

# Deep Learning Project 1

Spring 2025

**Competitions end 11:59pm, March 13, 2025.**

**Please upload your project report before 11:59pm, March 14, 2025.**

This project will comprise 15% of your overall grade. Please perform this project in groups of (at most) 3. If you are looking for project team members, please broadcast your interest in the Class Slack in the `find-a-teammate` channel.

## Goal

In this Kaggle competition you are tasked with coming up with a modified residual network (ResNet) architecture with the **highest test accuracy** on a held-out portion of a version of the CIFAR-10 image classification dataset, under the constraint that your model has **no more than 5 million parameters**.

## Details

Recall that a residual network (ResNet) architecture is any convolutional network with *skipped* connections. Here is a picture of ResNet-18:

The key component in ResNet models is a *residual block* that implements:

$$\text{ReLU}(S(x) + F(x))$$

where  $S(x)$  refers to the skipped connection and  $F(x)$  is a *block* that implements `conv -> BN -> relu -> conv -> BN`; here, “BN” stands for batch normalization. Chaining such blocks layer by layer gives a deep ResNet.

Hyperparameters (design variables) in such architectures include:

- $C_i$ , the number of channels in the  $i$ th layer.
- $F_i$ , the filter size in the  $i$ th layer
- $K_i$ , the kernel size in the  $i$ th skip connection
- $P$ , the pool size in the average pool layer,

etc.

You are free to experiment around and adjust these parameters to gain boosts in accuracy, as long as the *total* number of trainable parameters does not exceed 5 million. (You can

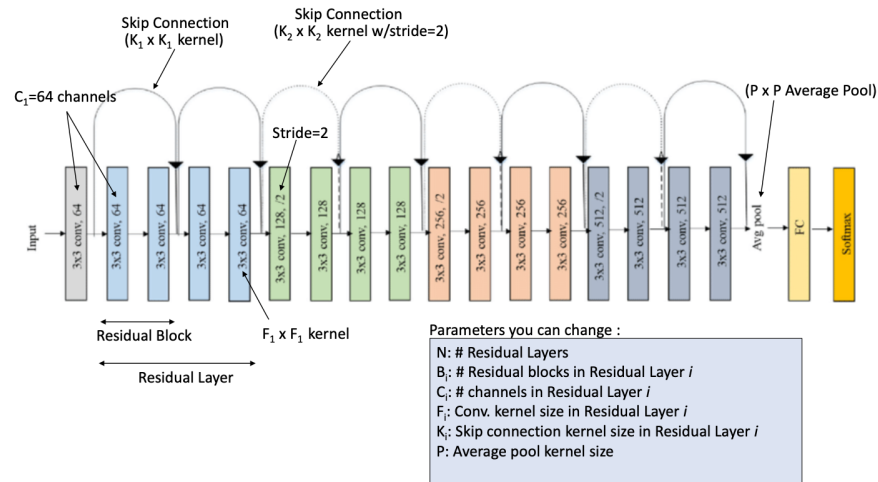


Figure 1: Block diagram of Resnet-18 model

use the `torchsummary.summary` function to check the number of parameters in your model.)

You are also free to experiment with:

- any optimizer (SGD, ADAM, RMSProp, etc)
- any data augmentation strategy
- any regularizer
- any choice of learning rate, batch size, epochs, scheduler, etc.

You are also free to use other tricks such as teacher-student distillation, or quantization, or etc.

You are **not** allowed to:

- simply load pre-trained model weights from the web, and fine-tune on the CIFAR-10 You **have** to be able to train your model from scratch.
- use other/bigger datasets such as ImageNet.

## Resources (optional)

This repository has excellent PyTorch code for training various ResNet models on CIFAR-10 from scratch. If you do use any parts of this repository, please include a clear citation. You are free to use any other online resources and/or techniques you like, as long as you include citations.

You can also use NYU HPC.

## Deliverables

This project has two main deliverables:

- A project report (7 points).
- A project codebase (7 points) in the form of a Github repository.

Teams finishing within the top 20% of the teams in the competition will be given 1 point,

There is a **bonus** 2 points for the top team, and a **bonus** 1 point for the runner-up team.

Projects will be graded on:

- clarity and quality of submitted project report;
- quality of final results. Aim for test accuracy of at least 80% as a minimum baseline; if your design is sensible, you should be able to achieve upwards of 90%;
- and clarity and quality of submitted codebase. Include notebooks with clear plots; in particular, we want to see your code execution. Include statements where you clearly print the final test accuracy and number of parameters.

## Project report

Your report has to be **no more than 4 pages long** including all figures, tables, and citations, **typeset in two-column AAAI 2024 camera-ready format**. Any report not in this format will not be graded. Here is a link to the format in LaTeX and MSWord formats; see the “Camera-ready” folder in this link for example documents.

Please upload a PDF of your report to Gradescope. **Only one team member should upload** on Gradescope, as long as they tag all other team members.

In your report, please include:

- Names of all team members
- A short overview of your project, along with a summary of your findings
- A methodology section that explains how you went about designing and training your models, pros and cons of different architectural choices, what lessons you learned during the design process, etc.
- A results section that reports your final test accuracy, model architecture, and number of parameters.
- Any relevant citations.

## Project codebase

In the first page of your report, please provide a link **on the first page** to a **publicly accessible** Github repository. Your repository should contain:

- the code necessary to reproduce the results in your report.
- well-documented code and/or Jupyter notebooks for easy visualization and verification of your work.

Once the competition is complete the TAs will reach out to the top performing teams to reproduce your code to make sure your results are valid. Therefore, readability and usability of your code is essential; please prioritize this.