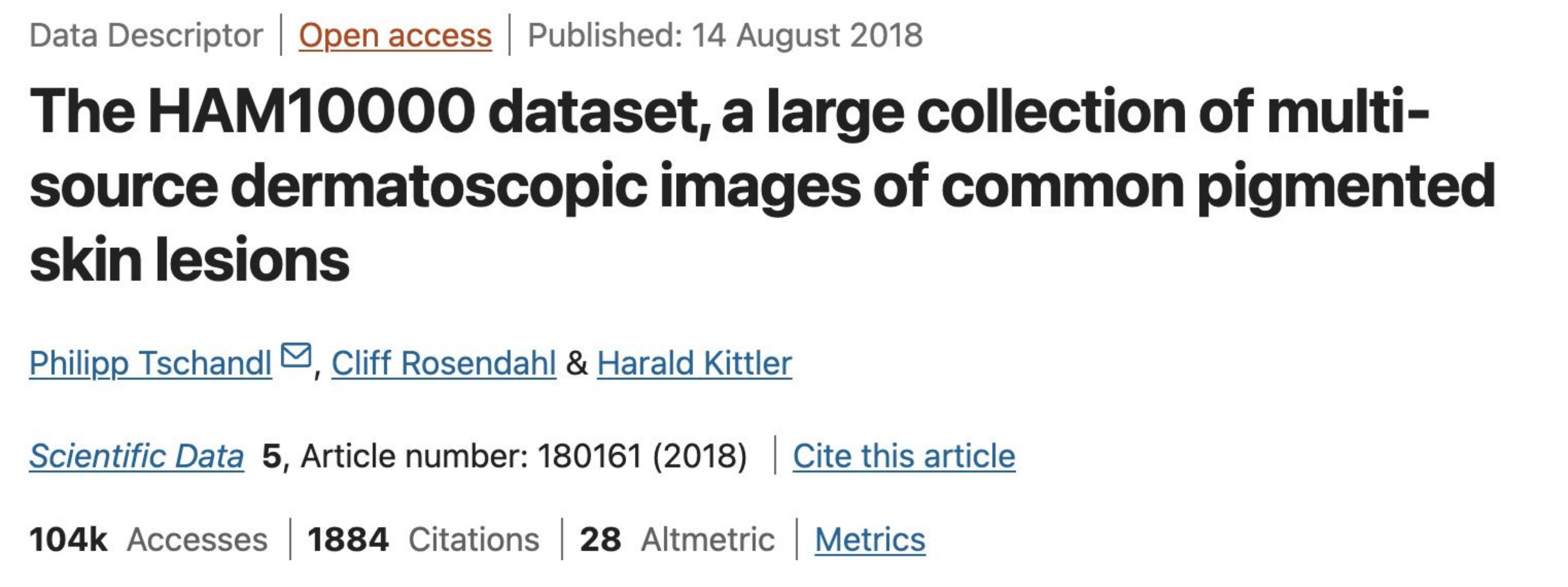
1. How exactly do you do this?



1. What is a validation dataset?

- The validation set helps determine how well a model generalizes to new, unseen data, and it's used to make adjustments to the model's parameters or architecture.
- Just how you can tell your brain to focus on certain things or avoid certain things, you can do the same with CNN with “hyperparameter tuning”
  - More on this to follow

1. Where did I get my data?



Data Descriptor | Open access | Published: 14 August 2018

The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions

Philipp Tschandl ✉, Cliff Rosendahl & Harald Kittler

Scientific Data **5**, Article number: 180161 (2018) | Cite this article

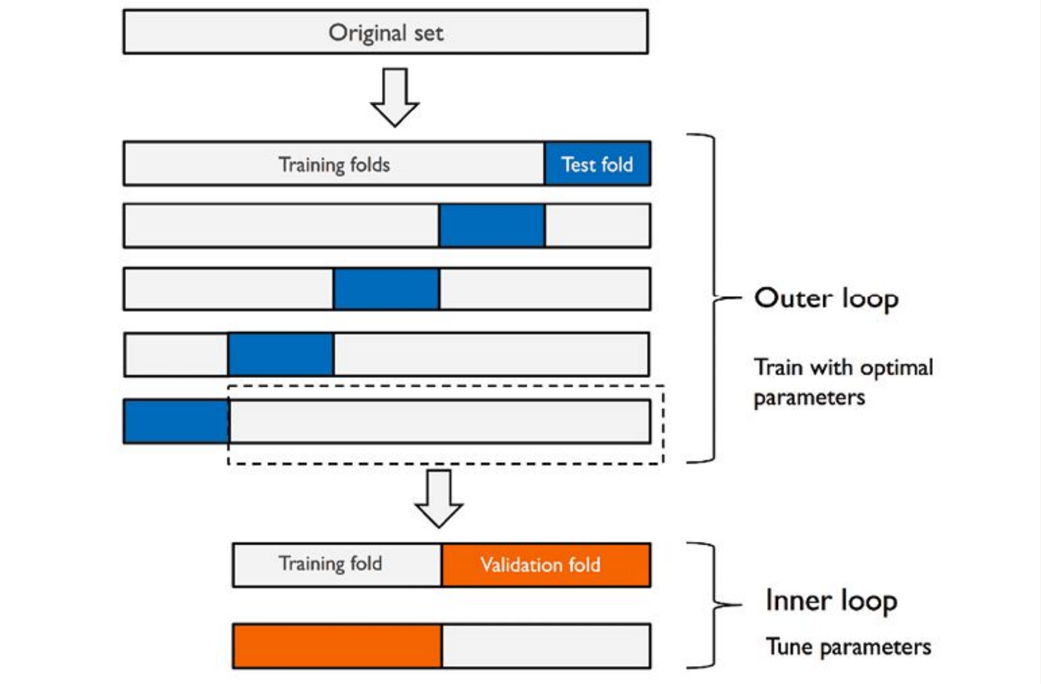
104k Accesses | **1884** Citations | **28** Altmetric | Metrics

1. Did I do Train/Validate/Test?

- **No**
- High variance in performance estimates: Results depend heavily on the specific data points that end up in each set
- Inefficient use of data: Only a portion of data is used for training
- Selection bias: A single random split may not be representative of the overall data distribution

1. Nested Cross-Validation

- Uses two nested loops of cross-validation
- Outer loop: Estimates generalization performance
- Inner loop: Performs hyperparameter optimization



1. Train/Test vs Cross Validation vs Nested Cross Validation

- Simple Train/Test Split
    - This is like having one practice game before the championship. You select players and develop a training program based on their performance in practice, then take them straight to the championship game. You might get lucky, but it's a risky approach!

1. Train/Test vs Cross Validation vs Nested Cross Validation

Standard Cross-Validation

- This is like having a series of practice games. You try different combinations of players across multiple games, getting a more reliable sense of who performs well consistently. But you're using the same training program for everyone.

1. Train/Test vs Cross Validation vs Nested Cross Validation

Nested Cross-Validation

- This is like running a complete tournament system:

Outer Loop (Tournament Structure):

- Imagine splitting your candidate players into 5 groups (outer fold). You'll run 5 separate mini-tournaments (inner fold), and each time one group sits out as the "final judges."

Inner Loop (Training Program Selection):

- Within each mini-tournament, you try different training programs (hyperparameters). For each program, you conduct several practice games, rotating which players participate. You select the training program that performs best overall in that mini-tournament (inner fold).

1. Train/Test vs Cross Validation vs Nested Cross Validation

Final Evaluation:

- After each mini-tournament selects its best training program, you test it against the group that sat out (who the program has never seen before).
- You do this five times, getting five independent measurements (outer fold) of how well your player selection and training program development process works.
- The advantage of nested cross-validation is that it tests not just your final team, but your entire team-building process.
- It helps you understand how reliable your player selection and training program development methodology is, rather than just evaluating a single team that might have gotten lucky in one practice game.

44
bibigo
비비고

1. What are hyperparameters?

- Hyperparameters are settings that determine how a machine learning algorithm learns, such as the number of layers in a neural network, the learning rate, or the regularization strength.
- Purpose:
  - They influence the model's architecture, learning speed, and overall performance.
- Set before training: They cannot be learned from the data
- Control the learning process: They influence how the model learns rather than what it learns
- Require tuning: Finding optimal values often involves experimentation

1. What are hyperparameters?

- Learning Rate: The step size taken when moving toward a solution. Too large causes overshooting; too small leads to slow learning. Like adjusting the speed of a car navigating to a destination.
- Number of Iterations/Epochs: How many times the algorithm processes the entire dataset. Too few means incomplete learning; too many can lead to memorization instead of generalization. Similar to deciding how many times to review material before an exam.
- Regularization Parameters: Controls that prevent models from becoming unnecessarily complex. Stronger regularization produces simpler models that generalize better but might miss subtle patterns. Like teaching someone to focus on core principles rather than memorizing exceptions.

1. How do you tune hyperparameters?

1. Manual tuning: Based on intuition, experience, and trial-and-error
2. Grid search: Exhaustively trying all combinations from a predefined set
3. Random search: Sampling random combinations, often more efficient than grid search
4. Bayesian optimization: Using past evaluations to guide the search process

1. Bayesian Optimization:

1. Surrogate Model: Typically uses Gaussian Processes to model the relationship between hyperparameters and model performance
2. Acquisition Function: Determines which points to sample next by balancing:
   a. Exploitation: Sampling where the model predicts high performance
   b. Exploration: Sampling uncertain regions to improve the surrogate model
3. Sequential Process: After each evaluation, the surrogate model is updated with the new information

1. Truffle Hunting: Bayesian Analogy

I am looking for truffles in a really really big forest

   a. Luckily, I have a truffle sniffing dog with me

2. Initial Exploration: Your dog sniffs at a few random spots in the forest.
3. Building a Mental Map: After each dig, your dog builds a more refined understanding of where truffles are likely to be found. It notices patterns like "truffles seem to grow near oak trees on north-facing slopes."
4. Smart Selection: Instead of checking everywhere, your dog considers both:

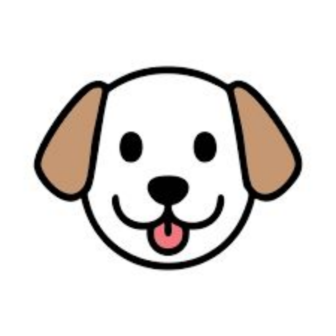
   a. Places that strongly match the pattern of previous finds (exploitation)

   b. Unexplored areas that might reveal new patterns (exploration)

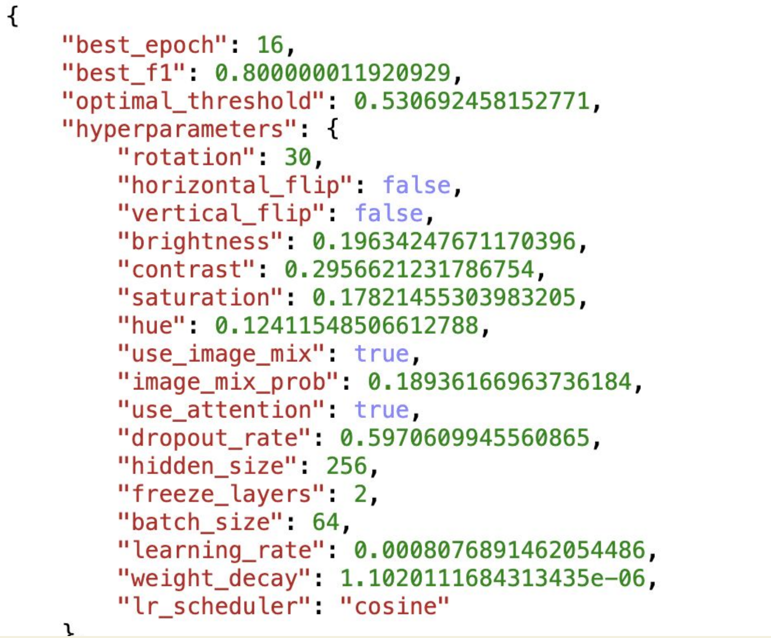
1. Truffle Hunting: Bayesian Analogy

5. Continuous Learning: With each new truffle you find (or don't find), your dog updates its mental map and makes increasingly better predictions about where to look next.

6. Diminishing Returns: As you find more truffles, your dog's understanding becomes more precise, and you spend less time digging in unpromising locations.

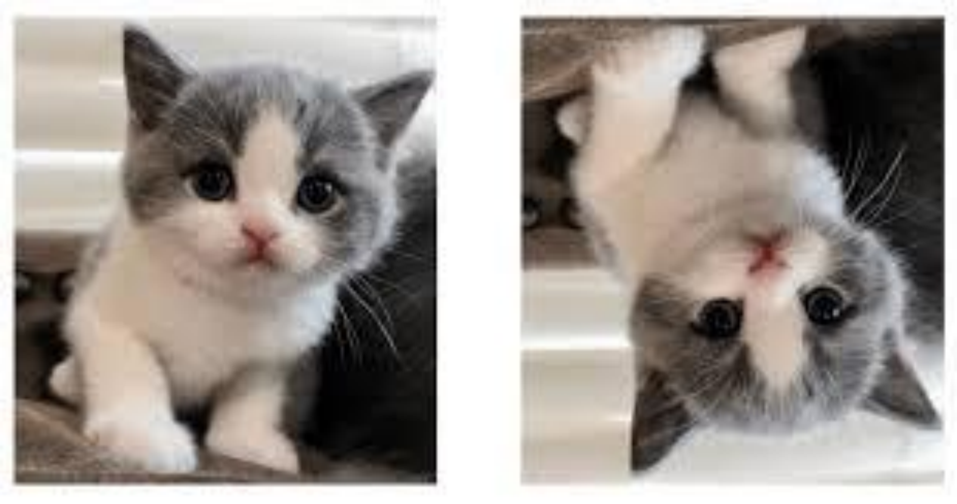


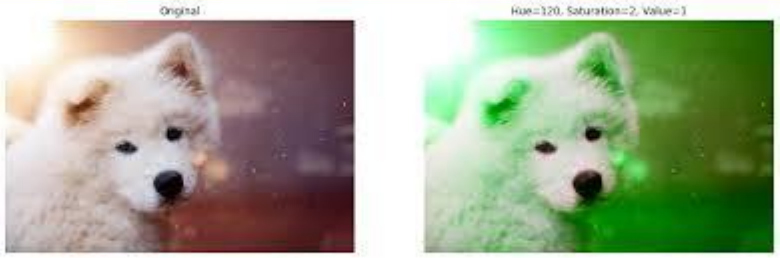
1. What hyperparameters did I tune?



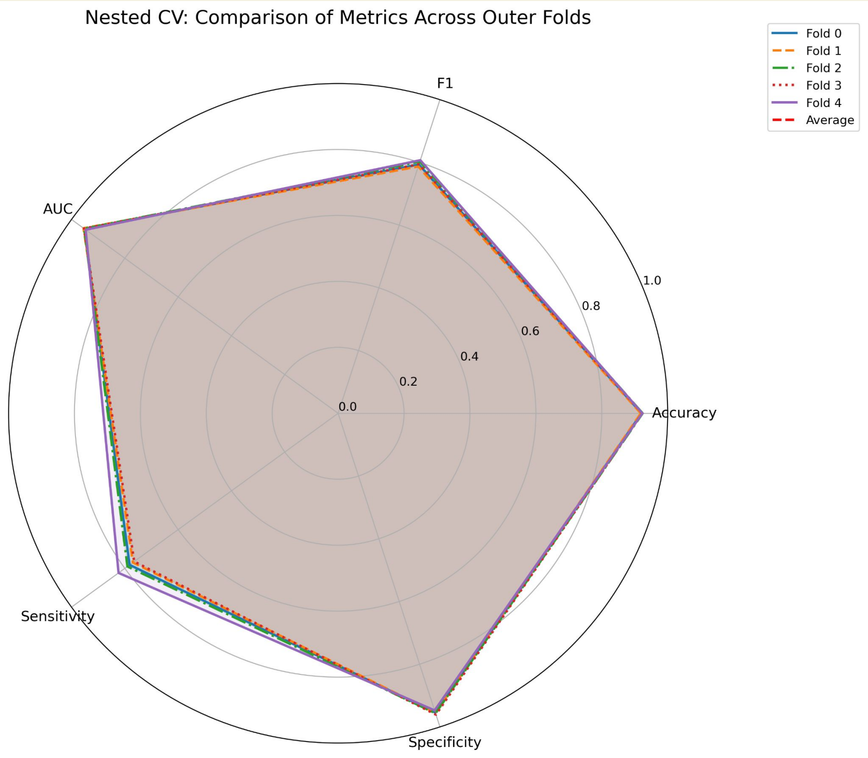
```json
{
    "best_epoch": 16,
    "best_f1": 0.800000011920929,
    "optimal_threshold": 0.530692458152771,
    "hyperparameters": {
        "rotation": 30,
        "horizontal_flip": false,
        "vertical_flip": false,
        "brightness": 0.19634247671170396,
        "contrast": 0.2956621231786754,
        "saturation": 0.17821455303983205,
        "hue": 0.12411548506612788,
        "use_image_mix": true,
        "image_mix_prob": 0.18936166963736184,
        "use_attention": true,
        "dropout_rate": 0.5970609945560865,
        "hidden_size": 256,
        "freeze_layers": 2,
        "batch_size": 64,
        "learning_rate": 0.0008076891462054486,
        "weight_decay": 1.1020111684313435e-06,
        "lr_scheduler": "cosine"
    }
```

1. Examples of Data Augmentation

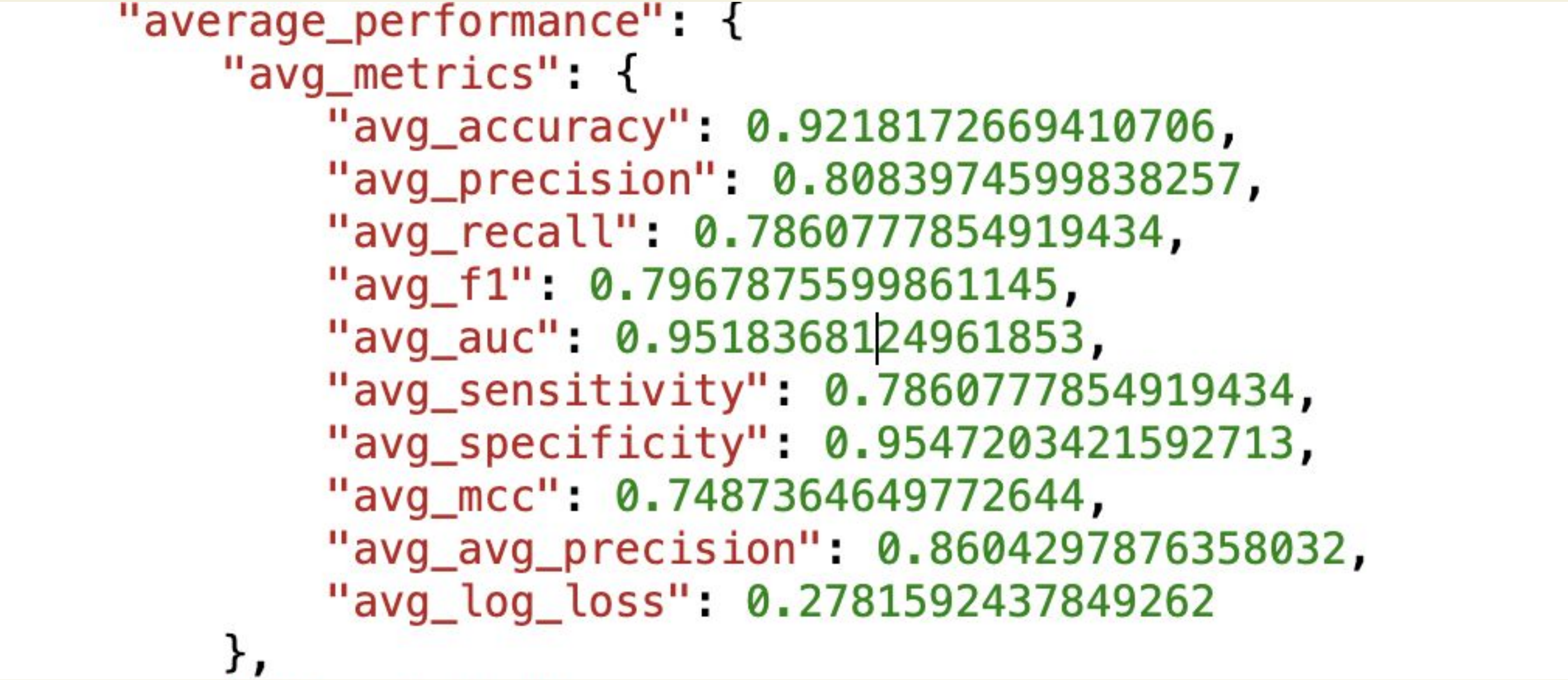




1. Results?

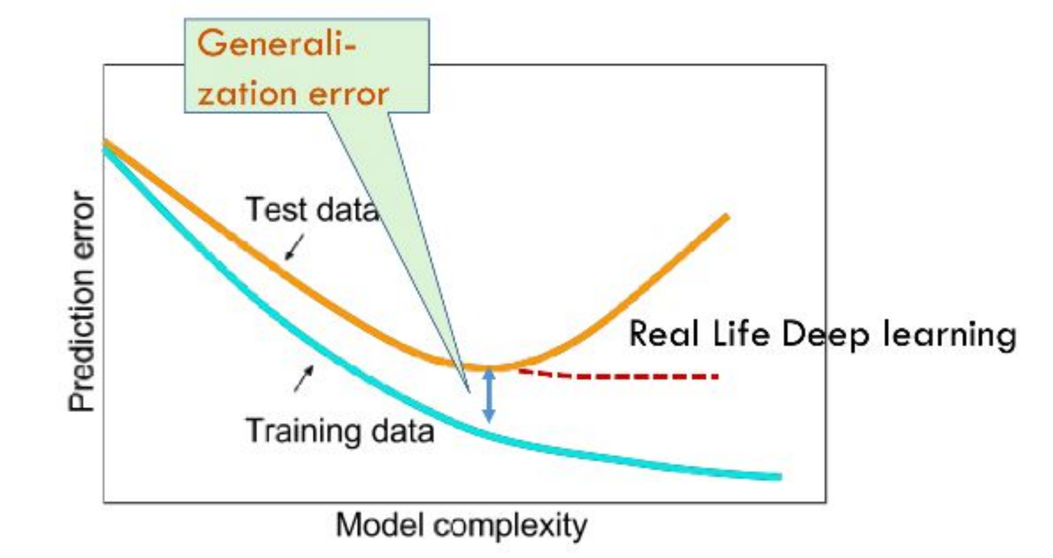

Nested CV: Comparison of Metrics Across Outer Folds
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Average
F1
AUC
Accuracy
Sensitivity
Specificity
0.0
0.2
0.4
0.6
0.8
1.0

1. Results?



```
"average_performance": {
    "avg_metrics": {
        "avg_accuracy": 0.9218172669410706,
        "avg_precision": 0.8083974599838257,
        "avg_recall": 0.7860777854919434,
        "avg_f1": 0.796787559986145,
        "avg_auc": 0.9518368124961853,
        "avg_sensitivity": 0.7860777854919434,
        "avg_specificity": 0.954720342159271З,
        "avg_mcc": 0.748736464977264,
        "avg_avg_precision": 0.8604297876358032,
        "avg_log_loss": 0.2781592437849262
    },
```

1. The issue with AI

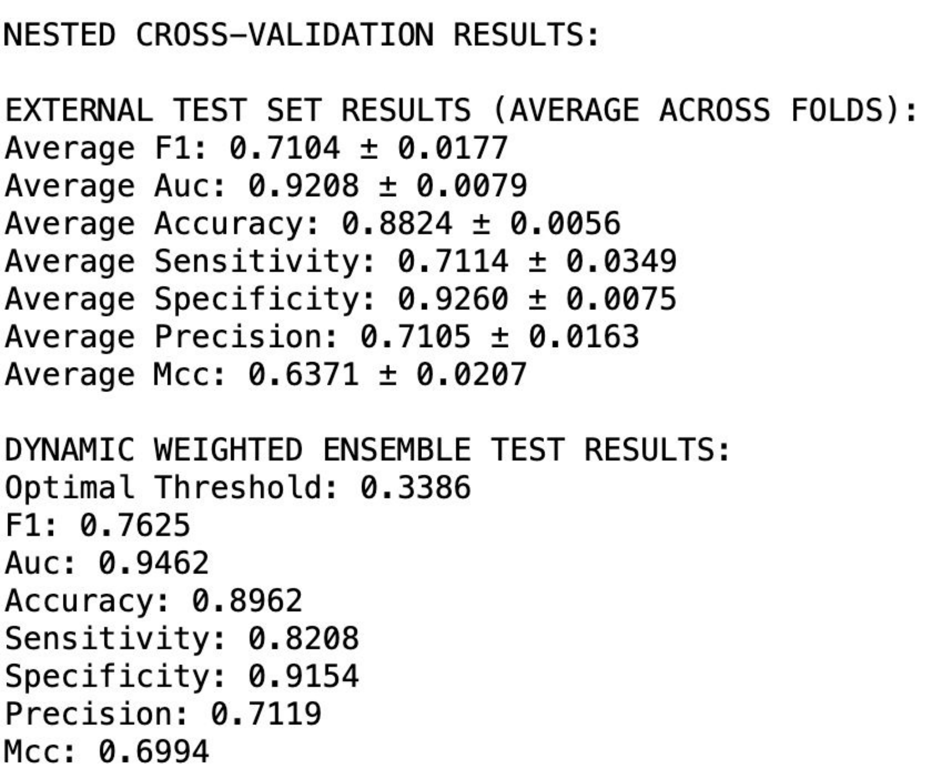


1. An external test set

- External test sets reveal whether a model has truly learned to recognize patterns or just memorized quirks in the training data.
- By testing on completely new images collected differently than the training set, we can discover if the model will actually work in real-world situations or if it's just good at one specific dataset.

MSK Dataset: (c) Anonymous; https://arxiv.org/abs/1710.05006; https://arxiv.org/abs/1902.03368

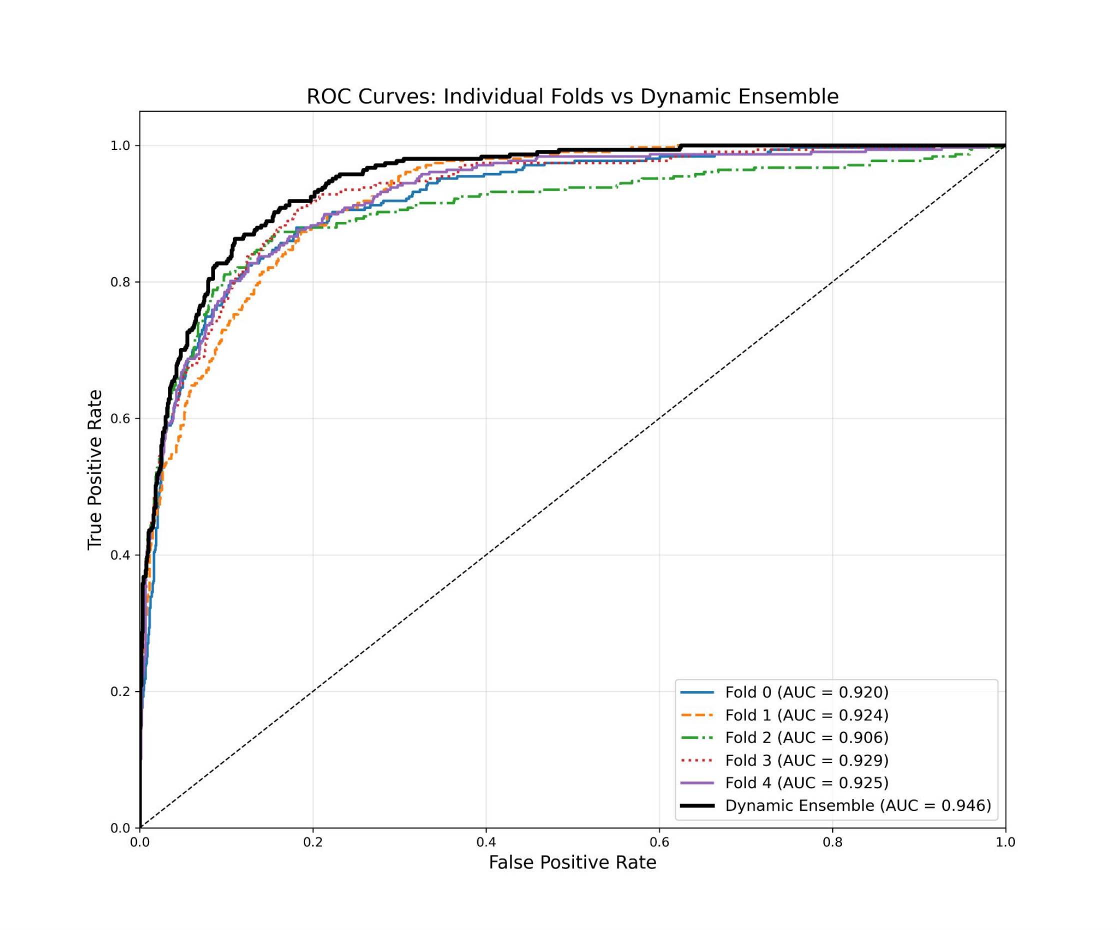
1. Results?

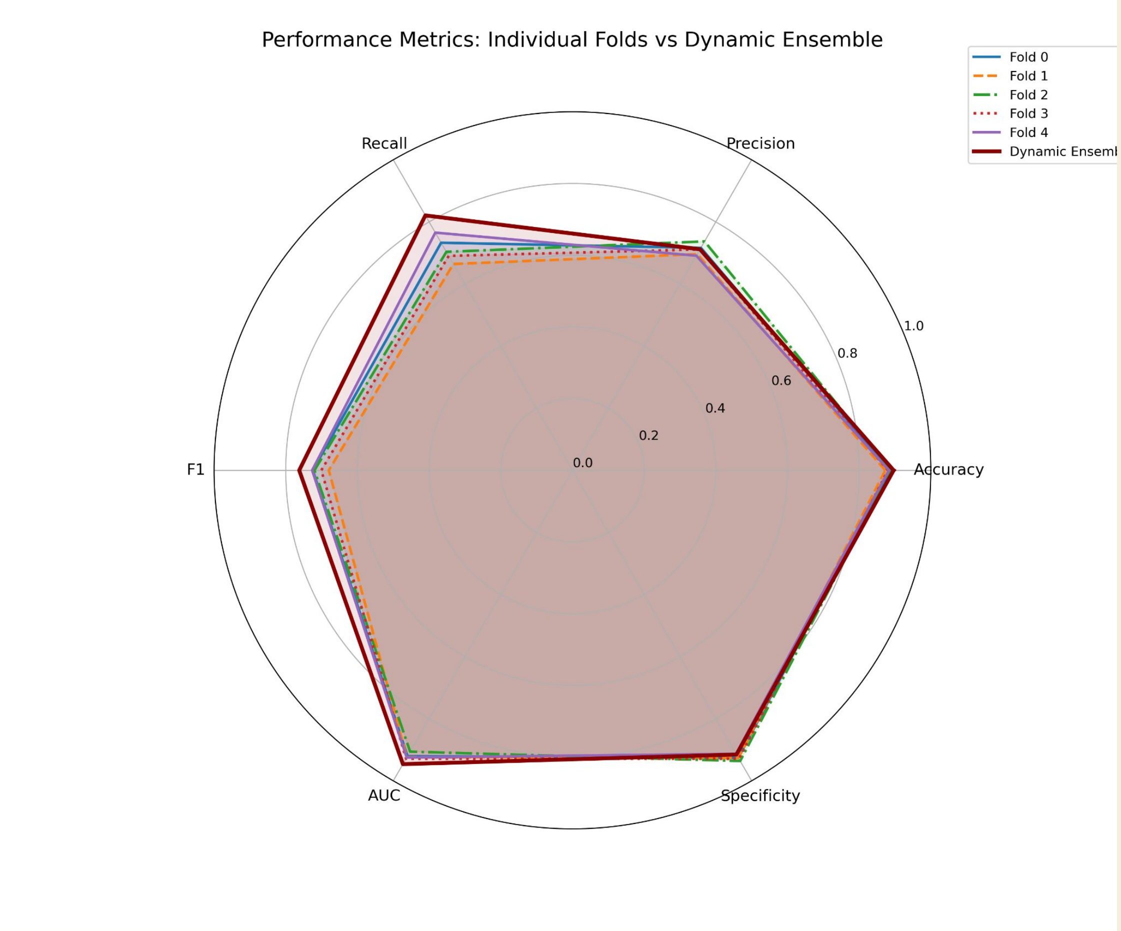


```
NESTED CROSS-VALIDATION RESULTS:

EXTERNAL TEST SET RESULTS (AVERAGE ACROSS FOLDS):
Average F1: 0.7104 ± 0.0177
Average Auc: 0.9208 ± 0.0079
Average Accuracy: 0.8824 ± 0.0056
Average Sensitivity: 0.7114 ± 0.0349
Average Specificity: 0.9260 ± 0.0075
Average Precision: 0.7105 ± 0.0163
Average Mcc: 0.6371 ± 0.0207

DYNAMIC WEIGHTED ENSEMBLE TEST RESULTS:
Optimal Threshold: 0.3386
F1: 0.7625
Auc: 0.9462
Accuracy: 0.8962
Sensitivity: 0.8208
Specificity: 0.9154
Precision: 0.7119
Mcc: 0.6994
```


ROC Curves: Individual Folds vs Dynamic Ensemble
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Fold 0 (AUC = 0.920)
Fold 1 (AUC = 0.924)
Fold 2 (AUC = 0.906)
Fold 3 (AUC = 0.929)
Fold 4 (AUC = 0.925)
Dynamic Ensemble (AUC = 0.946)



Performance Metrics: Individual Folds vs Dynamic Ensemble
Fold 0
Fold 1
Fold 2
Fold 3
Fold 4
Dynamic Ensembl
Recall
Precision
F1
Accuracy
AUC
Specificity
0.0
0.2
0.4
0.6
0.8
1.0

Binary Classification of Dermoscopic Lesions via Bayesian-Optimized Ensemble Convolution Neural Networks

That's all Folks!