# Split Federated Learning for Emotion Detection

Dinah Waref
*Digital Media Engineering and Technology*
*German University in Cairo*
Cairo, Egypt
dinah.shalaby@student.guc.edu.eg

Mohammed Salem
*Digital Media Engineering and Technology*
*German University in Cairo*
Cairo, Egypt
mohammed.salem@guc.edu.eg

2022 4th Novel Intelligent and Leading Emerging Sciences Conference (NILES) | 978-1-6654-5241-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/NILES56402.2022.9942417

***Abstract*—The benefits Machine Learning has provided for us are incredible but for better performance and accuracy, more data is needed. This data might include sensitive data such as one's face and expression and collecting such data is a huge privacy risk. Recognizing emotions is crucial for healthy social communication, It also provides a better understanding of the changing behaviours of customers and better metrics of how they like a product. This is why Emotion recognition is an incredible market currently facing significant gaps due to the lack of data. Using a hybrid Split Federated Learning model that overcomes the resource constraints of Federated Learning and reduces the computational time of Split Learning, We can decentralize the training process and keep the sensitive data safe. Our aim is to create an Emotion Classifier using Split Federated learning. We had to create new datasets using existing datasets and cropping them such that the lower part of the face is discarded and only the upper part of the face is visible. Our classifier is implemented using Split Federated Learning, which combines Split and Federated Learning. Our classifier gave accuracies 87%, 98%, 96%, 87% and 99% for FER2013plus, AffectNet, CKplus, ouluCASIA and KDEF cropped datasets respectively with SplitFed. The results were relatively good when compared to the centralized training approach accuracy results for the same datasets.**

***Index Terms*—Federated, Split, Learning, Decentralized, Emotions**

## I. Introduction

### A. Motivation

As users, we benefit from personalization which makes our lives easier and more efficient [1]. The problem lies here, for better research and customization, more data is needed to train the models for better performance and accuracy but to get this data from the user and store it in a centralized way to train the model is a huge privacy risk, especially for sensitive data like medical records, finance and personal data, for example, one's face and expression. Many Leading companies have taken the initiative to use emotion detection to their advantage [2].For example, Disney created an algorithm that can determine how the audience likes its movies by recognizing complex emotions and might even predict upcoming emotions. Another example is Affectiva which developed advanced emotion and object detection for in-car safety systems, to recognize whether the driver is drowsy, frustrated, happy or sad. In recent times, Covid-19 pandemic has impacted our lives greatly, it has also brought a mask mandate. Which is crucial to help us prevent the spread of the disease. Even before Covid, covering parts of the face is a practice done for religious reasons, cultural reasons and even as a mandate due to the nature of some jobs [3].

### B. Problem Statement

In this study, we will be developing a Split Federated Learning Hybrid Algorithm which will ensure the privacy of the users' data and offer better efficiency for the training process. The Algorithm aims to overcome the resource constraints of federated learning and decrease the computational time of split learning.

We will use Split Federated Learning to implement Human Emotion Classification based on the Eyes.

## II. Background

### A. Concepts Overview

*1) Federated Learning:* Federated Learning is a type of decentralized machine learning that allows collaborative learning between multiple servers or edge devices without sharing raw data, which enables the creation of a model that overcomes problems such as data privacy, rights and security.

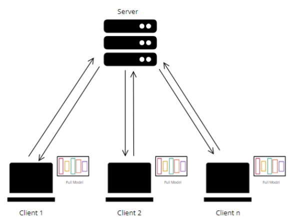


Fig. 1. Federated Learning Illustration

*2) Split Learning:* Split Learning is a recently proposed Federated Learning Technique that Splits the Neural Network and each segment lives on multiple servers or edge devices. During training, the edge device layers compute their outputs using the local data and then proceed to forward propagates the output which is called smashed data to the main server that holds the remaining part of the Split Neural Network.

*3) Split Federated Learning:* Split Federated Learning Architecture combines Split and Federated Learning in an effort to overcome the resource constraints of Federated Learning and decrease the computational time of Split Learning.
It splits the neural network as in the Split Learning into two parts. Then the training process is done by each client-server

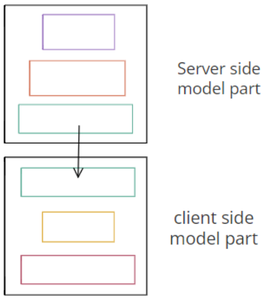


Fig. 2. Split Learning Illustration

pair simultaneously as in Federated Learning. After all the clients' data is used to update the parameters, The server-side weights are aggregated and updated.

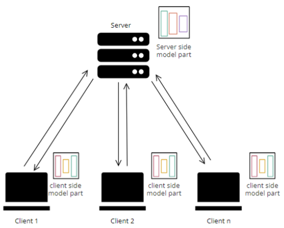


Fig. 3. Split Federated Learning Illustration

### B. Literature Review

In 2020 Turina et al [4] mentioned that Federated Learning and Split learning architectures both have their advantages and disadvantages set their goal which was to combine them to get the best out of both. The goal of this design is to reduce the computational power needed by each client in Federated Split Learning and Parallelize Split Learning.

Thapa et al [5] created SplitFed, the first Split Federated Learning framework, where each client does its computation in parallel and interacts with the main server. SplitFed results in model accuracy similar to that of Split Learning while supporting resource-constrained devices.

Romanini et al [6] created PyVertical which is a framework that uses a hybrid acrchitecture between split and federated learning for vertical federated learning

Abedi et al [7] proposed a framework, FedSL, that combines Federated Learning and Split Learning which allows the model to train on distributed sequential data. The implementation was done using Recurrent Neural Networks RNN on eICU dataset and demonstrated the success of the proposed algorithm.

Others such as Liu et al [8], Gao et al [9] and more such as [10] [11] took the initiative to evaluating the split federated learning architecture for Internet of Things.

## III. Methodology

### A. Data Pre-Processing

We created a script that first uses the Harr Classifiers in openCV to detect the face in the image. We first used the frontal face cascade classifier to detect the face, then if a face is found we crop our region of interest and store it. Then we use the eye pair classifier to detect our region of interest which in this case is the eyes, crop it and save the new cropped image. The cropping process is shown in Figure 4.

Fig. 4. Process of Cropping The Face Then Eyes

### B. SplitFed

SplitFed implements the server-side program starting with the FedAvg function which performs the federated averaging and another function for calculating the accuracy with the correct number of predictions divided by the number of predictions made. SplitFed then initializes one net server and n net model servers where n is the number of clients. Then the server-side training and testing functions are defined. Then the client-side program is defined where the client-side training and testing functions are defined. As for the Dataset preprocessing the dataset is read and then split into train and test which is then cast into the custom data class type. Then the training and testing processes begin, we iterate on the number of epochs and for each epoch we iterate on the number of clients. The Workflow of SplitFed is illustrated in Figure 5.

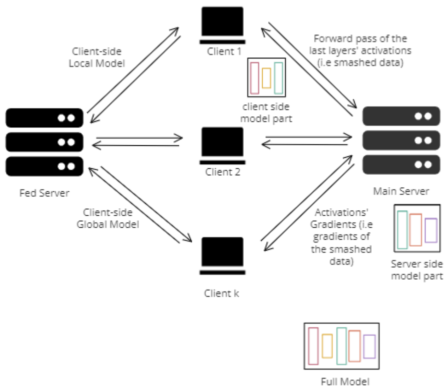


Fig. 5. SplitFed Workflow

### C. FedSL

FedSL starts by initializing the random seed and some parameters like the number of rounds, local epochs, learning rate, number of clients, batch size, number of classes and frac which is the percentage of clients that will participate in the process. Then the dataset is read, loaded split into train and test, then indices are added and then the dataset is split across the users. Then the model is built using a Recurrent Neural Network RNN that is split into two parts first and second.

Then the training and testing processes begin, we iterate on the number of epochs and for each epoch we iterate on the number of clients.The Workflow of FedSL is illustrated in Figure 6.

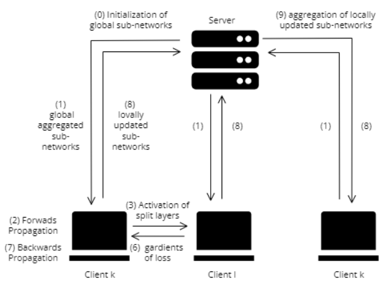


Fig. 6. FedSL Workflow

### D. PyVertical

PyVertical starts by defining the SplitNN class with the models, optimizers, data and remote tensors, inside the class function for the forward propagation of data and backwards propagation of gradients is defined as well. Then PyVertical makes sure that the data is correctly sorted by ids, and then Private Set Intersection is used to compute the intersection between the two sets while they are encrypted so no set is revealed to the other set except for the elements in the intersection in our case the ids. Then The model is defined which consists of two layers and 2 virtual workers are defined as well where one holds the labels while the other holds the data. The model is then split where each worker has a layer. The Workflow of PyVertical is illustrated in Figure 7.

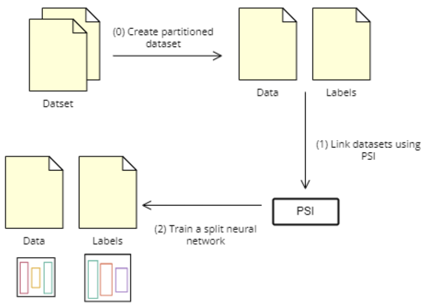


Fig. 7. PyVertical Workflow

## IV. Results

### A. Datasets

We used five different datasets to test and compare with. Figure 8 shows snapshots of the datasets after cropping and Table IV-A shows the details of the different datasets used.

### B. SplitFed

Table IV-B and Table IV-B shows the Accuracy of SFLV1 and SFLV2 respectively with 5 different emotion classification datasets.


Fig. 8. Face Emotion Classification Datasets

TABLE I
FACE EMOTION CLASSIFICATION DATASETS

| Dataset Name | Images | Subjects | Image | Date |
|---|---|---|---|---|
| FER2013plus [15] | 30,000 | 30,000 | 48x48 | 2016 |
| AffectNet [18] | 1,000,000 | 450,000 | 48x48 | 2017 |
| CKplus [20] | 980 | 123 | 48x48 | 2010 |
| KDEF [17] | 4900 | 70 | 562x762 | 1998 |
| ouluCASIA [19] | 1400 | 30 | 145x115 | 2011 |

### C. FedSL

Table IV-C shows the Accuracy of FedSL with 5 different emotion classification datasets.

### D. PyVertical

Table IV-D shows the Accuracy of PyVertical with 5 different emotion classification datasets.

### E. Comparison Graph

Figure 9 shows a comparison bar chart that compares the training accuracy of SplitFed Version 1, SplitFed Version 2, FedSL, PyVertical, Noraml Centralized learning, Federated Learning and Split Learning with the five cropped emotion classification datasets with 2 users. They are all comparable

TABLE II
THE RESULTS OF SFLV1 ON SEVERAL EMOTION CLASSIFICATION DATASETS

| Dataset | Epochs | clients | train accuracy | test accuracy |
|---|---|---|---|---|
| KDEF | 1000 | 2 | 99.7% | 70.4% |
| FER2013plus full | 100 | 2 | 87.27% | 51.67% |
| CKplus | 100 | 2 | 98.599% | 26.40% |
| ouluCASIA | 100 | 2 | 96.5% | 90.2% |
| AffectNet | 100 | 2 | 87.7% | 58.9% |

TABLE III
THE RESULTS OF SFLV2 ON SEVERAL EMOTION CLASSIFICATION DATASETS

| Dataset | Epochs | clients | train accuracy | test accuracy |
|---|---|---|---|---|
| KDEF | 100 | 2 | 96.37% | 64.28% |
| FER2013plus full | 100 | 2 | 90.74% | 52.73% |
| CKplus | 100 | 2 | 98.9% | 100% |
| ouluCASIA | 100 | 2 | 97.8% | 96.52% |
| AffectNet | 35 | 2 | 70.691 % | 52.4% |

TABLE IV
THE RESULTS OF FEDSL ON SEVERAL EMOTION CLASSIFICATION DATASETS

| Dataset | Epochs | clients | train accuracy | test accuracy |
|---|---|---|---|---|
| FER2013plus full | 20 | 2 | 36.34% | 36.51% |
| CKplus | 40 | 2 | 25.83% | 26.40% |
| ouluCASIA | 40 | 2 | 16.78% | 16.6% |
| AffectNet | 20 | 2 | 16.78% | 16.61% |

TABLE V
THE RESULTS OF PYVERTICAL ON SEVERAL EMOTION CLASSIFICATION DATASETS

| Dataset | Epochs | Clients | train accuracy | train loss |
|---|---|---|---|---|
| KDEF | 1000 | 2 | 95.5% | 0.198 |
| FER2013plus Full | 1000 | 2 | 99.97% | 0.009 |
| CKplus | 1000 | 2 | 100% | 0.007 |
| AffectNet | 895 | 2 | 99.91% | 0.015 |
| ouluCASIA | 1000 | 2 | 91.765% | 0.287 |

to the normal learning accuracy except for FedSL which has relatively low accuracies for all five datsets.

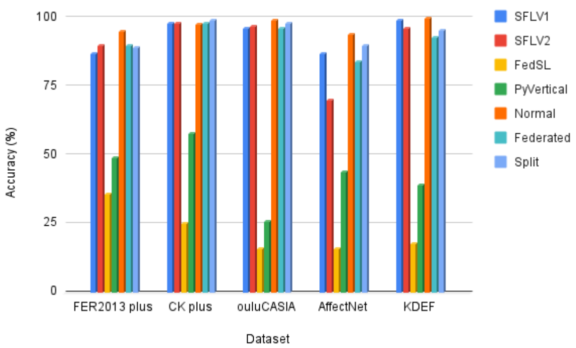


Fig. 9. Comparison Graph

## CONCLUSION

In this paper, we implement a Face Emotion classifier that classifies the emotion from a given image cropped so that only the upper part of the face is visible. The classifier was implemented using Split Federated Learning which is a novel technique that combines Split Learning and Federated Learning to minimize the computation time of Split Learning and overcome resource constraints of Federated Learning while preserving the privacy of the users' faces. We compare and test the three proposed Split Federated Learning algorithms SplitFed Version 1, SplitFed Version 2, FedSL and PyVertical. The overall performance of the classifier with SplitFed and PyVertical was good with relatively high accuracies up to 99 percent with some datasets, While FedSL's performance was not as good with relatively low accuracies with a maximum of 35 percent.

## REFERENCES

[1] Padgett, Curtis, and Garrison Cottrell. "Representing face images for emotion classification." Advances in neural information processing systems 9 (1996).
[2] Bouzakraoui, Moulay Smail, Abdelalim Sadiq, and Nourddine Enneya. "A Customer Emotion Recognition through Facial Expression using POEM descriptor and SVM classifier." Proceedings of the 2nd international Conference on Big Data, Cloud and Applications. 2017.
[3] Chua, Ming Hui, et al. "Face masks in the new COVID-19 normal: materials, testing, and perspectives." Research 2020 (2020).
[4] Turina, Valeria, et al. "Combining split and federated architectures for efficiency and privacy in deep learning." Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies. 2020.
[5] Thapa, Chandra, et al. "Splitfed: When federated learning meets split learning." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 8. 2022.
[6] Romanini, Daniele, et al. "Pyvertical: A vertical federated learning framework for multi-headed splitnn." arXiv preprint arXiv:2104.00489 (2021). Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271.
[7] Abedi, Ali, and Shehroz S. Khan. "FedSL: Federated Split Learning on Distributed Sequential Data in Recurrent Neural Networks." arXiv preprint arXiv:2011.03180 (2020).
[8] Liu, Xiaolan, Yansha Deng, and Toktam Mahmoodi. "Energy Efficient User Scheduling for Hybrid Split and Federated Learning in Wireless UAV Networks." IEEE ICC 2022. IEEE Communications Society, 2022.
[9] Gao, Yansong, et al. "End-to-end evaluation of federated learning and split learning for internet of things." arXiv preprint arXiv:2003.13376 (2020).
[10] Ads, Omar S., Mayar M. Alfares, and Mohammed A-M. Salem. "Multi-limb Split Learning for Tumor Classification on Vertically Distributed Data." 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE, 2021.
[11] Elshabrawy, Khaled M., Mayar M. Alfares, and Mohammad A-M. Salem. "Ensemble Federated Learning for Non-II D COVID-19 Detection." 2022 5th International Conference on Computing and Informatics (ICCI). IEEE, 2022.
[12] Kairouz, Peter, et al. "Advances and open problems in federated learning." Foundations and Trends® in Machine Learning 14.1–2 (2021): 1-210.
[13] Joshi, Praveen, et al. "Splitfed learning without client-side synchronization: Analyzing client-side split network portion size to overall performance." arXiv preprint arXiv:2109.09246 (2021).
[14] Pathak, Reese, and Martin J. Wainwright. "FedSplit: An algorithmic framework for fast federated optimization." Advances in Neural Information Processing Systems 33 (2020): 7057-7066.
[15] Zahara, Lutfiah, et al. "The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi." 2020 Fifth international conference on informatics and computing (ICIC). IEEE, 2020.
[16] Liliana, Dewi Yanti. "Emotion recognition from facial expression using deep convolutional neural network." Journal of physics: conference series. Vol. 1193. No. 1. IOP Publishing, 2019.
[17] Calvo, Manuel G., and Daniel Lundqvist. "Facial expressions of emotion (KDEF): Identification under different display-duration conditions." Behavior research methods 40.1 (2008): 109-115.
[18] Mollahosseini, Ali, Behzad Hasani, and Mohammad H. Mahoor. "Affectnet: A database for facial expression, valence, and arousal computing in the wild." IEEE Transactions on Affective Computing 10.1 (2017): 18-31.
[19] Zhao, Guoying, et al. "Facial expression recognition from near-infrared videos." Image and vision computing 29.9 (2011): 607-619.
[20] Adhikary, Abhijit, and Namas Bhandari. "PosEmotion-Combining Real-Time 2D Body Pose Estimation and Facial Emotion Recognition to Analyze Human Behavior." 2021 26th International Conference on Automation and Computing (ICAC). IEEE, 2021.