Efficiency and Privacy in Deep Learning Using Split Federated Learning

Dinah Waref Mohamed Lamy Shalaby (dinah.shalaby@guc.edu.eg)
Supervised By: Dr. Mohammed Abdel-Megeed Salem (mohammed.salem@guc.edu.eg)
Faculty of Media Engineering and Technology
German University in Cairo
2022

In this work, we aim to combine the Split Learning architecture and the Federated Learning architecture to implement a non-privacy invasive face emotion classification application. This combined architecture will offer us more security with decreased computational time.

Problem Statement

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

Motivation

As users we benefit from personalization which makes our lives easier and more efficient, as researchers the models can predict, analyze and capture patterns that would otherwise be time consuming for a human and even a human can miss them and of course companies have thrived in this new data era where it has better metrics for business and offering personalized services.

The problem lies here, for better research, metrics 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.

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. Further more in some cultures like in South Korea and China, wearing face masks is common to prevent fine dust or tiny air pollutants being inhaled. In addition to fine dust, young people in Asian countries tend to wear face masks as they offer a sense of security and anonymity as they attract less attention in addition it makes the wearer difficult to recognize. Some people choose to cover their faces by Niqab in the middle east or ghoonghat in India for religious purposes. Tuareg people cover their faces with Litham which is a mouth veil for functional reasons, to protect them from the sun and sand as they live in the Sahara desert in Africa. Like Tuareg, Berbers in Morocco, Bedouins in Egypt and Tubu people in Chad, all wear Litham to cover their faces for the same reasons.
It has hindered social communication by limiting the ability of inferring emotion from facial expressions. Recognizing emotions from facial expressions is important for initiating and maintaining healthy social relationships.

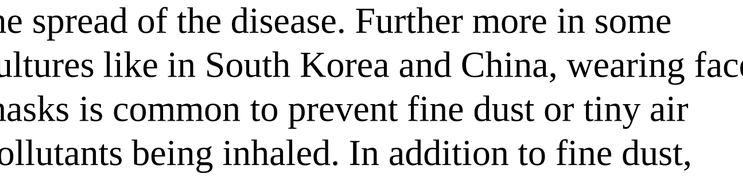


Figure (1): Niqab, Facemasks and Litham examples

Many Leading companies have taken the initiative to used emotion detection to their advantage. For example, Disney created an algorithm that can determine how the audience like it's 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.

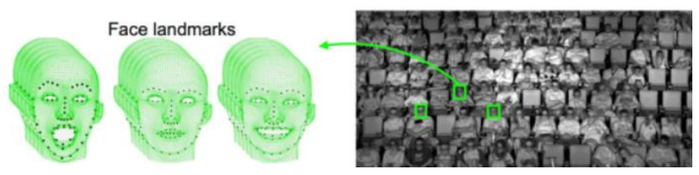


Figure (2): Disney using camera in cinemas to detect emotions

Dataset

The Karolinska directed emotional faces KDEF dataset contains 4900 pictures of human facial expressions. The set of pictures contains 70 individuals displaying 7 different emotional expressions.

Figure (1): The cropping process

We created a script that first detects the face in the image and then we crop the image using the bounding boxes detected by the classifier. Then we take the new cropped image and we further crop it's length so that the resulting image is of the upper part of the face i.e the eyes and forehead.

Figure (2): Snapshots of the KDEF dataset after cropping

Methodology

The Split Federated Architecture splits the neural network as in the SplitNN Algorithm into two parts. Then the training process is done by each client-server pair simultaneously. After all the clients' data is used to update the parameters, The server-side weights are aggregated and updated.

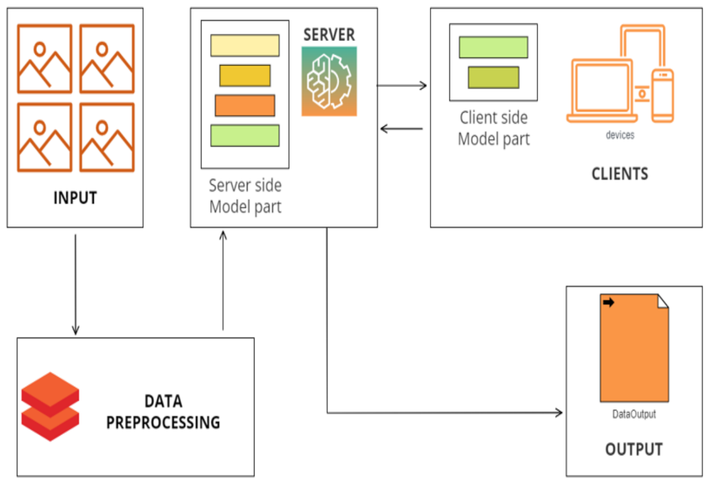


Figure (3): Block Diagram illustrating the methodology

▷ SplitFed:

The client propagates it's smashed data to the server which is used to update the model weights and then the gradients are used to update the client-side model.

▷ PyVertical:

The Algorithm starts by vertically partioning the dataset then calculates the intersection using Private Set Intersection (PSI) then starts the splitNN training.

▷ FedSL:

The Recurrent Neural Networks is split between clients and the server aggregates the model to create a global model.

Results

▷ SplitFed:

1)SFLV1: training accuracy was 99.7% and the testing accuracy was 70.4% with 2 clients and 1000 epochs.

2)SFLV2: training accuracy was 96.37% and the testing accuracy was 64.28% with 2 clients and 100 epochs

▷ PyVertical:

training accuracy was 95.5% with 2 clients and 1000 epochs

▷ FedSL:

training accuracy was 20% and the testing accuracy was 19% with 2 clients and 20 epochs

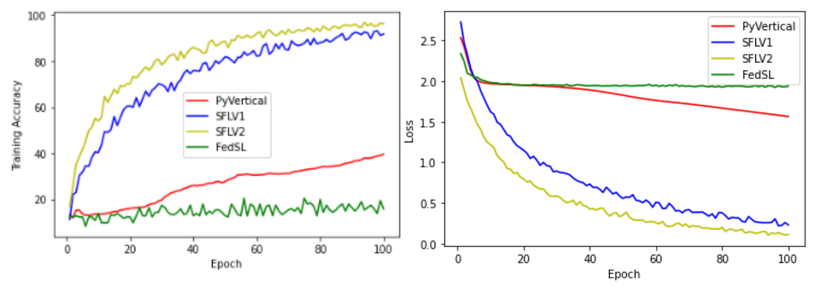
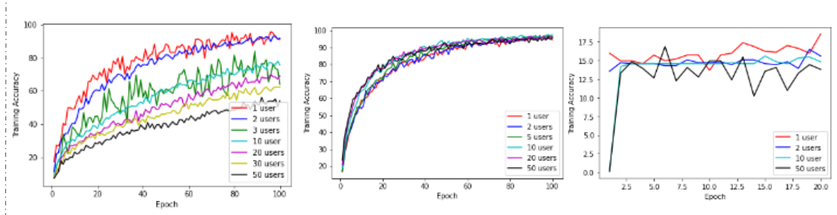


Figure (4): Accuracy & Loss comparison graphs on KDEF dataset

To further test and compare the algorithms we tested them on different number of users on the same dataset.



SFLV1 SFLV2 FedSL

Figure (5): Accuracy with different number of users comparison graphs on KDEF dataset