MARMARA UNIVERSITY

THE FACULTY OF ENGINEERING

CSE4088

Introduction to Machine Learning

Comprehensive Comparison of Machine Learning Methods on Image Classification Problem

---

Final Report

	Dept	Name Surname	Student Id
1	CSE	Ömer Kibar	150119037
2	CSE	Müslim Yılmaz	150119566

ABSTRACT

In this project, we studied various machine learning approaches and made comprehensive comparisons between them. Our study includes **supervised learning, semi-supervised learning**, and **federated learning**.

OVERVIEW OF THE PROJECT AND SCHEDULE

Objective 1. Determining CNN Architecture and supervised learning

Objective 2. Semi-supervised learning implementation and analysis

Objective 3. Federated learning implementation and analysis

Objective 4. Reports and presentation preparation.

***Figure 1**. Division of Labour graph.*

PROJECT ACCOMPLISHEMNT

1) Data Collection & Data Splitting & Data Preprocessing

The CIFAR-10 [1] data set was downloaded, and data preprocessing steps were applied, including **image resizing, one-hot encoding** and **normalization**. Additionally, dataset was divided into training, testing and validation subsets.

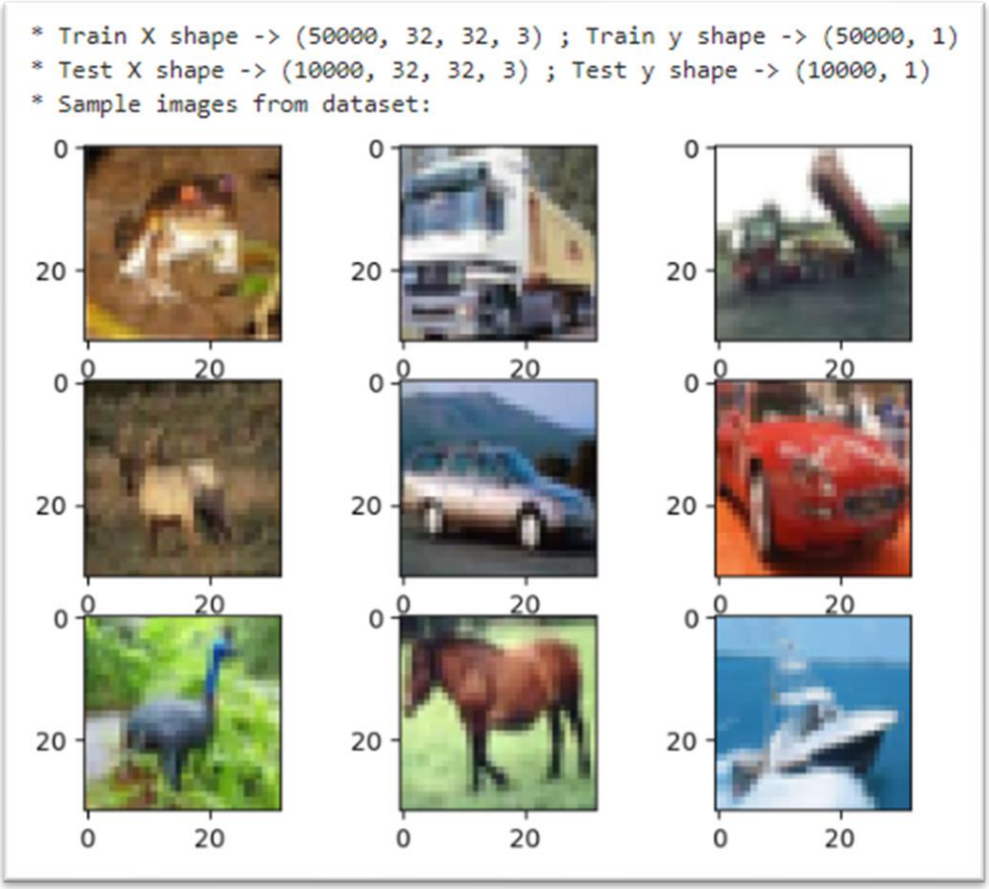
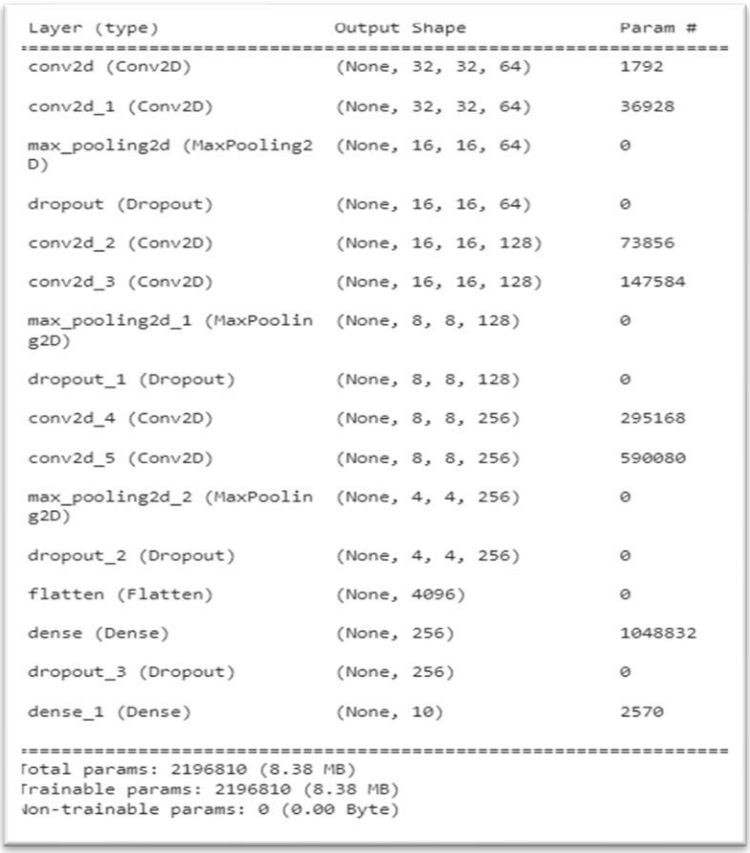


***Figure 2**. Example classes in CIFAR-10 dataset.*

2) Base CNN Model Designing for All Learning Methods.

A base convolutional neural network (CNN) model was created according to the CIFAR-10 dataset. This CNN model served as the base model across all learning methods, ensuring a fair comparison of outputs. The CNN model consisted of a combination of convolutional layers, a flatten layer, droput layers and dense layers. The model consists of 2,196,810 trainable parameters.

- **Convolutional Layers:** Model includes three sets of Conv2D layers with filter sizes 64,128 and 256. Dropout layers are added at the end of each set in order to prevent overfitting.
- **Flatten Layer**: Flatten layer is added to convert the three dimensions feature to one dimension vector to feed dense layers.
- **Dense Layers:** Two dense layers are used for classification purposes. The first one has 256 units and includes ReLU activation function. The second one contains 10 units and utilizes a softmax activation function to get class probabilities.
- **Droputs Layer**: Droupout layers added to prevent overfitting.



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 64)	1792
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 256)	1048832
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 10)	2570

Total params: 2196810 (8.38 MB)
Trainable params: 2196810 (8.38 MB)
Non-trainable params: 0 (0.00 Byte)

***Figure 3**. Parameters and layers on base CNN model.*

3) Model Training, Evaluation, Analysis for Centralized Supervised Learning

After creating the model, we train it with the following hyperparameters:

Learning rate: 0.001,

Batch size: 64,

Momentum: 0.9,

Epochs: 70

We use **stochastic gradient descent (SGD)** as the optimization algorithm and **cross-entropy** as the loss function in our centralized supervised method. CNN model reaches **85.48%** accuracy.

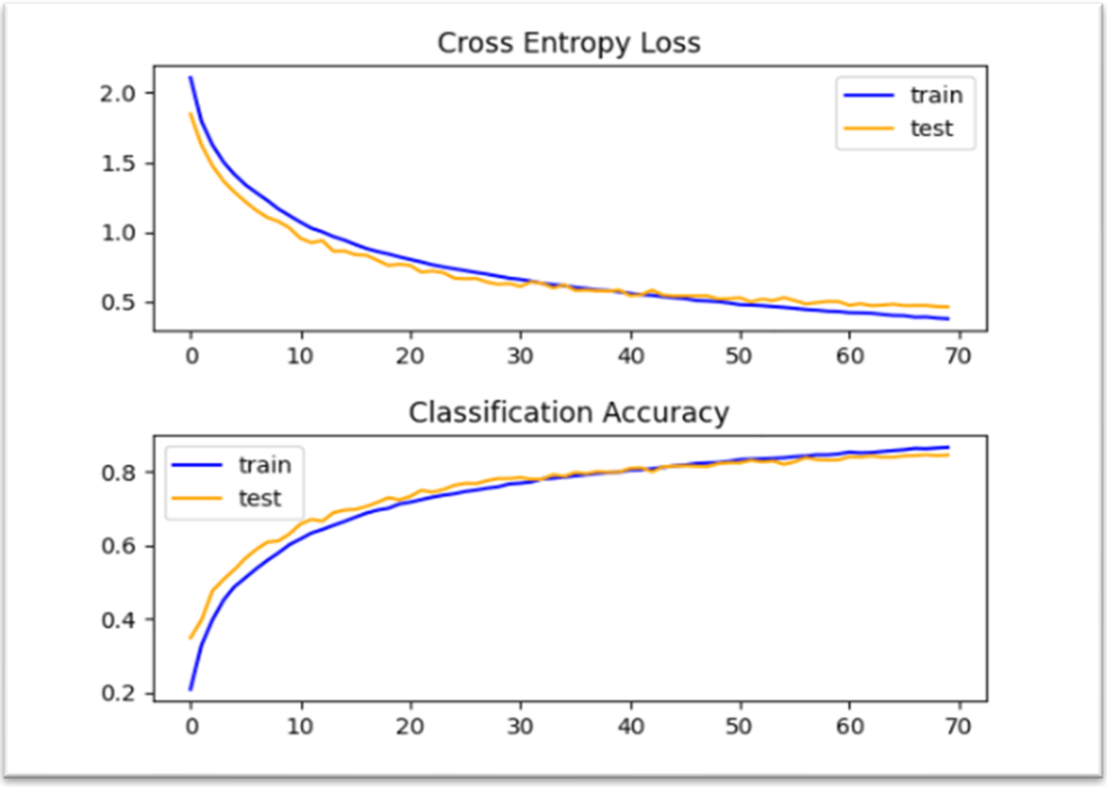


***Figure 4**. Centralized supervised learning loss and model accuracy values at given epochs.*

In figure 4, we observe that the model is not overfitting since training and test accuracy are following each other. Also, we see that test accuracy somehow was higher in the earlier epochs. After we see this behavior, we researched it and see that it is because of drop out layers used in CNN architecture.

4) Model Training, Evaluation, Analysis for Centralized Semi-Supervised Learning with Pseudo Labeling Technique

In our semi-supervised learning experiment, we prepared a setup to understand the effect of involving unlabeled data using pseudo-labeling technique. We split our dataset as in figure 4.

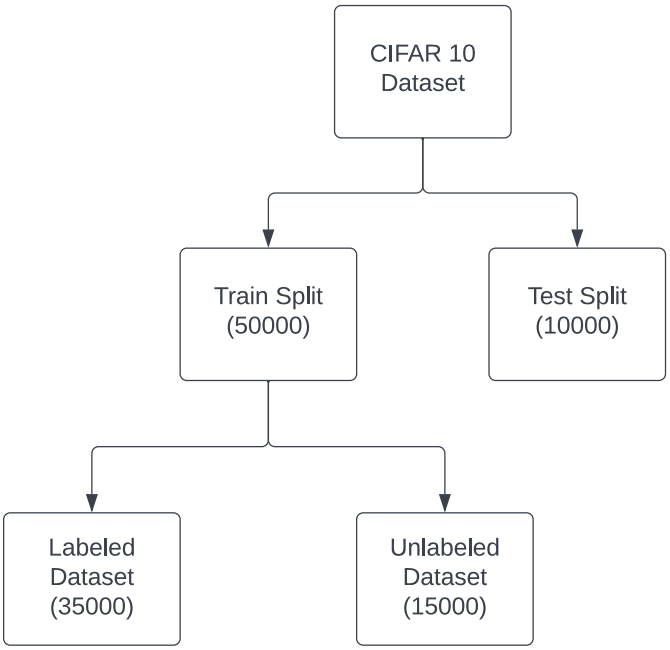


Figure 5*. Partitioning of the dataset employed in the semi-supervised methodology.*

In our first run we didn't use unlabeled data and trained model using only 35000 labeled images. In the second run we further trained pretrained model from the previous run by leveraging unlabeled data using pseudo-labeling technique. In pseudo-labeling, before each epoch model guesses on unlabeled data if models guess is higher than a confidence threshold, we give that image a pseudo label and append it in to labeled dataset. The hyperparameters remain consistent between the centralized version, with the empirical selection of a threshold value of **0.8**. Pretrained model which only uses labeled dataset achieved **81.40%** of accuracy. This accuracy increased to **84.12%** by involving unlabeled data using pseudo-labeling technique. This shows us that pseudo-labeling technique is effective and can be utilized to increase models' performance.

5) Model Training, Evaluation, Analysis for Centralized Federated Learning

In our experimental framework on federated learning, we simulate the application of federated learning techniques to systematically investigate their impact on the image classification.

Experimental Setup

- We simulate our federated learning on **100** devices.
- Each device is allocated **500** training images, resulting in a total of 50,000 training images, with an equal distribution among all devices.
- Sampling on devices adheres to the **IID** (independent and identically distributed) concept, signifying that the images on each device are independent of one another and are drawn from the same probability distribution and each device contains instances of dataset classes.
- We sampled **20** clients out of 100 in each communication round randomly for training.
- We sampled **5** clients out of 100 in each communication round randomly for evaluation.
- The simulation involves **75** communication rounds.
- On local devices, the training process comprises **10** local epochs with hyperparameters set as follows: learning rate = **0.01**, momentum = **0.9**, and batch size = **64**.
- On the server side, we employ the **Federated Averaging** (FedAvg) [2] algorithm to aggregate the incoming weights from the clients.
- The test set on the server side and used for the valuating the performance of the global model after each round.

Experimental Result

We reached approximately **%70** accuracy by using federated learning with the given setup.

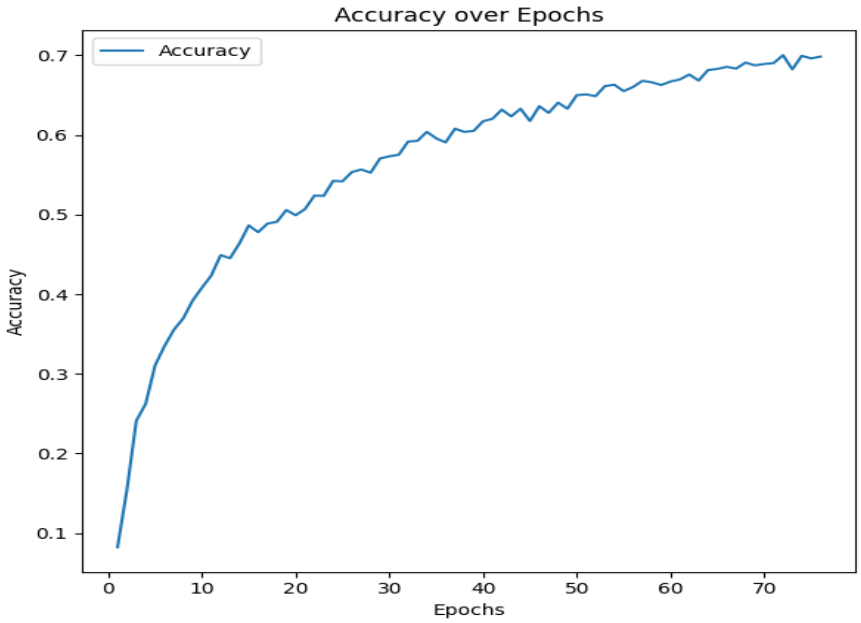


***Figure 6**. Depicts the changes in accuracy values over the given epochs.*

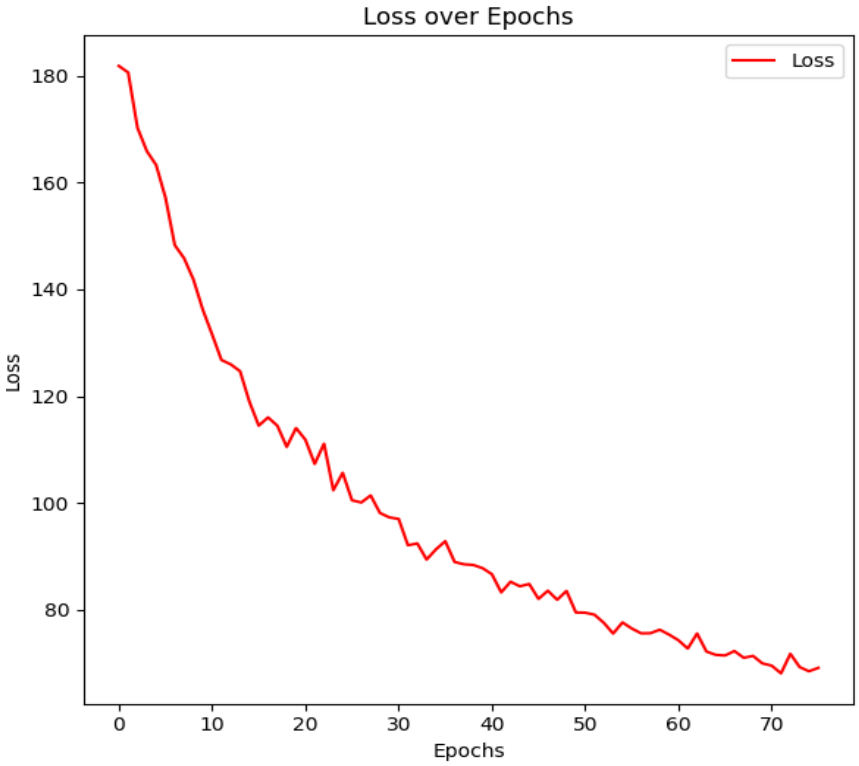


***Figure 7**. Depicts the changes in loss values over the given epochs.*

6) Comparison Statements

- **Statement about Supervised-Central vs Semi-Supervised Central**

In the centralized supervised approach, we achieved the highest accuracy at **85.48**%. In contrast, the semi-supervised approach yielded an accuracy of **84.12**% at the conclusion of the study. It is essential to note that in the semi-supervised scenario, **15,000** data points remain unlabeled, contributing to a **1.36**% decrease in accuracy compared to the centralized approach. On the other hand, we reduced the labeled data by **30%**, making the scenario more representative of real-world cases.

- **Statement about Supervised-Central vs Supervised-Federated**

In the centralized supervised approach, we attained the highest accuracy at **85.48%.** In contrast, the federated approach yielded an accuracy of **70%** at the conclusion of the study. It is crucial to highlight that in the federated learning scenario, there are notable advantages, particularly in terms of security and privacy where it makes the federated approach more realistic in the context of real-world scenarios.

The tradeoff between these models involves a careful consideration of factors beyond accuracy. While the centralized models may excel in accuracy metrics, the federated approach introduces crucial benefits in terms of privacy and security, making it more suitable for real-world applications. The decision-making process should thus involve a thoughtful evaluation of trade-offs between accuracy, data privacy, and security concerns based on the specific requirements of the application or system.

7) Summary

In this project, we explored various machine learning approaches, examining their strengths and weaknesses. We presented our experiment setups and results for each approach and conducted thorough comparisons among them.

8) References

[1] CIFAR-10:(2018). Kaggle. https://www.kaggle.com/c/cifar-10/

[2] H.Brendan, E. Moore, D. Ramage, S. Hamspon, B.Arcas "Communication-Efficient Learning of Deep Networks from Decentralized Data "