# Version Control System for Deep Learning

**Professor: Yug Yung Lee**

# Project Pre-proposal:

# Team Details:

**Team ID:** 6

**Name:** Dinesh Kumar, Kusam

**Class ID:** 14

**Name:** Pradeepika, Kolluru

**Class ID:** 12

**Name:** Sindhusha, Tiyyagura

**Class ID:** 24

**Name:** Sravan Kumar, Pagadala

**Class ID:** 21

# Project Goal and Objectives:

## Motivation:

The Motivation of our project is to create simple and flexible git-like interface and architecture for Deep Learning projects.

## Significance/ Uniqueness:

Existing Version control system(Git) is used to track a single file while our proposed system is also used to handle tracking set of files or folders required for deep learning projects.

## Objectives:

- The main objective of our proposed system is to bring collaboration, reproducibility and agility into existing deep learning workflow.
- To maintain track of changes to an individual experiment or set of files so that one can review specific versions.

## Scope of project:

Scope of our project is to reduce redundancy in version control and help users to keep an artifact clean, well-organised and allows flexible rolling back to previous versions.

## System Features:

- Automatic grouping of set of artifacts(modeling artifacts, source code, data sets) into a single version.
- Command line interface compatibility
- Creating a new environment for each run of experiment.

# Related Work:

1. Github:

   Github is a DVCS(Distributed version control system) used for tracking changes to the files and allows multiple developers to work simultaneously.
2. ModelHub: Deep learning Life Cycle Management

   In order to deal with rich set of artifacts, Modelhub adopts a modeling version control system, a domain specific language for seeking through model space and hosted service.
3. Towards Unified Data and Life Cycle Management for deep learning.

   This paper mainly concentrates on implementation of data and life cycle management system for deep learning. Firstly, proposes a high-level domain specific language to speed up the modeling process. Thereafter, a novel model versioning system and parameter archival storage system(PAS) are developed which reduces storage footprint.

# Bibliography:

[1] Hui Miao, Ang Li, Larry S. Davis, Amol Deshpande, “ModelHub: Deep Learning Lifecycle Management” 2017.
https://par.nsf.gov/servlets/purl/10041785

[2] Hui Miao, Ang Li, Larry S. Davis, Amol Deshpande, “Towards Unified Data and Lifecycle Management for Deep Learning” 2016
https://arxiv.org/pdf/1611.06224.pdf

# Project Plan:

# Prioritized Features/Technologies:

1. Using python language for developing git-like VCS server.
2. Using SSH key mechanism to authenticate users.
3. Using DAG(Directed Acyclic Graph) data structure for handling objects/commits of files.

# Project Increment 1:

Analysis of features and internal functionality.

# Project Increment 2:

User registration and authentication/login requests to be handled.

## Stories (Issues):

1. Client either requests for authentication or login to the server.
2. Server connects to backend database either to store or retrieve user details.
3. Checks validation and sends response to the client.

# Project Increment 3:

Discovering repositories for the user, handling few of the commands like init, pull, push, commit, add, clone with the new functionality that handles the set of artifacts as a single version.

## Stories (Issues):

1. Giving the list of repositories in the user workspace.
2. User requested commands are handled by separate command specific script files.

# Project Increment 4:

Handling multiple branches, merging version and diffing files.

## Stories (Issues):

1. Adding the flexibility of adding and managing branches for the client.
2. Using tools for merging and diffing of file contents.

# Project Timelines, Members, Task Responsibility:

**Dinesh:** User Registration, handling commit, push, pull commands.

**Pradeepika:** User authentication, handling branches for the project.

**Sindhusha:** discovering repositories of the user, handling merging and diff of files.

**Sravan:** handling init, clone, add commands, handling collaboration of users.