FedNet: Federated Implementation of Convolutional Neural Networks for Facial Expression Recognition

Md. Saiful Bari Siddiqui
Department of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka-1212, Bangladesh
ext.saiful.bari@bracu.ac.bd

Sanjida Ali Shusmita
Department of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka-1212, Bangladesh
sanjida.ali.shusmita@g.bracu.ac.bd

Shareea Sabreen
Department of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka-1212, Bangladesh
shareea.sabreen@g.bracu.ac.bd

Md. Golam Rabiul Alam
Department of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka-1212, Bangladesh
rabiul.alam@bracu.ac.bd

***Abstract*—Perhaps the most crucial aspect of any task involving Neural Networks is quality data that can be used to produce the best results. Edge devices like mobile phones and personal computers across the globe are exciting sources of quality data as data exist as islands in those devices and are guarded by strict privacy preserving laws. A clever means of connecting deep learning models to these disjointed data is Federated Learning (FL). This paper introduces FedNet, a novel approach towards Neural Network structures inspired by federated aggregation and averaging. The study was implemented from scratch for Facial Expression Recognition using two different approaches based on whether federated averaging was used or not. To carry out the task, Convolutional Neural Networks (CNN) were implemented using the Extended Cohn Kanade (CK+) dataset and the FER-2013 dataset. In this study, federated averaging based implementation of CNNs achieved 99.1% accuracy and a 100% accuracy for 8 and 7 emotion classifications respectively on CK+ dataset, beating the benchmarks for this dataset. It also achieved a respectable 65.6% accuracy on FER-2013 dataset without using any transfer learning, extra training data or even data augmentation. The accuracy for CK+ dataset beats the accuracy achieved using the same model without FL based averaging. This paper also demonstrates FedNet models showing significantly better resistance against over-fitting due to global averaging. The code and implementations of this study are publicly available at https://github.com/Saiful185/FedNet-Federated-Implementation-of-CNN-for-Facial-Expression-Recognition.**

***Index Terms*—Federated Learning, Convolutional Neural Networks, Data Augmentation, Overfitting.**

I. Introduction

In the present age, most mobile devices can capture images and thus have the ability to generate the amount of data required for deep learning model training. In most real-world applications, it is impractical to expect a central server to train a global model by gathering data from all connected mobile clients. The Federated Learning approach addresses these issues. FL is a creative technique to link machine learning models to data, independent of where it is stored. As a rule of thumb, FL sends the information to the system for training, not the other way around. All that is needed is for devices to commit to the federation process. This ML technique trains an algorithm without exchanging data samples by using decentralized edge devices or servers in huge numbers [1]. Typical centralized machine learning algorithms, which need all local data sets to be uploaded to a single server, as well as more traditional decentralized options, which often presume uniform distribution of local data samples, are not compatible with this strategy. Federated Learning is a privacy-preserving methodology for building models from distributed user data held on edge devices. Every participating device or client receives an initial model from a central server, runs stochastic gradient descent (SGD) on its local data set, and transmits the gradients back through using Federated Averaging technique in each federated learning round (FedAvg). The server then adds up all of the gradients from all of the clients and changes the initial model [2]. Data integrity occurs as islands on the world's edge devices, such as smartphones and other mobile devices, and is safeguarded by strict privacy restrictions. Federated Learning is a clever way to connect machine learning models to these disparate data sets, regardless of where they are stored, and, more crucially, without violating privacy rules. FL trains the model on the data rather than the data on the model, which is normal procedure. All that is required is for the device holding the data to be willing to participate in the federation process.

The FL architecture comprises of an or server that sits in the middle and supervises the training activities in its most basic form. Clients are typically devices at the edge of the

network, and there can be millions of them. On average in per training iteration, these devices talk with the server at least twice. For starters, they each receive the current global model's weights from the server after averaging (Fig. 3), then train it around their own local data to generate new variables, which are then sent back to the server for aggregation. This transmission cycle is repeated until an epoch number or accuracy condition is reached.

In this study, we used all these concepts as an inspiration to propose a novel neural network architecture, **FedNet**, for facial expression recognition. We divided our training data into multiple data shards and then performed a process similar to Federated Learning. The difference was that we had multiple subsets of the whole dataset (data shards) instead of local clients and machines. Here we have created a deep learning model from scratch using Tensorflow for image classification. This model is trained with extended CK+ and FER-2013 dataset. In this simulation, clients will be modeled by data shards, and CNN will be used to train all local models on the same system.

II. Related Works

Several authors have worked on image classification using Federated Learning, from which we can briefly overview some which are more related to our work. FaceNet2ExpNet [3] is a state-of-the-art model for facial image recognition using CK+ dataset. This study proposes a new distribution function to model the high-level neurons of the expression network. A two-stage training algorithm is designed based on this. User splits from the MNIST data set, according to McMahan, are created by sorting training samples by class labels and dividing them into pieces, with each client receiving two shards. They show that, although taking more cycles than identical clients, FedAvg accumulates to 99 percent accuracy on non-identical clients. Using the same sort-and-partition method [4]. In another research work Jun Luo and Shandong Wu found that FedSLD outperforms the top FL optimization algorithms in terms of convergence, resulting in a higher test score. Up to 5.50 percentage points of accuracy while they were working with medical image classifications [5]. In Beomyeol Jeon's work on Privacy-preserving Decentralized Aggregation for FL, their federated training approach on image classification application over benchmark data sets with 9 and 15 dispersed locations is demonstrated in the study. When compared to the standard centralized federated learning technique, the degradation in test accuracy is only
0.73 percent [6]. Another article explores the performance of federated models for histopathology image classification in a real-world setting. The authors give a comparison of the suggested method with regionally trained models, a centralized model, and two main FL model aggregation methods in this paper. [7]. Jaehoon Oh in his study of Enhanced Representation for Federated Image Classification, proposed a model which updates just the model's body except for the head during federated training. The head is fine-tuned for penalization during the evaluation process, and it is randomly initialized and never updated [8]. Extensive testing has revealed a consistent pattern in this research work. There has been more work on medical image classification data sets, for example, Yaoxin Zhuo and Baoxin Li proposed a learning method that directly connects learning across labeled and unlabeled clients, minimizing task knowledge deficits at unlabeled clients and encouraging discriminative information from unlabeled samples [1]. Agrawal and Mittalns' work on the FER-2013 data set, a detailed study of different kernel sizes and number of filters led to the proposal of two unique CNN architectures that attain a human-like accuracy of
65 percent using only CNN [9]. Furthermore, the smallest root mean square error for the JAFFE dataset was 0.1661 for valence and 0.1379 for arousal, according to a series of studies using face picture datasets from the Japanese female facial expression (JAFFE) dataset and the extended Cohn–Kanade (CK+) dataset on the research work of Lee and Kang [10].

While some of the preceding tactics outperform the more fundamental Fed-Avg strategy under rare circumstances, they don't necessarily transition well to more normal FL conditions. With minimum changes to the core FL architecture, our model tries to build an approach that focuses on better model aggregation. We used component-by-component parameter averaging, based on the ratio of data points provided by each client who participated. Our strategy has a better chance of being widely applicable, especially in real-world scenarios like image classification and handwritten digit recognition.

III. Proposed Model

A. Federated Learning (FL) Averaging:

The learning logic is divided into two loops: one for global iteration and another for repeating through the client's local training [11]. However, there is an implied third one that accounts for local epochs and will be handled by the model's epochs argument model.fit method. We began by creating a global model with a number of classes of 7 or 8. Then we approached the outer loop. Acquiring the global model's initialized weights first. The client's vocabulary order is then shuffled to ensure randomization. Following that, we began moving on client retraining. We built a new model object for each client, loaded it, and set its initialization phase weights to the global model's current parameters. Following that, the client was given training for 5 epochs. The new weights were scaled and added to the scaled local weight list after training. That would be the conclusion of the local training. Returning to the outer loop, we added up all of the scaled local training weights (by components of course) and modified the global model to these updated aggregates. An entire global training epoch has come to a close. We used the communication round to run 25-45 global training loops and evaluated the learned

global model after every communication round with our testing data.

B. Convolutional Neural Network (CNN):

CNN is a neural supervised classifier for detecting and classifying input after given the correct data. CNN has learned the art of categorizing photos for computer visions over the years, and it is now being applied in handwritten digit identification as well. CNN is an effective deep learning method for automated end-to-end prediction as evidenced by it as well. CNN essentially automatically pulls 'valuable' information from the provided information, making it extremely simple. Convolutional layer, pooling layer(s), and fully connected layer are the three basic layers of a CNN model [12].

(1) Convolutional Layer: This layer gathers high-level input features from input data and sends them on as feature mappings to the next layer.

(2) Pooling Layer: This layer is used to minimize the dimensions of data by applying pooling to the feature map and creating new feature maps with smaller dimensions. Within a stride, PL takes the maximum or average of the old feature map.

(3) Fully-Connected Layer: The FC layer is in charge of categorization at the end of the process. A prominent activation function known as the softmax function is used to produce probability scores for each class label.

In this paper, we start with an overview of the Convolutional Neural Network and its significance. For example, predictions of facial expressions ranging from 0(Angry) to 7(Surprised) were made using Extended CK+ and FER-2013, two famous datasets. The data was cleaned, scaled, and shaped before being used. A CNN model was developed with TensorFlow and then trained on the training dataset. Finally, the trained model was used to make predictions.

The overall structure and overview of our model can be visualized in **Figure 21**. **Figure 20** gives a detailed overview on the local models on which each data shard was trained.

IV. Model Aggregation and Functions

Federated Averaging: Everything we have done so far has been relatively normal for a deep learning pipeline. The novelty of FedNet lies in the Model aggregation and averaging part. The data to be used will be horizontally partitioned. Simply executing component wise parameter averaging, which then will be weighed based on the fraction of data instances belonging to each participating client in Federated Averaging (the vanilla algorithm for FL) will do the job here. Here is the federated averaging equation we're utilizing, which originates from one of those federated learning papers [13].

We calculate the weight parameters for every client in the equation on the right-hand side based on the loss values reported throughout every data point they trained. Each one of those factors were normalized and then added up component-wise on the left. This process has been broken down into three basic functions under.

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad \text{where} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w).$$

Fig. 1: Federated Averaging Equation

(1) Weight scaling factor: The information coming from a client compared to the overall training information stored by all clients is calculated. The batch size of the client was firstly determined and utilized to determine the number of data points. The amount of the global training data is estimated then. Finally, the scaling factor was determined as a fraction. This part is specific to the FedNet approach as this won't be the same using FL.

(2) Scale model weights: Each one of the local model's weights are scaled using it depending on the amount of their scaling factor determined before.

(3) Sum scaled weights: Adds up the scaled weights of all clients.

V. Experiments

A. FL implementation of CNN on CK+ Dataset

We used Extended [14] Cohn Kanade (CK+) dataset [15]for this study. Each image in CK+ Dataset is of size 690X480 or 690X490. We used the last 3 images from each sequence for all the classes apart from the class for neutral emotion. At the start of the task the images are reshaped and normalized as part of preprocessing. All the images were also resized to 69X48 pixels.

Fig. 2: Extended CK+ Dataset Samples

We used OpenCV package and imutils for this part. We binarized the labels to use cross entropy later. We split the data using Scikit-Learn and applied a 90-10 split. In the real- world implementation of FL, each federated member will have its own data coupled with it in isolation. But in our case with a single data-set, we have to create shards. So, we share the training set into 3 shards, one per client. That means the number of clients in our FL implementation is 3. Firstly, we zipped the data and label lists then randomized the resulting tuple list. Finally, we created shards from the tuple list based on the desired number of clients. To process each of the client's data into TensorFlow data set and batch them was our next step. Then the model is constructed. Our model for 8 emotion classification comprised of 4 Convolutional (32 to 256 kernels) and Maxpooling layer(2X2) blocks along with 1 dense hidden layer (128 neurons). Each block consisted of two convolutional layers and

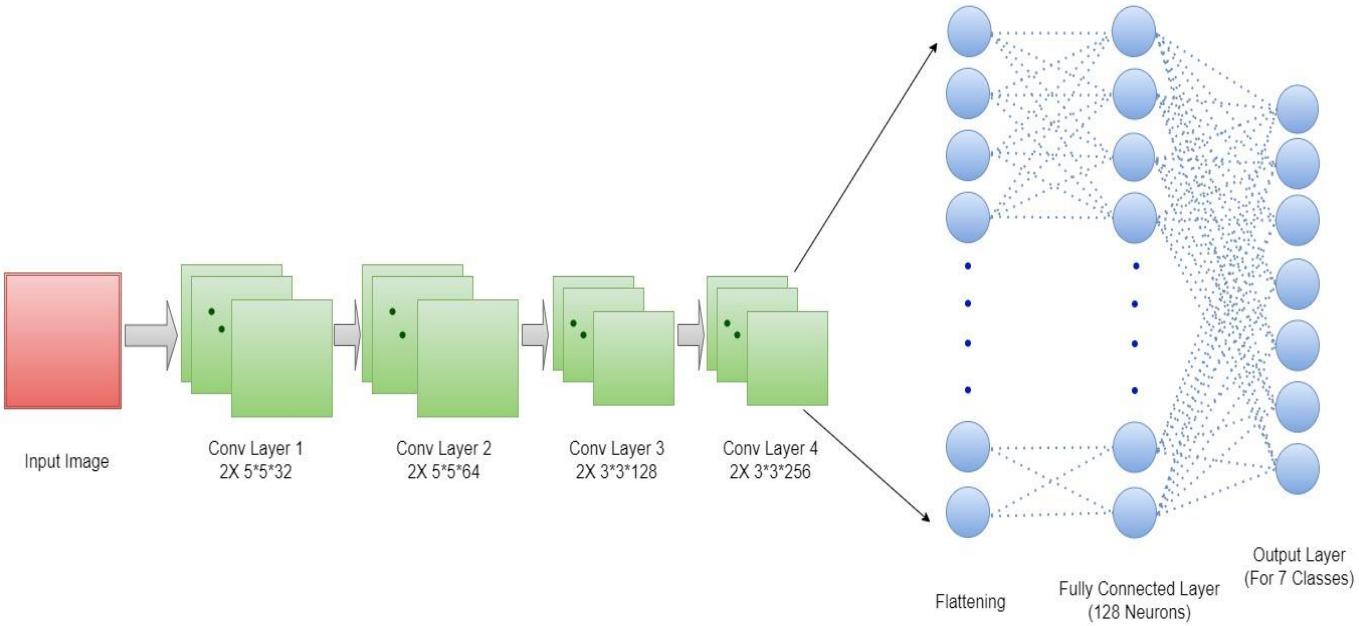

Fig. 20: Local Neural Network Architecture that Trains each Data Shard

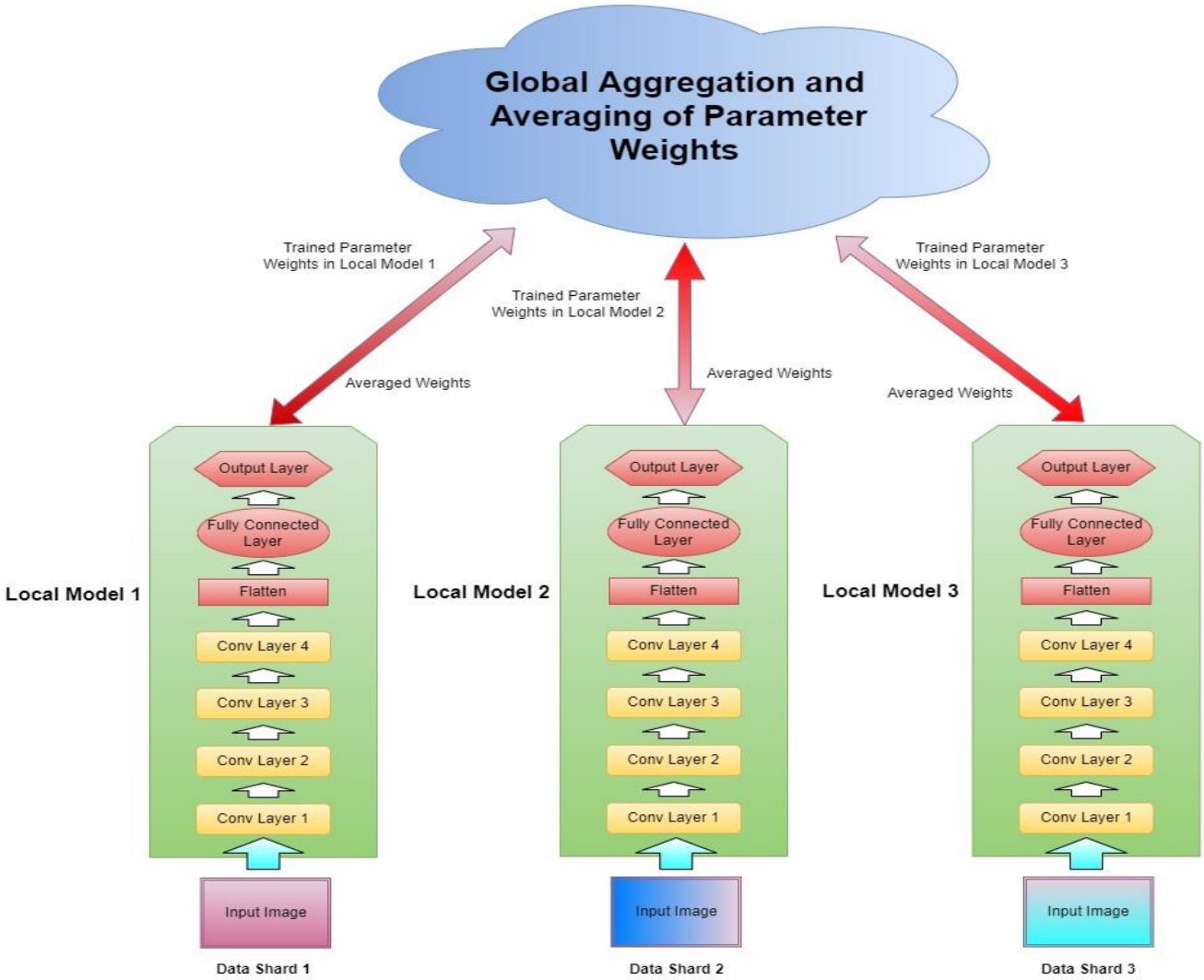

Fig. 21: FedNet at a Glance – Overview of the Model Including Averaging of Parameter Weights

one pooling layer. The first two blocks were created using 5*5 filters, and the next used 3*3 filters. After each Convolution layers batch normalization was applied. After each block, dropouts were used. This model summary is shown in Figure 20 and 21. The model for 7 emotion classification was much simpler.

Fig. 3: Federated Averaging Process

B. FL implementation of CNN on FER-2013 Dataset

FER-2013 [16] is a much larger yet more complex dataset consisting of a bit more than 28000 training images. Each of the images are of size 48X48. The implementation on Fer- 2013 dataset is pretty much the same. However, the resizing step is not needed as all the images are of same dimensions. Despite it being a 7-class classification, the model was the same as the 8-emotion classifier for CK+ dataset since FER is a complex dataset to train on. Rest of the code are pretty much the same as the implementation on CK+.

Fig. 4: FER2013 dataset samples

VI. RESULTS

We summarized our outcomes in **Table 1**. The overall observations are discussed below.

TABLE I
PERFORMANCE SUMMARY FOR ALL THE IMPLEMENTATIONS

Implementations	Accuracy	Log Loss	F1 Score
FedNet on Extended CK+ (7 Emotions)	1.0	0.0007	1.0
Basic CNN on Extended CK+ (7 Emotions)	0.9817	0.0324	0.9679
FedNet on Extended CK+ (8 Emotions)	0.9910	0.0603	0.9936
Basic CNN on Extended CK+ (8 Emotions)	0.9775	0.1520	0.9803
FedNet on FER-2013	0.65657	1.50528	0.6503
Basic CNN on FER-2013	0.6551	1.7852	0.6407

1. With FedNet approach, the accuracy is **99.1%** for 8 class classification using our CNN model on Extended CK+ dataset. In case of 7 class classification, it achieves a **100%** accuracy. These results are superior to the current SOTA models in both cases. It also outperforms the exact same model implemented without Federated aggregation, which achieves **97.75%** accuracy for 8 emotion classification.

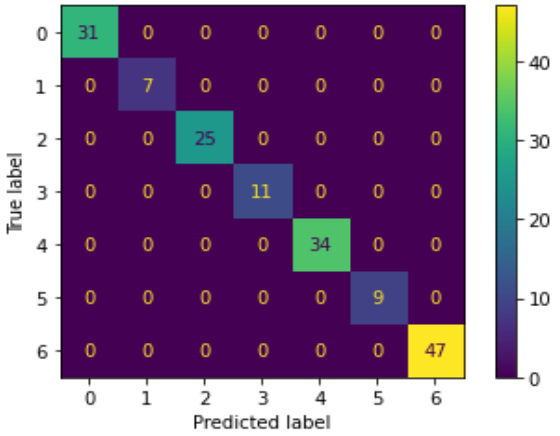

Fig. 5: Confusion Matrix for 7 emotion Classification on Extended CK+ Dataset

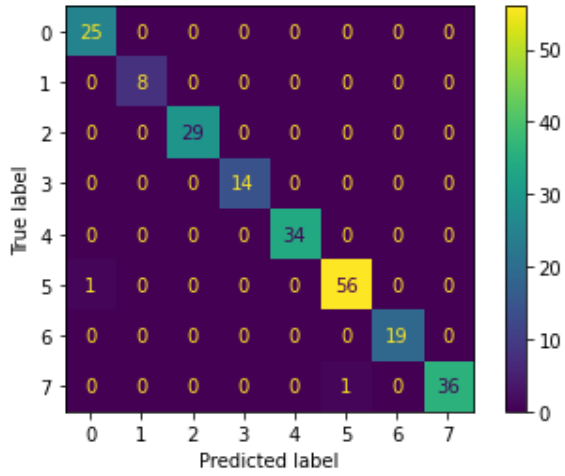

Fig. 6: Confusion Matrix for 8 emotion Classification on Extended CK+ Dataset

2. In case of FER-2013 dataset, the accuracy is **65.657%** for 7 emotion classification which is competitive with other models without any use of Data Augmentation and it once again beats the implementation without federated averaging at **65.51%.**

3. Typically, FedNet based approach is slower in learning compared to basic Neural networks. That's because every single model is fed with less data in case of FL but during each comm. round several local models are being trained simultaneously. For this reason, FedNet approaches in a single machine will generally have a longer training period.

4. The Log loss however, is significantly lower with FedNet implementations. The reason being the fact that there are multiple models being trained at the same time and averaging their trained weights gives better generalization on validation and test set, reducing the loss in the process. FedNet performs exceptionally in terms of F1 score too, ensuring its effectiveness on unbalanced datasets.

5. One important observation is that in case of Federated implementations the tendency to overfit is much less compared to basic neural networks. That's because we are

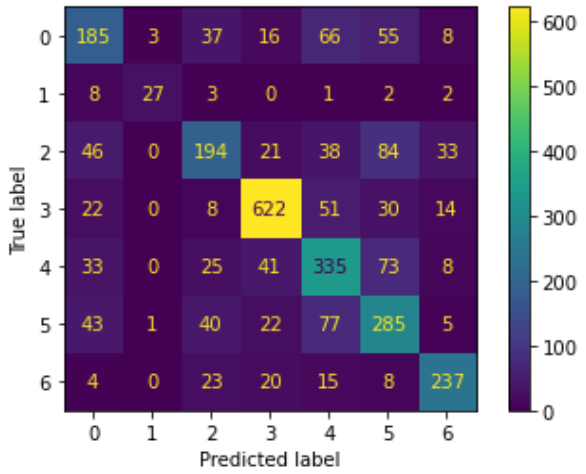

Fig. 7: Confusion Matrix for 7 emotion Classification on FER-2013 Dataset

averaging weights for every client in each comm round. So, it cannot possibly overfit for a particular client's data. This works naturally against overfitting, which is not possible in case of basic neural networks. Figures **8 to 19** depicts this situation clearly, where we can see log loss decreasing with each epoch in case of Fed Net implementations **(see Figure 10),** but starting to increase after a few epochs again in case of basic NN implementations **(see Figure 11).** In case of Extended CK+ the accuracy and loss curves for FedNet is much smoother compared to the implementations without FedNet **(Figure 12-19).** This happens because of overfitting that takes place without the federated averaging. This property of our proposed FedNet approach is the most important outcome of this study, as it demonstrates a novel approach to counter one of the major obstacles that classical machine learning and deep learning models face - Overfitting.

TABLE II
PERFORMANCE COMPARISON OF FEDNET WITH DIFFERENT SOTA MODELS FOR EXTENDED CK+ DATASET BASED ON ACCURACY

Benchmark Models	Extra Training Data	8-Emotion	7-Emotion
FN2EN	NO	0.968	—
ViT + SE	YES	—	0.998
FAN	NO	—	0.997
DeepEmotion	NO	—	0.98
FedNet	**NO**	**0.991**	**1.0**

VII. CONCLUSION

Federated Aggregation was always known as a creative way to connect ML models to diverse data sets irrespective of where they are stored and more significantly without violating privacy rules. Our study demonstrates the fact that federated averaging with a suitable neural network structure not only does that but also can produce significant performance gains. The smoothing effect that global averaging has on the training process can be groundbreaking. Everything that is required is for the machine containing the data to be prepared to cooperate in the federation process. In this paper, the building blocks of Federated Learning (FL) were implemented and used as an inspiration to create FedNet, a simple but effective approach. FedNet was used to train from scratch using the Extended CK+ dataset and FER 2013 dataset. The outcomes of our model were excellent, on par or surpassing the current best models **(See Table 2)** with 99.1% test accuracy for 8 emotion classification and 100% for 7 emotion classification on the CK+ dataset. FedNet performing better than both current best models and its own version without global averaging proves our hypothesis of FedNet being able to better generalize resisting overfitting to be true. This property of FedNet not only proposes a unique way to reduce overfitting - a major obstacle to training deep learning models properly, but also provides plenty of scope for future research and experimentation on different domains using this federated averaging concept.

REFERENCES

[1] Y. Zhuo and B. Li, "Fedns: Improving federated learning for collaborative image classification on mobile clients," *CIDSE, Arizona State University*, 2021.

[2] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019.

[3] H. Ding, S. K. Zhou, and R. Chellappa, "Facenet2expnet: Regularizing a deep face recognition net for expression recognition," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, 2017, pp. 118–12610.1109/FG.2017.23

[4] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konecn ý, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *Carnegie Mellon University*, 2019. [Online]. Available: https://leaf.cmu.edu

[5] J. Luo and S. Wu, "Fedsld: Federated learning with shared label distribution for medical image classification," 2021.

[6] B. Jeon, S. M. Ferdous, M. R. Rahman, and A. Walid, "Privacy-preserving decentralized aggregation for federated learning," 2020.

[7] G. N. Gunesli, M. Bilal, S. E. A. Raza, and N. M. Rajpoot, "Fed-dropoutavg: Generalizable federated learning for histopathology image classification," November, 2021.

[8] J. Oh, S. Kim, and S.-Y. Yun, "Fedbabu: Towards enhanced representation for federated image classification," June, 2021.

[9] A. Agrawal and N. Mittal, "Using cnn for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy," January, 2019. [Online]. Available:https://doi.org/10.1007/s00371-019-01630-9

[10] H.-S. Lee and B.-Y. Kang, "Continuous emotion estimation of facial expressions on jaffe and ck+ datasets for human–robot interaction," November, 2019. [Online]. Available:https://doi.org/10.1007/s11370-019-00301-x

[11] S. Tijani, "Federated learning: A step by step implementation in tensorflow," Sep 2020. [Online]. Available:https://towardsdatascience.com/federated-learning-a-step-by-step-implementation-in-tensorflow-aac568283399

[12] S. Rajwal, "Classification of handwritten digits using cnn," Jul 2021. [Online]. Available:https://www.analyticsvidhya.com/blog/2021/07/classification-of-handwritten-digits-using-cnn/

[13] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2017.

[14] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000, pp. 46–5310.1109/AFGR.2000.840611

[15] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 94–10110.1109/CVPRW.2010.5543262

[16] L. Zahara, P. Musa, E. Prasetyo Wibowo, I. Karim, and S. Bahri Musa, "The facial emotion recognition (fer-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (cnn) algorithm based raspberry pi," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–9 10.1109/ICIC50835.2020.9288560

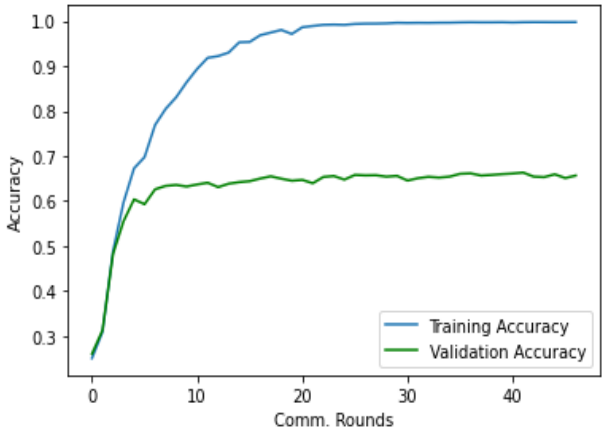

Fig. 8: Training and Validation Accuracy for FedNet implementation on FER- 2013 Dataset

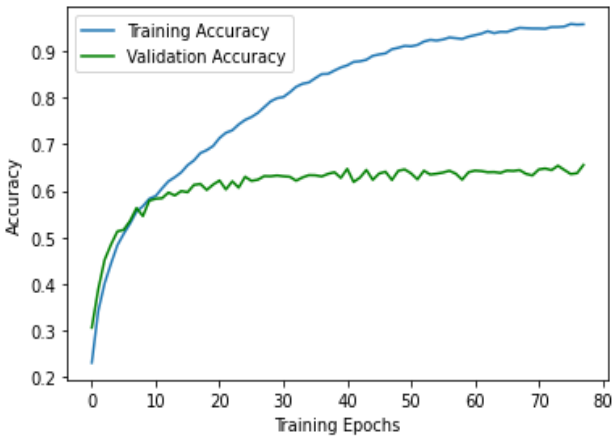

Fig. 9: Training and Validation Accuracy for implementation without Federated Averaging on FER-2013 Dataset

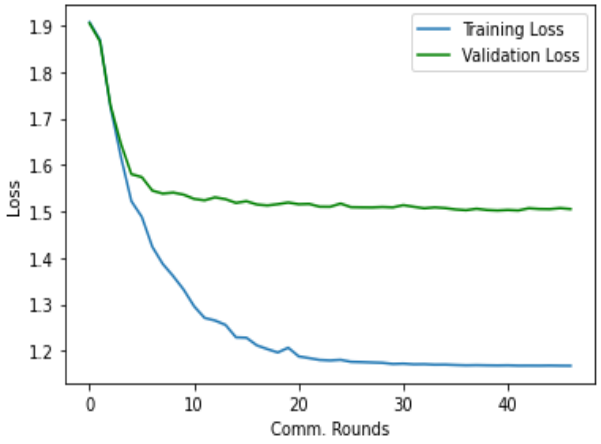

Fig. 10: Training and Validation Loss for FedNet Implementation on FER-2013 Dataset.

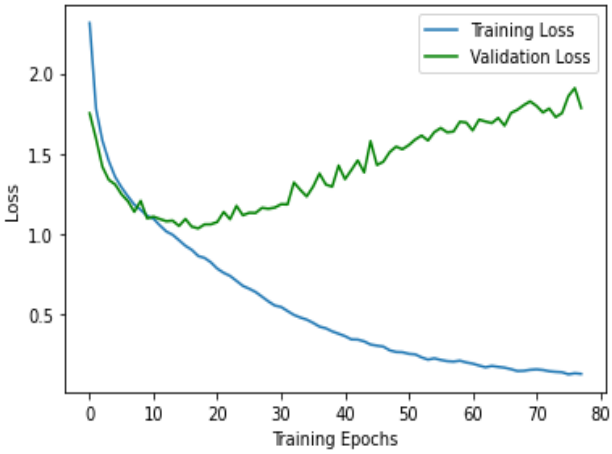

Fig. 11: Training and Validation Loss for Implementation without Federated Averaging on FER-2013 Dataset

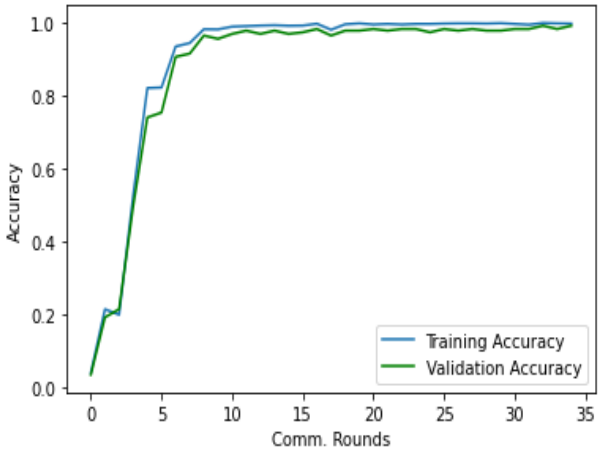

Fig. 12: Training and Validation Accuracy for FedNet implementation on Extended CK+ Dataset (8 Emotions)

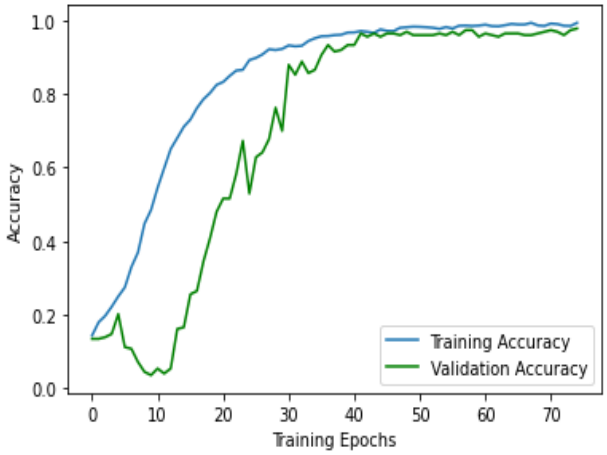

Fig. 13: Training and Validation Accuracy for Implementation without FedNet on Extended CK+ Dataset (8 Emotions)

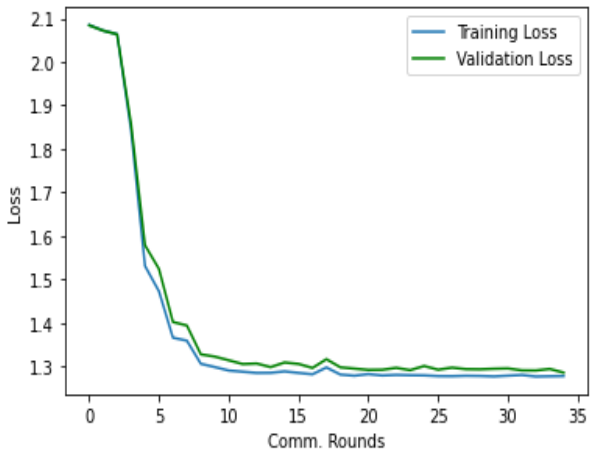

Fig. 14: Training and Validation Loss for FedNet implementation on Extended CK+ Dataset (8 Emotions)

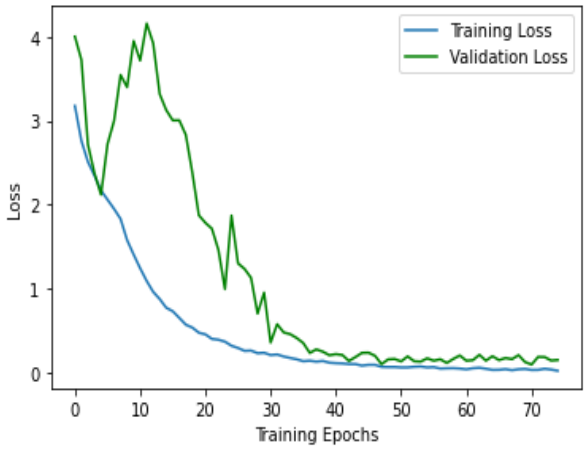

Fig. 15: Training and Validation Loss implementation without FedNet on Extended CK+ Dataset (8 Emotions)

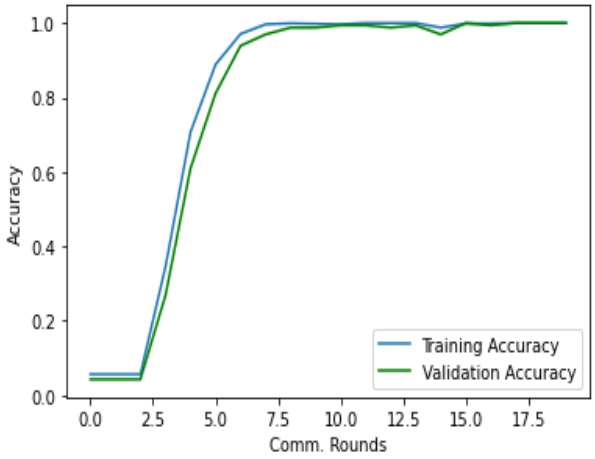

Fig. 16: Training and Validation Accuracy for FedNet implementation on Extended CK+ Dataset (7 Emotions)

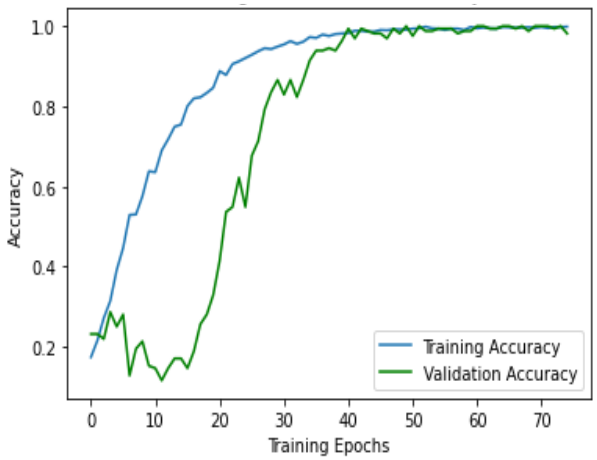

Fig. 17: Training and Validation Accuracy for implementation without FedNet on Extended CK+ Dataset (7 Emotions)

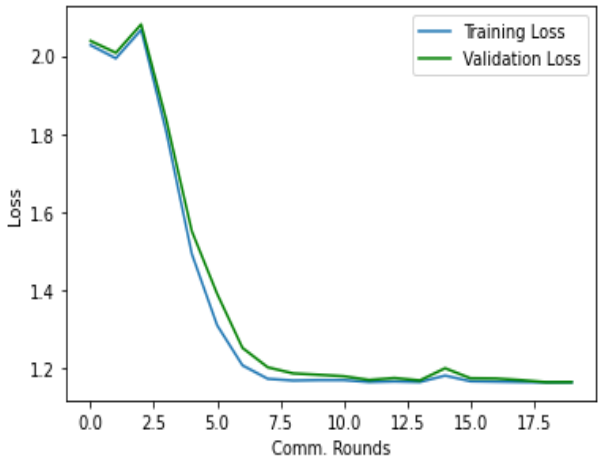

Fig. 18: Training and Validation Loss for FedNet implementation on Extended CK+ Dataset (7 Emotions)

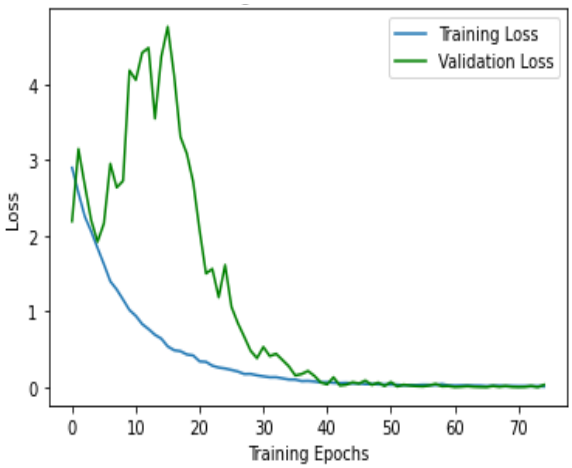

Fig. 19: Training and Validation Loss for implementation without FedNet on Extended CK+ Dataset (7 Emotions)