5-2-2021

Multi-scale, class-generic, privacy-preserving video

Zhixiang Zhang
King's College London

Thomas Cilloni
University of Mississippi

Charles Walter
University of Mississippi

Charles Fleming
University of Mississippi

Recommended Citation

Zhang, Z., Cilloni, T., Walter, C., & Fleming, C. (2021). Multi-scale, class-generic, privacy-preserving video. Electronics, 10(10), 1172. https://doi.org/10.3390/electronics10101172

electronics

MDPI

Article

Multi-Scale, Class-Generic, Privacy-Preserving Video

Zhixiang Zhang [1], Thomas Cilloni [2], Charles Walter [2] and Charles Fleming [2,*]

[1] Department of Computing, King's College London, London SW7 2AZ, UK; zhixiang.zhang@kcl.ac.uk
[2] Department of Computer and Information Science, University of Mississippi, Oxford, MS 38677, USA; tcilloni@go.olemiss.edu (T.C.); cwwalter@olemiss.edu (C.W.)
* Correspondence: fleming@olemiss.edu

Abstract: In recent years, high-performance video recording devices have become ubiquitous, posing an unprecedented challenge to preserving personal privacy. As a result, privacy-preserving video systems have been receiving increased attention. In this paper, we present a novel privacy-preserving video algorithm that uses semantic segmentation to identify regions of interest, which are then anonymized with an adaptive blurring algorithm. This algorithm addresses two of the most important shortcomings of existing solutions: it is multi-scale, meaning it can identify and uniformly anonymize objects of different scales in the same image, and it is class-generic, so it can be used to anonymize any class of objects of interest. We show experimentally that our algorithm achieves excellent anonymity while preserving meaning in the visual data processed.

Keywords: privacy-preserving video; video anonymization; computer systems; data privacy

Citation: Zhang, Z.; Cilloni, T.; Walter, C.; Fleming, C. Multi-Scale, Class-Generic, Privacy-Preserving Video. *Electronics* **2021**, *10*, 1172. https://doi.org/10.3390/electronics10101172

Academic Editor: Younho Lee

Received: 5 January 2021
Accepted: 6 May 2021
Published: 14 May 2021

1. Introduction

Video capture devices have become ubiquitous [1]. Modern cities are now densely covered by advanced surveillance cameras networks [2] and mobile devices with video capture capabilities are inexpensive and readily available in almost every country in the world. Even entry-level smartphones have the ability to record videos in Full High Definition (FHD) resolution (1920 × 1080 pixels) and frame rates up to 30 frames per second (FPS). In addition, advances in machine learning for visual data understanding mean that large amounts of recorded video can be processed quickly and easily, and semantic information extracted automatically. The net result of these advances is that personal privacy is rapidly shrinking.

Constructing a video anonymization system is a common solution to protect privacy in systems that deal with visual or audio data [3,4]. The most common approach is to process a raw video or a set of images by applying multiple privacy filters. These filters either obfuscate sensitive information or completely replace it with unidentifiable versions of the that same data [2]. Two general types of algorithms have been developed. The first are global algorithms that apply a uniform transformation to the whole image, such as Gaussian blur, superpixelation, downsampling, or wavelet decomposition [5–8]. These methods are fast and simple to implement but have several downsides. First, because they are applied uniformly across an image, they do not provide the same level of anonymity to objects at different distances. For example, if a face is three feet from a camera, it will be much clearer than a face that is several yards away. In fact, to achieve sufficient anonymity for very near objects, it may be necessary to blur the image to the point that the most distant objects become indistinguishable from the background [8]. Second, because they transform the entire image, they may destroy information required for the task the video is recorded for. For example, blurring traffic camera data to anonymize faces may reduce license plate recognition rates.

The second type of algorithm is machine learning based. These algorithms recognize certain features in images and apply local filters, masks, or transformations [1,9,10]. While

these algorithms solve some of the problems with global algorithms, they also suffer from multiple shortcomings. The first is that they are generally specialized to detect and anonymize a particular aspect of the image, in almost all cases faces. While faces are definitely an important privacy feature, other aspects of the image may also be sensitive: license plates, street signs, car make and model, etc. Unfortunately most of these algorithms do not easily generalize to other classes of objects. For example, face detection and writing detection models are architecturally very different (see [11] for a good example of a state-of-the-art text detector). The other major drawback is that these types of algorithms have problems with multi-scale detection [12]. As a result, while faces in the foreground may be well recognized, smaller scale faces, such as those in the background, may be missed. Other related systems can be found in [13–17].

In this paper, we propose a different technique. Rather than developing a detector for a specific class of objects, we use semantic segmentation, which generates pixel-level class labels for the entire image, using the DeepLab algorithm [18,19]. This algorithm has several advantages. First, it can be trained on one or more classes, ranging from text to faces, allowing the use of a single model to anonymize a wide range of classes, or even multiple classes at the same time. Second, it is multi-scale, meaning it can correctly classify pixels belonging to objects for a wide variety of scales. Based on the output from the semantic segmentation stage, we perform a scale-dependent Gaussian blur on the pixels of interest. The resulting system gives us an extremely flexible method to effectively anonymize a wide range of object classes at a wide range of scales, without negatively affecting the performance in the task for which the video was recorded.

To demonstrate the viability and flexibility of the system, we first show that we can train DeepLab to label pixels for a wide range of classes and scales. We then consider two tasks: human action recognition and license plate recognition. For human action recognition, we anonymize the human subject in the standard UCF101 dataset, and show that this has only a minimal effect on the action recognition rate. We repeat this at various scales. We then consider license plate recognition and show that our algorithm allows us to completely anonymize license plates in the Chinese City Parking Dataset (CCPD).

2. Background and Related Works

2.1. Semantic Image Segmentation

Semantic image segmentation is one of the fundamental topics in the field of computer vision [18]. The objective of semantic segmentation is to cluster all parts of an image that belong to the same object [20]. In pixel-level semantic image segmentation, every pixel in the target image should be classified as belonging to a certain object class and be labeled accordingly [19]. Generally, this results in an image "mask", with pixel classes indicated by the value of the corresponding pixel in the mask (see Figure 1). Different from object detection, semantic image segmentation does not distinguish different instances of the same class of objects [21].

Figure 1. An example of semantic image segmentation taken from the Pascal Visual Object Class dataset. The image on the right is a mask, where each pixel is numbered according to the image class.

Up until five years ago, traditional image segmentation algorithms heavily relying on domain knowledge (i.e., that did not apply neural networks) were regarded as the mainstream approach to computer vision tasks by the scientific community [20]. In these traditional approaches, a fundamental part of the process was choosing the features. Pixel colors, histograms of oriented gradients (HOG), scale-invariant feature transformations (SIFT), bag-of-visual-words (BOV), poselets, and textons were among the most frequently chosen features [20]. Picking several features for each pixel in high-resolution images leads to high computational loads in the model training process. Therefore, pre-processing methods of dimensionality reduction, such as image down-sampling and principal component analysis (PCA), were often used prior to semantic image segmentation [22].

In recent years, researchers have made numerous attempts to use deep-learning techniques in training of semantic image segmentation systems. The fundamental idea is to handle a trained neural network as a convolution and apply it on the input pixel data, thus efficiently implementing the sliding window process [20]. Published papers (e.g., [23,24]) show that the use of deep-learning techniques enhance many features of semantic image segmentation models. Moreover, these new deep-learning based semantic segmentation models have significant advantages on segmentation accuracy and efficiency over models trained with traditional approaches [18,24]. Semantic segmentation with deep neural networks is a well-studied topic. An excellent survey of these methods can be found in [25]. Some of the more recent methods include: MobileNetv3 [26], SVCNet [27], CFNet [28], and HFCNet [29].

2.2. DeepLab

In this project, we utilize DeepLab to implement the analyzer component. DeepLab is a deep-learning based semantic image segmentation model developed by Google, delivering high performance on most commonly used computer vision testing datasets, such as PASCAL VOC 2012 and Cityscapes [19].

DeepLab combines networks trained for image classification with the "atrous convolution", atrous spatial pyramid pooling (ASPP), Deep Convolutional Neural Networks (DCNN), and fully-connected Conditional Random Fields (CRF). Atrous convolutions enable this model to explicitly control the resolution at which feature responses are computed with DCNNs and allows the model to incorporate a larger context without an increase in computational requirements. It is also notable that the model has the capacity to provide robust segmentation features at multiple scales by making use of ASPP [30]. Incoming convolutional feature layers can be probed by ASPP with multi-sampling-rates filters and effective fields-of-views. Finally, DeepLab achieves high accuracy in localizing entities by combining methods from DCNNs and probabilistic graphical models, to which a fully-connected CRF is applied to eradicate any loss of localization accuracy [23]. Thanks to all these techniques, DeepLab can produce semantic predictions with a pixel-level accuracy and detailed segmentation maps along objects' boundaries. An illustration of the DeepLab network is shown in Figure 2.

As some of the components of DeepLab are complex compared to other DNNs, we review how these components work.

- Atrous Convolutions are a type of convolution that introduces a new parameter called the "dilation rate". While normal convolutional filters map each filter coefficient onto adjacent pixels, atrous convolutions allow for spacing between kernel values. For example, a 3×3 kernel with a dilation rate of 2 will convolve each filter weight with every other pixel (in a checkerboard pattern), effectively turning it into a 5×5 filter while maintaining the 3×3 filter computational cost.
- Atrous Spatial Pyramid Pooling (ASPP) uses multiple atrous convolutions, each with different dilation rates, to capture image information at different scales.
- Fully Connected Conditional Random Fields (CRF) are used to smooth segmentation maps as a post-processing step. These models have two terms. The first one corresponds to the softmax probability of the pixel class assigned to each pixel. The second

is a "penalty term" that penalizes pixels that are close together but have different labels. Labels are assigned by finding the maximal probability label assignments under this model.

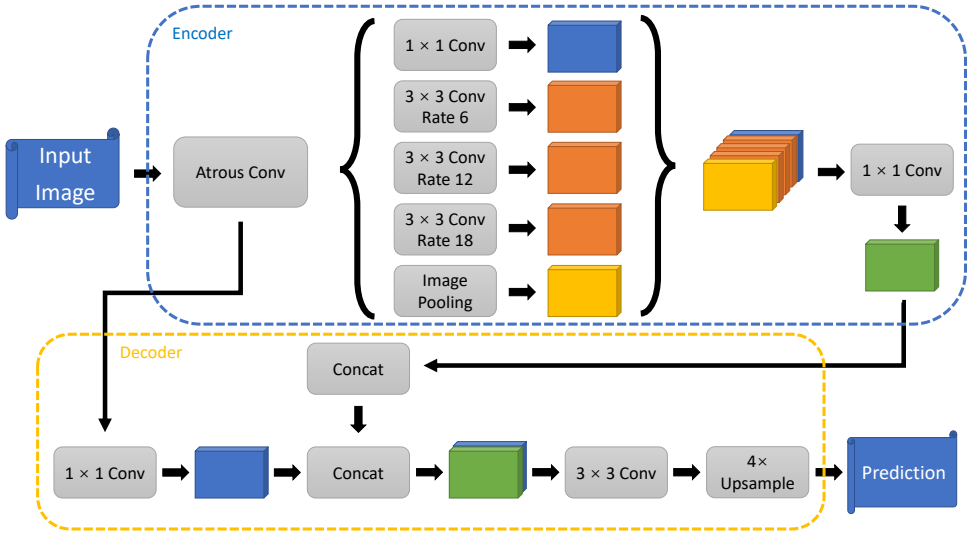

Figure 2. A high-level illustration of DeepLabv3+. The general structure is an atrous convolution, followed by atrous pyramid pooling, with results from both layers concatenated and both used as inputs to the final layers.

Several upgraded models of DeepLab have been developed and open-sourced by Google since the its first release. The specific version we chose for this project is DeepLabv3+, released in February 2018, and the latest at the time of the experiments. DeepLabv3+'s new features include a new encode–decoder structure, the module Xception, and atrous separable convolutions. By using the earlier versions of DeepLab for the encoder module and adding an effective decoder module to refine object boundaries [31], the model can achieve good performance in capturing sharp object boundaries. Additionally, the use of the Xception model, which has shown promising image classification and object detection results [24], allows the new model to be faster and have a better accuracy. The effectiveness of DeepLabv3+ is demonstrated by its accuracy of 89.0% and 82.1% on PASCAL VOC 2012 and Cityscapes datasets, respectively [18].

2.3. Gaussian Blur Algorithm

Gaussian blur is a convolution filter that can provide anonymity to the applied images [5]. Due to its simplicity and practicability, it is widely used in many image processing-related applications, such as Adobe Photoshop [3].

Convolutional filters are one of the most fundamental image processing techniques. Convolutional filters are usually separately applied to every single pixel in the target image. In each convolution, the feature values of a pixel and its neighboring pixels are captured by a fixed-size convolution kernel [5]. According to the position of a pixel in the convolution kernel, this will be assigned a specific weight. Finally, a new feature value will be calculated and will overwrite the original value. This is calculated as the weighted average of the captured feature values. Various visual transformations, such as image sharpening, embossing, and image obfuscation, can be achieved by applying convolutional filters with different distributions of weights in the convolution kernel [32].

Gaussian blur is a convolutional filter whose kernel weights follow a normal (Gaussian) distribution [32]. Since the pixel matrix of a 2D image is two-dimensional, a 2D normal distribution is used in the Gaussian blurring algorithm [5]. Similar to a one-dimensional normal distribution, if a neighbor pixel is located close to the source pixel in the original image, the weight of that pixel will be higher than those that are more distant, which means

it contributes more to the final result of the new feature value of the source pixel. This schema of distributed weights gives the Gaussian blur algorithm the ability to provide smooth image obfuscation.

Equation (1), called the "Gaussian function", shows the density equation of a two-dimensional normal distribution [5].

$$G(x,y) = \frac{1}{\pi\sigma_x^2 + \pi\sigma_y^2} e^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)} \quad (1)$$

In in this equation, (x, y) refers to a coordinate position in the convolution kernel, $(x_0; y_0)$ is the coordinate of the kernel's center, and σ_x and σ_y refer to the standard deviations in the directions of the abscissa and ordinate, respectively [5]. In this case, the coordinates of the kernel's center are always (0, 0), while the standard deviations in the two directions the same and are replaced by σ. Consequently, the previous function can be simplified to Equation (2):

$$G(x,y) = \frac{1}{\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2)$$

Because we are implementing this convolution filter for a discretized image, we need to discretize the Gaussian filter as well. This is done by approximating the continuous filter as an R $\times$ R matrix of coefficients, where R is odd. These coefficients are the values of the Gaussian kernel at discrete points around the center. This filter is convolved with the image, and the current pixel value is replaced by this weighted average of the surrounding pixels. Because our system handles anonymization at different scales, the value of R will vary, as well as the value of σ.

3. Design and Implementation

3.1. System Design

Our system has two stages. The first stage, which we call the analyzer, takes the original image and generates a semantic segmentation label mask. This mask, along with the original image, is fed into the anonymizer, which adaptively generates a Gaussian blurring filter based on the size of the region to be blurred. Figure 3 illustrates this basic architecture.

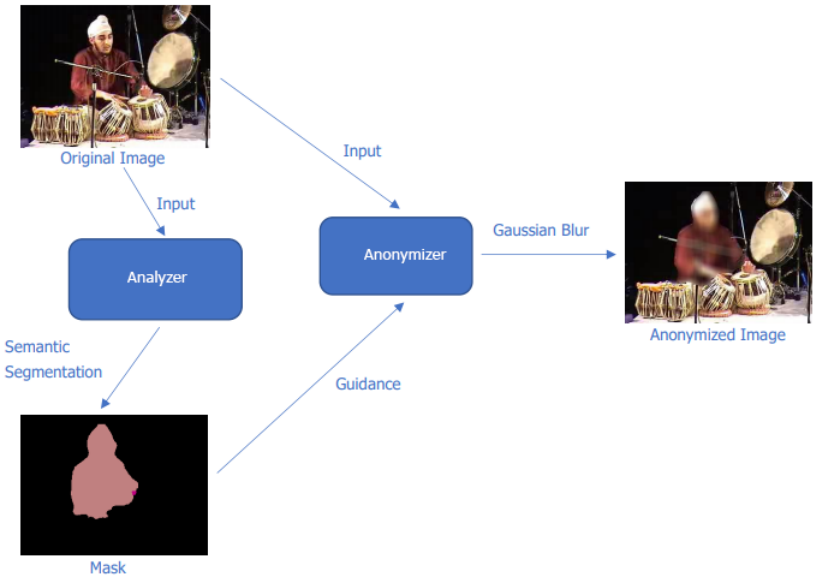

Figure 3. This system diagram shows the various processing stages of the algorithm. The raw image is fed into the semantic segmentor, which generates a pixel level label mask. These are both combined in the anonymization step to generate the final anonymized image.

3.2. Analyzer

The analyzer component performs several tasks. The first is to convert the input data into the standard format (standard 24 bit RGB bitmap) for the semantic segmentation component. Because our system can handle either video or images, video input is decompressed and converted to individual frames, which are fed into the semantic segmenter. These frames will be recombined into the output video at the end of the anonymization process.

The second task of the analyzer is to generate the pixel label mask image (see Figure 4). This is done using the semantic segmentation algorithm available in DeepLabv3+, the newest version of DeepLab developed and open-sourced by Google. The output mask image is the same dimension as the input image, with each pixel set to the identified class value or zero if the pixel was not identified as belonging to any of the known classes. This mask, along with the original image, is then passed to the anonymizer.

Figure 4. Examples of the image masks from DeepLabv3+. These images come from the Pascal Visual Objects Category dataset. Each color denotes a different class label, with black being background.

Because our implementation follows the guidance provided by Google's official documentation, the model training process strictly follows the training protocols used in [18,33]. In this section, only some fundamentally important methods and parameter settings are listed. The complete versions of the training protocol can be found in [18,33]. A "poly" learning rate policy was employed in the training. The initial learning rate is set to 0.007. More details of the "poly" learning rate policy can be found in [19,34]. The output stride was set to 16.

As DeepLabv3+ uses large-rate atrous convolutions, we must choose a large crop size. If our chosen crop size is too small, DeepLabv3+ can be affected [18,33]. Therefore, a large crop size (513×513) is used by the model for training. With the purpose to enrich the training dataset, we apply data augmentation by flipping and scaling the input images. The scaling factor is in the range of 0.5–2.0 and the flipping can be to the right or to the left. In addition, the choices of the scaling factor and the flipping direction are randomized [33].

Our implemented DeepLabv3+ system is trained with the augmented PASCAL VOC dataset. In the original PASCAL VOC dataset, made of 1464 training samples, 1449 validation samples and 1456 testing samples, images are annotated with their content at pixel-level. For the training phase, extra annotations provided by Dr.Sleep [35] are used for augmentation. As a result, there are 10,582 augmented training images in the dataset used [18].

The trained DeepLabv3+ model in our proposed system has the ability to perform semantic image segmentation by classifying pixels into 21 different classes of object (one of which is the background class). Each pixel of the output image contains a value that represents one class of objects [35]. For example, for each pixel classified into the class "Person" in a segmented image, the output image contains the RGB value (192, 128, 128).

The mean of the intersection-over-union of pixels across the 21 classes (mIOU) is the performance measure. For this implementation, the trained model can achieve a 77.31% mIOU accuracy on the Pascal VOC 2012 validation dataset [18].

3.3. Anonymizer

The anonymization algorithm used by the anonymizer is the Gaussian Blur, which replaces the feature value of a source pixel with the weighted average (following a normal distribution) of its neighboring pixels [5]. We implemented the anonymizer with Python.

The core idea behind our implementation is same as that of general Gaussian Blur [5] and the pixel features we chose are the RGB values, which means that the Gaussian kernel needs to apply a convolution to the same pixel three times to get its new R, G, and B values. Different levels of object obfuscation can be achieved by choosing varying convolution kernels. These are defined by two modifiable parameters: the radius (r) and sigma (s) for the distribution of weights [5]. However, it is important to note that a large radius value generates a larger kernel, which requires more pixels when calculating the weighted averages. This means that the difference between the replacement values of two adjacent pixels are narrowed, and the relative visual effect is an image that looks more blurred. The value of the sigma parameter for the two-dimensional Gaussian can also be increased, resulting in a flatter peak and increasing the blurring effect [5]. Figure 5 shows how tweaking these two parameters affects the blurring effect.

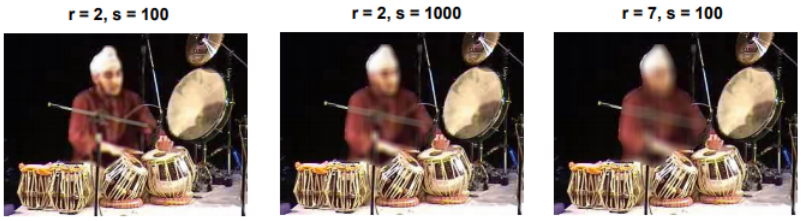

Figure 5. Different visual effects with different parameters of Gaussian kernel.

When applying the convolution filter, there are two issues that must be considered. The first is how to handle pixels on edges. Handling edges is important because if we simply apply the filter naively using pixels that are external to the object, the edges of the object become mixed with the background and no longer are clearly differentiated. This can have a negative impact on object detectors and action recognition classifiers. To solve this problem, we used a symmetry strategy to fill in the missing values. In the final implementation, for every kernel value not included in the object, a replacement value is taken from another pixel in the object. The position of the alternative pixel is chosen by symmetry on either the x or y axis, relative to the position of the source pixel.

The second problem is selecting the correct filter radius and sigma. Since we can detect objects of the same class at different scales, there is no single radius that works for all filters. A filter radius suitable for small-scale objects will not adequately anonymize large scale objects, while a filter radius for large-scale objects smooths small-scale objects too much and results in excessive artifacts when dealing with edge pixels. To solve this, we compute a bounding box for each object, and set the filter radius to 1/4 of the average length of the two sides, rounded to the nearest odd number. Given this filter width, we set sigma equal to 10 times the radius, a value that we experimentally determined.

4. Evaluation

To evaluate our system, we look at several different features. First, to show that it can be used to anonymize very different classes of objects, we consider two different datasets: UCF101, a human action recognition dataset, and the Chinese City Parking Dataset (CCPD), a dataset of license plate photos. We then consider two different use cases. The first is the case where we want to anonymize objects in the scene without negatively impacting machine learning of other features of the video, a key capability for any anonymization system. For this case, we use the UCF101 dataset and demonstrate that we can anonymize the human figures in the dataset with minimal impact on action recognition classification rates. For the second use case, we want to completely anonymize an object so that it cannot be recognized by a machine learning algorithm. In this latter case, we show that we can anonymize the license plates in the CCPD dataset to the degree that they cannot be recognized even when the machine learning algorithm is trained with blurred data. Finally, we consider the performance on scaled objects by repeating the UCF101 experiments with multiple scaled versions of the original data.

To compare the performance of our system against a standard benchmark, we ran identical experiments with a global Gaussian blur algorithm. To maintain the equivalent level of privacy with our adaptive algorithm, we chose set the filter radius and σ for the Gaussian blur to the maximum of all calculated radii and σ on the dataset being anonymized for the adaptive algorithm.

4.1. Datasets

UCF101 is an action recognition dataset composed of 13,320 realistic human action videos, collected from YouTube and classified into 101 action categories. The UCF101 dataset features a wide range of different actions and camera motions that are often present, as well as a variety of different objects, objects of different sizes, various viewpoints, illumination conditions, etc. [36].

CCPD (Chinese City Parking Dataset) is an open-source dataset for license plate detection and recognition [37]. It includes over 200,000 images of parked cars in a variety of lighting and weather conditions, with bounding boxes around their license plates. For the purpose of testing the system, 20,000 images from CCPD were chosen as our test dataset. We refer to CCPD* as the subset of 20,000 samples chosen. The remaining images were used to train DeepLabv3+ to label license plates, a class that was not included in the original model. Examples of the CCPD dataset can be seen in Figure 6.

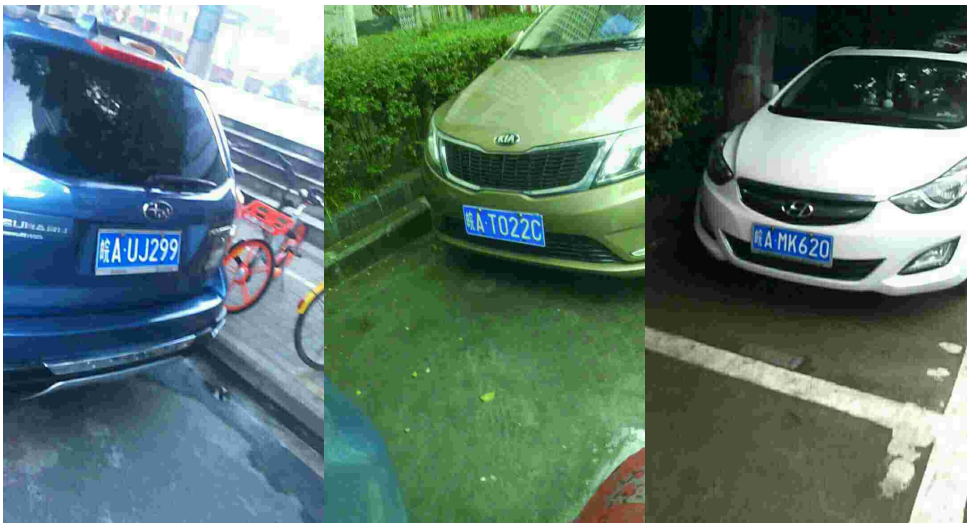

Figure 6. Sample images from the CCPD dataset. These license plates were collected in mainland China and are uniform nationally. Each is dark blue with white letters consisting of a province code character, city letter, and a unique ID number for the automobile consisting of letters and numbers.

4.2. Experiments

4.2.1. UCF101 Action Recognition

The first set of tests are designed to check whether the utility of original video data is maintained after being processed by the anonymizer. These tests are conducted using the blurred UCF101 dataset. The 'utility' of visual data refers to the amount of useful information that can be extracted from it. Concretely, preserving utility in the anonymized videos from UCF101 means the blurred videos can still be used for some task, such as action recognition.

For this test, we used a deep-learning based action recognition model called temporal segment network (TSN) to perform action recognition on the blurred UCF101 dataset. More details of its working principles can be found in [38]. In previously published experiments, TSN achieved a 93.5% action recognition accuracy on the original UCF101 dataset in the "RGB + Flow" mode (where "RGB" refers to the RGB video stream and "Flow" refers to how the input was processed, in a stream manner). In our testing, we trained the TSN

model with the blurred UCF101 training dataset and measured the action recognition accuracy on the same dataset by following established guidelines [38].

The results for this experiment can be seen in Table 1. The base accuracy of TSN on this dataset was 93.5%. After training with anonymized training data, the recognition rate fell to 88.9 %. While some accuracy was lost, the algorithm was still reasonably accurate. Anonymized data specific algorithms (e.g., [7]) could potentially perform identically to the original algorithm.

Table 1. TSN performance comparison comparing our multi-scale algorithm vs. a single-scale global Gaussian blur.

Dataset	Accuracy	
	Multi-Scale	**Single-Scale**
Original UCF101	93.5%	93.5%
Blurred UCF101	88.9%	40%
1/2 Blurred UCF101	88.9%	36.2%
1/4 Blurred UCF101	88.9%	31.1%
1/8 Blurred UCF101	88.9%	28.6%

To demonstrate the ability of our system to handle multi-scale data, we performed a second round of experiments with the UCF101 dataset. The test was performed by first downsizing the original UFC101 videos to 1/2, 1/4, and 1/8 of their original size. Each frame of these downsized videos was then placed in the center of a black image the same size as the original image. This created a set of videos the same dimensions as the original videos, but with human actors a fraction of their original size. We then performed the same anonymization and classification tasks from the previous experiment. The results, included in Table 1, show that the scale of the objects has no effect on the anonymization process. All human figures were detected and anonymized, and the recognition rate remained similar to the full sized test, with only small, gradual deterioration, likely due to the loss of information from the down-scaling process.

In comparison, the global Gaussian blur algorithm seriously deteriorated the performance of the classifier, with results ranging from 40% to 28.6%. This is primarily due to the need to maintain equivalent privacy, which results in selecting parameters that correspond to the worst (most highly blurred) case for the adaptive algorithm.

The failure cases primarily occurred in instances where numerous objects of the same class overlapped, which resulted in a degenerate filter that resulted in an a video that was too blurred to recognize the action taking place. Examples of this can be seen in Figure 7. The first of these examples is correctly labeled "Marching Band" and the second should be labeled "Military Parade". However, as can be seen from the masks, the labeled human figures overlap to such a degree that the entire image is treated as a single large instance of the human class.

Figure 7. Examples of failure cases from the the UCF101 dataset. The first example is labeled "Marching Band" and the second is labeled "Military Parade". In both cases, the clutter of same-label objects results in a degenerative blurring filter.

4.2.2. CCPD*

For the CCPD* dataset, we consider the case where the objects being anonymized are sensitive in nature, and we specifically want to prevent a machine learning algorithm

from recognizing them. Different from the previous scenario, in this case, the successful outcome of the anonymization system is to be checked with a machine-learning license plate recognition system. We implemented this test with an open-source license plates detection and recognition model [39], which is used to detect the existence of a license plate in each of the images in the CCPD* and to detect the license plate number. This code implements the algorithm discussed in [40], which has a reported recognition accuracy of 98.4%. For this experiment, DeepLabv3+ was retrained to label license plates using the remaining 180,000 license plate images from CCPD.

The results of this experiment can be seen in Table 2. We split the results into two parts: detection and recognition. The detection and recognition model [40] used was able to detect 100% of the license plates in the CCPD* dataset and recognize the license plate number 97.8% of the time. After training, DeepLabv3+ was able to detect 98.3% of the license plates in CCPD*. After anonymization, the detection rate for our model dropped to 10.7% with a recognition rate of 2.8%. The model used to recognize the license plates is a joint detection/recognition model, so blurring the text of the license plate reduces both detection and recognition of the license plate digits.

Table 2. Detection and Recognition rates on the 20,000 image CCPD* dataset. Base detection and recognition rates are the performance of the classifier from Zhang and Huang [40]. The DeepLabv3+ detection rate is the percent of the test set where a license plate was detected. Post-anonymization detection and recognition rates are the rates for the classifier from Zhang and Huang [40] on the test dataset after anonymization.

Task	Accuracy
Base Detection Rate	100%
Base Recognition Rate	97.8%
DeepLabv3+ Detection Rate	98.3%
Post-anonymization Detection Rate	10.7%
Post-anonymized Recogntion Rate	2.8%

The failure cases in the CCPD dataset primarily revolved around two cases: inability to detect the rectangular shape of the license plate and failures due to apparent changes in the color of the plate, both of which resulted in DeepLabv3+ failing to detect the plate. Examples of this can be seen in Figure 8. In the first example, the low light conditions rendered the outline of the indiscernible. In the second example, the lighting significantly modifies the color of the plate. This dataset was collected in mainland China, where license plates are uniformly dark blue. We theorize that the absence of this blue color resulted in this license plate not being detected.

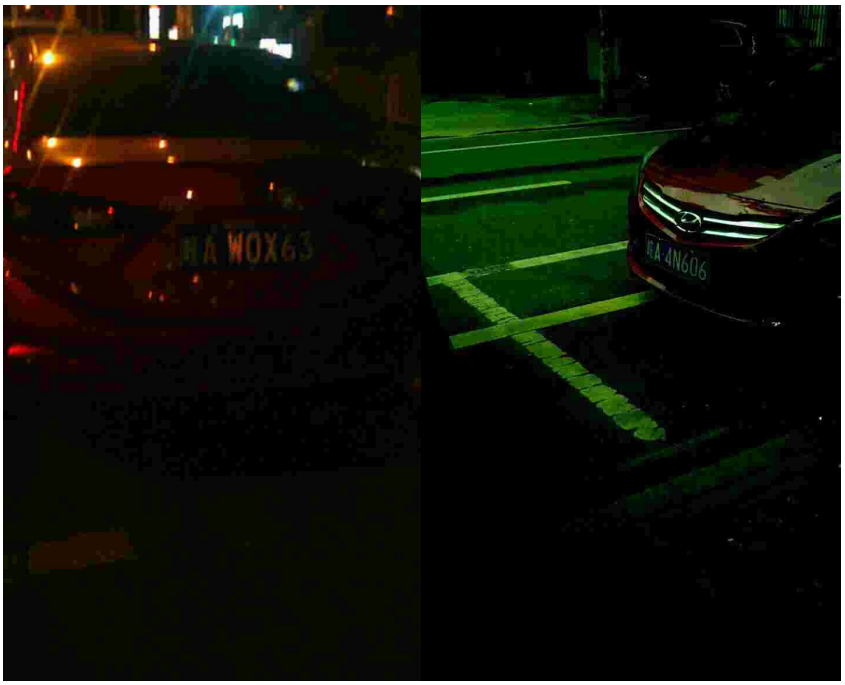

Figure 8. Examples of failure cases from the the CCPD dataset. In the first example, the low light leaves no clear outline of the plate. In the second example, the plate can be seen, but the lighting conditions render the color unrecognizable. In both cases, DeepLabv3+ fails to detect the plate.

5. Conclusions and Future Work

In this paper, we describe a flexible anonymization algorithm based on semantic segmentation with DeepLabv3+ and adaptive Gaussian blurring. This system addresses several issues with existing video anonymization systems, namely the lack of flexibility in object class recognition and the inability handle multi-scale objects. We then show that this system worked for several practical use cases, and at a variety of scales. This flexibility and adaptability means that our algorithm can be used in many practical situations where video anonymization is needed.

While this system is extremely practical, there are several areas where future work can be done. One such area would be to explore different anonymization layers, which may be more suitable for some specific applications. We also feel it would be useful to consider different use cases, and particularly cases where changes to the machine learning algorithm for the vision task could be modified in tandem with the anonymization algorithm to provide both anonymization and higher accuracy for the vision task.

Another issue that needs to be addressed is that the current algorithm estimates the size of objects simply by their bounding box. In cases where objects in the images are distorted by camera perspective, or take up significant depth in the image, the resulting filter may over blur all or part of the object. While, with knowledge of the object class, we could attempt to estimate orientation or similar information, this is further complicated by occlusion. Additionally, depth of field effects can result in initial blurring, which will again result in over-blurring of the object. As we can see from the global Gaussian results, this can seriously decrease the accuracy rate of the machine learning algorithm.

Additionally, further evaluation of this algorithm would be useful. While we show that it works well for anonymized action recognition and anonymizing license plates, there are many other privacy crucial cases that could be considered. We also believe that it would be interesting to explore different parameter and hyperparameter choices for the DeepLabv3+ model, to determine their effect on the final anonymization.

Author Contributions: All authors designed the project and drafted the manuscript, collected the data, wrote the code and performed the analysis. All participated in finalizing and approved the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data sets used in this work are cited and publicly available.

Conflicts of Interest: The authors declare no conflict of interests.

References

1. Ren, Z.; Lee, Y.J.; Ryoo, M.S. Learning to anonymize faces for privacy preserving action detection. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
2. Dufaux, F.; Ebrahimi, T. A framework for the validation of privacy protection solutions in video surveillance. In Proceedings of the 2010 IEEE International Conference on Multimedia and Expo, Singapore, 19–23 July 2010; pp. 66–71. [CrossRef]
3. Padilla-López, J.R.; Chaaraoui, A.A.; Flórez-Revuelta, F. Visual privacy protection methods: A survey. *Expert Syst. Appl.* **2015**, *42*, 4177–4195. [CrossRef]
4. Olade, I; Champion, C.; Liang, H.; Fleming, C. The Smart2 Speaker Blocker: An Open-Source Privacy Filter for Connected Home Speakers. *arXiv* **2020**, arXiv:1901.04879v3.
5. Gedraite, E.; Hadad, M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In Proceedings of the ELMAR-2011, Zadar, Croatia, 14–16 September 2011; pp. 393–396.
6. Thomas, R.E.; Banu, S.K.; Tripathy, B.K. Image anonymization using clustering with pixelization. *Int. J. Eng. Technol.* **2018**, *7*, 990–993. [CrossRef]
7. Ryoo, M.S.; Rothrock, B.; Fleming, C.; Yang, H.J. Privacy-preserving human activity recognition from extreme low resolution. In Proceedings of the 2017 AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
8. Yu, C.; Fleming, C.; Liang, H.N. Scale Invariant Privacy Preserving Video via Wavelet Decomposition. *Int. J. Des. Anal. Tools Integr. Circuits Syst.* **2018**, *7*, 56–58.
9. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. Towards open-set identity preserving face synthesis. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6713–6722.
10. Li, T.; Lin, L. Anonymousnet: Natural face de-identification with measurable privacy. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019.
11. He, W.; Zhang, X.Y.; Yin, F.; Luo, Z.; Ogier, J.M.; Liu, C.L. Realtime multi-scale scene text detection with scale-based region proposal network. *Pattern Recognit.* **2020**, *98*, 107026. [CrossRef]
12. Hao, Z.; Liu, Y.; Qin, H.; Yan, J.; Li, X.; Hu, X. Scale-aware face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6186–6195.
13. Matthews, C.E.; Kuncheva, L.I.; Yousefi, P. Classification and comparison of on-line video summarisation methods. *Mach. Vis. Appl.* **2019**, *30*, 507–518. [CrossRef]
14. Fan, J.; Luo, H.; Hacid, M.S.; Bertino, E. A novel approach for privacy-preserving video sharing. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 31 October–5 November 2005; pp. 609–616.
15. Yousefi, P.; Kuncheva, L.I. Selective keyframe summarisation for egocentric videos based on semantic concept search. In Proceedings of the 2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS), Sophia Antipolis, France, 12–14 December 2018; pp. 19–24.
16. Wu, Z.; Wang, Z.; Wang, Z.; Jin, H. Towards privacy-preserving visual recognition via adversarial training: A pilot study. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 606–624.
17. Fleming, C.; Peterson, P.; Kline, E.; Reiher, P. Data Tethers: Preventing information leakage by enforcing environmental data access policies. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 835–840.
18. 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 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
19. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
20. Thoma, M. A Survey of Semantic Segmentation. *arXiv* **2016**, arXiv:1602.06541.
21. Learned-Miller, E.; Huang, G.B.; Roychowdhury, A.; Li, H.; Gang, H. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2016.
22. Hammer, B.; Biehl, M.; Bunte, K.; Mokbel, B. A general framework for dimensionality reduction for large data sets. In Proceedings of the 2011 International Conference on Advances in Self-Organizing Maps, Espoo, Finland, 13–15 June 2011.

23. Krähenbühl, P.; Koltun, V. Efficient inference in fully connected CRFs with gaussian edge potentials. In *Advances in Neural Information Processing Systems 24*; Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2011; pp. 109–117.
24. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
25. Hao, S.; Zhou, Y.; Guo, Y. A Brief Survey on Semantic Segmentation with Deep Learning. *Neurocomputing* **2020**, *406*, 302–321. [CrossRef]
26. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1314–1324.
27. Ding, H.; Jiang, X.; Shuai, B.; Liu, A.Q.; Wang, G. Semantic correlation promoted shape-variant context for segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8885–8894.
28. Zhang, H.; Zhang, H.; Wang, C.; Xie, J. Co-occurrent features in semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 548–557.
29. Yang, T.; Wu, Y.; Zhao, J.; Guan, L. Semantic segmentation via highly fused convolutional network with multiple soft cost functions. *Cogn. Syst. Res.* **2019**, *53*, 20–30. [CrossRef]
30. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2169–2178. [CrossRef]
31. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
32. Erdélyi, Á.; Winkler, T.; Rinner, B. Privacy protection vs. utility in visual data. *Multimed. Tools Appl.* **2018**, *77*, 2285–2312. [CrossRef]
33. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
34. Wei, L.; Rabinovich, A.; Berg, A.C. ParseNet: Looking Wider to See Better. *arXiv* **2015**, arXiv:1506.04579v2.
35. Dr.Sleep. DeepLab-ResNet-TensorFlow. Available online: https://github.com/DrSleep/tensorflow-deeplab-resnet (accessed on 11 May 2019).
36. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv* **2012**, arXiv:1212.0402.
37. Xu, Z.; Yang, W.; Meng, A.; Lu, N.; Huang, H. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 255–271.
38. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Val Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016.
39. ShadowN1ght. License Plate Detection and Recognition Model (Implemented on Tensorflow). Available online: https://blog.csdn.net/shadown1ght/article/details/78571187 (accessed on 8 May 2019).
40. Zhang, Y.; Huang, C. A robust chinese license plate detection and recognition systemin natural scenes. In Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), Wuxi, China, 19–21 July 2019; pp. 137–142. [CrossRef]