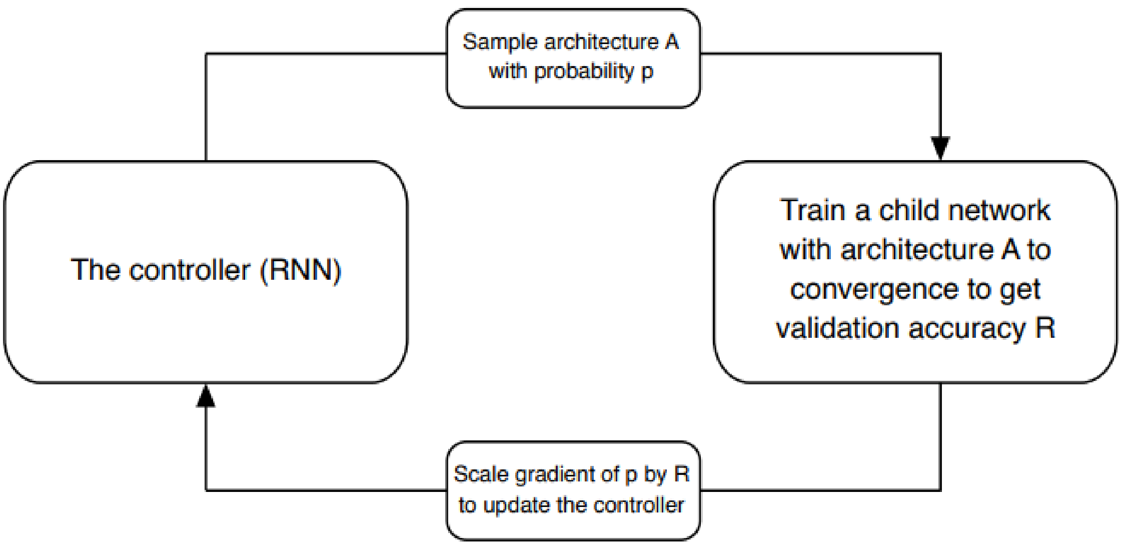
DATA SCIENCE - MA 305

CLASSIFICATION OF COVID19 X-RAY IMAGES USING NasNetMobile

Report by : TEAM 7
Ishita Saxena (230041011)
Rishabh Anand (230041029)
Pranab Mishra (230041027)

Overall Objective

The primary goal is to use a Recurrent Neural Network (RNN) as a **controller** to automatically find an optimal convolutional cell architecture. This controller is trained with a **reinforcement learning (RL)** algorithm to maximize the accuracy (the **reward**) of the architectures it generates. The best-found cell architecture, learned on a small dataset (CIFAR-10), is then transferred to a large-scale dataset (ImageNet) by stacking it to form a full-sized "NASNet" model.



Step 1: The NASNet Search Space

The controller does not design an entire network. Instead, it designs the detailed structure of two repeating building blocks, or "cells" :

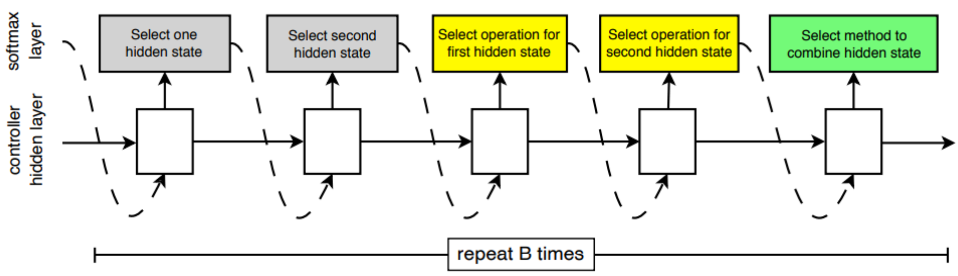
1. **Normal Cell:** A convolutional cell that takes a feature map as input and returns a feature map of the **same spatial dimensions**.
2. **Reduction Cell:** A cell that returns a feature map with its **height and width reduced by a factor of two** (acting like a pooling layer or a strided convolution) .

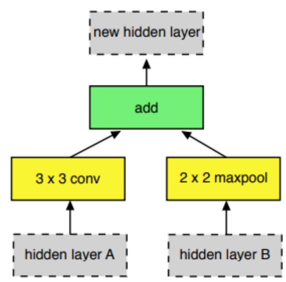
Each cell is a small directed acyclic graph (DAG) composed of **B blocks** (the paper uses B=5). Each block takes two inputs, applies a different operation to each, and combines them.

For each of the B blocks, the controller RNN must make **5 discrete predictions** from a set of choices :

- **Prediction 1 (Input A):** Select a hidden state from the previous B-1 blocks or the two initial cell inputs.
- **Prediction 2 (Operation for A):** Select an operation from a set (e.g., 3x3 separable conv, 5x5 max pool, identity, 1x1 conv, etc.).
- **Prediction 3 (Input B):** Select a hidden state (same choices as #1).
- **Prediction 4 (Operation for B):** Select an operation from the same set.
- **Prediction 5 (Combination):** Select a method to combine the outputs of A and B (e.g., element-wise addition, concatenation).

The output of the cell is the concatenation of all B block outputs. This design, repeated for both a Normal and a Reduction cell, defines the complete search space.





Set of operations to be applied to hidden states-

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv

Step 2: The Controller as a Policy Network

The controller is an **LSTM (Long Short-Term Memory) RNN** . It acts as a *policy network* π with trainable parameters (weights) θ_c.

- **Action Generation:** The controller autoregressively predicts the sequence of 5 choices for each of the `B` blocks, for both the Normal and Reduction cells. For a cell with `B=5` blocks, this is `2*(5*5) = 50` discrete predictions in total.
- **Output Mechanism:** At each prediction step, the LSTM's output is fed into a **softmax classifier** to produce a probability distribution over the possible discrete choices (e.g., the probability of picking a 3x3 conv vs. a 5x5 pool).
- **Action Sampling:** The controller *samples* from this probability distribution to make a choice. The sequence of all sampled choices $a_{1:T}$ (where `T=50`) defines one complete child architecture.

Step 3: Training with Reinforcement Learning (PPO)

This is the core mathematical process used to train the controller.

1. **Sample and Run:** The controller, using its current policy samples a child architecture `a`.

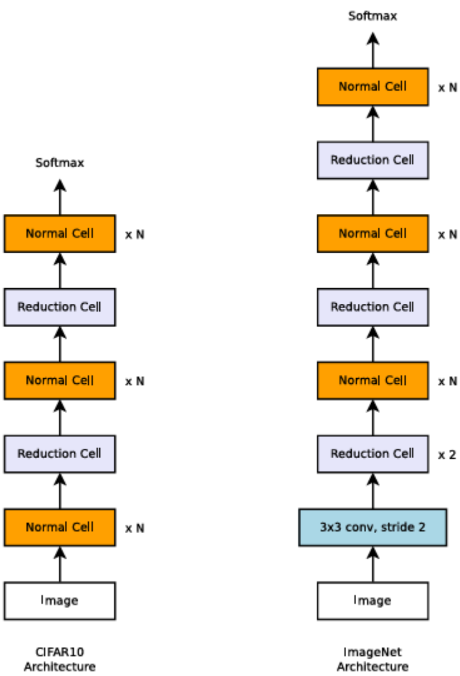
2. **Get Reward:** This child network is built and trained on the CIFAR-10 dataset for a fixed number of epochs. Its final validation accuracy is recorded as the **Reward (R)** .
3. **Objective Function:** The controller's goal is to maximize its Expected Reward over all possible architectures it can generate.
4. **Policy Gradient:** To maximize this, the paper uses the **REINFORCE** algorithm, which updates the controller's parameters in the direction that increases the reward . The gradient is: Update = (How Good Was This Design?) * (What Decisions Did I Make?)

This process is repeated thousands of times (the paper used 12,800 architectures), with the controller's policy θ_c iteratively improving to find cells that yield higher and higher accuracy.

Step 4: Transfer and Scalability

Once the RL search on CIFAR-10 is complete, the single best cell architecture (named **NASNet-A**, **-B**, or **-C**) is saved.

- **Final Network Construction:** A full-size network for ImageNet is built by stacking these exact same cells in a predefined pattern (e.g., N Normal Cells, 1 Reduction Cell, N Normal Cells, 1 Reduction Cell, etc.).



- **Scaling:** The authors create different model sizes (like **NASNet-Mobile** or the large NASNet-A) by varying two main factors:
    1. **N:** The number of times the Normal Cell is repeated in each stack.
    2. **F:** The number of filters in the initial "stem" and subsequent cells.
- **Final Training:** This final, large NASNet is then trained from scratch on the ImageNet dataset for a full training run (no more RL or controller involved).

Model	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V1 [59]	6.6M	1,448 M	69.8 †	89.9
MobileNet-224 [24]	4.2 M	569 M	70.6	89.5
ShuffleNet (2x) [70]	~ 5M	524 M	70.9	89.8
NASNet-A (4 @ 1056)	**5.3 M**	**564 M**	**74.0**	**91.6**
NASNet-B (4 @ 1536)	5.3M	488 M	72.8	91.3
NASNet-C (3 @ 960)	4.9M	558 M	72.5	91.0

OUR APPROACH

1. Introduction

Chest X-ray (CXR) imaging is widely used for screening and diagnosis of respiratory diseases, including COVID-19. However, medical X-ray images frequently suffer from low contrast, noise and intensity variation, which can make learning more difficult.

Image enhancement methods can potentially improve lung region visibility and boost classification performance.

We have studied the effect of three enhancement techniques on COVID-19 chest X-ray classification performance using NASNetMobile:

- **Histogram Equalization (HE)**
- **CLAHE (Contrast Limited Adaptive Histogram Equalization)**

- **Image Complement (negative transformation)**

2. Dataset

- **Dataset**: COVIDx CXR-2
- **Train images**:
    - Positive Label : 57199
    - Negative Label : 10664
    - After Random Oversampling total : 57199 + 57199 = 114,398
- **Test images**: 4241 + 4241 = 8482
- **Train/Validation split**: 80/20

{**Handling Class Imbalance -** The dataset was imbalanced, with more negative cases. We applied **oversampling** to minority class. Additionally, class weights were computed dynamically to ensure balanced gradient contributions.}

3. Methodology

3.1 Image Enhancement Pipeline

A) Histogram Equalization

- **Effect:** Increases overall contrast, making lung structures more visible across the whole image.

- **How:** Redistributes pixel intensity values so darker and lighter regions spread more evenly.

B) CLAHE

- **Effect:** Enhances fine details and textures in local lung regions without amplifying noise too much.

- **How:** Applies histogram equalization in small tiles and limits contrast boost using a clipping threshold.

C) Image Complement (Negative Transformation)

- **Effect:** Highlights soft tissues and abnormal opacities by flipping brightness patterns.

- **How:** Inverts pixel values (new = 255 − old), turning light areas dark and vice-versa.

Combined Strategy:
Histogram Equalization → CLAHE → Complement → NASNet Preprocess

Data Augmentation:
Rotation, zoom, translation, horizontal flip augmentations added to combat overfitting.

3.2 Model Architecture

Model: **NASNetMobile** (1.4M parameters), pretrained on ImageNet
Additional Layers:

1. Global Average Pooling
2. Dense layer (512 units, ReLU)
3. Dropout (0.5)
4. Final sigmoid output layer (binary classification)

3.2 Training Strategy

- Image input size : 224 * 224* 3

- Stage 1 (Feature extraction) - Train with backbone frozen (8 epochs, LR=1e-4)
  Stage 2 (Fine tuning) - Unfreeze last 100 layers, fine-tune (5 epochs, LR=1e-5)

- Loss: Binary Cross-Entropy
  Optimizer: Adam
  Metrics: Accuracy

4. Results

Training Performance

- Validation accuracy improved to **91.63%**
- Validation loss decreased consistently during fine-tuning

5. Discussion

- Robust enhancement pipeline highlights lung regions
- High COVID-positive recall (clinically useful for screening)

6. Conclusion

This project investigated how contrast enhancement techniques help COVID-19 CXR classification using NASNetMobile.

7. Future Enhancements

- **Lung segmentation (U-Net/DeepLabv3+)** to isolate lung regions before classification, reducing background noise and focusing the model on pathology-relevant areas.
- **Ensemble models** (NASNet + EfficientNet/DenseNet) for more robust prediction.
- **Advanced loss functions** (Focal Loss, weighted BCE) to better handle class imbalance.
- **Explainability** using Grad-CAM to verify the model focuses on lung abnormalities.
- **Cross-dataset validation** to improve generalization and real-world reliability.