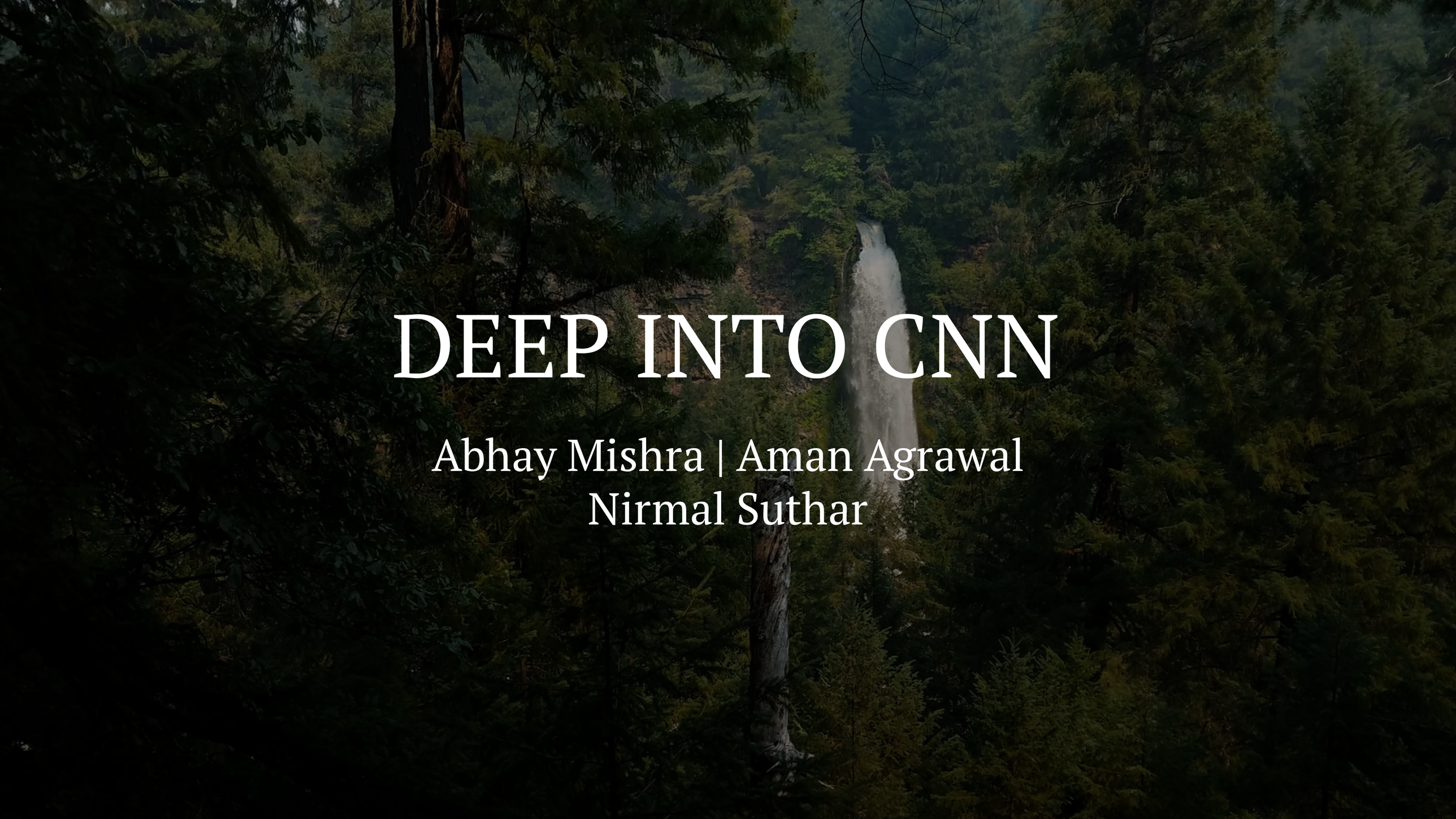

DEEP INTO CNN

Abhay Mishra | Aman Agrawal
Nirmal Suthar

GOAL

- Introduce different types of Neural networks.
  - MLP
  - CNN
- Why do we need different Networks?:
  - Ineffectiveness of Existing Ones, New Needs
  - Solution Architecture
  - Effectiveness of Proposed Network , Performance Comparison
- 2 Competitions
  - One on Numerical Data and other on Image Data
- 2 Paper Implementation

PRE-REQS AND TIME REQS

- Googling Skills
- Python, Basic Linear Algebra, Differentiation (MTH 102) : Matrix Multiplication, Calculus etc.
- Deep Learning by Andrew Ng , Coursera : Week 6 : Familiarity With linear and logistic Regression, Minimal familiarity with ML terminology
- 8-10 hours/week
- Reading material : Notes , Articles, Documentation
- Practice Material : Theory(Minimal), Programming: Colab Tutorials : NO SUBMISSION

MINIMAL

- Regression And Shallow NN Using Python - *Week 1-2*

  Hackathon 1 : Classification On Numerical Data
- NN With Pytorch - *Week 2*
- Intro To CNN - *Week 2-3*

  Hackathon 1 Ends

  Hackathon 2: Classification On RGB-images

  Paper 1: Alexnet/VGG/Inception/Xception : Choose One You Want
- Optimization Methods - *Week 4-5*

  Hackathon 2 Ends

  Submit Paper 1 Implementation
- GAN + Autoencoders/Any Advanced Concept - *Week 5-6*

  (~~₹20~~) Paper 2: GAN/Any Advanced Paper
- Collaborate With Others For Implementing More Architecture

  Submit Paper 2 at the end of project

RATIFICATION

- At Least 2 different submissions in each Hackathon:
  - Hackathon 1 : Regression,NN
  - Hackathon 2: Baseline CNN model , LeNet (Or any Other deep architecture),Transfer learning :
- 2 Paper Implementation for Y19 and 1 for Y20 (Optional)
- 3 Github Repo: All your work (Practice + Theory/Notes) : As a backup : If you fail to complete the above two show us your previous work.

ROADMAP

1. Setup
   - Colab,kaggle(Recommended)
     - Training Large Models, Free GPU
     - kaggle api for datasets
   - Local System : Anaconda
     - Conda to install packages and manage environment
     - Recommended for low training requirements and Good laptops

ROADMAP

CONTINUED

2. Numerical data : Multi layer Perceptron (MLP) : Week 1-2
   - Regression : Week 1 : : Python Implementation
     - Gradient Descent,relu layer, MSE loss
     - Binary Classification,sigmoid layer,BCE loss
     - Multiclass Classification,softmax layer, NLL loss
   - MLP + PyTorch : Week 1-2
     - Linear Algebra, Single Layer NN, Training, - Inference and Validation : Illustrated Through Pytorch
     - Implement 1-hidden layer NN using PyTorch but train in python.

Hackathon 1 Classification On Numerical Data Starts

ROADMAP

CONTINUED

3. Intro to CNN : Week 2-3

- Simple Feed-forward Network : Week 2-3
  - Flatten image first and then treat as numerical data
- Convolutional Neural Networks : Week 3
  - Use Spatial Information
- Compare results with MLP on MNIST data
- LeNet : Week 3
  - Convolution + [Pooling] + Fully connected layers

Hackathon 1 Ends
Hackathon 2 : Image Classification Starts
Paper 1 : SOTA Model Implementation

ROADMAP

CONTINUED

4. Optimization of Neural Nets : Week 4-5
    - Optimizer variation :
        - SGD with Momentum, Nesterov and Adam
    - Overfitting and Regularization
        - L1, L2
        - Batch-Norm
    - Hyperparameter tuning
        - Variable learning rate,
        - Weight Initalization : Xavier, He Normal

Hackathon 2 Ends
Paper 1 Submission

ROADMAP

CONTINUED

5. Advanced Topics : Week 5-6
    - Autoencoders
        - Convert High dimension to Low dimension data
        - Should be able to convert Low to high with minimum error
        - MLP :
            - First flatten images i.e. convert to numerical data
            - As a (ineffective) compression method
        - Convolution :
            - For Denoising images
            - Uses Transposed Convolutions

ROADMAP

CONTINUED

5. Advanced Topics : Week 5-6
    - Generative Adversarial networks : (~~Y20~~)
        - Generate new data points as efficiently possible
        - Generator : Genearate fake data
        - Discriminator : Recognize fake data and penalize Genearator
        - Generator and Discriminator Compete with Each Other !!!

Paper 2 Implementation : Start from Week $5\frac{1}{2}$ Itself
Submit In the End

COLLOBORATION & GENERAL

- Form a group and implement one more paper/architecture if you have time : Week 6
- We will help in gathering resources for the same.
- Github repo
  - minimal clean code in jupyter notebook format.
  - Do not print large outputs.
  - Proper Folder,File structure(For both your and our convenience)
- Google


THANK YOU