Stay Home Safe with Starving Federated Data

Session TS10-B (Learning Algorithm Development, Analysis and Interpretability)

Jaechul (Harry) Roh (HKUST)

Dr. Yajun Fang (Universal Village)

Table of Content

Sections 1 & 2: Background Information

- Adversarial Attack / Defense
- FGSM / FFGSM / Square Attack
- Federated Learning
- Face Recognition

Section 3. Our Approach

- FLATS (Method 1 / Method 2)
- Adversarial Batch Ratio

Section 4. Experiments and Results

- Benign / Robust Federated Learning
- Data Manipulation

Section 5 & 6: Summary and Evaluation

- Limitations / Novel Findings

IEEE 6th International Conference on Universal Village · UV2022 · Session TS7-A

Stay Home Safe with Starving Federated Data

Jaechul Roh
Dept. of Electronic and Computer Engineering
HKUST
Hong Kong, Hong Kong
jroh@connect.ust.hk

Yajun Fang*
Universal Village Society
1 Broadway, Cambridge, MA, 02142
yjfang@mit.edu

***Abstract*—Over the past few years, the field of adversarial attack received numerous attention from various researchers with the help of successful attack success rate against well-known deep neural networks that were acknowledged to achieve high classification ability in various tasks. However, majority of the experiments were completed under a single model, which we believe it may not be an ideal case in a real-life situation. In this paper, we introduce a novel federated adversarial training method for smart home face recognition, named FLATS, where we observed some interesting findings that may not be easily noticed in a traditional adversarial attack to federated learning experiments. By applying different variations to the hyperparameters, we have spotted that our method can make the global model to be robust given a starving federated environment. Our code can be found on `https://github.com/jcroh0508/FLATS`.**

***Index Terms*—adversarial attack, robustness, federated learning, smart home, face recognition**

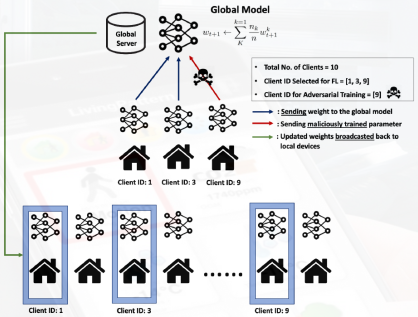


Fig. 1. General Architecture of FLATS

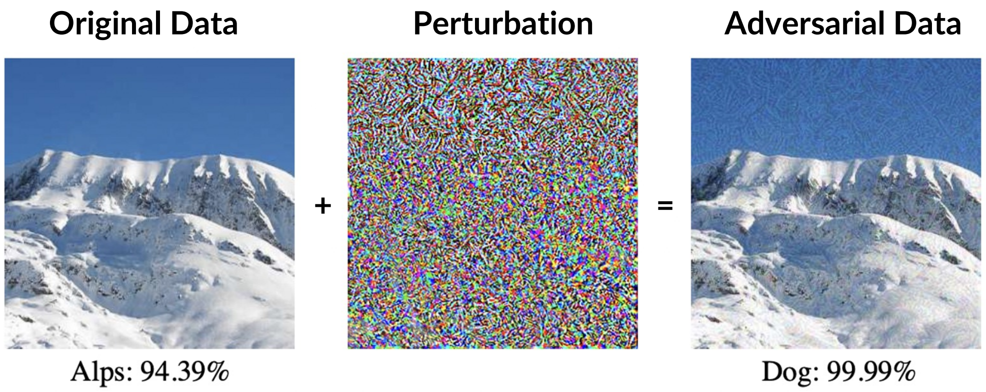
I. Introduction

The introduction of Deep Neural Networks (DNNs) to the field of machine learning grasped the attention of numerous researchers by achieving the classification ability to almost

Sections 1 & 2: Background Information

Adversarial Attack

- **Adversarial Examples** input data with an imperceptible change
- **Adversarial Examples** = Original data (x) + Perturbation with noise (ϵ)
- **Adversarial Attack** induce misclassification in purpose to make machine learning models more **ROBUST**



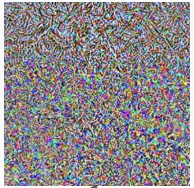
Real-Life Adversarial Attack (Smart Home)

(1) INPUT (x)

x ("DiCaprio")

+

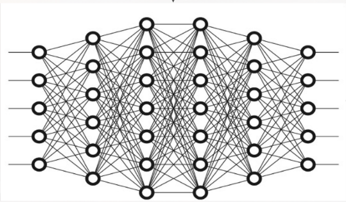
Adversarial Noise



(2) ADVERSARIAL EXAMPLE (x^*)

x^*

(3) DEEP NEURAL NETWORK



"David Beckham"

(4) OUTPUT

DOOR OPEN

(5) ACTION

(6) CONSEQUENCE

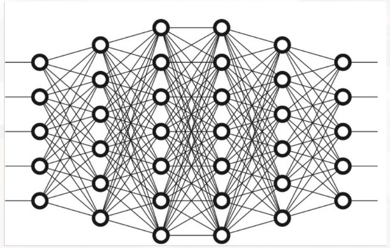
Adversarial Defense (Training)

ATTACK (Step 1)

Input

Adversarial Examples

Deep Learning Model



Output

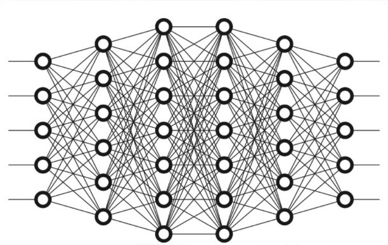
"David Beckham"

DEFENSE (Step 2)

Clean Data

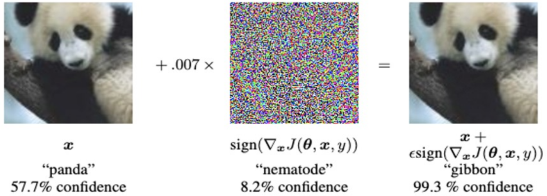
+

Adversarial Examples



"Leonardo DiCaprio"

FGSM (Fast Gradient Sign Method)



Goodfellow et al. 2014

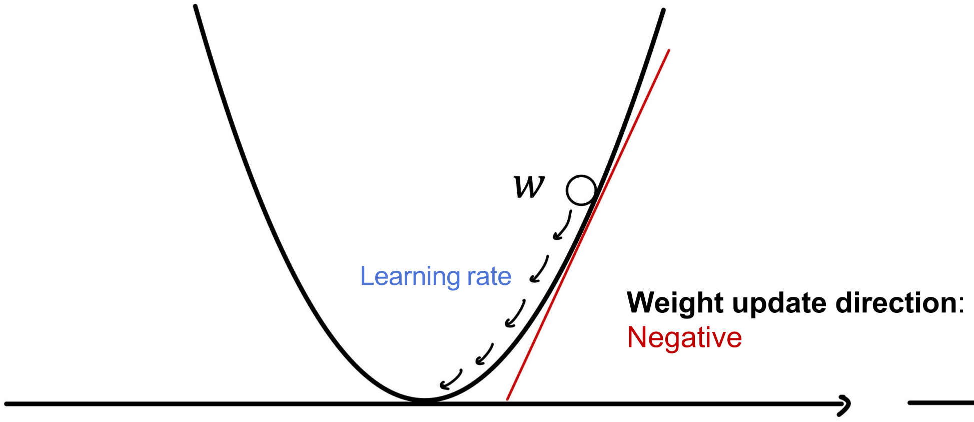
- **Gradient Descent Algorithm**

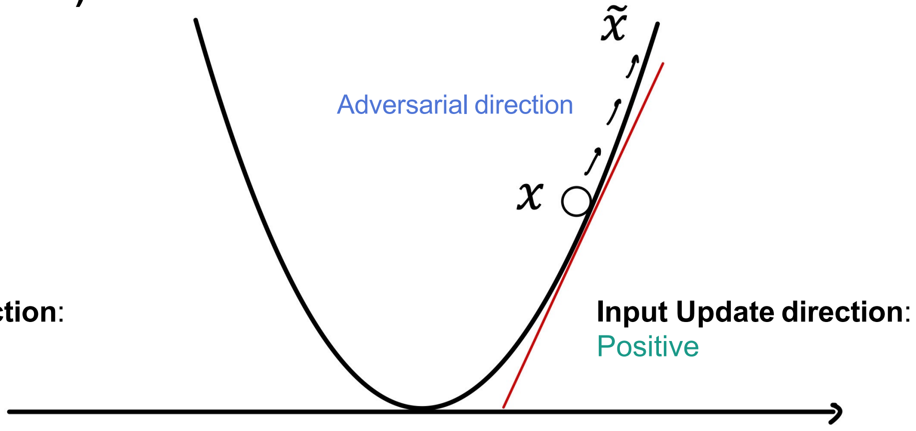
  OPPOSITE direction of the gradient of the cost function

- **Fast Gradient Sign Method (FGSM)**

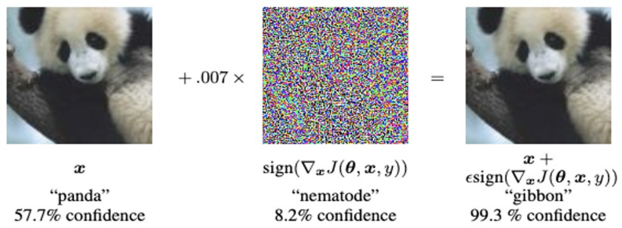
  SAME direction of the gradient of the cost function

sign(gradient): +





FGSM (Cont.)



Goodfellow et al. 2014

$$\underbrace{x + \epsilon \cdot sign(\underbrace{\nabla_x \underbrace{J(\theta, x, y)}_{\text{Loss Function}}}_{\text{Loss Function Gradient}})}_{\text{Adversarial Example}}$$

FGSM (Cont.)

Input
(panda image)

Loss Function

$$L(x, x^*) = \epsilon \cdot sign(\nabla_x J(\theta, x, y))$$

x ("panda")

x^* ("gibbon")

Parameters of
the model

Output
("panda" label)

Adversarial Boundary

Deciding perturbation

FGSM uses the ***"max norm constraint":***

$$(In\ all\ definitions, x = (x_1, x_1, \dots, x_n))$$

$$L^\infty\ distance:\ \| x \|_\infty = \max_{1 \le i \le n} |x_i| \qquad L^1\ distance:\ \| x \|_1 = \sum_{i=1}^{n} |x_i|$$

L^∞ : moving **many pixels as possible** but only by a small number

L^1 : **summed absolute value difference** between x and x^*

Adversarial Defense (FGSM)

$$\underbrace{\tilde{J}(\theta, x, y)}_{(3)} = \alpha \cdot \underbrace{J(\theta, x, y)}_{(1)} + (1 - \alpha) \cdot \underbrace{J(\theta, \tilde{x}, y)}_{(2)}$$

(1) $\tilde{J}(\theta, x, y)$: loss function of the original data

(2) $J(\theta, \tilde{x}, y)$: loss function of the adversarial example

(3) $J(\theta, x, y)$: loss function of both original data and adversarial example

α : proportion of applying loss between original data and adversarial example

Universal Village Society

Fast Adversarial Training using FGSM (FFGSM)

Efficient training techniques added to FGSM

- Cyclic learning rates
- Mixed-precision training

FAST IS BETTER THAN FREE:
REVISITING ADVERSARIAL TRAINING

Eric Wong*
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
ericwong@cs.cmu.edu

Leslie Rice*
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
larice@cs.cmu.edu

J. Zico Kolter
Computer Science Department
Carnegie Mellon University and
Bosch Center for Artifical Intelligence
Pittsburgh, PA 15213, USA
zkolter@cs.cmu.edu

Method	Standard accuracy	PGD ($\epsilon = 8/255$)	Time (min)
FGSM + DAWNBench			
+ zero init	85.18%	0.00%	12.37
+ early stopping	71.14%	38.86%	7.89
+ previous init	86.02%	42.37%	12.21
+ random init	85.32%	44.01%	12.33
+ $\alpha = 10/255$ step size	83.81%	46.06%	12.17
+ $\alpha = 16/255$ step size	86.05%	0.00%	12.06
+ early stopping	70.93%	40.38%	8.81
"Free" ($m = 8$) (Shafahi et al., 2019)[1]	85.96%	46.33%	785
+ DAWNBench	78.38%	46.18%	20.91
PGD-7 (Madry et al., 2017)[2]	87.30%	45.80%	4965.71
+ DAWNBench	82.46%	50.69%	68.8

Square Attack (Black-Box Attack)

Key Concept of Square Attack

- Based on **randomized** search scheme
- Perturbation situated at **boundary of feasible set**

Square Attack: a query-efficient black-box adversarial attack via random search

Maksym Andriushchenko*[1], Francesco Croce*[2], Nicolas Flammarion[1], and Matthias Hein[2]

[1] EPFL
[2] University of Tübingen

Table 2. Results of **untargeted** attacks on ImageNet with a limit of 10,000 queries. For the l_∞-attack we set the norm bound $\epsilon = 0.05$ and for the l_2-attack $\epsilon = 5$. Models: normally trained **I**: Inception v3, **R**: ResNet-50, **V**: VGG-16-BN. The Square Attack outperforms for both threat models all other methods in terms of success rate and query efficiency. The missing entries correspond to the results taken from the original paper where some models were not reported

Norm	Attack	Failure rate			Avg. queries			Med. queries		
		I	R	V	I	R	V	I	R	V
l_∞	Bandits [31]	3.4%	1.4%	2.0%	957	727	394	218	136	36
	Parsimonious [49]	1.5%	-	-	722	-	-	237	-	-
	DFO_c–CMA [39]	0.8%	**0.0%**	0.1%	630	270	219	259	143	107
	DFO_d–Diag. CMA [39]	2.3%	1.2%	0.5%	424	417	211	**20**	20	2
	SignHunter [2]	1.0%	0.1%	0.3%	471	129	95	95	39	43
	Square Attack	**0.3%**	**0.0%**	**0.0%**	**197**	**73**	**31**	24	**11**	**1**
l_2	Bandits [31]	9.8%	6.8%	10.2%	1486	939	511	660	392	196
	SimBA-DCT [28]	35.5%	12.7%	7.9%	**651**	**582**	452	564	467	360
	Square Attack	**7.1%**	**0.7%**	**0.8%**	1100	616	**377**	**385**	**170**	**109**

Federated Learning

Step 1: Train local model
Step 2: Send parameters to global server
Step 3: Federate all the parameters
Step 4: Send back the updated parameters to the local devices

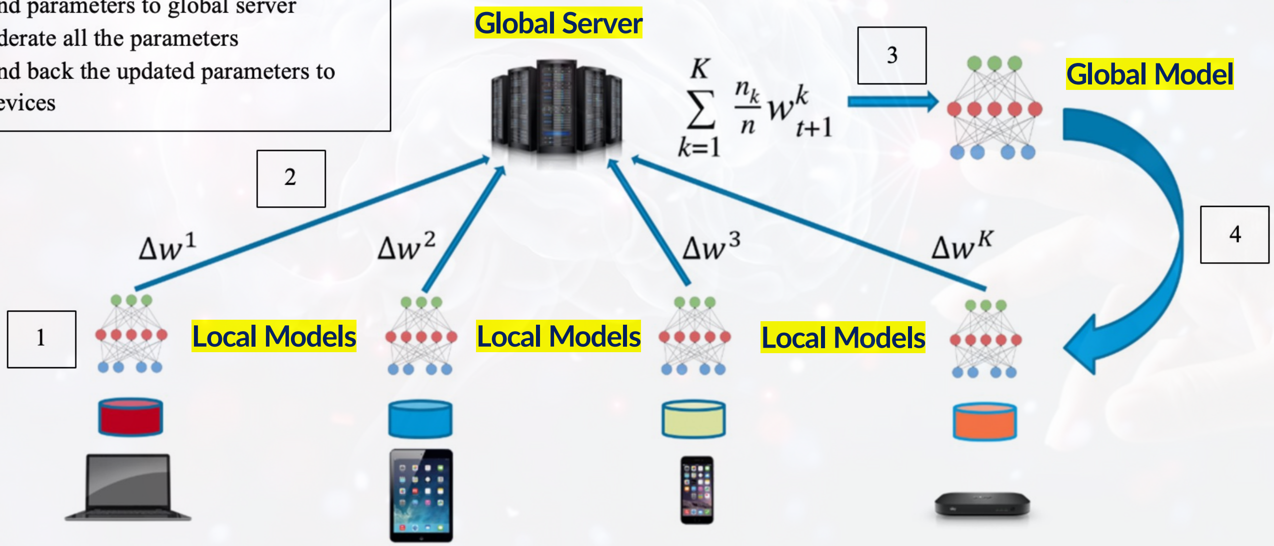


image source: https://proandroiddev.com/federated-learning-e79e054c33ef

Brendan McMahan

Advantages of Federated Learning

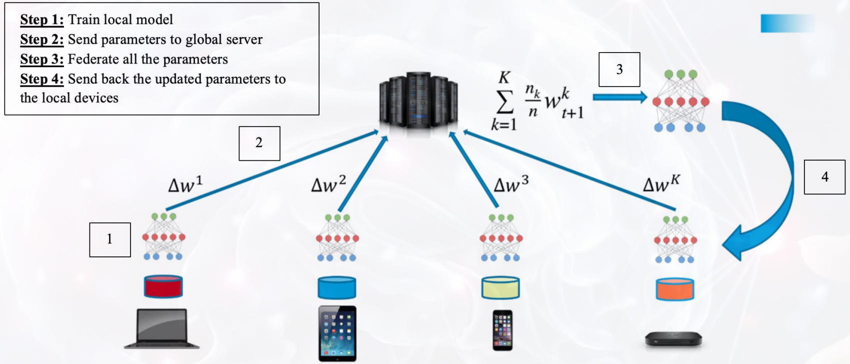


image source: https://proandroiddev.com/federated-learning-e79e054c33ef

1. **Data Security**: local models do not have to send their private data

2. **Hardware Efficiency**: training only conducted within distributed local devices

3. **Data Diversity**: Wider range of data utilized in each training process

Federated Averaging (FedAvg) Algorithm

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \ldots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w^k_{t+1} \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w^k_{t+1}$

ClientUpdate(k, w): *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

$$w_{t+1} \leftarrow \sum_{K}^{k=1} \frac{n_k}{n} w^k_{t+1}$$

n	Total Data Size
K	Total No. of Clients
n_k	Data Size of Client k
w^k_{t+1}	Weight of Client k at Time Step $t+1$
w_{t+1}	Global Aggregated Parameter

Face Recognition

Killing Two Birds with One Stone: Efficient and Robust Training of Face Recognition CNNs by Partial FC

Xiang An [1,3] Jiankang Deng * [2,3] Jia Guo [3]
Ziyong Feng [1] XuHan Zhu [4] Jing Yang [3] Tongliang Liu [5]
[1]DeepGlint [2]Huawei [3]InsightFace
[4]Peng Cheng Laboratory [5]University of Sydney
{xiangan,ziyongfeng}@deepglint.com, tongliang.liu@sydney.edu.au
{jiankangdeng,guojia,zhuxuhan.research,y.jing2016}@gmail.com

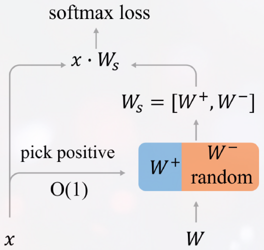


Figure 1. PFC picks the positive center by using the label and randomly selects a significantly reduced number of negative centers to calculate partial image-to-class similarities. PFC kills two birds (efficiency and robustness) with one stone (partial sampling).

ArcFace: Additive Angular Margin Loss for Deep Face Recognition

Jiankang Deng * [1,2,3] Jia Guo * [2] Niannan Xue [1] Stefanos Zafeiriou [1,3]
[1]Imperial College London [2]InsightFace [3]FaceSoft
{j.deng16, n.xue15, s.zafeiriou}@imperial.ac.uk , guojia@gmail.com

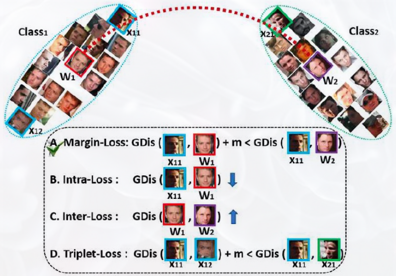


Figure 1. Based on the centre [15] and feature [35] normalisation, all identities are distributed on a hypersphere. To enhance intra-class compactness and inter-class discrepancy, we consider four kinds of Geodesic Distance (GDis) constraint. (A) Margin-Loss: insert a geodesic distance margin between the sample and centres. (B) Intra-Loss: decrease the geodesic distance between the sample and the corresponding centre. (C) Inter-Loss: increase the geodesic distance between different centres. (D) Triplet-Loss: insert a geodesic distance margin between triplet samples. In this paper, we propose an Additive Angular Margin Loss (ArcFace), which is exactly corresponded to the geodesic distance (Arc) margin penalty in (A), to enhance the discriminative power of face recognition model. Extensive experimental results show that the strategy of (A) is most effective.

CosFace: Large Margin Cosine Loss for Deep Face Recognition

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li*, and Wei Liu*
Tencent AI Lab
{hawelwang,yitongwang,encorezhou,denisji,sagazhou,michaelzfli}@tencent.com
gongdihong@gmail.com wliu@ee.columbia.edu

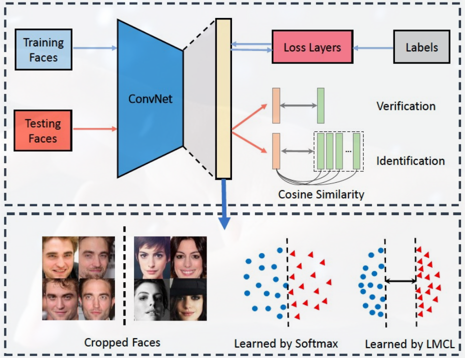
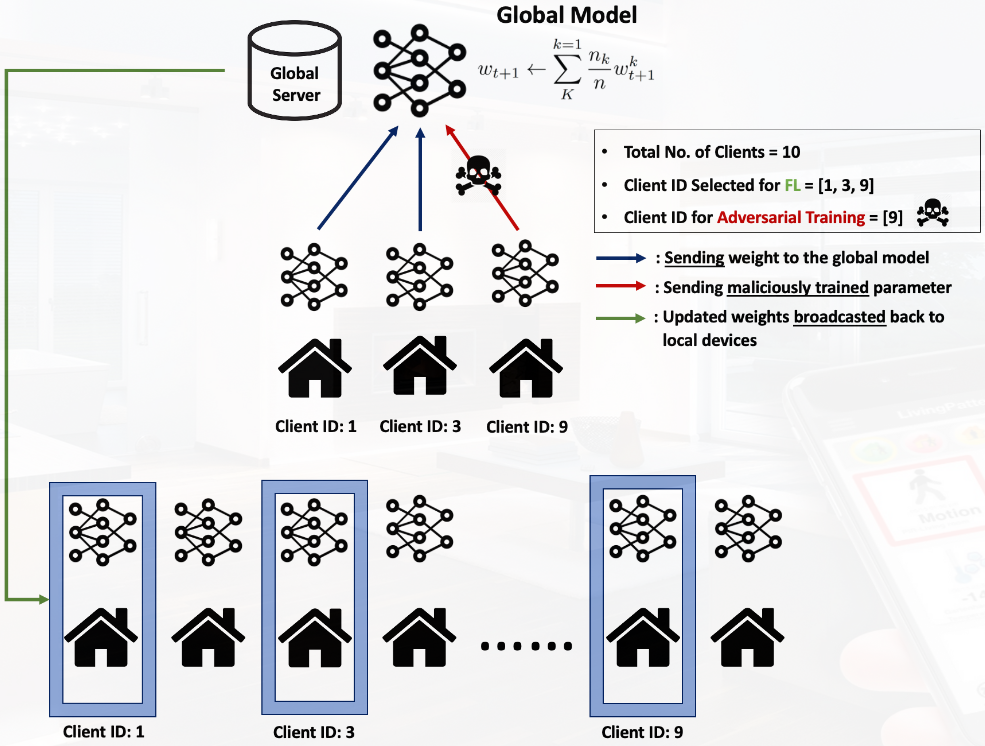


Figure 1. An overview of the proposed CosFace framework. In the training phase, the discriminative face features are learned with a large margin between different classes. In the testing phase, the testing data is fed into CosFace to extract face features which are later used to compute the cosine similarity score to perform face verification and identification.

Section 3: Our Approach

FLATS : Federated Learning Adversarial Training for Smart Home Face Recognition System



FLATS (Method 1)

Algorithm 1 FLATS (Method I)

N = Total global rounds
J = Total no. of clients
d = Total data size
d_j = Data size of client j
n = No. of clients selected every round
n_a = No. of clients to go through adversarial training
w_g = Global model parameter
$Clients \leftarrow [w_1, w_2, w_3, ..., w_k]$
$RoundClients \leftarrow []$
$AdvClients \leftarrow []$

for N **do**
  $UpdatedWeights \leftarrow []$
  (1) $RoundClients \leftarrow$ Random($Clients$, n)
  (2) $AdvClients \leftarrow$ Random($RoundClients$, n_a)
  for $i \leftarrow RoundClients$ **do**
    (3) **if** i is in $AdvClients$ **then**
      $newW \leftarrow$ AdvTraining($Clients$[i])
      $UpdatedWeights \leftarrow newW$
    (4) **else**
      $newW \leftarrow$ ClientUpdate(i, $Clients$[i])
      $UpdatedWeights \leftarrow$newW
    end if
  end for
  (5) $w_g \leftarrow$ FedAvg($UpdatedWeights$)
end for

Guarantee for Adversarial Training in Each Global Round

(1) Randomly select client IDs to be trained at each global round

(2) Randomly select client IDs for adversarial training

(3) If the ID is in *AdvClients:*

- Adversarial Training

(4) It not:

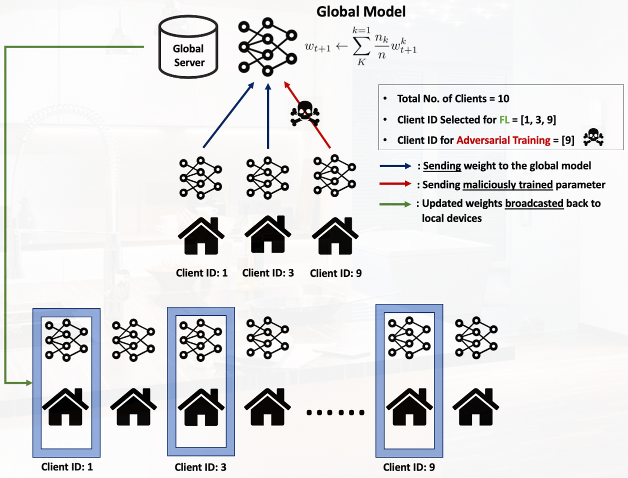
- Standard Training

(5) `FedAvg`

- Save **"global parameter"**
- Broadcast back to local devices

FLATS (Method 2)

No Guarantee for Adversarial Training in Each Global Round



Algorithm 2 FLATS (Method II)

N = Total global rounds
J = Total no. of clients
d = Total data size
d_j = Data size of client j
n = No. of clients selected every round
n_a = No. of clients to go through adversarial training
w_g = Global model parameter
$Clients \leftarrow [w_1, w_2, w_3, ..., w_k]$
$RoundClients \leftarrow []$
$AdvClients \leftarrow []$

$\boxed{AdvClients \leftarrow \text{Random}(RoundClients,\ n_a)}$
for N **do**
    $UpdatedWeights \leftarrow []$
    $RoundClients \leftarrow \text{Random}(Clients,\ n)$
    for $i \leftarrow RoundClients$ **do**
        if i is in $AdvClients$ **then**
            $newW \leftarrow \text{AdvTraining}(Clients[\text{i}])$
            $UpdatedWeights \leftarrow newW$
        else
            $newW \leftarrow \text{ClientUpdate}(i,\ Clients[\text{i}])$
            $UpdatedWeights \leftarrow newW$
        end if
    end for
    $w_g \leftarrow \texttt{FedAvg}(UpdatedWeights)$
end for

Select client IDs for Adversarial Training

in the BEGINNING

Adversarial / Clean Batch Ratio

`E.g. clean_train_batch_ratio = 0.25`

BATCH | 25% | 75% | BATCH

Clean Images | Adversarial Examples

```
ratio = int(len(train_loader) * clean_train_batch_ratio)
# Attack if current id is the selected id
if (client_id in attack_id_selected) and (idx >= ratio):
    # Create adversarial example
    images = atk(images, labels)

optimizer.zero_grad()
outs = local_model(images)
_, preds = torch.max(outs, 1)

# outs = torch.exp(outs)
loss = criterion(outs, labels)
loss.backward()
optimizer.step()
local_loss += loss.item()
```

Switch to Adversarial Example
AFTER specific "batch" ratio

Backpropagation
based on Adversarial Examples

Algorithm 1 FLATS (Method I)

for N **do**
  $UpdatedWeights \leftarrow []$
  $RoundClients \leftarrow \text{Random}(Clients, n)$
  $AdvClients \leftarrow \text{Random}(RoundClients, n_a)$
  for $i \leftarrow RoundClients$ **do**
    if i is in $AdvClients$ **then**
      $newW \leftarrow \text{AdvTraining}(Clients[\text{i}])$
      $UpdatedWeights \leftarrow newW$
    else
      $newW \leftarrow \text{ClientUpdate}(i, Clients[\text{i}])$
      $UpdatedWeights \leftarrow \text{newW}$
    end if
  end for
  $w_g \leftarrow \texttt{FedAvg}(UpdatedWeights)$
end for

Section 4: Experiments and Results

Starving Dataset

1. Data Size

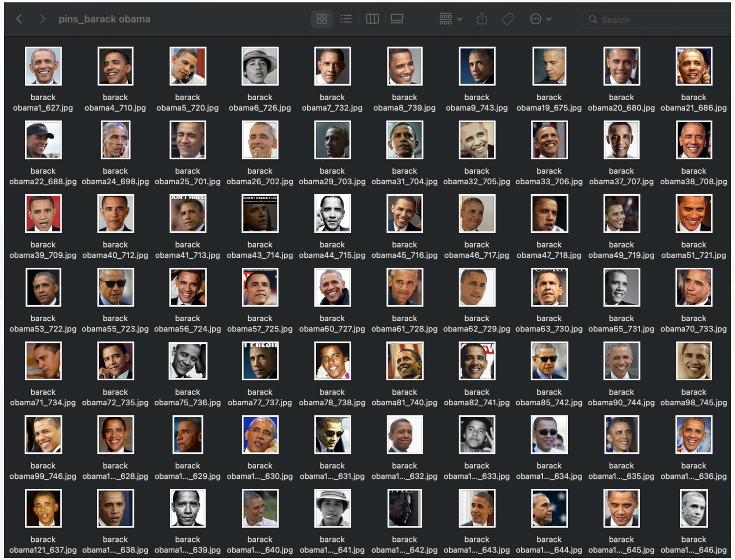
```
In [8]:
print("No. of All images: ", len(dataset))
print("Size of fist image: ", dataset[0][0].size())

No. of All images:  17534
Size of fist image:  torch.Size([3, 224, 224])
```

TOTAL Client = 5

Data size for each client = around 3506 (IID)

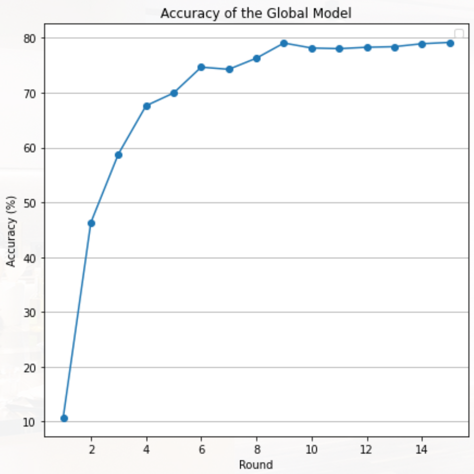
2. Model: ResNet-34 (97.8% classification accuracy)

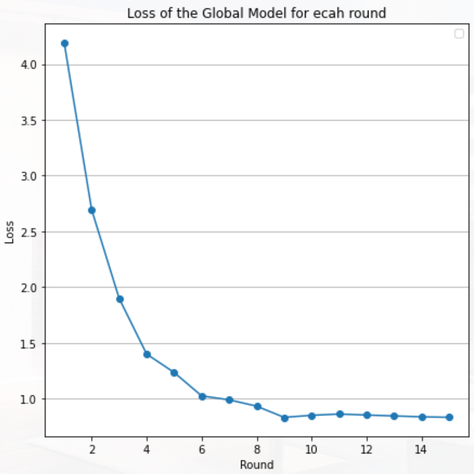
```python
class FaceRecog(nn.Module):
    def __init__(self, num_classes, pretrained=True):
        super(FaceRecog, self).__init__()

        # Pretrained resnet34
        self.resnet34 = models.resnet34(pretrained=True)
        for param in self.resnet34.parameters():
            param.requires_grad = False

        modified_fc = nn.Linear(in_features = fc_in_features, out_features=num_classes)
        self.resnet34.fc = modified_fc

    def forward(self, x):
        return self.resnet34(x)

    def summary(self, input_size):
        return summary(self, input_size)
```



4.1 Benign Federated Learning

Benign Federated Learning (IID)





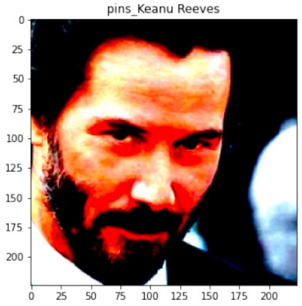
7 total clients

5 (71%) clients selected randomly

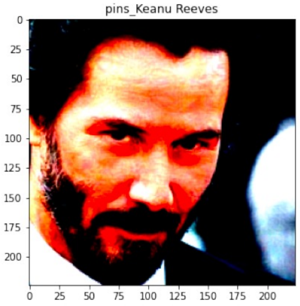
5 epochs per client

15 global rounds

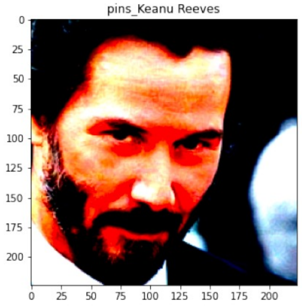
Adversarial Examples



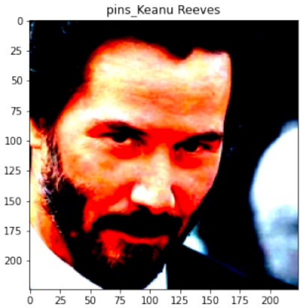
Original Image



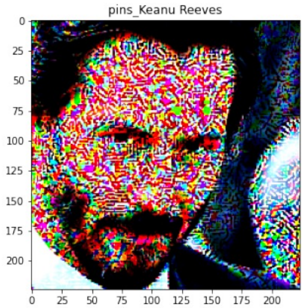
FGSM (ϵ=8/255)



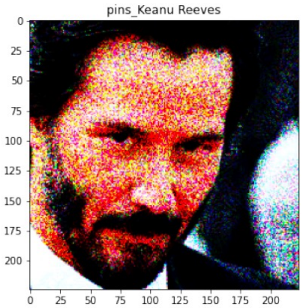
FFGSM (ϵ=8/255)



Square (ϵ=8/255, $n_{queries} = 2000$)



FGSM (ϵ=0.9)



FFGSM (ϵ=0.9)

Robust Acc. of Benign FL Model (IID)

CATASTROPHIC

Robust Acc.

TABLE I
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF BENIGN FEDERATED LEARNING METHOD

Global Acc.(%)	Robust Acc.(%)	Global Rounds	Clients Selected	Total Clients	Selected Proportion (%)
81.9	N/A	7	5	5	100
82.5	6.5	10	4	5	80
79.6	6.7	10	8	10	80
73.2	4.2	10	12	15	80
78.6	N/A	10	3	6	50
76.1	5.0	10	4	8	50
61.5	3.4	10	8	16	50
2.70	2.7	10	10	20	50
71.8	4.4	10	1	5	20
48.5	3.8	10	2	10	20
40.8	2.3	10	3	15	20
34.1	2.2	10	4	20	20

Adversarial Example: FGSM (ϵ = 8/255)

Benign FL Model vs. Robust FL Model (FLATS)

TABLE I
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF BENIGN FEDERATED LEARNING METHOD

Global Acc.(%)	Robust Acc.(%)	Global Rounds	Clients Selected	Total Clients	Selected Proportion (%)
81.9	N/A	7	5	5	100
82.5	6.5	10	4	5	80
79.6	6.7	10	8	10	80
73.2	4.2	10	12	15	80
78.6	N/A	10	3	6	50
76.1	5.0	10	4	8	50
61.5	3.4	10	8	16	50
2.70	2.7	10	10	20	50
71.8	4.4	10	1	5	20
48.5	3.8	10	2	10	20
40.8	2.3	10	3	15	20
34.1	2.2	10	4	20	20

Overall Increase

in Robust Acc.

TABLE II
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

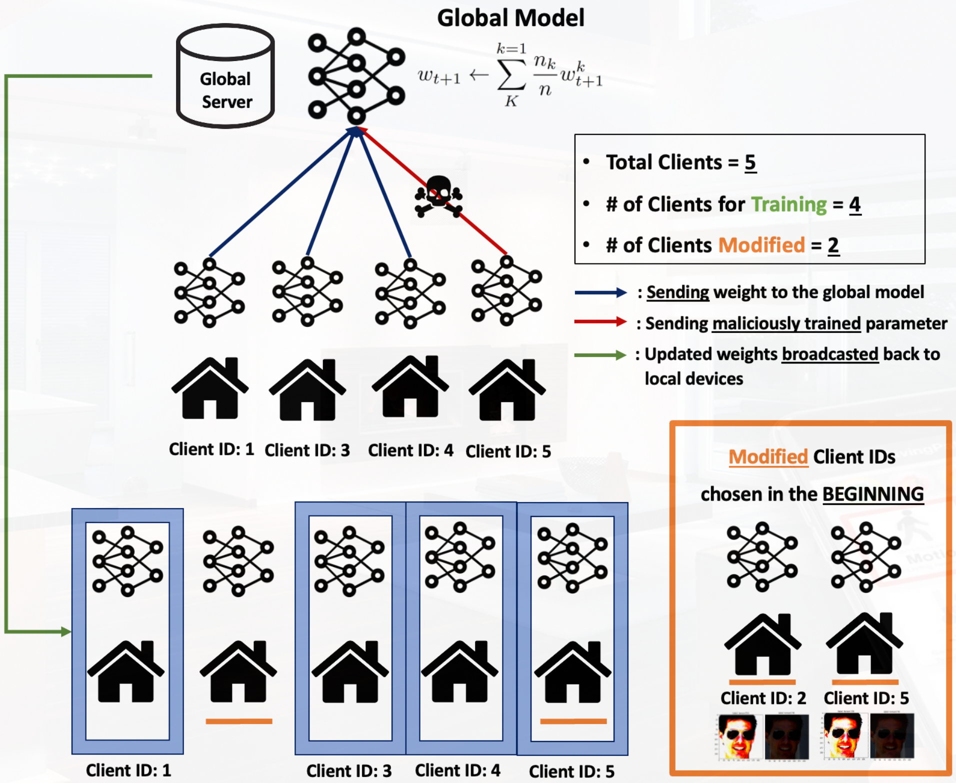
[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Adversarial Training: FFGSM (AlexNet, ϵ = 8/255)

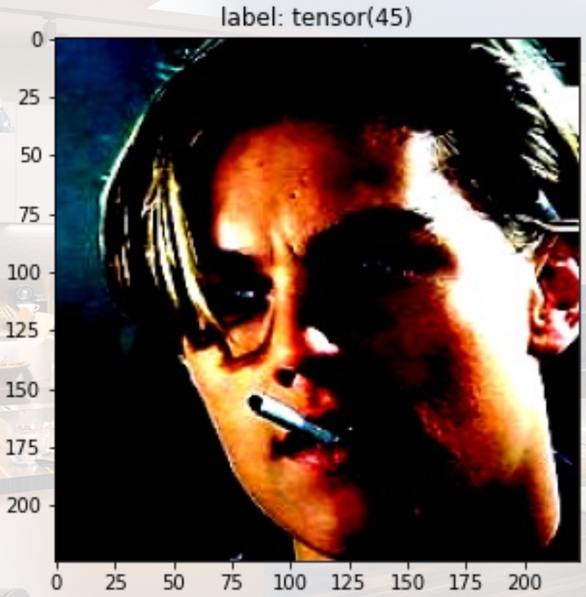
4.2 Data Manipulation (Non-IID)

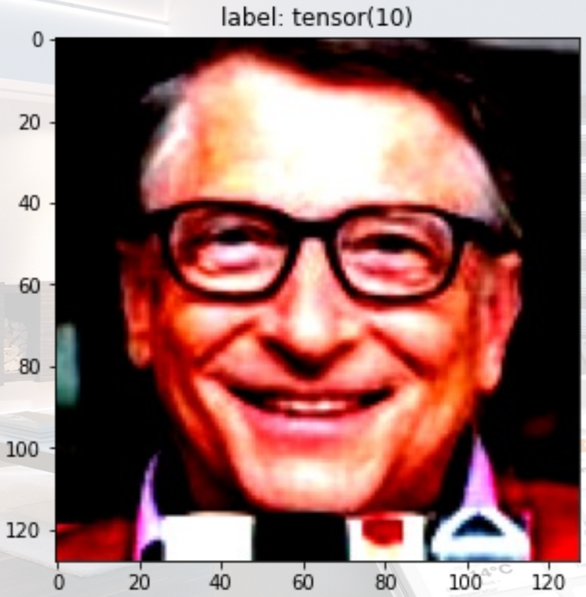
1. Pixel | 2. “Eye” Cover | 3. Brightness | 4. Test Data Augmentation

Default Setting

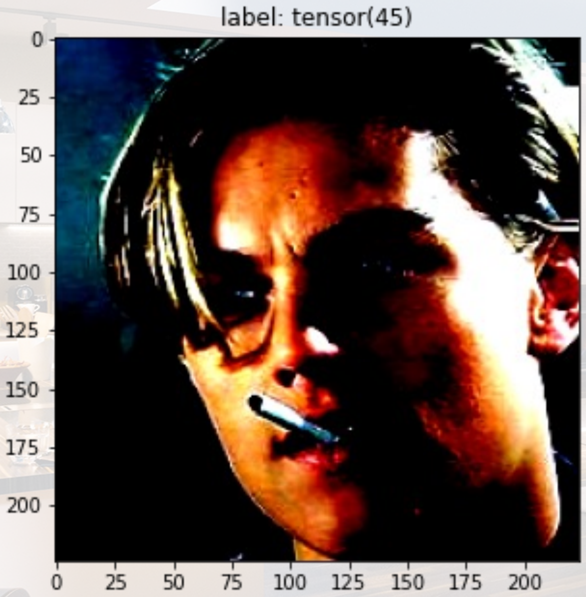


1.1 Pixel Comparison

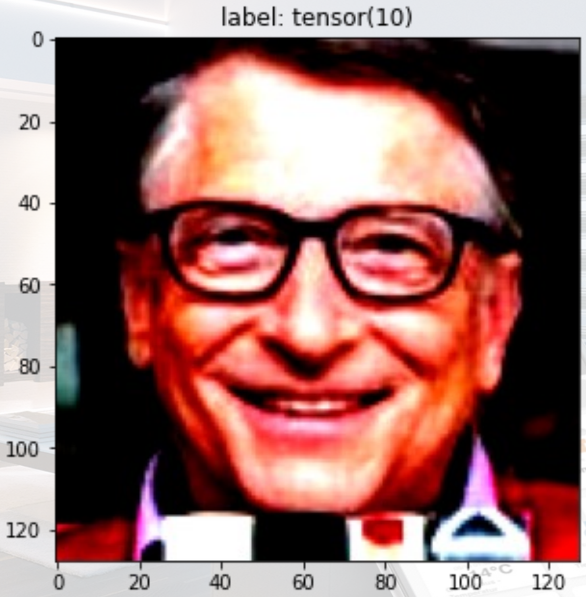




1.1 Pixel Comparison



3 x 224 x 224



3 x 128 x 128

1.2 Pixel Modification

TABLE II
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

No Modifications

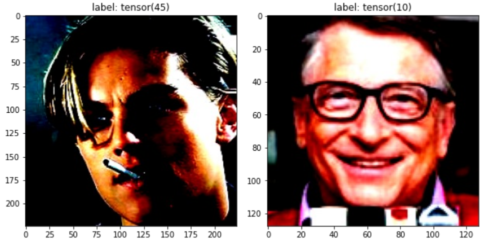
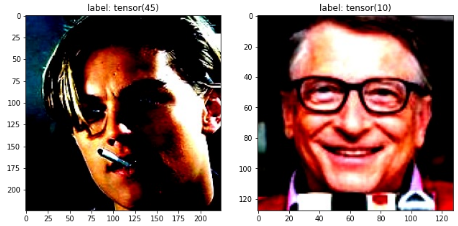


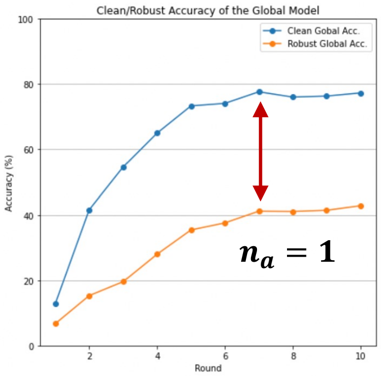
TABLE III
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS PIXEL MODIFIED TO $3 \times 128 \times 128$

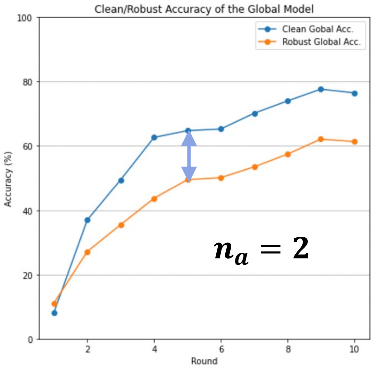
n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	**77.3**	**42.2**	**41.7**	**48.4**
2	50	76.5	60.7	61.6	64.5
3	50	61.6	63.7	64.5	64.1
4	**25**	57.6	61.7	63.2	62.4

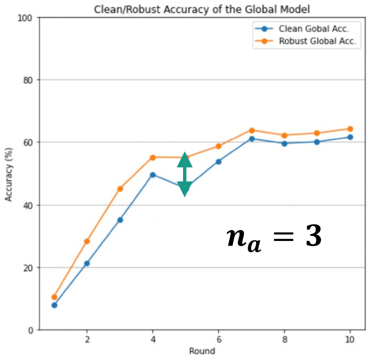
[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

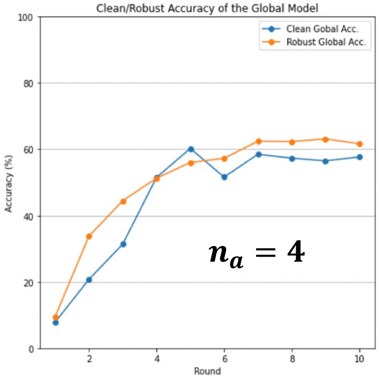
1.2 Pixel Modification





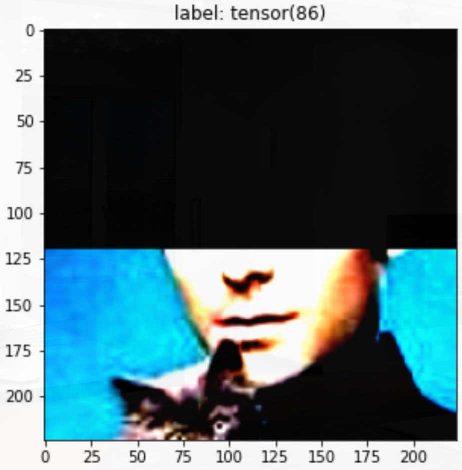


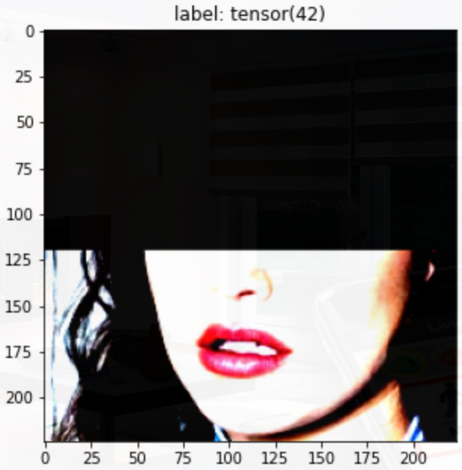




n_a = Adv. Trained Clients

2.1 “Eye” Area Covered





2.2 Pixel vs. Eye Covered

TABLE II
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

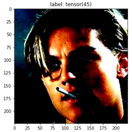
No Modifications

TABLE III
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS PIXEL MODIFIED TO $3 \times 128 \times 128$

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	77.3	42.2	41.7	48.4
2	50	76.5	60.7	61.6	64.5
3	50	61.6	63.7	64.5	64.1
4	**25**	57.6	61.7	63.2	62.4

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Pixel



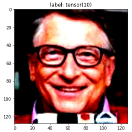
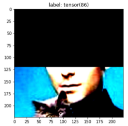


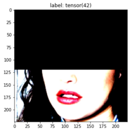
TABLE IV
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS CONSIST DATA WITH "EYE AREA" COVERED

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	73.6	39.3	40.6	47.3
2	50	77.4	60.6	61.6	64.9
3	50	69.1	60.6	61.3	63.0
4	**25**	63.8	63.1	63.5	63.4

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Eye Covered





2.2 Pixel vs. Eye Covered

TABLE II
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

No Modifications

TABLE III
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS PIXEL MODIFIED TO $3 \times 128 \times 128$

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	77.3	42.2	41.7	48.4
2	50	76.5	60.7	61.6	64.5
3	50	61.6	63.7	64.5	64.1
4	**25**	57.6	61.7	63.2	62.4

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Pixel

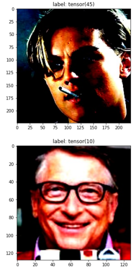
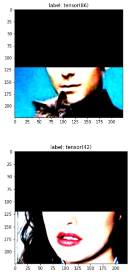


TABLE IV
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS CONSIST DATA WITH "EYE AREA" COVERED

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	73.6	39.3	40.6	47.3
2	50	77.4	60.6	61.6	64.9
3	50	69.1	60.6	61.3	63.0
4	**25**	63.8	63.1	63.5	63.4

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Eye Covered



2.2 Pixel vs. Eye Covered

TABLE II

GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

[a] n_a: No. of clients to go through adversarial training

[b] ABR: Adversarial training batch ratio

No Modifications

TABLE III

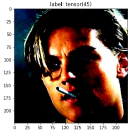
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS PIXEL MODIFIED TO $3 \times 128 \times 128$

n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	77.3	42.2	41.7	48.4
2	50	76.5	60.7	61.6	64.5
3	50	61.6	63.7	64.5	64.1
4	**25**	57.6	61.7	63.2	62.4

[a] n_a: No. of clients to go through adversarial training

[b] ABR: Adversarial training batch ratio

Pixel



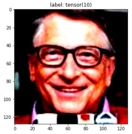


TABLE IV

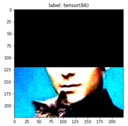
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS CONSIST DATA WITH "EYE AREA" COVERED

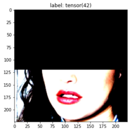
n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	73.6	39.3	40.6	47.3
2	50	77.4	60.6	61.6	64.9
3	50	69.1	60.6	61.3	63.0
4	**25**	63.8	63.1	63.5	63.4

[a] n_a: No. of clients to go through adversarial training

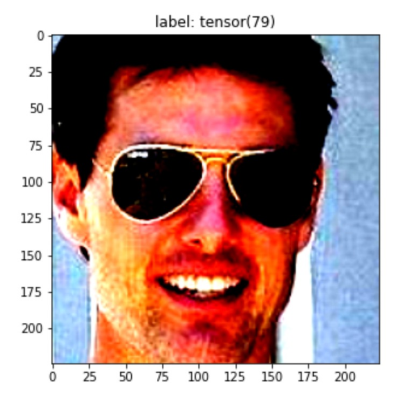
[b] ABR: Adversarial training batch ratio

Eye Covered

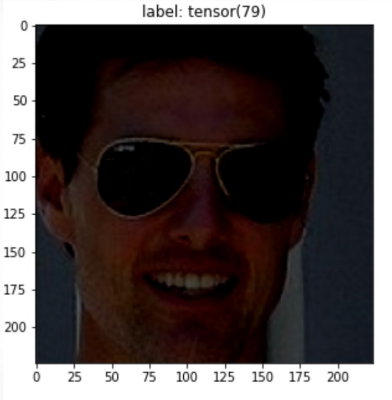




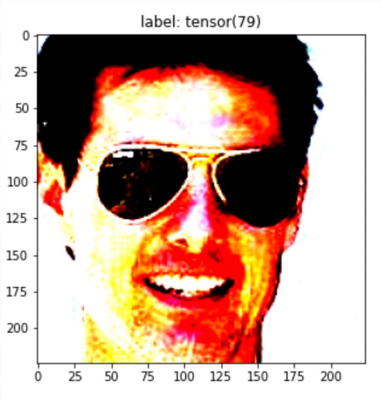
3.1 Brightness Comparison



Original



Brightness Factor = 0.15



Brightness Factor = 2.30

3.2 Brightness Modification (Dark)

TABLE II
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

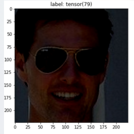
[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

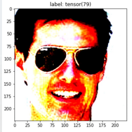
No Modifications

TABLE V
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS WITH BRIGHTNESS MODIFIED

n_a[a]	ABR[b](%)	BR[c]	GA[d](%)	Robust Acc.(%)		
				FGSM [2]	FFGSM [3]	Square [5]
1	50	0.15	79.9	57.9	59.1	63.3
2	50	0.15	65.6	70.6	71.3	70.3
3	50	0.15	23.6	74.1	75.2	65.4
4	25	0.15	67.5	72.5	73.5	72.5
1	50	2.30	72.6	60.4	61.3	63.8
2	50	2.30	57.4	67.8	68.4	66.4
3	50	2.30	18.7	68.7	69.6	60.1
4	50	2.30	10.4	69.0	69.5	58.3
4	25	2.30	39.4	69.4	70.0	64.3

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] BR: Brightness Ratio (0.15: dark / 2.30: bright)
[d] GA: Global Accuracy





3.2 Brightness Modification (Bright)

TABLE II

GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (IID). ADVERSARIALLY TRAINED WITH FFGSM (ϵ=8/255, α=10/255)

n_a [a]	ABR [b] (%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	25	85.1	47.9	49.2	56.2
1	50	85.7	54.1	54.2	60.5
1	75	85.1	51.6	52.5	58.6
2	25	82.8	65.2	65.3	68.7
2	50	83.1	66.7	67.9	71.0
2	75	80.7	67.4	67.9	68.1
3	25	83.0	64.9	65.0	68.45
3	50	73.9	71.9	72.3	72.5
4	25	71.5	70.2	70.9	71.2
4	50	30.7	74.1	75.0	66.6

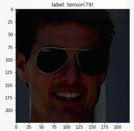
[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

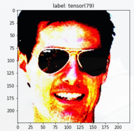
No Modifications

TABLE V

GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS WITH BRIGHTNESS MODIFIED

n_a [a]	ABR [b] (%)	BR [c]	GA [d] (%)	Robust Acc.(%)		
				FGSM [2]	FFGSM [3]	Square [5]
1	50	0.15	79.9	57.9	59.1	63.3
2	50	0.15	65.6	70.6	71.3	70.3
3	50	0.15	23.6	74.1	75.2	65.4
4	25	0.15	67.5	72.5	73.5	72.5
1	50	2.30	72.6	60.4	61.3	63.8
2	50	2.30	57.4	67.8	68.4	66.4
3	50	2.30	18.7	68.7	69.6	60.1
4	50	2.30	10.4	69.0	69.5	58.3
4	25	2.30	39.4	69.4	70.0	64.3





[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] BR: Brightness Ratio (0.15: dark / 2.30: bright)
[d] GA: Global Accuracy

3.3 Brightness vs. (Pixel, Eye Covered)

TABLE V

GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS WITH BRIGHTNESS MODIFIED

n_a[a]	ABR[b](%)	BR[c]	GA[d](%)	Robust Acc.(%)		
				FGSM [2]	FFGSM [3]	Square [5]
1	50	0.15	79.9	57.9	59.1	63.3
2	50	0.15	65.6	70.6	71.3	70.3
3	50	0.15	23.6	74.1	75.2	65.4
4	25	0.15	67.5	72.5	73.5	72.5
1	50	2.30	72.6	60.4	61.3	63.8
2	50	2.30	57.4	67.8	68.4	66.4
3	50	2.30	18.7	68.7	69.6	60.1
4	50	2.30	10.4	69.0	69.5	58.3
4	25	2.30	39.4	69.4	70.0	64.3

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] BR: Brightness Ratio (0.15: dark / 2.30: bright)
[d] GA: Global Accuracy

Brightness Modified

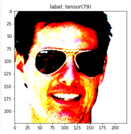
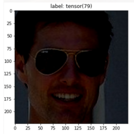



TABLE III

GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS PIXEL MODIFIED TO $3 \times 128 \times 128$

n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	77.3	42.2	41.7	48.4
2	50	76.5	60.7	61.6	64.5
3	50	61.6	63.7	64.5	64.1
4	**25**	57.6	61.7	63.2	62.4

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Pixel

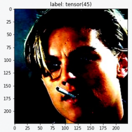
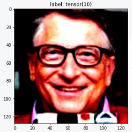



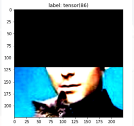
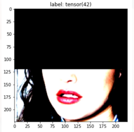
TABLE IV

GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS CONSIST DATA WITH "EYE AREA" COVERED

n_a[a]	ABR[b](%)	Global Acc.(%)	Robust Acc.(%)		
			FGSM [2]	FFGSM [3]	Square [5]
1	50	73.6	39.3	40.6	47.3
2	50	77.4	60.6	61.6	64.9
3	50	69.1	60.6	61.3	63.0
4	**25**	63.8	63.1	63.5	63.4

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio

Eye Covered

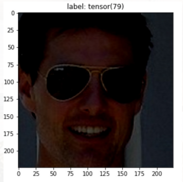
*Surprisingly "**ROBUST**" global aggregated model*

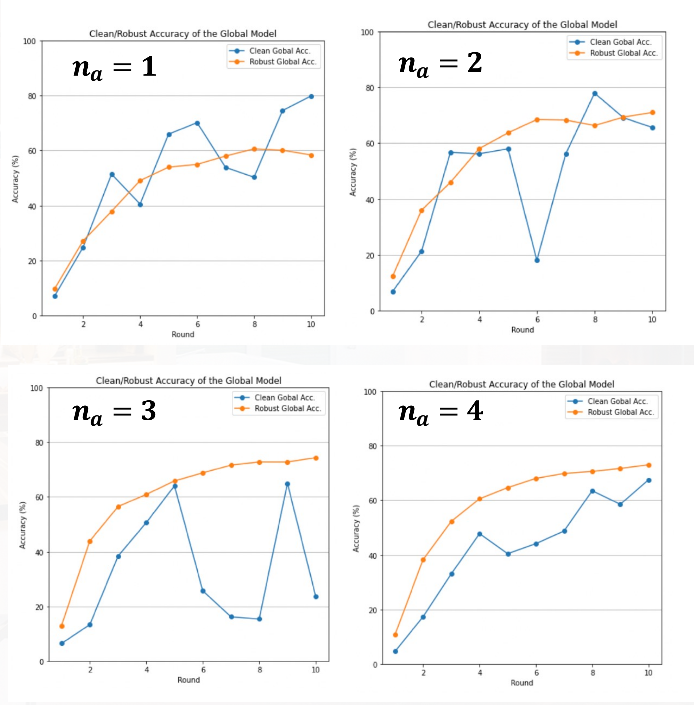
3.3 Brightness vs. (Pixel, Eye Covered)

Key Points

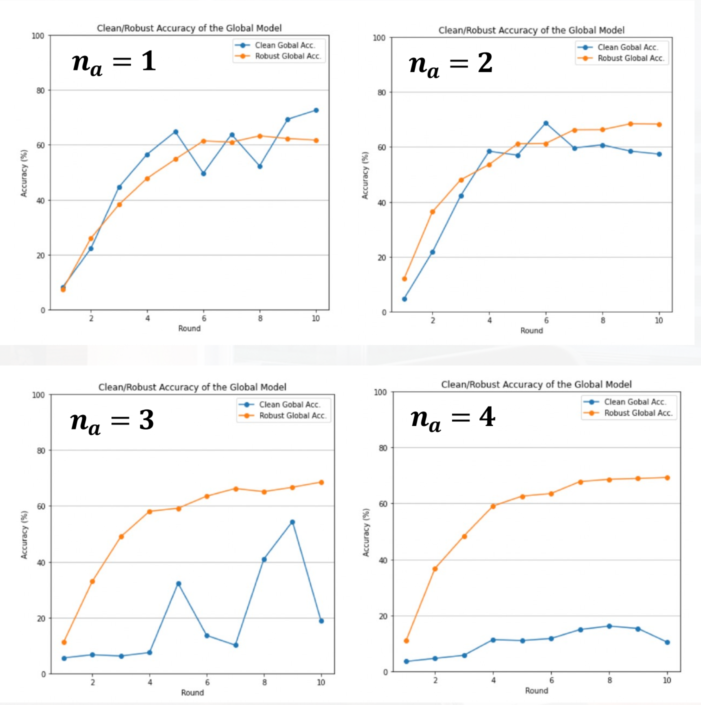
Fluctuating Global Acc. (%)

Stable Increasing Trend in Robust Acc. (%)

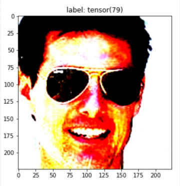




"Dark" Images



"Bright" Images



n_a = Adv. Trained Clients

Test Images: can be considered as ***different "race"***

4. Augmented Test Data (Dark)

TABLE V
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS WITH BRIGHTNESS MODIFIED

n_a[a]	ABR[b](%)	BR[c]	GA[d](%)	Robust Acc.(%)		
				FGSM [2]	FFGSM [3]	Square [5]
1	50	0.15	79.9	57.9	59.1	63.3
2	50	0.15	65.6	70.6	71.3	70.3
3	50	0.15	23.6	74.1	75.2	65.4
4	25	0.15	67.5	72.5	73.5	72.5
1	50	2.30	72.6	60.4	61.3	63.8
2	50	2.30	57.4	67.8	68.4	66.4
3	50	2.30	18.7	68.7	69.6	60.1
4	50	2.30	10.4	69.0	69.5	58.3
4	25	2.30	39.4	69.4	70.0	64.3

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] BR: Brightness Ratio (0.15: dark / 2.30: bright)
[d] GA: Global Accuracy

Brightness Modified

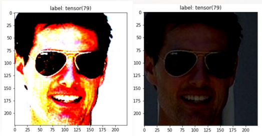
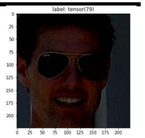


TABLE VI
ROBUST FEDERATED LEARNING (NON-IID) WITH TWO RANDOM CLIENTS CONSIST OF "DARK" IMAGES. EVALUATED ON AUGMENTED TEST DATA

n_a[a]	ABR[b](%)	TDT[c]	GA[d](%)	Robust Acc.(%)		
				FGSM	FFGSM	Square
1	50	Bright + Clean	54.4	57.0	56.9	61.5
1	50	Bright + Dark + Clean	55.9	64.5	66.5	67.7
1	50	Dark + Clean	61.2	56.7	58.2	61.9
2	50	Bright + Clean	36.5	71.7	72.5	63.8
2	50	Bright + Dark + Clean	57.8	65.2	66.1	69.1
2	50	Dark + Clean	57.2	67.2	69.7	69.7
3	50	Bright + Clean	62.5	70.7	72.6	72.5
3	50	Bright + Dark + Clean	57.5	68.6	70.3	69.6
3	50	Dark + Clean	58.3	72.2	72.5	71.5
4	25	Bright + Clean	49.1	69.7	71.1	66.5
4	25	Bright + Dark + Clean	59.1	71.5	72.0	71.2
4	25	Dark + Clean	50.4	70.0	70.3	65.8

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] TDT: Test Data Type
[d] GA: Global Accuracy



Test Images: can be considered as different "race"

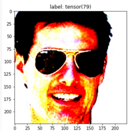
4. Augmented Test Data (Bright)

TABLE V
GLOBAL ACC.(%) AND ROBUST ACC.(%) OF ROBUST FEDERATED LEARNING (NON-IID). TWO RANDOM CLIENTS WITH BRIGHTNESS MODIFIED

n_a[a]	ABR[b](%)	BR[c]	GA[d](%)	Robust Acc.(%)		
				FGSM [2]	FFGSM [3]	Square [5]
1	50	0.15	79.9	57.9	59.1	63.3
2	50	0.15	65.6	70.6	71.3	70.3
3	50	0.15	23.6	74.1	75.2	65.4
4	25	0.15	67.5	72.5	73.5	72.5
1	50	2.30	72.6	60.4	61.3	63.8
2	50	2.30	57.4	67.8	68.4	66.4
3	50	2.30	18.7	68.7	69.6	60.1
4	50	2.30	10.4	69.0	69.5	58.3
4	25	2.30	39.4	69.4	70.0	64.3

[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] BR: Brightness Ratio (0.15: dark / 2.30: bright)
[d] GA: Global Accuracy

Brightness Modified



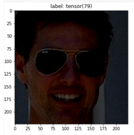
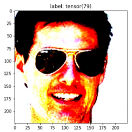


TABLE VII
ROBUST FEDERATED LEARNING (NON-IID) WITH TWO RANDOM CLIENTS CONSIST OF "BRIGHT" IMAGES. EVALUATED ON AUGMENTED TEST DATA

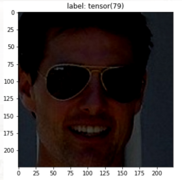
n_a[a]	ABR[b](%)	TDT[c]	GA[d](%)	Robust Acc.(%)		
				FGSM	FFGSM	Square
1	50	Bright + Clean	55.0	58.1	60.4	60.8
1	50	Bright + Dark + Clean	47.8	66.1	66.3	64.5
1	50	Dark + Clean	47.0	62.2	63.9	62.1
2	50	Bright + Clean	64.2	66.3	66.9	68.6
2	50	Bright + Dark + Clean	54.3	68.4	70.3	68.2
2	50	Dark + Clean	56.8	66.8	67.2	68.5
3	50	Bright + Clean	62.8	69.9	71.6	70.7
3	50	Bright + Dark + Clean	44.8	70.5	71.1	62.0
3	50	Dark + Clean	49.2	68.2	68.3	66.4
4	50	Bright + Clean	38.6	69.9	70.1	59.2
4	50	Bright + Dark + Clean	47.4	71.2	71.9	63.9
3	50	Dark + Clean	36.1	70.2	70.5	61.8

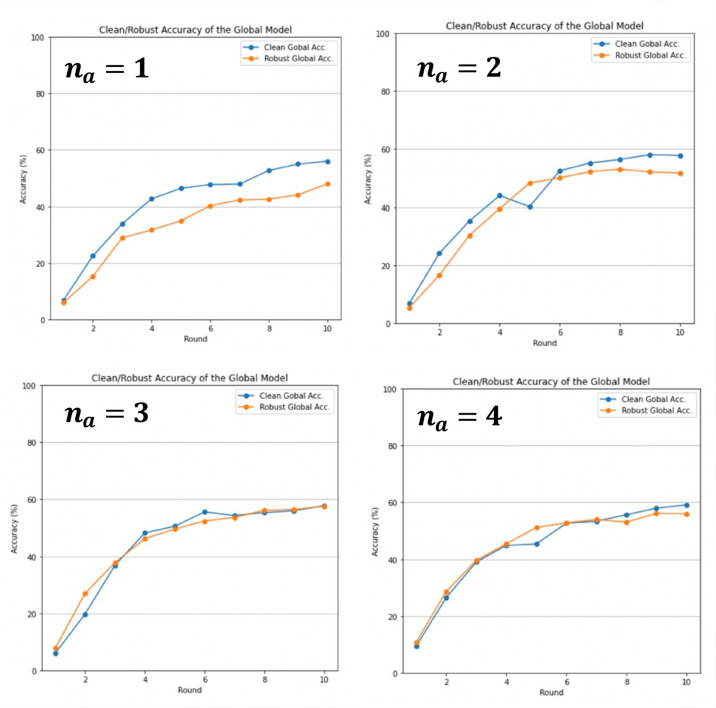
[a] n_a: No. of clients to go through adversarial training
[b] ABR: Adversarial training batch ratio
[c] TDT: Test Data Type
[d] GA: Global Accuracy



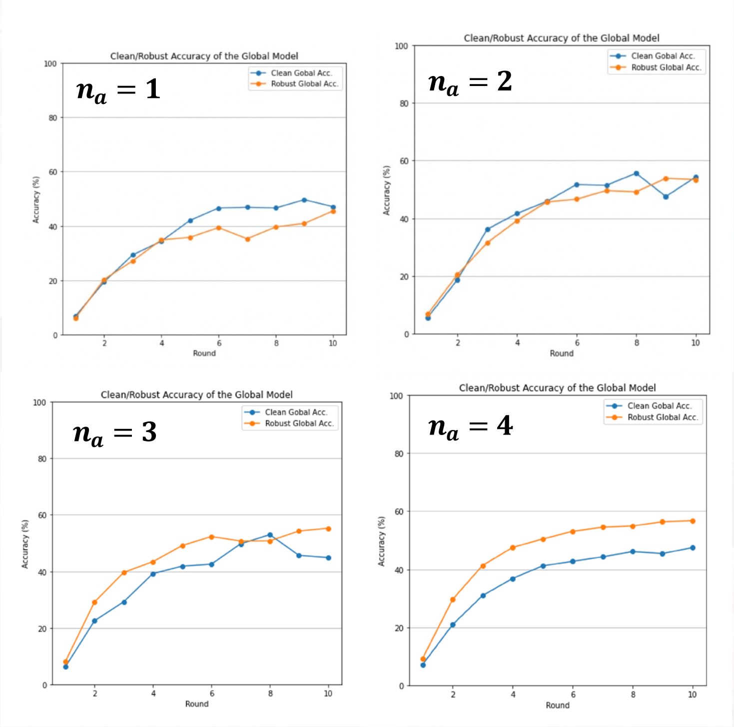
4. Augmented Test Data

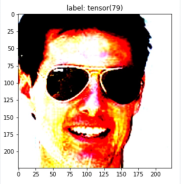
n_a = Adv. Trained Clients





"Dark" Images





"Bright" Images

Section 5 & 6: Summary and Evaluation

Limitations

1. Utilization of ResNet

- Instead of using SOTA face recognition models

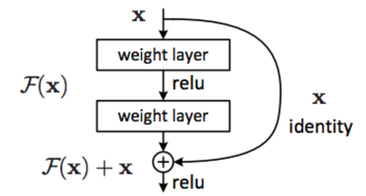


Figure 2. Residual learning: a building block.

2. Starving Federated Data

- Limited amount of data distributed → Bias and Overfitting

3. Single Weight Averaging Method

- Only used `FedAvg` for the entire experiment

$$w_{t+1} \leftarrow \sum_{K}^{k=1} \frac{n_k}{n} w_{t+1}^{k}$$

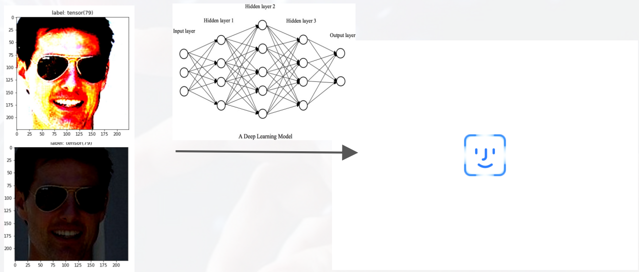
Key / Novel Findings

1. STARVING FEDERATED DATA

- **FLATS:** more ROBUST global model against adversarial examples
- More **REALISTIC** experiment

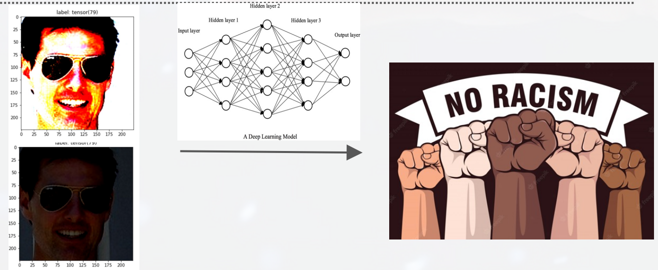
2. ROBUSTNESS with DATA MODIFICATION

- Increased BOTH Global Acc.(%) and Robust Acc.(%)
- **Broaden spectrum** to general CV / Face Recognition training
- Needs to be considered as **COMMON PRACTICE**



3. ALLEVIATE FAIRNESS ISSUE

- Augmented Test Data → considered as "race" mixed test data
- Reduce BIAS in classification



Reference

- Attack & Defense (1): Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).
- Kariya, Mahendra. “Paper Discussion: Explaining and Harnessing Adversarial Examples.” *Medium*, Medium, 16 Nov. 2018, https://medium.com/@mahendrakariya/paper-discussion-explaining-and-harnessing-adversarial-examples-908a1b7123b5.
- Dong, Yinpeng, et al. "Boosting adversarial attacks with momentum." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- **1-dimensional example:** “Explaining And Harnessing Adversarial Examples 논문-리뷰.” *Explaining And Harnessing Adversarial Examples* 논문-리뷰, 10 July 2020, velog.io/@miai0112/Explaining-And-Harnessing-Adversarial-Examples-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0.
- Pan, Zhixin, and Prabhat Mishra. "Fast Approximate Spectral Normalization for Robust Deep Neural Networks." *arXiv preprint arXiv:2103.13815* (2021).
- Cho, Yoon Sang. “Adversarial Attacks and Defenses in Deep Learing.” *Adversarial Attacks and Defenses in Deep Learing*, 2020, pp. 5–32, dmqm.korea.ac.kr/activity/seminar/289.

Q & A

Thank You!