Article

Federated Learning for Clients' Data Privacy Assurance in Food Service Industry

Hamed Taheri Gorji [1,2], Mahdi Saeedi [1,2], Erum Mushtaq [3], Hossein Kashani Zadeh [1,2], Kaylee Husarik [1,2], Seyed Mojtaba Shahabi [4], Jianwei Qin [5], Diane E. Chan [5], Insuck Baek [5], Moon S. Kim [5], Alireza Akhbardeh [2], Stanislav Sokolov [2], Salman Avestimehr [3,6], Nicholas MacKinnon [2], Fartash Vasefi [2] and Kouhyar Tavakolian [1,*]

1   Biomedical Engineering Program, University of North Dakota, Grand Forks, ND 58202, USA; hamed.taherigorji@und.edu (H.T.G.); mahdi.saeedi@und.edu (M.S.); hossein.kashanizadeh@ndus.edu (H.K.Z.); kaylee.husarik@und.edu (K.H.)
2   SafetySpect Inc., 4200 James Ray Dr., Grand Forks, ND 58202, USA; aakhbardeh@safetyspect.com (A.A.); ssokolov@safetyspect.com (S.S.); nmackinnon@safetyspect.com (N.M.); fvasefi@safetyspect.com (F.V.)
3   Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California (USC), Los Angeles, CA 90089, USA; emushtaq@usc.edu (E.M.); avestime@usc.edu (S.A.)
4   School of Electrical Engineering & Computer Science, University of North Dakota, Grand Forks, ND 58202, USA
5   USDA/ARS Environmental Microbial and Food Safety Laboratory, Beltsville Agricultural Research Center, Beltsville, MD 20705, USA; jianwei.qin@usda.gov (J.Q.); diane.chan@usda.gov (D.E.C.); insuck.baek@usda.gov (I.B.); moon.kim@usda.gov (M.S.K.)
6   FedML Inc., 26618 Nokomis RD, Rancho Palos Verdes, CA 90275, USA
*   Correspondence: kouhyar@und.edu

Abstract: The food service industry must ensure that service facilities are free of foodborne pathogens hosted by organic residues and biofilms. Foodborne diseases put customers at risk and compromise the reputations of service providers. Fluorescence imaging, empowered by state-of-the-art artificial intelligence (AI) algorithms, can detect invisible residues. However, using AI requires large datasets that are most effective when collected from actual users, raising concerns about data privacy and possible leakage of sensitive information. In this study, we employed a decentralized privacy-preserving technology to address client data privacy issues. When federated learning (FL) is used, there is no need for data sharing across clients or data centralization on a server. We used FL and a new fluorescence imaging technology and applied two deep learning models, MobileNetv3 and DeepLabv3+, to identify and segment invisible residues on food preparation equipment and surfaces. We used FedML as our FL framework and Fedavg as the aggregation algorithm. The model achieved training and testing accuracies of 95.83% and 94.94% for classification between *clean* and *contamination* frames, respectively, and resulted in intersection over union (IoU) scores of 91.23% and 89.45% for training and testing, respectively, of segmentation of the contaminated areas. The results demonstrated that using federated learning combined with fluorescence imaging and deep learning algorithms can improve the performance of cleanliness auditing systems while assuring client data privacy.

Keywords: federated learning; deep learning; FedML; CSI-D; contamination detection; food service

1. Introduction

Foodborne illness contributes significantly to morbidity and mortality as a public health issue. In 2010, contaminated food caused around 600 million cases of foodborne disease and 420 thousand fatalities worldwide [1]. According to the US Centers for Disease Control and Prevention (CDC), approximately 48 million Americans are sickened each year, 128,000 are hospitalized, and 3000 are killed by foodborne illnesses [2]. According to the CDC, restaurants and institutional kitchens are the places with the most foodborne

disease-associated outbreaks [3]. Since restaurants usually handle a broad range of raw foods, the risk of cross-contamination leading to outbreaks of foodborne disease grows significantly. Cross-contamination in restaurants during food preparation phases can be caused by inadequate hygiene and sanitization methods and contaminated equipment and surfaces.

Bacteria can adhere to food contact surfaces, facilitated by the formation of protective film coatings from organic components and nutrients that can impede the sanitization of these surfaces [4,5]. Contamination detection in restaurants and institutional kitchens is primarily limited to visual examination during the inspection. Another way of testing is swabbing food contact surfaces for adenosine triphosphate (ATP) testing or for laboratory culture [6]. In many cases, the visual inspection method is insufficient because the contaminated areas are invisible to the naked eye. When it comes to testing large surfaces, swabs are not the best option. They can be time-consuming and costly since they can only contact a tiny fraction of the surface, even if swabbed back and forth.

Newer developments in imaging technology and machine learning (ML), especially deep learning algorithms, have been proposed to improve food safety and surface cleanliness. The authors of [7] proposed a system for the early detection of apple bruises using hyperspectral imaging (HSI) with shortwave infrared (SWIR) illumination and a line-scan camera combined with a partial least squares discriminant analysis (PLS-DA) classifier. In [8], real-time detection of parasites in shell-off cooked clams using transillumination imaging and binary decision trees was proposed. A method for detecting the neurotoxin acrylamide in potato chips was suggested by [9], using an SVM classifier on spatial domain statistical features extracted from images. The authors of [10] proposed foreign object detection (FOD) in meat products using a sequential deep learning framework. In [11], a three-level classification of cleanliness in restrooms as dirty, average, and clean using deep convolutional neural networks (CNN) was proposed. In a previous research project, our group combined two deep learning algorithms (EfficientNet-B0 and U-Net) with fluorescence imaging to automatically identify and segment fecal contamination on meat carcasses [12]. In another work, we used Xception and DeepLabV3+ deep learning algorithms with multiwavelength fluorescence imaging to identify and segment contaminated areas in images of equipment and surfaces in the food-service industry [13]. Moreover, our research group has continued to innovate and expand the field with numerous studies that leverage this technology to a variety of cleanliness and sanitization contexts [14–16].

ML models generally and deep learning algorithms specifically need vast and diversified sets of training data to achieve accurate and reliable performance. These datasets are usually collected from multiple devices, such as cameras, sensors, cell phones, etc., at the interface or edge between the real world and the abstract world of data. The data from these "edge" devices are then brought together on a central server or storage system. The ML models use this data to train themselves and eventually predict outcomes with new data. However, these centralized methods can be troublesome if the collected data contain sensitive information or the centralization is too costly. Centralized data collection may create privacy issues for people and liability problems for corporations if data is not handled correctly. Since people are increasingly concerned about data privacy and security, governments and organizations have begun enforcing legislation to safeguard personal information [17–19].

Since the most frequent sites of foodborne illness outbreaks are restaurants and institutional kitchens, sanitization and infection prevention in these facilities are vital. New technologies such as the SafetySpect sanitization inspection device [20] can help restaurants and institutional kitchens keep their environment safer and improve their level of cleanliness. However, companies and institutions are concerned that using new technologies can increase privacy risks, and leaks of sensitive information (e.g., the presence of contamination in the kitchen or dining area) could jeopardize their reputation. There is a need to convince such organizations and facilities that their sensitive information and data will be safe and secure when participating in shared-data projects.

New avenues of ML research have opened up with the advent of federated learning (FL) [21]. FL allows ML algorithms to learn from data without transferring data to a centralized server. FL lets the training data remain distributed on multiple clients' devices while the ML models are downloaded and the models trained on the decentralized local data. The updated model parameters are then returned to the centralized server without transferring any of the client's raw data. FL allows several clients to work together to train an ML model without sharing any private data or sensitive information. In order to train the FL model, it is necessary to have a coordinating agent that is responsible for handling the information exchange [22]. FL spreads the computational power requirements across all clients while preserving client privacy and eliminating the cost of data transfer and storage needed with centralized processing. Each client's computational power is used to analyze their own data.

The application of FL has recently gained attention in various "edge" device applications, such as in mobile devices for text prediction on virtual keyboards [23], improving the quality of search suggestions [24], and emoji suggestions based on the text typed on a keyboard [25]. Another notable application of FL is in the medical field, where data typically includes private and sensitive patient information that cannot be shared outside of an organization, limiting its use in public research. For example, in [26], the authors proposed a framework to investigate changes in the subcortical brain caused by neurological disorders. In [27], a heterogeneous FL method was proposed for training ML models for electroencephalography data classification, and the effectiveness of FL for the identification of COVID-19 using chest X-ray images was investigated in [28]. FL has also been investigated in many other areas [29], but to our knowledge, there are no publications focusing on using FL in food safety and cleanliness assessments in the food service industry. In this study, we use a fluorescence-imaging technology combined with an FL-based deep learning model for the detection of contamination and image segmentation of contamination in fluorescence images of a variety of food-handling surfaces. Combining FL-based deep learning models with fluorescence imaging for contamination detection could improve the level of safety and cleanliness in institutional kitchens and restaurants while assuring safety and privacy of client data.

In this paper, the data collection technology and process are explained in Section 2. The FL framework and deep learning algorithms used for the classification and segmentation of contamination are described in Section 3. The results and performance evaluation of this study are shown in Section 4. Section 5 concludes with the main achievements of this study.

2. Material

2.1. Data Collection Technology

A portable automated imaging inspection system for "contamination, sanitization inspection, and disinfection" (CSI-D) has been developed (SafetySpect Inc., Grand Forks, ND, USA). This handheld system provides mobility and flexibility for fluorescence-based detection of organic residue, bacterial biofilms, saliva, and respiratory droplets on a variety of surfaces [20]. CSI-D is able to detect, disinfect, and document the presence of contamination on food preparation surfaces that might potentially harbor pathogens or disease organisms. The illumination methods of the CSI-D includes 270 nm and 405 nm light-emitting diode (LED) arrays, programmed to rapidly turn on and off to allow image capture and removal of ambient light reflectance from fluorescence images of the surface. CSI-D uses two cameras for fluorescence imaging: an RGB camera to capture images of various organic residues and a UV camera to capture saliva and respiratory droplets as well as aromatic amino acids such as tryptophan and other residues.

2.2. Data Collection

We collected data from seven Edgewood LTCF kitchens in North Dakota, a kitchen facility that prepares meals and snacks for multiple public schools in Grand Forks, ND, and two restaurants in Los Angeles, CA. We discussed cleaning procedures, high-touch and

high-risk locations, and any perceived sanitization problems with each facility manager prior to data collection. We used CSI-D to record videos from a variety of high-risk regions, including doorknobs, garbage cans, oven and refrigerator door handles, chopping boards, and preparation tables. All videos were recorded at a resolution of 1024 × 768 and a capture rate of 24 frames per second (FPS). We analyzed 1 h and 58 min of video.

In Figures 1 and 2, we have examples of kitchen equipment and other high-touch areas showing clean surfaces (Figure 1) and surfaces with contamination (Figure 2). Many materials both fluoresce and reflect ambient light that can be detected by the CSI-D scanner. It is important to consider both the fluorescence and the differences between irregular fluorescence on a surface and regular fluorescence on that surface. The irregular patterns of fluorescence are more consistently associated with contamination. By looking for differences between patterns on a surface and patterns on a surface, we can identify the presence of contamination.

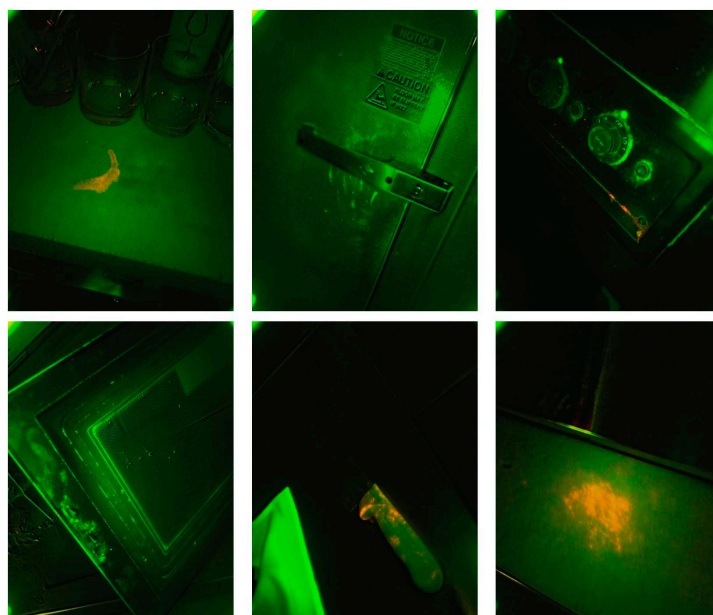**Figure 1.** Six CSI-D fluorescence images of clean surfaces.

Figure 2. Six CSI-D fluorescence images of contamination on different surfaces.

3. Methodology

A standard FL system comprises a central server and multiple individual clients with their own raw datasets that are stored locally. The central server is responsible for coordinating the training by selecting a random set of clients, randomly initializing the model's global weights, broadcasting it to the clients, collecting and aggregating the updated weights received from the clients after training the model (local weights), and finally, sending the updated global model back to the clients for the next round of training. This process will be repeated until the model meets the criteria defined by the system owner. From the client viewpoint, they receive the global model broadcast by the server and train this model over several epochs using locally stored data, then send the updated model back to the server for aggregation. Clients then wait to receive the updated global model for the next round of training. It is worth emphasizing that the clients only share the updated model weights with the server and that the server has no access to the client's private data. Figure 3 shows a concise illustration of an FL system. To implement an FL system, several algorithms and open-source frameworks have been proposed [30–37]. In this study, we use the Federated Averaging (FedAvg) algorithm and FedML framework to implement our FL system. We also considered other aggregation algorithms. Most of the models we considered are based on FedAvg with slight modifications that help address particular data problems, such as unfavorable convergence behavior or objective inconsistency. Some models required actual distributed learning and so could not be tested in our simulation environment. We had a good convergence with FedAvg, so we did not expect the modified models to improve our performance. We did try FedProx as an aggregation algorithm, but for our data, it did not perform as well as FedAvg, and because FedAvg is a well-known and accepted model, we decided not to proceed further with other models.

MobileNetV3 is used for the classification of the *clean* and *contamination* frames and DeepLabv3+ for the precise segmentation of contaminated areas in each video frame.
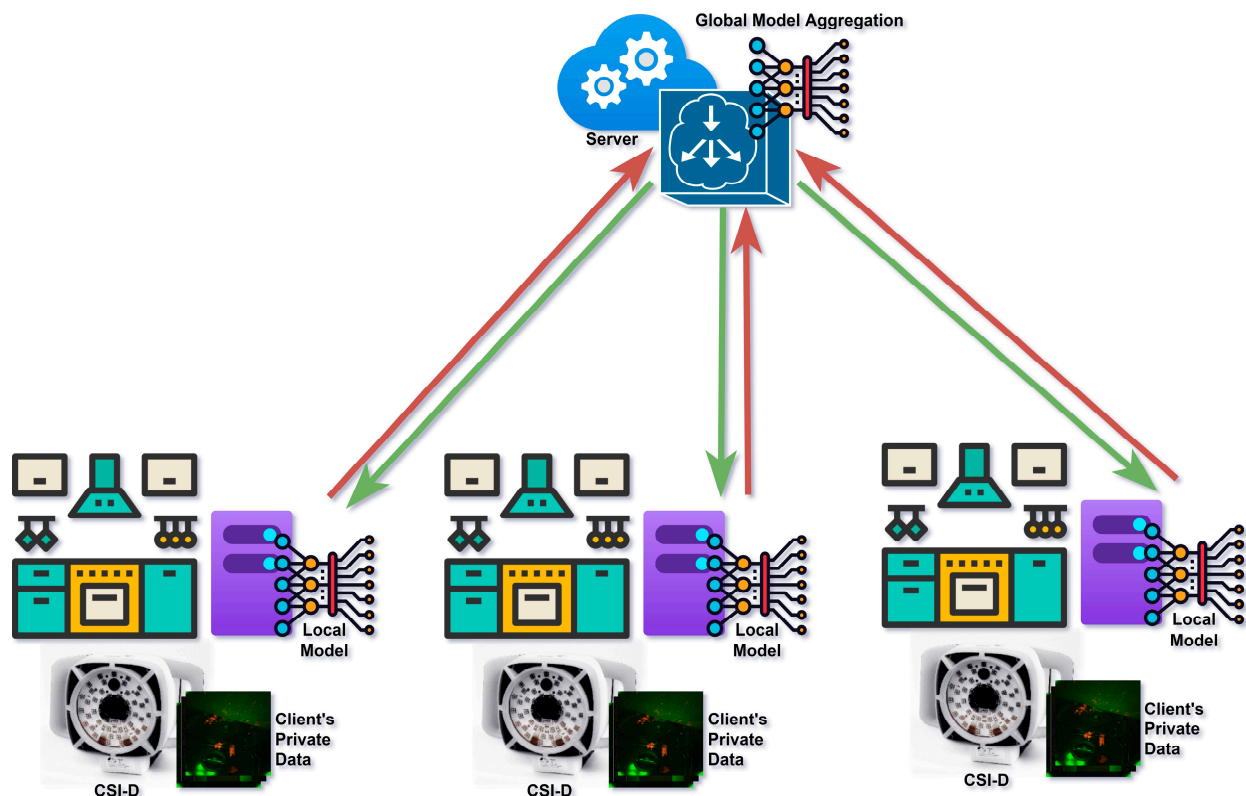

Figure 3. A concise illustration of an FL system.

3.1. Federated Averaging (FedAvg)

FedAvg [21] is the most common and well-known method used in the implementation of FL to aggregate local weights coming from clients and to update the global model on the server. Generally, FedAvg comprises four steps. In the first step, the global model is sent synchronously from the server to a randomly chosen subset of clients. Then each client, in parallel, performs the gradient descent steps during training on their local data and then updates the model with a determined learning rate, epoch number, and batch size, as shown in (1). After that, each client returns the updated model weights to the server. Finally, the server aggregates the client model weights (2) and updates the global model, sending it back to the clients for the next round. These four steps make one round of communication and are repeated several times until the global model converges. In each communication round, the FedAvg takes a weighted average of the weights from the local models based on the size of each client's local dataset. This means clients with more training data contribute more to the global model update.

$$w \leftarrow w - \eta \nabla \ell(w; b) \tag{1}$$

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{2}$$

In (1), η is the local learning rate, w is the model weights, b is the local batch size, and $\ell()$ is the local loss function. In (2), K is the total number of selected clients, k shows each client, n denotes the total number of data samples, and n_k is the sample size of each client. Despite the simplicity of FedAvg, this algorithm has proven to be not only robust enough to cope with unbalanced data that is not "independent and identically distributed" (non-IID) but also able to decrease the frequency and the amount of data transferred to that server that is normally required for training a deep neural network model [21].

3.2. FedML

Any problem being framed in terms of a federated machine learning scenario will have several aspects in common. This characteristic allows developers to adopt an open-source model to expand the capabilities of machine learning. One of the frameworks that enable deployments and research using FL is FedML [34]. FedML aims to address challenges in the space of algorithmic development by offering APIs for secure aggregation, communication, and benchmarking tools.

FedML is designed with two layers of high-level and low-level APIs. Low-level APIs handle the establishment of secure communication between servers and clients. High-level APIs enable the developer to handle model type and data manipulations. FedML's main goal is to provide an adequate platform that supports algorithm and development and has standard benchmarking tools. These benchmarking tools can serve as models for setting up benchmarking with additional model types. The FedML platform currently provides FedAvg, FedOpt, FedNova, and FedNAS as FL algorithms. However, the authors have built-in mechanisms to allow outside developers to add their own optimization algorithm and run standard tests on it.

A federated learning environment has two aspects: infrastructure and training/aggregation. The infrastructure establishes secure communication across the nodes and handles the initiation of the training. The training and aggregation aspect is the part that usually requires more fine-tuning compared to the infrastructure. This is what motivated the development of FedML, where the design principle allows the developers to standardize benchmarking. A key feature of FedML is its support for a diverse set of topologies dictating how nodes can communicate with one another. The current version of FedML supports standalone simulation, distributed computation, topology customization, flexible and customizable message passing, and custom algorithm implementation. In this study, we deployed the standalone simulation capabilities of FedML to study the training capabilities on fluorescence images of 10 clients.

3.3. Contamination Classification

To classify CSI-D recorded video frames into *clean* and *contamination* categories, we first selected video excerpts and converted them to frames that were labeled *clean* and *contamination*. We then fed the labeled frames to a CNN model, MobileNetV3 [38], developed by Google, which is the next generation of the MobileNet family [39–41]. MobileNetV3 uses a novel architecture and a mix of complementary search methods to provide high accuracy and a computationally efficient model for mobile computer vision tasks. There are two models of MobileNetV3, MobileNetV3-Small and MobileNetV3-Large, designed for low or high resource usage needs, respectively. The architecture of these two models differs in terms of the number of blocks, expansion sizes, activation functions, etc.

Classification Model Architecture

MobileNetV3 depends on AutoML to identify the best architecture feasible in a search space for the given tasks. MnasNet [41] and NetAdapt [42] are used sequentially to exploit the search space more efficiently and discover and optimize the network architecture. MnasNet first uses reinforcement learning to identify the optimum global network structure by optimizing each network block, and then the NetAdapt algorithm tries to fine-tune the architecture by adjusting the number of filters per layer. Combining these two techniques can determine the best model for a specific hardware platform. MobileNetV3 also incorporates a squeeze-and-excitation block [43] into its core architecture to improve the network representational ability (emphasizing informative features and suppressing less helpful features) via adaptive channel-wise feature recalibration by explicitly modeling between the channels' interdependencies. MobileNetV3 uses an activation function called hard-swish, a modified version of the swish [44] activation function. Swish nonlinearity uses a sigmoid function, which is not computationally efficient. Hard-swish replaces the sigmoid with its piece-wise linear hard analog, as shown in (3):

$$hard\text{-}swish[x] = x\frac{ReLU6(x+3)}{6} \tag{3}$$

This replacement not only eliminates any numerical precision loss caused by various implementations of the approximate sigmoid but can also significantly minimize latency cost by reducing the number of memory accesses. In addition to the modifications mentioned above, some computationally expensive layers at the beginning and end of MobileNetV3 were redesigned to reduce the cost of feature generation while maintaining accuracy.

To train the MobileNetV3 for classification between *clean* and *contamination* frames, we needed to choose a suitable loss function and optimizer. Since we are dealing with a binary classification task, binary cross-entropy (BCE) [45] was chosen as the model loss function. Comparing predicted probability to actual class labels using BCE yields a result that is either 0 or 1. It then creates a score that penalizes the probability, depending on the difference between the predicted and actual values. If the predicted value is far from the actual value, the BCE loss will be increased. Equation (4) shows the definition of BCE.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \log(p(y_i)) + (1 - y_i) \times \log(1 - p(y_i)) \tag{4}$$

where N shows the training sample size, y_i is the assigned label, $p(y_i)$ shows the predicted probability of class 1 (*Contamination*), and $1 - p(y_i)$ represents the predicted probability of class 0 (*Clean*).

After the loss function is defined, the model requires an optimization algorithm to minimize the loss by changing the model weights and learning rate. We used the Adam (Adaptive Moment Estimation) optimizer [46] as a straightforward and computationally efficient approach for first-order gradient-based stochastic optimization. Adam estimates the first- and second-order moments to calculate individual adaptive learning rates for different parameters. Adam's mathematical expressions include:

$$v_t \leftarrow \beta_1 v_{t-1} + (1 - \beta_1) g_t \tag{5}$$

$$s_t \leftarrow \beta_2 s_{t-1} + (1 - \beta_2) g_t^2 \tag{6}$$

$$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1}) \tag{7}$$

$$\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_1^t} \ , \quad \hat{s}_t \leftarrow \frac{s_t}{1 - \beta_2^t} \tag{8}$$

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta \hat{v}_t}{\sqrt{\hat{s}_t + \epsilon}} \tag{9}$$

where g_t is the gradient and step t, v_t and s_t denote the exponential moving average of g_t and g_t^2, respectively. β_1 and β_2 are smoothing parameters for the first and second-order moments, $f()$ is the loss function to minimize, θ represents the parameter (weights), η shows the learning rate, and ϵ is a small number.

3.4. Contamination Segmentation

The precise segmentation of contaminated regions on a variety of surfaces is critical, since identifying *contamination* video frames using a classification model does not necessarily lead to detecting all contaminated areas. There might be many tiny contaminated spots strewn across a surface, making it difficult to recognize them all during a live inspection. Restaurants and kitchens usually contain many surfaces and objects that may create background fluorescence or reflection artifacts, making it more likely that inspectors could overlook some regions of contamination during an inspection. Segmenting and pseudo-coloring contaminated regions can make it easier for inspectors to identify and not miss any contamination. This is why we focused on the segmentation of contaminated regions in video frames already classified as *contamination* for the second goal of this study.

3.4.1. Semantic Segmentation and Pixel-Level Annotation

Instead of relying on threshold-based techniques, which are prone to error, we employed a semantic segmentation approach to perform accurate pixel-level classification in each frame classified as *contamination*. To categorize every pixel of a frame into a specific class (in our case, green fluorescence, red fluorescence, and background), pixel-level annotation is required. The annotated data is used to train a deep CNN to classify pixels, and the trained model can then be used to predict the probable class of pixels in unseen video frames (test set).

We used MATLAB R2021b image labeler for annotating when building the semantic segmentation training and testing datasets. Images can be annotated quickly and easily by sketching shapes that can be assigned region of interest (ROI) labels. In MATLAB image labeler, a rectangle, line, polygon, and projected cuboid can be used to construct a ground truth annotation for a single image or a series of images. As mentioned above, we have three different classes, and all pixels need to be annotated accordingly. Four image labelers annotated a total of 17,859 frames, supervised by two experts present throughout the data collection and another expert who provided review and training remotely.

3.4.2. Semantic Segmentation Model Architecture

To accomplish the semantic segmentation task, we used DeepLabv3+, a state-of-the-art semantic segmentation algorithm developed by Google researchers [47]. The DeepLabv3+ architecture comprises an encoder and a decoder. Multi-scale contextual information from the image is encoded, and object boundaries are precisely and accurately recovered by the decoder module. The encoder comprises three essential components: ResNet, atrous convolution, and atrous spatial pyramid pooling. In this study, we employed ResNet50 as the network backbone to extract features. Atrous convolution is a helpful technique that enables the model to directly modify the resolution of features generated by deep convolutional neural networks and alter the filter's field of view to collect multi-scale information. The mathematical expression of atrous convolution is as follows:

$$y[i] = \sum_k x[i + r{\cdot}k]w[k] \tag{10}$$

where r is the atrous rate (by changing it, the field of view of the filters can be modified adaptively), w depicts the convolution filter, and i and k show the pixel locations.

Atrous spatial pyramid pooling (ASPP), the other main module of DeepLabv3+, is used to resample the features extracted from the model backbone at several rates before convolution. This is equivalent to scanning the original image with several filters, each with a complimentary effective field of view, in order to capture objects and valuable visual context at different scales. We chose the default DeepLabv3+ dilation rate of 6 since contamination might occur in a very small or very large region. The outputs of the ASPP are concatenated and pass through a 1×1 convolution layer with 256 filters that can generate rich semantic information.

A bilinear upsampling factor of four is then applied to the encoder's features before passing them to the decoder section and concatenating them with low-level features from the backbone. The low-level features from the backbone network are subjected to a 1×1 convolution layer that limits the number of channels to prevent the encoder features' importance from being outweighed and complicating the training process. Once the low-level features and the encoder's rich features are combined, a few 3×3 convolution layers are used to improve the generated features, and finally, a bilinear upsampling by a factor of four is applied to generate the segmentation output.

4. Experimental Settings

To implement federated learning for the classification of *clean* and *contamination* frames, we used PyTorch v1.11.0. We resized all the images to (300, 300) for training and testing purposes. To implement FL for semantic segmentation, we used Tensorflow v2.2.0. Because semantic segmentation is computationally more expensive than classification, we resized all the images to (256, 256). Since the real-world implementation of the resulting FL models on our scanning systems is an important goal, we need to be able to develop models that can be deployed on the GPUs or TPUs that will be built into device processor systems. Some of the systems work well with Tensorflow, and some models work better in PyTorch when multi-GPU processors are used. We decided to ensure that we had experience deploying our models based on both frameworks. We have previously used PyTorch and Tensorflow in both our semantic segmentation and our classification research. In this FL project, we decided to use Pytorch for classification and Tensorflow for semantic segmentation. We did not have any intrinsic reason to select one over the other for these tasks since we have not finalized our system processor selection for our scanners. The FL framework was trained and tested using eight NVIDIA Tesla V100 GPUs with 32 GB RAM on a Red Hat Enterprise Linux Server 7.9 operating system.

5. Results and Discussion

5.1. Federated Learning Classification Model Performance

Performance testing of classification between the clean and contamination frames using FL was carried out on data from ten clients (facilities), including 44,185 *clean* frames and 46,882 *contamination* frames. The dataset description is shown in Table 1. As can be seen in Table 1., the establishments range from small restaurants to institutional kitchens. They also vary in the types of surfaces to be scanned and the relative cleanliness, as well as measurement issues discussed above. That is why the number of frames selected for model development varies between clients. We used data from eight initial clients (35,858 *clean* frames and 36,523 *contamination* frames) for training and validation of the model and data from two new clients (8327 *clean* frames and 10,359 *contamination* frames) for final testing. Data from the eight clients were randomly assigned to training (70%) and validation (30%) sets. The FedAvg algorithm requires that in each round, a subset of clients be selected for training the model and then evaluated on all clients before starting the next round. In this study, for each round of communication, four out of eight clients were randomly chosen for the training subset for each of the 150 rounds. We then validated the updated model on the validation data sets of eight clients. For each client, the training data and validation data were separate, so we could see how well the new model was generalized.

In each round, clients download the global model and train it on their local dataset over 100 epochs, with a local batch size of 32, using binary cross-entropy as the model loss function. We used Adam as the model optimizer with a learning rate of 1×10^{-5} and weight decay of 1×10^{-6}. Each client then returns the locally trained model to the server and the local models are aggregated using the FedAvg algorithm, and an updated global model is sent back to all clients. The updated global model is validated using each client's validation dataset before starting the next training round.

Table 1. Description of datasets.

	No. of Clients	No. of "Clean" Frames	No. of "Contamination" Frames	Total No. of Frames
Training/ Validation	1	708	3242	3950
	2	1585	7207	8792
	3	2740	1815	4555
	4	3893	2574	6467
	5	2679	2320	4999
	6	11,897	8629	20,526
	7	11,041	6486	17,527
	8	1315	4250	5565
External Testing	9	6354	7606	13,960
	10	1973	2753	4726

The FL classification model was evaluated using the six metrics of accuracy, precision, recall, specificity, F-score, and area under the curve (AUC). The first five metrics are defined as follows:

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \qquad (11)$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}}, \quad \text{F}_{\text{score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \qquad (12)$$

where TP, TN, FP, and FN show true positive, true negative, false positive, and false negative, respectively. In our case, TP shows correctly identified *contamination* frames, and TN shows *clean* frames. FP denotes the *clean* frame wrongly classified as *contamination*, and FN shows the *contamination* frame misclassified as *clean*.

We trained and validated the FL classification model over 150 rounds, and the updated global model was validated using the clients' validation dataset. Figure 4 shows the model accuracy and loss for each round of communication. The accuracy and loss of each round are the average accuracies and average losses of all eight clients.
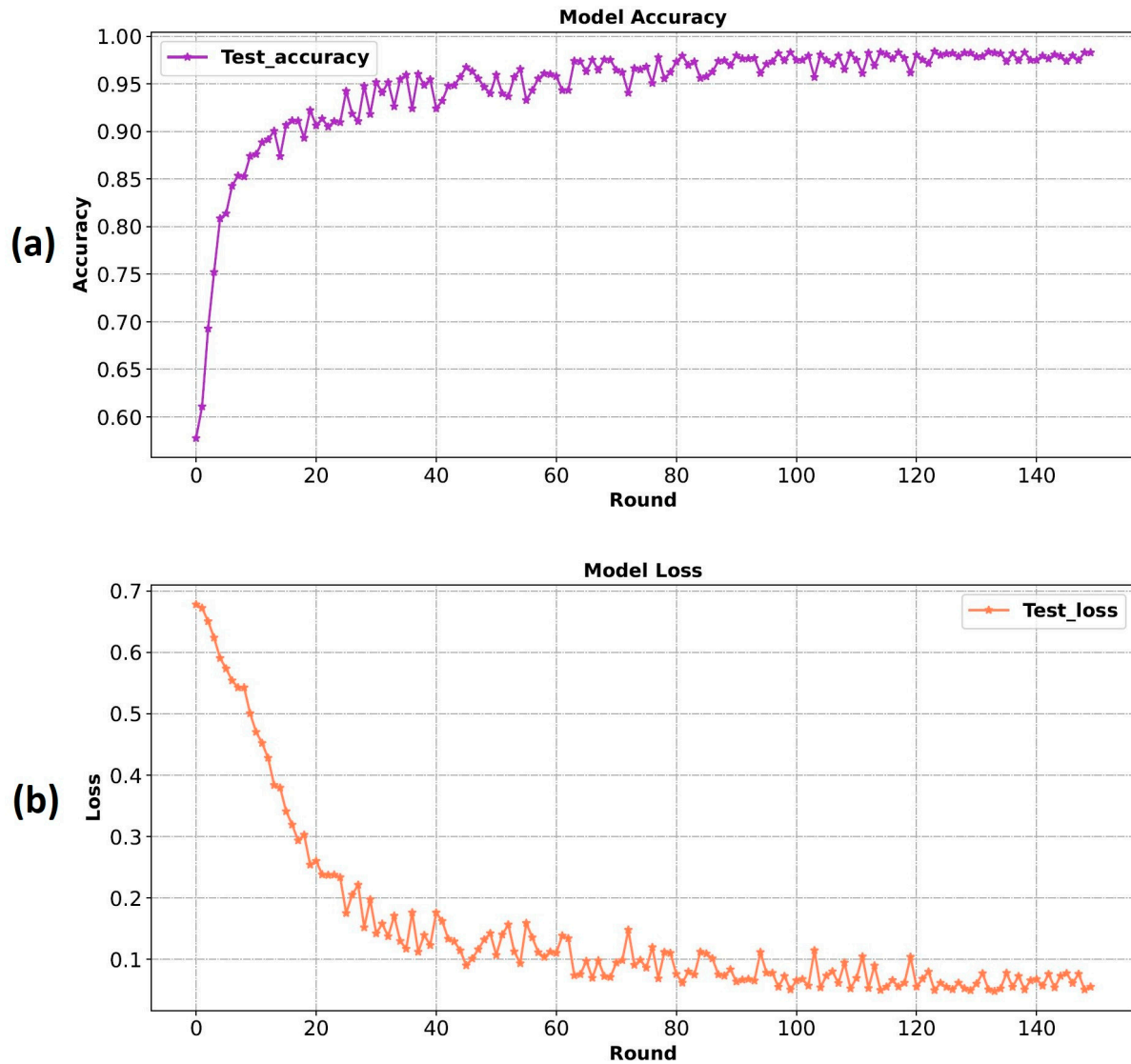
(a)

(b)

Figure 4. (**a**) FL model accuracy during training and validation. (**b**) FL model loss during training and validation.

After the model was trained and validated over 150 rounds, we tested it on two new clients. The FL model could identify *clean* and *contamination* frames with accuracies of 95.84% (precision of 96.88%, recall of 95.44%, specificity of 96.32%, F-score of 96.15%), and 94.92% (precision of 96.11%, recall of 95.13%, specificity of 96.63%, F-score of 95.62%) for clients 9 and 10, respectively. Figure 5 shows the FL model confusion matrix on the two new client data sets. The rows show the true label of *clean* and *contamination* frames, and the columns show the predicted labels.
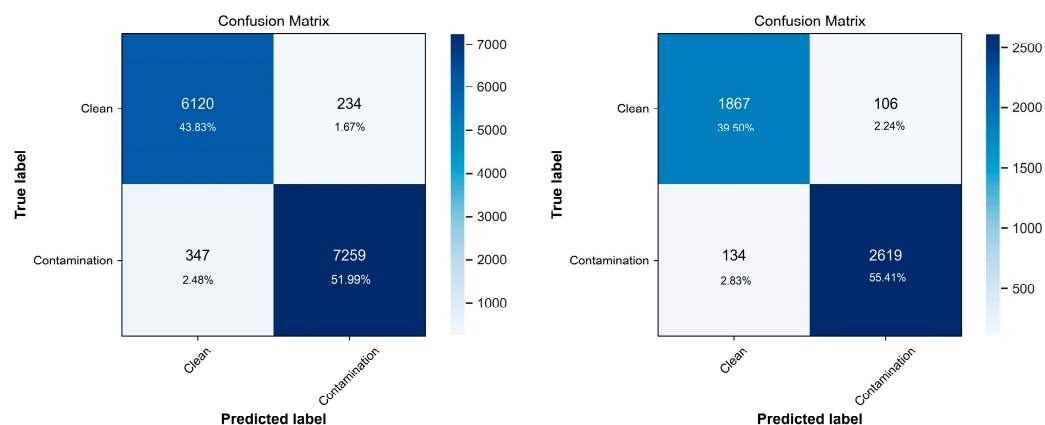
Figure 5. FL model confusion matrix when applied to two new client data sets.

5.2. Federated Learning Semantic Segmentation Model Performance

We employed DeepLabv3+ to perform FL semantic segmentation to precisely identify contamination areas on various surfaces. We used 12,000 annotated *contamination* frames from eight clients to train and validate the model and 5859 (3770 and 2089) annotated frames from two clients for testing to evaluate how effectively the FL model could generalize when confronted with a new client's data. We randomly split the data from the eight clients to train (70%) and validate (30%) the FL model. Similar to the frame classification, we trained the model on four randomly chosen clients in each round and validated it on all eight clients after model aggregation and before moving on to the next round.

We trained and validated the FL semantic segmentation model over 150 rounds. For each round, after downloading the global model, each client trains it on its local data for over 100 epochs with a local batch size of 16. For each client, we used categorical cross-entropy as the model loss function and Adam as the model optimizer with a learning rate of 1×10^{-5}. The locally trained models are sent to the server and aggregated using the FedAvg method, and a new global model is sent back to all clients to validate the model on the validation set and then start the next round.

To evaluate the FL semantic segmentation model performance, we used five metrics: intersection over union (IoU), precision, recall, specificity, and F-score. IoU is a common metric in semantic segmentation problems that measures the overlap between the ground truth (regions annotated by a human expert) and the model prediction. The following equation shows the IoU definition.

$$\text{IoU} = \frac{\text{TP}}{(\text{TP} + \text{FP} + \text{FN})} \tag{13}$$

The reason for not using accuracy as a semantic segmentation evaluation metric is that contamination usually affects a tiny fraction of the frames, and accuracy would always be more than 99%, making it inappropriate for evaluating the model's performance. The FL semantic segmentation model achieved a mean IoU of 91.23% and 89.45% for segmenting frames into the background, green fluorescence, and red fluorescence classes for clients 9 and 10, respectively.

In Figure 6, we show six image frames from a range of kitchen surfaces in the two new client datasets and the corresponding ground truth and model outputs to show how accurately the FL semantic segmentation algorithm was able to segment the red and green fluorescence contamination. The first row is the raw image captured via CSI-D, the second row depicts the FL model segmentation results, and the last row shows the human expert annotation (ground truth). When comparing FL model output to ground truth, it can be shown that the model successfully distinguishes and segments green and red fluorescence on various surfaces, including (Figure 6, left to right) under-sink plumbing, kitchen wall, kitchen countertop, a microwave door inner side, toaster, and refrigerator door handle.

This comparison shows the model's ability to recognize either red fluorescence or green fluorescence contamination, or a mixture of these two types of contamination, that takes the shape of minute drops, sprays, and splashes and larger regions such as spills or stains.
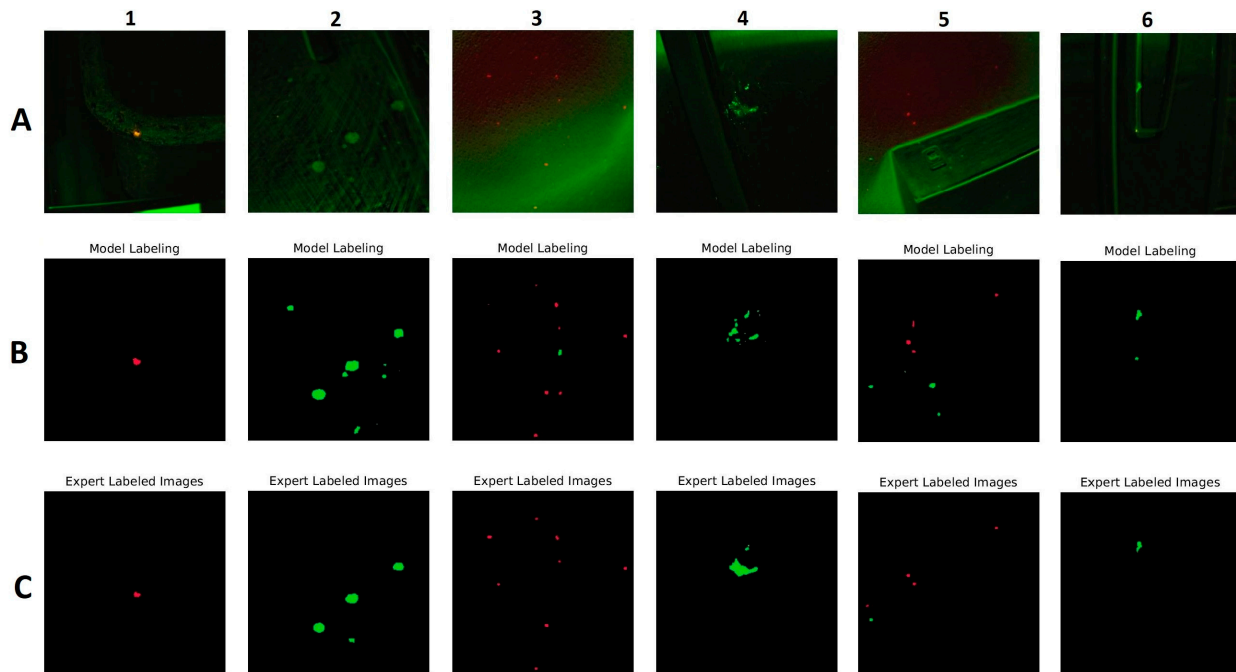

Figure 6. FL semantic segmentation model performance on new client's dataset. (**A**) Raw frames captured by CIS-D. (**B**) Segmented frames by FL model. (**C**) Annotated frames by human experts.

5.3. Privacy and Performance Trade-Off

As already discussed, the primary goal of using FL is to improve clients' data privacy. However, there would be a privacy and model performance trade-off. We considered two scenarios to evaluate at what cost we can have more privacy compared to a centralized approach. In the first scenario (S1), we combined data from eight clients and used it for training and evaluation of the model (81,961 frames) and tested the trained model on clients 9 and 10 (the two most recent data we collected, 9106 frames). The first scenario was performed for both classification and segmentation tasks. For the classification task, we achieved 96.36% accuracy, 97.32% precision, 95.96% recall, 96.84% specificity, and 96.64% F-score for testing the model on client 9. By testing the model on client 10, we could achieve an accuracy of 95.68%, a precision of 96.63%, a recall of 95.93%, a specificity of 95.34%, and an F-score of 96.28%. For the segmentation task, we could also achieve a mean IoU of 93.21% for client 9 and a mean IoU of 91.73% for client 10.

In the second scenario (S2), we combined data from all ten clients (91,067 frames) and used 70% of it for training, 20% for validation, and 10% for testing. We believe this scenario could result in the highest performance because the data from all the clients could be part of training, validation, and testing. For the classification task, the model had an accuracy of 97.81%, precision of 98.03%, recall of 97.71%, specificity of 97.91%, and F-score of 97.87%. For the segmentation task, the model had a mean IoU of 95.87. For more clarification, the comparison between the model performance for the FL and centralized (S1 and S2) approach is shown in Figures 7 and 8. If we consider client 9 in Figure 7, by testing the model trained using the FL approach, we could achieve an accuracy of 95.84, which is 0.52% less than the S1 and 1.97% less than S2. This comparison shows that by losing a negligible amount of performance, we can preserve the privacy of the client's data. Similarly, for the segmentation task, in Figure 8, by comparing the result for client 10 we can see the difference between the FL approach, S1, and S2 is 2.28% and 6.42%, respectively, which is reasonably acceptable performance.
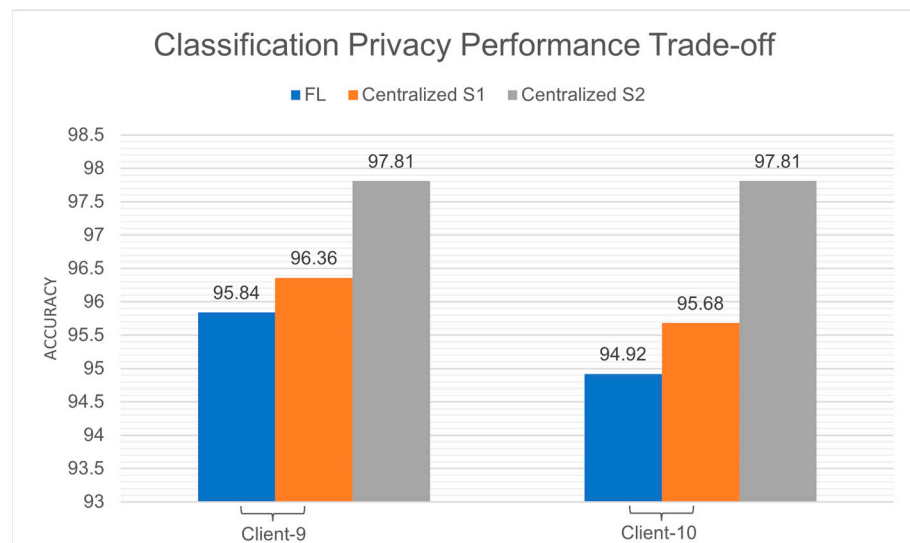
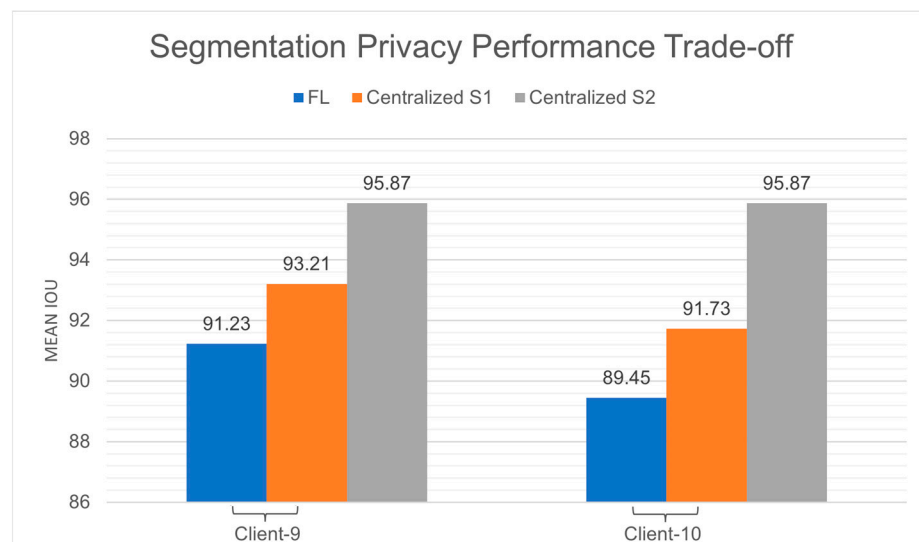**Figure 7.** Privacy performance trade-off for the classification task.

Figure 8. Privacy performance trade-off for the segmentation task.

Sanitization and infection prevention is of the utmost importance in food service establishments such as restaurants and institutional kitchens, since they are the most common places where outbreaks of foodborne illnesses occur. They must prevent food contamination and also must deal with potential infection spread among workers and customers. Beyond zero tolerance legal requirements and damage to institutional, brand, or restaurant reputation, loss of trust between workers and customers can be very costly [48]. Restaurants and other institutional kitchens may benefit from new technology such as the SafetySpect sanitization inspection technology, which can assist these establishments in maintaining a safer environment and a higher level of cleanliness. However, there might be doubts and concerns between companies and institutions that using new technologies such as CSI-D could raise privacy issues, and the leak of sensitive information could harm their reputations. To clear such doubts and concerns, this paper focuses on incorporating a data safety approach into the sanitization monitoring technology. We used federated learning, a new paradigm in machine learning that trains a model on client devices so that no private data is shared with other parties. FL can play a crucial role in designing scalable artificial intelligence (AI) technology when it is challenging to build up data-sharing agreements across clients or organizations. Recently, FL has been shown to be useful in

several areas [29,49]. We believe this is the first study where the main focus is on using FL in the food service industry, or for identification and segmentation of contamination.

However, implementing federated learning in the food service industry is not without its challenges. The deployment of FL requires a robust and reliable infrastructure that supports secure and efficient data communication. A secure and reliable communication channel is vital to maintain the integrity and privacy of the data during the exchange of updated model parameters between the central server and client devices. Further, the client devices must possess adequate computational capabilities to locally train complex deep learning models. This might be challenging, given that most devices used in the food service industry may not be equipped with high-end computational resources. Another hurdle is dealing with the data heterogeneity intrinsic to FL due to diverse, decentralized data sources. This heterogeneity could influence the model's performance, necessitating the development of techniques to mitigate its effects. Lastly, coordinating the training process among numerous client devices, ensuring synchronization, handling device failures or dropouts, and integrating new clients into the learning process all require robust protocols.

The main purpose of this research is a proof of concept for using cutting-edge imaging technology together with state-of-the-art deep learning algorithms and federated learning to enhance cleanliness assurance in the food service industry, while assuring client data remains private and within the client's information technology domain. We have shown that deep learning-based models trained using the FL approach could achieve results comparable to those of the centralized methods. In this particular case, training models using the FL approach can rapidly detect contamination using a portable handheld scanning device, paving the way for more precise, rapid, reliable, and secure examinations. This demonstrates how FL can be applied in other sensing and classification technologies.

There are trade-offs between privacy and performance, and it is important to consider the need for training classification algorithms for deployment on devices with more limited processing capabilities.

While centralized models can give the best performance, based on our experience discussing food safety issues with the industry, it is unrealistic to expect that clients would be willing to share any data that could negatively reflect on their sanitization procedures and potentially impact business value [50,51]. Our comparison of the benefit of FL vs. a theoretical scenario where all clients would be willing to share all their data is intended to illustrate what might be achievable with sufficient participation. While the magnitude of improvement that we discuss here seems relatively small, the benefit can be significant. An increase in accuracy means a decrease in false positives and false negatives. If we can reduce false positives and false negatives from 5% to 2%, that is a 60% reduction in contamination on food preparation or food production surfaces that can host pathogenic organisms. While false positives have a small impact in that they require additional cleaning, false negatives can result in contamination of entire batches of food, which can result in product recalls, illness, or death of customers and attendant legal consequences, as well as major damage to brand equity [48]. Thus, there is quite a benefit to a customer that participates in improving the model if privacy can be maintained. Other privacy preserving schemes such as data anonymization, differential privacy, or cryptographic methods [52–54] all involve some transfer of data outside of an organization, so there is potential for reconstruction of original data. There is also some loss in the utility of models generated by this data. We selected FL because we felt it had the level of security that clients would find acceptable and the improvement in model utility that justifies participation.

Many edge computing devices employ chip-based processing units, such as the Google Coral tensor processing unit (TPU), for high-performance neural net inferencing. Currently, these require quantized model parameters, and these can limit model classification or segmentation performance. This creates a trade-off between performance, speed, and privacy. As edge device processing unit technology improves, it may be possible to reduce or eliminate quantization, further improving the usefulness of FL. In this study, we show that using the principle of FL, with a negligible percentage of performance, clients can

ensure that their private local data is safe while benefiting from an accurate and reliable contamination detection model.

The impact of the number of clients selected in each round of FL training is also a bit of a trade-off. Using fewer clients for training on each round usually means many more rounds of communication to reach the maximal model performance. With more clients, we might reach maximal model performance sooner, but the generalization may be worse.

6. Conclusions

This study presents a federated learning model for identifying and segmenting organic-based contamination and biofilms on various surfaces. We captured video frames using a fluorescence-imaging technology developed by SafetySpect Inc. and used two state-of-the-art deep learning algorithms, MobileNetV3 and DeepLabv3+, for classification and semantic segmentation. FL allows a model to be trained without transferring client datasets to a central server, helping to address client data privacy and confidentiality concerns. In this study, we used FedML, with some modifications, as our FL framework and FedAvg as the aggregation algorithm. We collected data from 10 institutional kitchens and restaurants. We trained and validated the model on eight clients, including 35,858 *clean* and 36,523 *contamination* frames, and tested it on two clients (*clean*: 8327, *contamination*: 10,359). For differentiation between *clean* and *contamination* frames, the model achieved respective accuracies of 95.84% and 94.92% for clients 9 and 10. The FL-based semantic segmentation model was trained and validated on 12,000 annotated contamination frames from eight clients and tested on 5859 (3770 and 2089) annotated frames from two new clients. The model resulted in a mean IoU score of 91.23% for client 9 and 89.45% for client 10.

The results demonstrated that using new fluorescence imaging technology combined with FL-based deep learning models can not only improve the performance of cleanliness and safety auditing systems for food preparation facilities but also can improve data privacy assurance for clients in the food-service industry.

Author Contributions: Conceptualization, H.T.G., N.M., M.S.K., F.V., A.A. and K.T.; methodology, H.T.G., M.S., E.M., F.V., A.A., S.A. and K.T.; validation, M.S.K., F.V., A.A., J.Q. and K.T.; formal analysis, H.T.G. and E.M.; investigation, N.M., F.V., S.A., K.T. and M.S.K.; resources, M.S.K., D.E.C. and J.Q.; data collection, and curation, H.T.G., H.K.Z. and S.S.; writing—original draft preparation, H.T.G., M.S. and K.H.; writing—review and editing, N.M., H.T.G., D.E.C., F.V., S.A., K.T., J.Q., A.A. and I.B.; visualization, S.M.S. and H.T.G.; supervision, K.T., N.M., F.V., M.S.K. and I.B.; project administration, F.V. and K.T.; funding acquisition, N.M., F.V. and K.T. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: F.V., N.M., S.S., K.H., K.T., H.K.Z., M.S. and H.T.G. serve as employees of or consultants to or hold stocks or stock options in SafetySpect Inc. F.V. and N.M. are inventors on US patent application US20210228757A1. The remaining authors declare no conflict of interest.

References

1. Pires, S.M.; Desta, B.N.; Mughini-Gras, L.; Mmbaga, B.T.; Fayemi, O.E.; Salvador, E.M.; Gobena, T.; Majowicz, S.E.; Hald, T.; Hoejskov, P.S. Burden of foodborne diseases: Think global, act local. *Curr. Opin. Food Sci.* **2021**, *39*, 152–159. [CrossRef] [PubMed]
2. CDC. Estimates of Foodborne Illness in the United States. Available online: https://www.cdc.gov/foodborneburden/index.html (accessed on 1 August 2023).
3. Dewey-Mattia, D.; Manikonda, K.; Hall, A.J.; Wise, M.E.; Crowe, S.J. Surveillance for foodborne disease outbreaks—United States, 2009–2015. *MMWR Surveill. Summ.* **2018**, *67*, 1. [CrossRef]
4. Abban, S.; Jakobsen, M.; Jespersen, L. Attachment behaviour of Escherichia coli K12 and Salmonella Typhimurium P6 on food contact surfaces for food transportation. *Food Microbiol.* **2012**, *31*, 139–147. [CrossRef]
5. Quan, Y.; Kim, H.-Y.; Shin, I.-S. Bactericidal activity of strong acidic hypochlorous water against *Escherichia coli* O157: H7 and *Listeria monocytogenes* in biofilms attached to stainless steel. *Food Sci. Biotechnol.* **2017**, *26*, 841–846. [CrossRef] [PubMed]
6. Verran, J.; Redfern, J.; Smith, L.; Whitehead, K. A critical evaluation of sampling methods used for assessing microorganisms on surfaces. *Food Bioprod. Process.* **2010**, *88*, 335–340. [CrossRef]
7. Keresztes, J.C.; Goodarzi, M.; Saeys, W. Real-time pixel based early apple bruise detection using short wave infrared hyperspectral imaging in combination with calibration and glare correction techniques. *Food Control* **2016**, *66*, 215–226. [CrossRef]
8. Coelho, P.A.; Torres, S.N.; Ramírez, W.E.; Gutiérrez, P.A.; Toro, C.A.; Soto, J.G.; Sbarbaro, D.G.; Pezoa, J.E. A machine vision system for automatic detection of parasites Edotea magellanica in shell-off cooked clam Mulinia edulis. *J. Food Eng.* **2016**, *181*, 84–91. [CrossRef]
9. Dutta, M.K.; Singh, A.; Ghosal, S. A computer vision based technique for identification of acrylamide in potato chips. *Comput. Electron. Agric.* **2015**, *119*, 40–50. [CrossRef]
10. Al-Sarayreh, M.; Reis, M.M.; Yan, W.Q.; Klette, R. A sequential CNN approach for foreign object detection in hyperspectral images. In *Computer Analysis of Images and Patterns*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; pp. 271–283.
11. Jayasinghe, L.; Wijerathne, N.; Yuen, C. A deep learning approach for classification of cleanliness in restrooms. In Proceedings of the 2018 International Conference on Intelligent and Advanced System (ICIAS), Kuala Lumpur, Malaysia, 13–14 August 2018; pp. 1–6.
12. Gorji, H.T.; Shahabi, S.M.; Sharma, A.; Tande, L.Q.; Husarik, K.; Qin, J.; Chan, D.E.; Baek, I.; Kim, M.S.; MacKinnon, N. Combining deep learning and fluorescence imaging to automatically identify fecal contamination on meat carcasses. *Sci. Rep.* **2022**, *12*, 2392. [CrossRef]
13. Taheri Gorji, H.; Van Kessel, J.A.S.; Haley, B.J.; Husarik, K.; Sonnier, J.; Kashani Zadeh, H.; Chan, D.E.; Qin, J.; Baek, I.; Kim, M.S. Deep Learning and Multiwavelength Fluorescence Imaging for Cleanliness Assessment and Disinfection in Food Services. *Front. Sens.* **2022**, *3*, 25.
14. Husarik, K.; Gorji, H.T.; Qin, J.; Chan, D.E.; Baek, I.; Kim, M.S.; Thompson, M.S.; MacKinnon, N.; Sokolov, S.; Vasefi, F. Cleanliness assessment in long-term care facilities using deep learning and multiwavelength fluorescence imaging. In Proceedings of the Sensing for Agriculture and Food Quality and Safety XV, Orlando, FL, USA, 30 April–5 May 2023; pp. 41–53.
15. Propp, C.; Woods, L.; Gorji, H.T.; Husarik, K.; Sueker, M.; Qin, J.; Baek, I.; Kim, M.S.; Chan, D.E.; Sokolov, S. Dual-excitation fluorescence imaging system for contamination detection in food facilities. In Proceedings of the Sensing for Agriculture and Food Quality and Safety XV, Orlando, FL, USA, 30 April–5 May 2023; pp. 135–142.
16. Woods, L.; Propp, C.; Sueker, M.; Husarik, K.; Gorji, H.T.; Qin, J.; Baek, I.; Kim, M.S.; Chan, D.E.; Sokolov, S. Efficacy of sanitization in healthcare using deep learning and multiwavelength fluorescence imaging. In Proceedings of the Sensing for Agriculture and Food Quality and Safety XV, Orlando, FL, USA, 30 April–5 May 2023; pp. 143–148.
17. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). In *A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10, pp. 10–5555.
18. Gaff, B.M.; Sussman, H.E.; Geetter, J. Privacy and big data. *Computer* **2014**, *47*, 7–9. [CrossRef]
19. Pardau, S.L. The California consumer privacy act: Towards a European-style privacy regime in the United States. *J. Tech. Law Policy* **2018**, *23*, 68.
20. Sueker, M.; Stromsodt, K.; Gorji, H.T.; Vasefi, F.; Khan, N.; Schmit, T.; Varma, R.; Mackinnon, N.; Sokolov, S.; Akhbardeh, A. Handheld Multispectral Fluorescence Imaging System to Detect and Disinfect Surface Contamination. *Sensors* **2021**, *21*, 7222. [CrossRef] [PubMed]
21. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
22. Rodríguez-Barroso, N.; López, D.J.; Luzón, M.; Herrera, F.; Martínez-Cámara, E. Survey on Federated Learning Threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *arXiv* **2022**, arXiv:2201.08135. [CrossRef]
23. Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; Ramage, D. Federated learning for mobile keyboard prediction. *arXiv* **2018**, arXiv:1811.03604.
24. Yang, T.; Andrew, G.; Eichner, H.; Sun, H.; Li, W.; Kong, N.; Ramage, D.; Beaufays, F. Applied federated learning: Improving google keyboard query suggestions. *arXiv* **2018**, arXiv:1812.02903.

25. Ramaswamy, S.; Mathews, R.; Rao, K.; Beaufays, F. Federated learning for emoji prediction in a mobile keyboard. *arXiv* **2019**, arXiv:1906.04329.

26. Silva, S.; Gutman, B.A.; Romero, E.; Thompson, P.M.; Altmann, A.; Lorenzi, M. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In Proceedings of the 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 270–274.

27. Gao, D.; Ju, C.; Wei, X.; Liu, Y.; Chen, T.; Yang, Q. Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography. *arXiv* **2019**, arXiv:1909.05784.

28. Liu, B.; Yan, B.; Zhou, Y.; Yang, Y.; Zhang, Y. Experiments of federated learning for COVID-19 chest X-ray images. *arXiv* **2020**, arXiv:2007.05592.

29. Li, L.; Fan, Y.; Tse, M.; Lin, K.-Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]

30. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3347–3366. [CrossRef]

31. Colom-Cadena, M.; Tulloch, J.; Jackson, R.J.; Catterson, J.H.; Rose, J.; Davies, C.; Hooley, M.; Anton-Fernandez, A.; Dunnett, S.; Tempelaar, R. TMEM97 increases in synapses and is a potential synaptic Aβ binding partner in human Alzheimer's disease. *bioRxiv* **2021**. [CrossRef]

32. Ziller, A.; Trask, A.; Lopardo, A.; Szymkow, B.; Wagner, B.; Bluemke, E.; Nounahon, J.-M.; Passerat-Palmbach, J.; Prakash, K.; Rose, N. Pysyft: A library for easy federated learning. In *Federated Learning Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 111–139.

33. Beutel, D.J.; Topal, T.; Mathur, A.; Qiu, X.; Parcollet, T.; de Gusmão, P.P.; Lane, N.D. Flower: A friendly federated learning research framework. *arXiv* **2020**, arXiv:2007.14390.

34. He, C.; Li, S.; So, J.; Zeng, X.; Zhang, M.; Wang, H.; Wang, X.; Vepakomma, P.; Singh, A.; Qiu, H. Fedml: A research library and benchmark for federated machine learning. *arXiv* **2020**, arXiv:2007.13518.

35. Wen, Y.; Li, W.; Roth, H.; Dogra, P. Federated Learning Powered by NVIDIA Clara. Available online: https://developer.nvidia.com/blog/federated-learning-clara/ (accessed on 12 April 2023).

36. Liu, Y.; Fan, T.; Chen, T.; Xu, Q.; Yang, Q. FATE: An industrial grade platform for collaborative learning with data protection. *J. Mach. Learn. Res.* **2021**, *22*, 1–6.

37. Caldas, S.; Duddu, S.M.K.; Wu, P.; Li, T.; Konečný, J.; McMahan, H.B.; Smith, V.; Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv* **2018**, arXiv:1812.01097.

38. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.

39. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.

40. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

41. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2820–2828.

42. Yang, T.-J.; Howard, A.; Chen, B.; Zhang, X.; Go, A.; Sandler, M.; Sze, V.; Adam, H. Netadapt: Platform-aware neural network adaptation for mobile applications. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 285–300.

43. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

44. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.

45. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4.

46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

47. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.

48. Brand Finance. Food Safety. 2018. Available online: https://brandirectory.com/download-report/Food%20Safety%20Report.pdf (accessed on 4 July 2022).

49. Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* **2020**, *8*, 140699–140725. [CrossRef]

50. Open Data Institute. Why Businesses Aren't Sharing More Data. Available online: https://theodi.org/article/why-businesses-arent-sharing-more-data/ (accessed on 21 March 2023).

51. Stitzlein, C.; Fielke, S.; Waldner, F.; Sanderson, T. Reputational risk associated with big data research and development: An interdisciplinary perspective. *Sustainability* **2021**, *13*, 9280. [CrossRef]

52. Murthy, S.; Bakar, A.A.; Rahim, F.A.; Ramli, R. A comparative study of data anonymization techniques. In Proceedings of the 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Washington, DC, USA, 27–29 May 2019; pp. 306–309.
53. Dwork, C. Differential privacy: A survey of results. In Proceedings of the Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, 25–29 April 2008; pp. 1–19.
54. Bhanot, R.; Hans, R. A review and comparative analysis of various encryption algorithms. *Int. J. Secur. Its Appl.* **2015**, *9*, 289–306. [CrossRef]