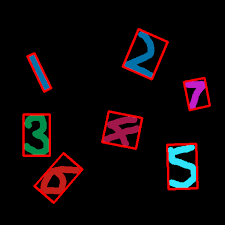
|  |
| --- |
| Digits recognition on meter-like device  A real world OCR application |
| Chenyu Ran 2019/10/24 |



## Introduce to digits recognition on meter-like device

## Introduction

This report concerns the recognition of digits on meter-like device. The digits are currency issuance number, which are used to represent how much electricity you have used. Recognition by human-being has always been an expensive way on this kind of task, and will also includes unexpected mistakes. We come up with this idea since image processing techniques and digits recognition are so mature that they beat human-being’s performance on many of image-based tasks.



(1). Application procedure

a. Pictures with 5 digits inside each are auto-taken by cameras.

b. Pictures are saved and then uploaded to PC.

c. Digit recognition model is applied to predict the digits inside each picture.

d. Results of prediction are shown on PC application.

### 

(2). Algorithm details

a. Image preprocessing

a.1. Image cropping and splitting

Crop the area which includes target digits corresponding to different input image size.





Then image is split into 5 single digit sub-images.

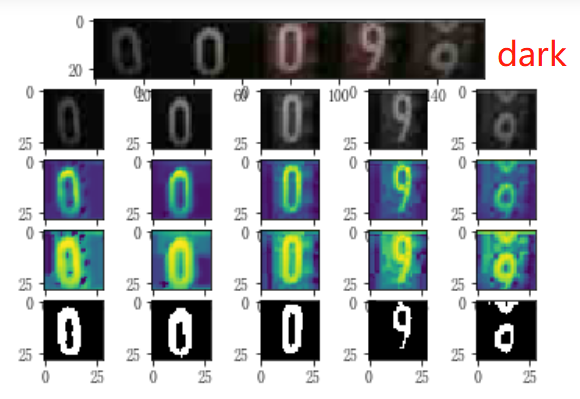
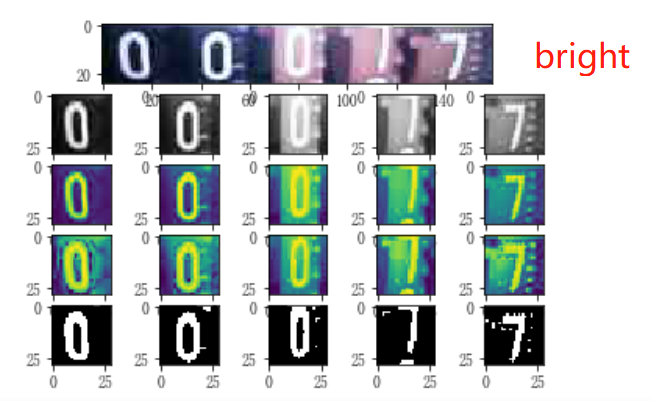
a.2. Denoise

Lighting spot and other useless information are removed from pictures taken by photographer to ensure the consistency of different inputs to increase the accuracy of prediction.



a.3. Histogram equalization

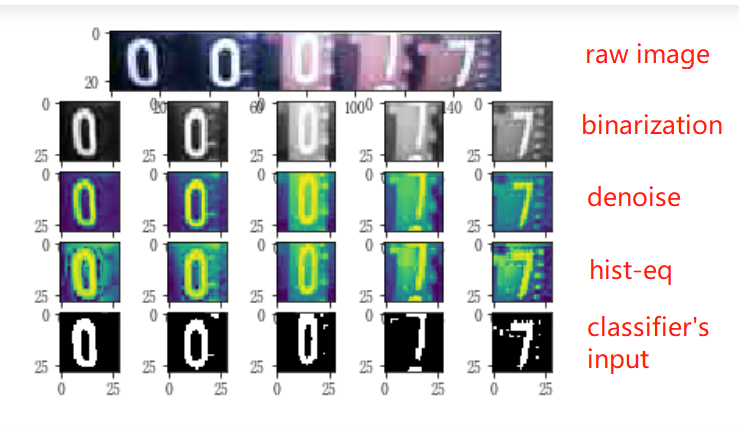
Due to the difference between difference pictures’ environments, the strengths of lighting are also not consistent. So we need to normalize the brightness of each picture. Thus histogram equalization is used to tackle with this issue.

a.4. Image size normalization.

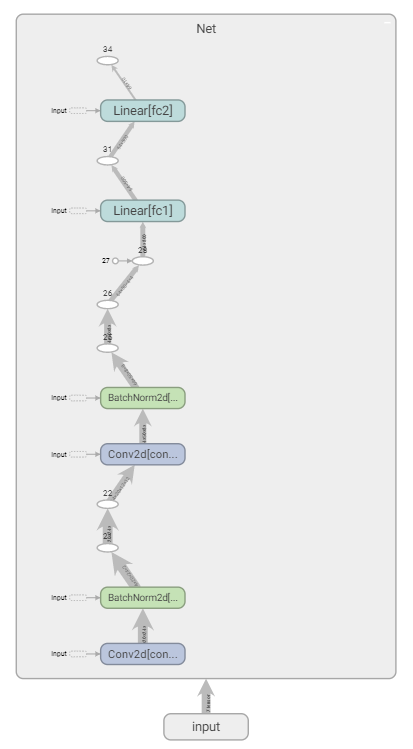
All images are resized to standard size 28 pixels by 28 pixels.

a.5. Full procedure of image pre-processing.

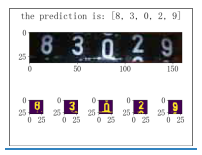


b. CNN model in detail

*class* Net(nn.Module):  
 *def \_\_init\_\_*(*self*):  
 *super*(Net, *self*).\_\_init\_\_()  
 *self*.conv1 = nn.Conv2d(1, 20, 5, 1)  
 *self*.conv2 = nn.Conv2d(20, 50, 5, 1)  
 *self*.fc1 = nn.Linear(4 \* 4 \* 50, 500)  
 *self*.fc2 = nn.Linear(500, 10)  
 *self*.bn1 = nn.BatchNorm2d(20)  
 *self*.bn2 = nn.BatchNorm2d(50)  
  
 *def* forward(*self*, x):  
 x = F.relu(*self*.bn1(*self*.conv1(x)))  
 x = F.max\_pool2d(x, 2, 2)  
 x = F.relu(*self*.bn2(*self*.conv2(x)))  
 x = F.max\_pool2d(x, 2, 2)  
 x = x.view(-1, 4 \* 4 \* 50)  
 x = F.relu(*self*.fc1(x))  
 x = *self*.fc2(x)  
 *return* F.log\_softmax(x, dim=1)



c. Making predictions

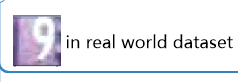


(3). Difficulties and experience

a. Bias between training/test dataset and real world application’s dataset

The enviroment of training/test dataset is more stable because the images are collected by experienced photographer, wheras the real world dataset might be a lot different from training/test dataset. Another important issue is the pattern of digit between two kinds of dataset are also different. For example, digit 9 are shown as below.





Solution:

a.1. Fine tune the parameters to fit to real world dataset. Otherwise, due to the two datasets' different pattern, the performance of the classifier might go down quite much.

a.2. Make sure the training pictures and test pictures and real world application’s pictures consistent next time.

b. Digit transition

Sometimes the camera might take pictures with two half digit in a single image at the meantime.



Solution:

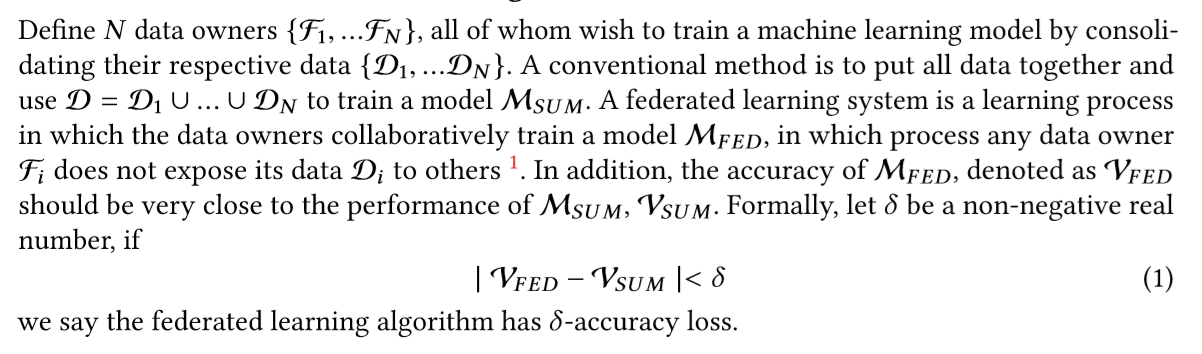
Define the strategy when this situation occurs. For example, always choose the upper part digit when digit transition occurs. This needs to be done when labelling the images.

1. Federated Machine Learning: Concept and Applications

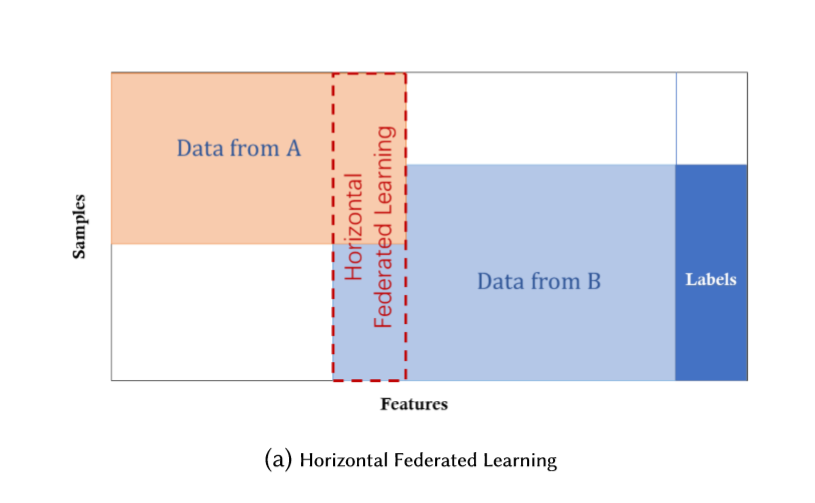
Today’s AI still faces two major challenges. One is that in most industries, data exists in the

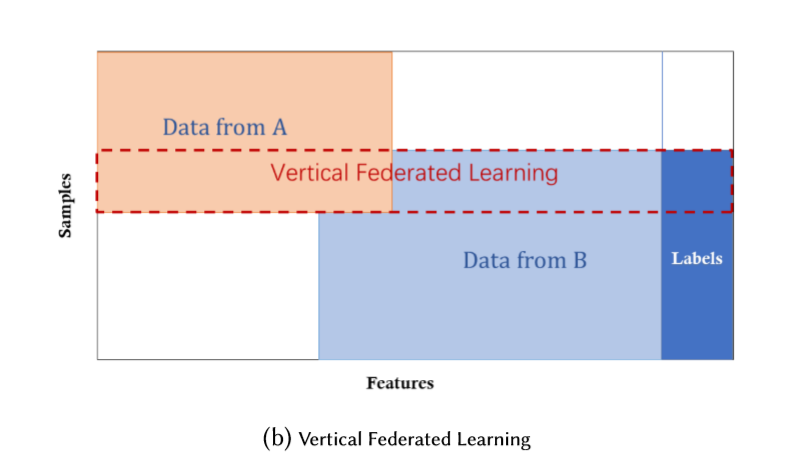
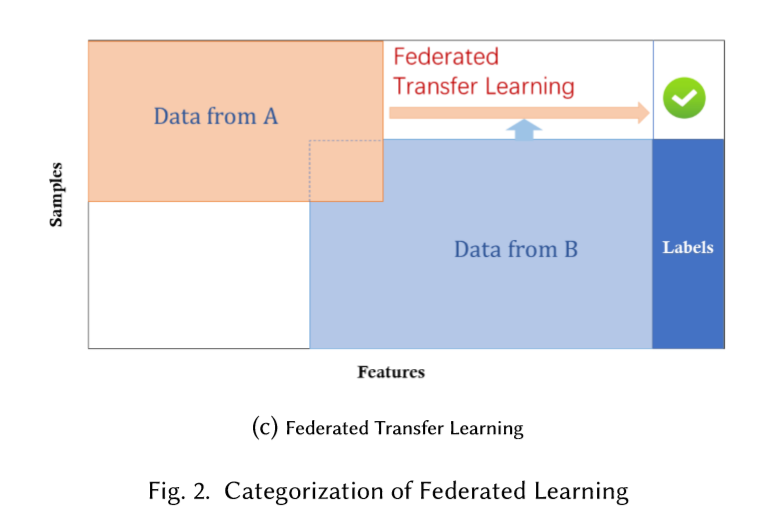
form of isolated islands. The other is the strengthening of data privacy and security. We propose a possible solution to these challenges: secure federated learning. Beyond the federated learning framework first proposed by Google in 2016, we introduce a comprehensive secure federated learning framework, which includes horizontal federated learning, vertical federated learning and federated transfer learning. We provide definitions, architectures and applications for the federated learning framework, and provide a comprehensive survey of existing works on this subject. In addition, we propose building data networks among organizations based on federated mechanisms as an effective solution to allow knowledge to be shared without compromising user privacy.

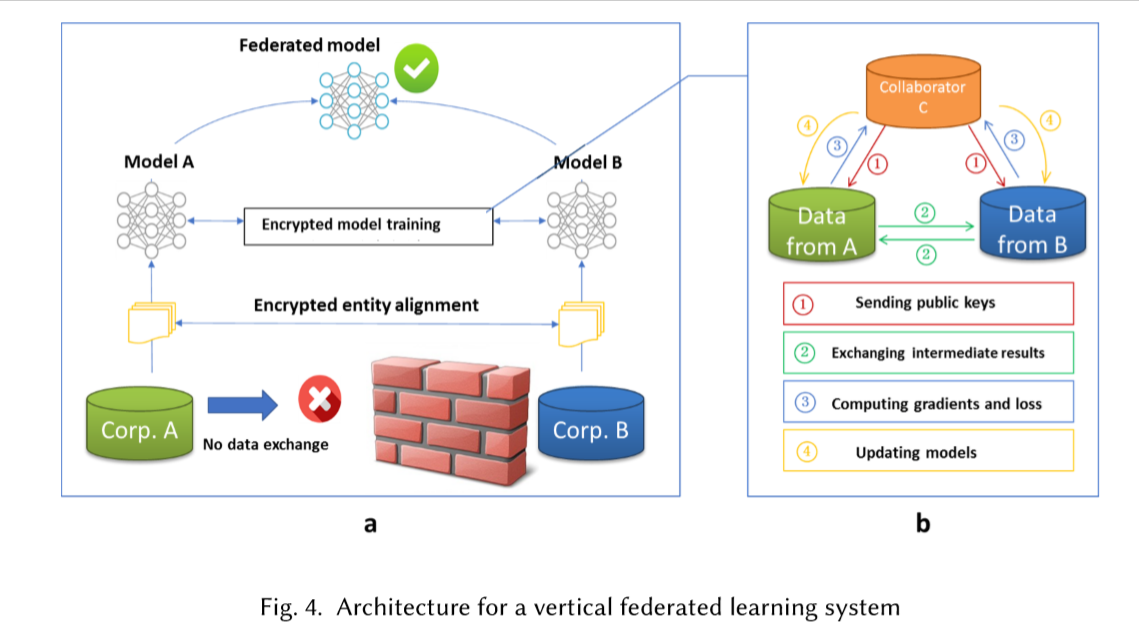
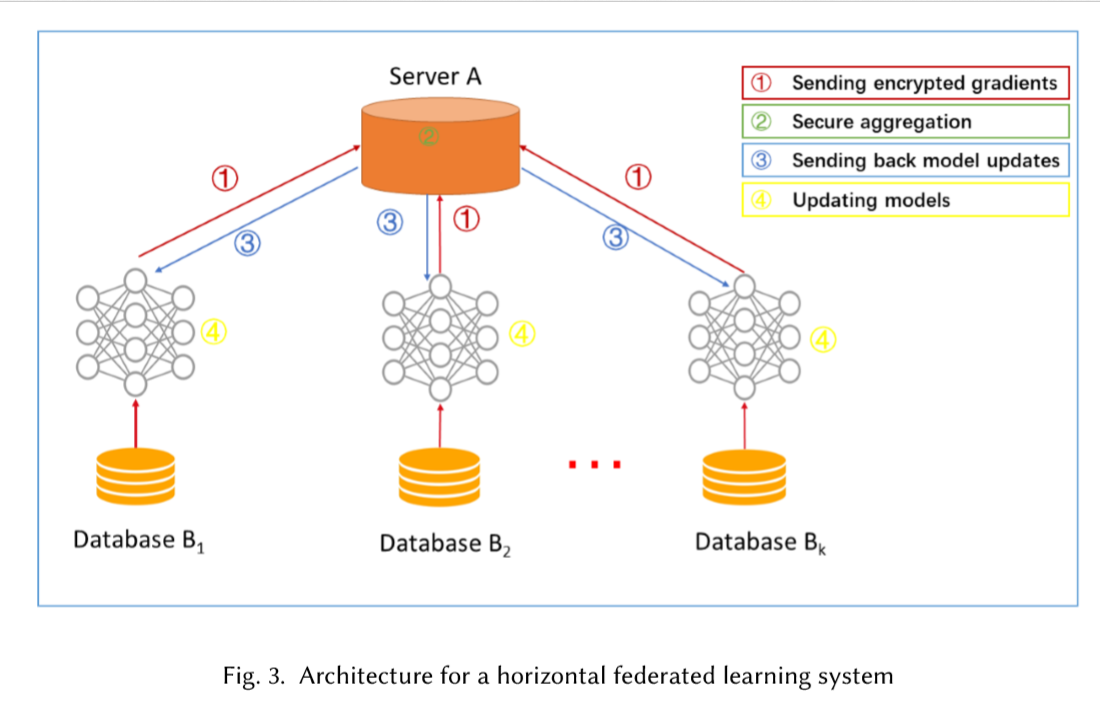
1. Definition of Federated Learning:



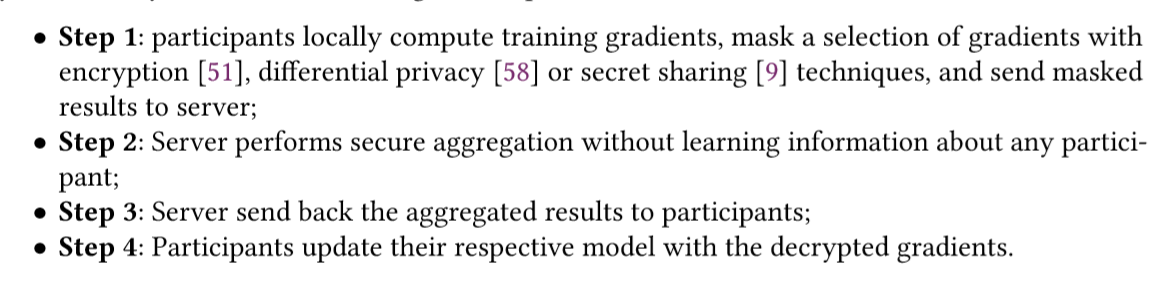
1. Categorization of Federal Learning





1. The training process of such a system usually contain the following four steps:



1. Related works Federated learning enables multiple parties to collaboratively construct a machine learning model while keeping their private training data private. As a novel technology, federated learning has several threads of originality, some of which are rooted on existing fields.
2. Federal Learning vs. Distributed Machine Learning

