Proceedings of the Ninth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2021)

Pterodactyl: Two-Step Redaction of Images for Robust Face Deidentification

Abdullah B. Alshaibani and Alexander J. Quinn
Purdue University, 610 Purdue Mall, West Lafayette, IN 47907
aalshai@purdue.edu, aq@purdue.edu

Abstract

Redacting faces in images is trivial when the number of faces is small, and the annotator is trusted. For large batches, automated face detection has been the only currently viable solution, yet even the best ML-based solutions have error rates that would be unacceptable for sensitive applications. Crowd-based face detection/redaction systems exist, yet the process and the cost make them infeasible. We present Pterodactyl, a system for detecting (and redacting) faces at scale. It uses the AdaptiveFocus filter, which splits the image into smaller regions and uses machine learning to select a median filter for each region to hide the facial identities in the image while simultaneously allowing those faces to be detectable by crowd workers. The filter uses a convolutional neural network trained on images associated with the median filter level that allows detection and prevents identification. This filter allows Pterodactyl to achieve human-level detection with just 14% as much crowd labor as another recent crowd-based face detection/redaction system (IntoFocus). Our evaluation found that the redaction accuracy was higher than a commercial machine-based application, and on par with IntoFocus,, while requiring 86% less crowd work (number of comparable tasks).

Introduction

Machine learning face detection has made significant gains in recent years, yet for some applications, such as redaction of sensitive images, failure to detect even a single face may incur significant human costs. Also, some of the most accurate publicly available systems have inherent privacy risks, as expressed in their policies (Amazon 2021).

For images that require the highest possible privacy, the following approaches could solve the problem:

1. Build a face detector in-house (at the cost of lower than perfect accuracy).
2. Hire a team to manually redact the images (at the expense of possible disclosure).
3. Adapt a crowd-based privacy-preserving redaction system for in-house use (at the expense of a potentially large team and high cost).

This paper presents *Pterodactyl*, a crowd-based system for redaction of faces in images. It uses the AdaptiveFocus filter, which obfuscates an image to allow face detection while preventing untrusted crowd workers from identifying any face in the image. Together, the system permits redaction of faces with human-level accuracy, without disclosing identifiable faces to crowd workers.

The contributions of this paper are as follows.

1. We present Pterodactyl, a system for enhancing the quality of image redaction annotations that applies a set of rules and restrictions to increase early face detection and prevent facial identity disclosure.
2. We present the AdaptiveFocus filter, an image filter that combines machine learning and the median filter to allow privacy-preserving face redaction in images at a lower cost.
3. We evaluate the system and filter by comparing crowd-based alternative and automated face detection systems.

Related Work

Pterodactyl builds on our prior work with the IntoFocus system 2020. As part of tuning parameters for that system, we performed a face perception study on Amazon Mechanical Turk. We then presented crowd workers with median filtered images and tasked them with detecting and identifying the faces in the photos. Using the data collected from that study, we extracted seven different median filter levels. The filter levels allowed crowd workers to detect faces but did not allow them to identify the faces.

Pterodactyl achieves a very similar end result, but substantially reduces cost by eliminating the need for iterations. The AdaptiveFocus filter uses the data generated by that face perception study to train the Convolutional Neural Network (CNN).

Another important work in image redaction using the crowd is the work by Kaur et al. 2017. They introduced the field and showed that it is possible to build a system that allows crowd workers to perform redaction tasks without having access to all the information. With this work, they not only opened the doors but also provided the path. Furthermore, their plan of equally segmented regions in a single image gave us an idea of building the AdaptiveFocus filter.

Face detection is an active problem in computer vision (Yadav and Priyanka 2021). In their work, they categorize face detectors into three categories. The first is called CNN Cascade Face detectors; they start by creating image pyramids and use a sliding window as input to the CNN (Li et al. 2015; Qin et al. 2016). The second is a Region-based Face detectors; they propose a region for input to the CNN (Ren et al. 2017; Girshick et al. 2014). Finally, Proposal Free Network, which does not require region proposals (Redmon and Farhadi 2017; Liu et al. 2016). These approaches are all possible in providing the regions to apply the AdaptiveFocus filter. But relying on such methods would limit the AdaptiveFocus filter to the limitations of the proposed methods.

There are several benchmark datasets where researchers are trying to achieve the highest possible success rate. The MogFace face detector (Liu et al. 2021) achieved the state-of-the-art performance on Wider Face (Yang et al. 2016), FDDB (Jain and Learned-Miller 2010), Pascal Face (Agarwal, Awan, and Roth 2004), and AFW (Zhu and Ramanan 2012). They achieved near-perfect results on FDDB, Pascal Face, and AFW. In the Wider face dataset, they reached 97.7%, 96.9%, and 93.8% in the easy, medium, and hard image sets, respectively. These results show that machine face detection has not yet achieved perfection. State-of-the-art methods also require large amounts of processing power and training time to build.

Instead, the images used to train the AdaptiveFocus filter will apply the Sliding Window approach (Glumov, Kolomiyetz, and Sergeyev 1995), similar to Cascade methods, but for a classification problem (Gouk and Blake 2014) instead of a detection problem (Seo and Ko 2004). The filter would need to assign a filter level appropriate for all the regions, not just those containing a face.

The Pterodactyl System

The Pterodactyl System is an improvement over the IntoFocus system (Alshaibani et al. 2020). The IntoFocus system focused on achieving the highest possible detection rate with the lowest possible identification rate. The downside of the approach is that it ignores the time and team size required (at least 21 different people to redact a single image). Because of the required team size, it becomes hard to manage on crowdsourcing platforms. 57.4% (283 of the assignments) of the participants failed their attention check image. Based on the cost of the task, $\$0.75$, and the associated fees, the extra assignments would be $\$297.15$. The Pterodactyl System focuses on reducing the cost while maintaining a non-significantly different performance in detection and identification.

The Pterodactyl system uses a combination of rules and requirements to ensure the quality of the results. The first requirement is to use at least three crowd workers. That ensures that people of differing detection abilities perform the detection task. The second requirement is assigning a qualification that blocks crowd workers from working on the same images again. This makes sure that crowd workers cannot see the same image more than once. The third requirement is that each image set is seeded with an image that contains at least two faces. This detects if a crowd worker is not adding ellipses on all the visible faces in the images. It also informs us if the instructions need to be improved.

In addition to the above requirements, the following rules were also added when analyzing the attention check images. 1) All the faces need to be redacted. 2) The number of ellipses added is equal to the number of faces in the image. 3) A single ellipse does not intersect with three or more faces. 4) None of the ellipses goes beyond a 100% increase in the width and height of the face it is covering. These added protections are only applied to the attention check images where the ground truth information already exists. These rules were added to make sure that workers are following the task requirements.

AdaptiveFocus Median Filter

The task was to create an image filter that would allow people to detect but not identify faces in images using the data gathered in the perception study (Alshaibani et al. 2020). The AdaptiveFocus filter 1 will take an image as input, assign appropriate median filters to obfuscate the faces and return a thoroughly obfuscated image that allows face detection and prevents face identification. The AdaptiveFocus filter needs to solve the following problems: 1) Image restrictions and requirements 2) Find the locations of faces in the image 3) Which filter level to use.

Image Requirements

The face perception study performed by Alshaibani et al. 2020 requires that the image be 640×640 pixels, and their collected data is only valid under that image size. Because the AdaptiveFocus filter is based on the data collected in that study, the AdaptiveFocus filter undergoes the exact requirements. In theory, the AdaptiveFocus filter can be applied to larger image sizes, but a size threshold exists where the AdaptiveFocus filter no longer works for an image. That threshold was not explored in this paper.

Finding The Face Locations

The general approach in face detection/redaction systems is to extract regions with a high probability of containing faces and redacting those regions. However, the purpose of the AdaptiveFocus filter is to obfuscate the image so people can perform the detection task. Therefore, the filter does not need to detect the locations of the faces; it needs to assign the correct filter levels to all the regions of the image to allow detection and prevent identification. To solve that problem, the image is segmented into square tiles, and each tile is assigned a filter level based on its content and the content of the surrounding regions. Based on the data collected in the perception study (Alshaibani et al. 2020) the smallest detectable face has a size of 209 $pixels^2$ and the image has a size requirement of 640×640. Therefore, we selected the size for tiles to be $16 \times 16 = 256$ pixels, and a small face will fit in one tile. To have a fixed tile space, the size of all the images is increased (in width or height), so the size will be 640×640. Thus, each image will contain 40×40 tiles, and each of the tiles will be evaluated and assigned a filter level.

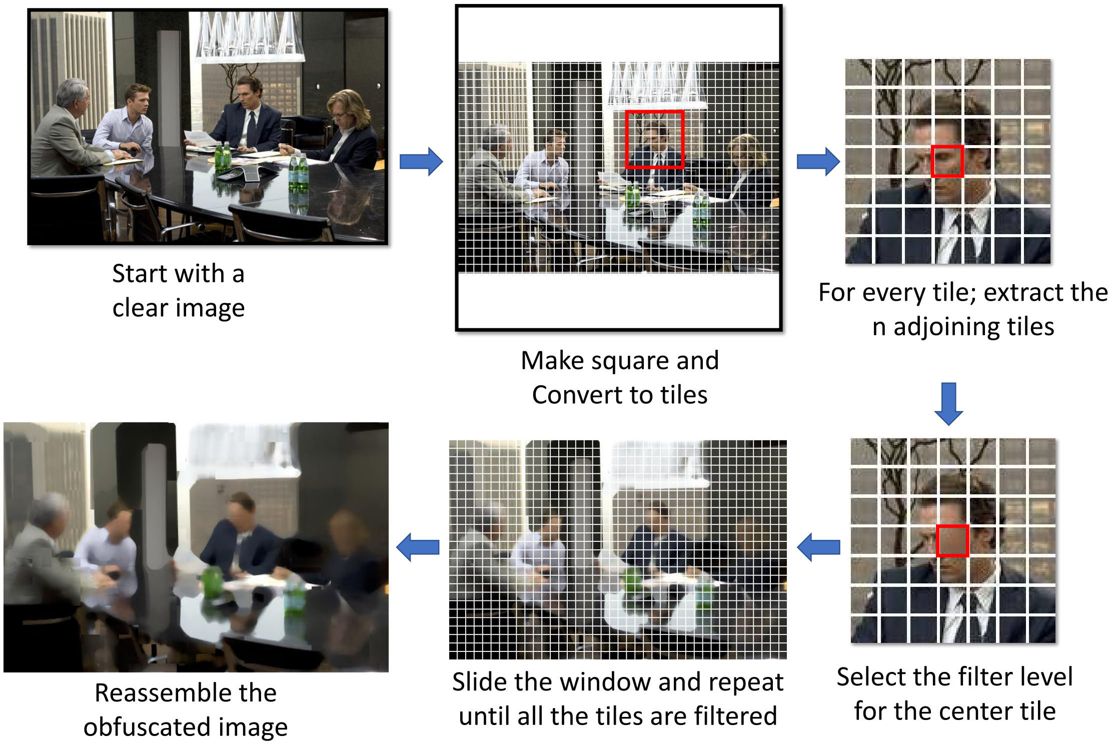

Start with a clear image
Make square and Convert to tiles
For every tile; extract the n adjoining tiles
Select the filter level for the center tile
Slide the window and repeat until all the tiles are filtered
Reassemble the obfuscated image

Figure 1: This figure shows the process an image takes in the AdaptiveFocus filter. Starting with the clear image, it is resized, made square, and each tile is separated. Next, a window of size $n \times n$ centered at each tile slides across the image and assigns a filter level to each tile. Finally, the tiles are reassembled, and an obfuscated image is created.

Selecting The Filter Levels

The filter level selection is based on the results of the face perception study of median filtered faces (Alshaibani et al. 2020). The study presented people with median filtered images containing people and tasked them with detecting and identifying the faces. The AdaptiveFocus filter utilizes the detection point of inflection where 100% of the study participants could detect the faces as the appropriate filter level for a specific face. For example, sequentially, a face has been tested three times with median filters 25, 29, and 31 (ksize), with detection of 100%, 100%, and 92%. The filter level selected for that face will be 29. Creating a filter map for each image (figure 2) using the inflection points of all the faces in the images. The filter map shows the filter level to use for each face to be detectable and not identifiable. The color range for the filter map goes from black (no filter required) to white (highest filter required).

We train a Convolutional Neural Network (CNN) on the tiles containing faces as input. It outputs the filter level required for each tile. The neural network needs to answer the question, “If this tile contains a face, what filter level needs to be used to allow detection and prevent identification?”. If the tile has a face, the proposed filter level will allow that face to be detectable but not identifiable. If the tile does not contain a face, it will still suggest a filter level that would allow a face to be detectable and not identifiable. We chose this approach because current face detection algorithms (Liu et al. 2021) are not yet perfect. With this approach, the filter will obfuscate the entire image, allow face detection on all the regions containing a face, and obscure the areas that do not have faces.

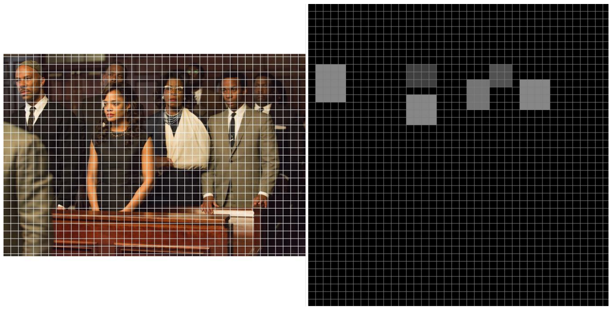


Figure 2: Shows an example of a filter map. The left image shows the image to be redacted. The white lines separate the tiles. The image on the right is the ground truth filter map for that image. The black region means that there are no faces in that region. The brighter regions specify the filter levels required for that face to be detectable but not identifiable. The darker the region, the lower the filter required; the brighter the region, the higher the filter required.

For the CNN to correctly select the correct filter level, it will need to analyze the tile to be classified and n adjacent tiles. We evaluated different values of n. We split the inflection point data into a training and validation set. We trained the neural networks starting from $n = 3$ to find the value

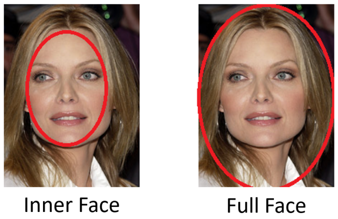


Figure 3: This figure shows the difference between the inner face and full-face images. In the inner face, only the inside of the face is used. In the full face, the full face is covered, including hair, beard, and hats.

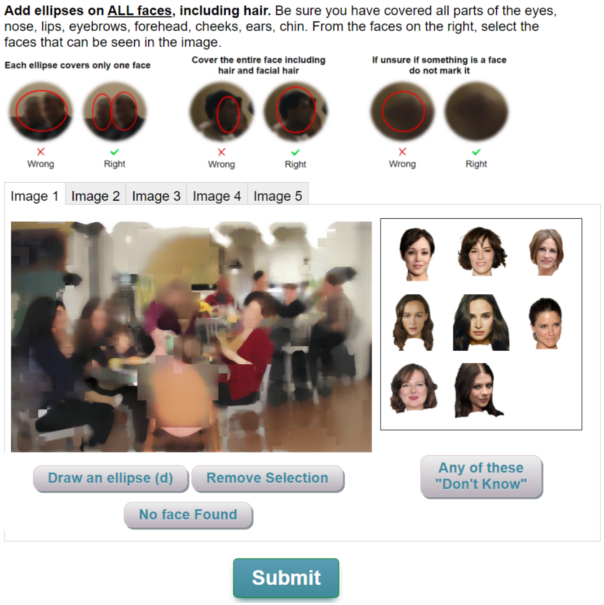
of n where the classification accuracy no longer increases. Based on the results of the CNN training, it was found that before $n = 7$, there were significant accuracy increases, and after $n = 7$, the increase was less than 0.5%. Thus, the chosen value for n was 7. Since each tile was 16×16 pixels, the classification model will input square images with a width and height of $16 \times 7 = 112$ pixels.

The CNN model used for the classification task was based on the VGG16 classifier (Simonyan and Zisserman 2015). It was compiled using the categorical cross-entropy loss function and the stochastic gradient descent(Qian 1999) optimizer. The training occurred over 100 epochs. The dataset used was the inflection point dataset, and it was split into 75% training and 25% validation, with a total of 9 different classes. The classes were faces detectable at 7, 13, 17, 23, 29, 35, 41, 53, and 85. The images were augmented using horizontal and vertical flip, rotations (Shorten and Khoshgoftaar 2019), Gaussian blur, noise injection(Moreno-Barea et al. 2018), and applying multiple augmentations together. Photometric augmentations (Taylor and Nitschke 2017) were considered, but they were avoided because lighting and color might affect the appropriate filter levels for a tile. The dataset started with 60 images containing 185 faces. With the augmentation and splitting of the tiles, we generated 219876 images across the nine classes. Training the CNN took 32.4 days (777.8 hours). There was no class for no faces detected because **the face detection and redaction are performed by crowd workers, not the neural network**.

Moreover, there were three different variants of the AdaptiveFocus filter. The first variant contained images of the full face, including hair, beards, and hats 3. The second variant included only the inner face, excluding any hair, beards, or hats 3. This final variant had two separate CNNs, one for the higher filter levels (29, 35, 41, and 85) and a second for the lower filter levels (7, 13, 17, and 23). This variant applies the AdaptiveFocus filter into the IntoFocus process (Alshaibani et al. 2020) to increase the probability of face detection. The variants were created by using transfer learning on the full face variant for 30 epochs.

Experiment Setup

The experiment (Figure 4) to evaluate the Pterodactyl system and the AdaptiveFocus filter was designed to follow the experiment performed to evaluate the IntoFocus system and method (Alshaibani et al. 2020). The experiment aims to assess the Pterodactyl system combined with the AdaptiveFocus filter and compare the results with the 7-stage IntoFocus method and system. We hypothesize that the Pterodactyl system will maintain a non-significantly different identification and detection with the highest method.


Add ellipses on ALL faces, including hair. Be sure you have covered all parts of the eyes, nose, lips, eyebrows, forehead, cheeks, ears, chin. From the faces on the right, select the faces that can be seen in the image.

Each ellipse covers only one face
Cover the entire face including hair and facial hair
If unsure if something is a face do not mark it
✗ Wrong
✓ Right
✗ Wrong
✓ Right
✗ Wrong
✓ Right
Image 1
Image 2
Image 3
Image 4
Image 5
Draw an ellipse (d)
Remove Selection
Any of these "Don't Know"
No face Found
Submit

Figure 4: This figure shows the interface used to evaluate the Pterodactyl system and the AdaptiveFocus filter. Participants were tasked with detecting all the faces in five filtered images and identifying a face in each image.

AdaptiveFocus Filters

In the experiment, we evaluate three different models of the AdaptiveFocus filters. The first is trained on detecting face regions, including hair, hats, and beards. The second model is trained on detecting the inner face regions, excluding hair. The third is trained to work on two stages, where the faces detected on the first stage are redacted before progressing to the second stage. Each of the filters is presented to three different crowd workers. Any crowd worker who worked on a specific image set could not work on that set again for all the available conditions.

Control Conditions

The controls in this experiment are automated face detection solutions and the IntoFocus method system (Alshaibani et al. 2020). The system proposes using a 7-stage redaction process, starting from the highest filter level, and iteratively reducing the filter level, while asking the crowd to redact faces in the images. Before the start of each stage, all the faces that crowd workers previously detected are redacted. Then, the process is repeated seven times until the entire image is redacted.

Face Identification

In each image, we present participants with a set of eight faces, one of which is a subject located in the obfuscated image. The goal of the methods being evaluated is to hide the identities of the faces from the participants. Out of the eight faces, only one face is located in each of the images. The participants are allowed to select multiple faces if they can narrow the list of possible faces. If the participants could not narrow down the list of faces, they were asked to choose the "don't know" button.

The eight faces have similar facial features (hair color, hair length, skin tone) to provide k-anonymity(Sweeney 2002) for those faces. This ensures that if a participant identified a face, it was because the facial features were visible.

Face Detection

To measure face detection, the participants need to add ellipses that encompass the inner regions of each face. Covering part of the face did not count towards a detected face. This ensured that the participant could detect the face and not just add ellipses to the images.

Participants

We divided the experimentation into two categories. The first category employed crowd workers on Amazon Mechanical Turk (AMT) (Amazon 2017) to perform the redaction task. The second category engaged in-person participants to complete the redaction task.

In the first category, crowd workers must have a 90% task success rate to participate. Each Human Intelligence Task (HIT) rewarded $0.75, and the requirement was to redact only five images. There were 248 unique crowd workers. The cost of the experiment was $621, with an average of $9.32 per hour. The task took an average time of 4.8 minutes to complete.

In the second category, the task for each participant was to redact 50 images. The task took the participants an average of 21 minutes to complete. There were only three participants. One participant performed the tasks using the AdaptiveFocus (full face) method, and two participants performed the AdaptiveFocus (two-step) method. The participants used the same interface and performed the same tasks as the first category.

Attention Check

The Attention Check proposed in (Alshaibani et al. 2020) is a combination of rules that reduces the risks of malicious participants. An image–where the detection task was purposefully made trivial–was seeded in each HIT (Marcus et al. 2012; Huang and Fu 2013) to ensure that the task is completed correctly. The Pterodactyl system uses the rules proposed in the Pterodactyl system section .

Dataset

In the experiment, we used the IMDB image dataset (Rothe, Timofte, and Van Gool 2015). The experiment contained ten different HITs; each hit contained five images, four of which were being evaluated, and the other was to test for attention check. Thus, making a total of 40 images for evaluation and ten images for attention check.

Experiment Task

Participants are required to perform two tasks. First, they needed to add ellipses on all the visible faces. Second, they needed to attempt to try to identify one of the faces in the image. When a participant performs the task on a specific image set, they can not complete the task on that image set again. Even if the filter method is changed, this is achieved by assigning a qualification on AMT that blocks them from accessing HITs containing the same images.

Evaluation

All the crowd-based methods will be tested using both the Pterodactyl system and the IntoFocus system. The AdaptiveFocus filter was designed to perform with annotators and crowd workers. To evaluate, crowd workers were hired on Amazon Mechanical Turk (AMT), and annotators were hired for an in-person evaluation. The results were also compared with the IntoFocus method (Alshaibani et al. 2020), Microsoft Azure's face detector (Microsoft 2021), Face++ (Face++ 2021), and the MTCNN pre-trained face detector (Zhang et al. 2016).

Results

The evaluation is separated into the following sections: detection, identification, time, and cost (table1). The first part compares all the methods, with the AdaptiveFocus methods using the Pterodactyl system and the IntoFocus method using the IntoFocus system. The second part compares the AdaptiveFocus method and the IntoFocus method (Alshaibani et al. 2020) using both the IntoFocus system and the Pterodactyl system (table1).

Identification

Identification in the evaluation is the number of images where the faces were identified during the experiment. All the AdaptiveFocus-based methods had lower identification rates than the IntoFocus method.

Detection

Detection in the evaluation is adding an ellipse or a bounding box covering the entire face. The IntoFocus method had the highest detection rate of all the methods, missing only one face. The annotator-based methods only used a single annotator, which negatively affected the results by not having multiple views on a single image. The two annotators in the AdaptiveFocus (two-step) method did not detect faces detected by the other AdaptiveFocus methods.

Cost

The cost in the evaluation of how much it costs to perform the detection/redaction of a single image. The AdaptiveFocus methods had the lowest cost among the crowd-based methods. It was seven times less than the IntoFocus method.

Time

The time is the evaluation of the time needed to detect/redact all the faces. The Crowd-based methods required crowd

	Automated			Crowd-based			Annotators	
Method	Face++	Azure	MTCNN	IntoFocus	AdaptiveFocus (full faces)	AdaptiveFocus (inner faces)	AdaptiveFocus (full faces)	AdaptiveFocus (two-step)
Detected faces	66.2%	32.3%	93.9%	**99.5%**	96%	92.5%	93.9%	99%
Identified Faces	-	-	-	42.5%	10%	**7.5%**	-	-
Cost/image (USD)	-	-	-	$\geq$ 3.9375	$\geq$ **0.5625**	$\geq$ **0.5625**	-	-
Time	-	-	-	$\geq$ 35 minutes	$\geq$ **5 minutes**	$\geq$ **5 minutes**	-	-

Table 1: The table above shows the results of the experiment. The automated methods are pure machine-based methods. Crowd-based methods are methods that were evaluated with crowd workers On AMT. The annotator-based methods are methods that were evaluated with in-person annotators. Identified faces are the faces that the crowd workers were able to identify. The detected faces are the number of faces that were detected by each method. The cost is how much it would cost to redact a single image. The time is the time required to redact a single image fully. The time required for the crowd-based methods is to redact five images because of the attention check requirement. The time and cost per image for the crowd-based methods are greater than or equal because attention check requires at least three participants to perform the detection task correctly for the image to finish.

	Method	Identified Faces	Detected Faces
IntoFocus System	IntoFocus Method	42.5%	**99.5%**
	AdaptiveFocus (full faces)	17.5%	82.8%
	AdaptiveFocus (inner faces)	20%	78.8%
Pterodactyl System	IntoFocus Method	**17.5%**	**99.5%**
	AdaptiveFocus (full faces)	**10%**	**96%**
	AdaptiveFocus (inner faces)	**7.5%**	**92.4%**

Table 2: The table above shows the results of the comparison between the Pterodactyl system and the IntoFocus system. Identified faces are the faces that the crowd workers were able to identify. The detected faces are the number of faces that were detected by each method.

workers to perform the task at a certain level of accuracy for their input to pass the attention check requirements. That is why the time needed to redact a single image is greater than or equal because that is the minimum time required based on one minute per image for a crowd worker to perform the task. The AdaptiveFocus-based methods were seven times faster than the IntoFocus method.

System Analysis

This section compares the IntoFocus system with the Pterodactyl system. The two systems are compared in face detection and face identification using the IntoFocus, AdaptiveFocus (full face), and AdaptiveFocus (inner face) methods.

Detection The detection results of the IntoFocus system and the Pterodactyl system (table 2) show that, even when the system is changed, the IntoFocus detection results do not change. In comparison, the AdaptiveFocus methods had a significant increase in detection for both methods. These results show the importance of the Pterodactyl system for the AdaptiveFocus system, but With the IntoFocus process, it does not affect the detection results.

Identification The identification results (table 2) show that the Pterodactyl system significantly improves the results of all the methods. With the IntoFocus method gaining the most significant decrease in terms of identification. However, even when using the Pterodactyl system, the IntoFocus method still has a higher disclosure rate than the AdaptiveFocus-based methods.

Discussion

The system analysis results show that with the addition of the Pterodactyl system rules, the AdaptiveFocus-based methods significantly increase detection and a significant decrease in identification. While the IntoFocus method did not improve detection, the identification results decreased by 25%. In the IntoFocus method, the images went through several iterations of redaction. This process allowed the faces that were not adequately redacted in the first stage they were detectable to be redacted in the following stages at a lower filter level. However, this allows some of the faces to be identifiable to some of the crowd workers. This can be seen when comparing the identification and detection results for the IntoFocus method. It shows that crowd workers identified 25% more faces that would not have been identifiable if those faces were redacted at an earlier stage. Nevertheless, the number of faces detected remains the same. This shows that some faces can be detected at a later stage where they are identifiable, and the image is clearer because of the iterative process. On the other hand, the AdaptiveFocus methods are not iterative processes (except for the two-step method), so if a face is not detected in the initial run, it will not be detected at a later stage. That is why it can be seen that the ratio of faces detected increases significantly with the added rules. This shows that with the AdaptiveFocus methods, a single crowd worker who does not follow the task requirements can cause faces not to be redacted. However, with the addition of the rules proposed in the Pterodactyl system, the results significantly improve identification and detection. The results show that the IntoFocus method has the highest detection overall, but the number of faces identified and the time and cost per image were very high. On the other hand, the crowd-based AdaptiveFocus methods have significantly lower time, cost, and identification rates than IntoFocus, and a higher detection rate than the automated methods. Finally, the annotator-based AdaptiveFocus methods had detections close to the IntoFocus method.

There was no increase in detection between the IntoFocus (IntoFocus system) and the IntoFocus (Pterodactyl system). But there was a significant increase in face identity preservation. This is because the strict rules have increased the number of faces redacted at earlier stages, thus decreasing the chances of disclosure.

In the AdaptiveFocus (full faces), crowd workers had more face detections than the single annotator. Another observation was that the faces the crowd workers did not detect were also not detected by the annotator. These results show that since a total of three crowd workers had their detections aggregated, they were able to achieve a higher detection rate than a single annotator. This points to the observation by Alshaibani et al. 2020 that people have differing abilities when detecting faces in filtered images.

AdaptiveFocus (full face) vs. AdaptiveFocus (inner face), the difference between the methods is that in the training set for the CNN, one used the full face, including hair, beards, and hats. While the other only used the inner face without the hair, beards, and hats. Because the inner face method's training set only includes faces and the lack of other features in the images, the filter could not accurately filter the surrounding regions, making the task harder for the crowd workers to accurately detect the faces.

In creating the AdaptiveFocus filter, the images were split into 40×40 tiles. This approach was the most viable because the data set was small (185 faces), and one of the categories only had four faces. This was because of our condition of the point of inflection. If a face did not reach that point, it was not used in the training/validation. In creating the filter, there were multiple possible approaches. The most viable was to use a face detector to redact the detectable faces and then submit the image to the crowd workers. This approach does work and is not a flawed approach, but in the results of the AdaptiveFocus (full faces) method, none of the faces missed by crowd workers were detected by the MTCNN face detector (Zhang et al. 2016).

Another approach was to incorporate the MTCNN face detector (Zhang et al. 2016) into the filter and select the regions for applying the AdaptiveFocus filter. With this approach, the remaining parts of the images would have a static predefined filter. However, when this approach was attempted, a single predefined filter could not obfuscate all the remaining faces and prevent identification.

Conclusion

In this paper, we presented the Pterodactyl system and the AdaptiveFocus image filter. We showed that the Pterodactyl system increases the performance of the IntoFocus face redaction system (Alshaibani et al. 2020). We demonstrated that the AdaptiveFocus filter yields lower face identification rates when compared to the IntoFocus method and high face detection rates. Based on the results, the AdaptiveFocus filter aims to determine the right blur level for each tile to maintain the guarantees of privacy and efficacy.

References

Agarwal, S.; Awan, A.; and Roth, D. 2004. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(11): 1475–1490. ISSN 1939-3539. doi: 10.1109/TPAMI.2004.108. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Alshaibani, A.; Carrell, S.; Tseng, L.-H.; Shin, J.; and Quinn, A. 2020. Privacy-Preserving Face Redaction Using Crowdsourcing. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 8(1): 13–22. URL https://ojs.aaai.org/index.php/HCOMP/article/view/7459. Number: 1.

Amazon. 2017. Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome. Accessed: 2021-06-12.

Amazon. 2021. Data protection in Amazon Rekognition - Amazon Rekognition. https://docs.aws.amazon.com/rekognition/latest/dg/data-protection.html. Accessed: 2021-03-15.

Face++. 2021. Face Detection - Face++ Cognitive Services. https://www.faceplusplus.com/face-detection/. Accessed: 2021-06-23.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. doi: 10.1109/CVPR.2014.81. ISSN: 1063-6919.

Glumov, N. I.; Kolomiyetz, E. I.; and Sergeyev, V. V. 1995. Detection of objects on the image using a sliding window mode. *Optics & Laser Technology* 27(4): 241–249. ISSN 0030-3992. doi:https://doi.org/10.1016/0030-3992(95)93752-D. URL https://www.sciencedirect.com/science/article/pii/003039929593752D.

Gouk, H. G. R.; and Blake, A. M. 2014. Fast Sliding Window Classification with Convolutional Neural Networks. In *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*, IVCNZ '14, 114–118. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-3184-5. doi:10.1145/2683405.2683429. URL http://doi.org/10.1145/2683405.2683429.

Huang, S.-W.; and Fu, W.-T. 2013. Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 conference on Computer supported cooperative work*, CSCW '13, 639–648. San Antonio, Texas, USA: Association for Computing Machinery. ISBN 9781450313315. doi:10.1145/2441776.2441847. URL https://doi.org/10.1145/2441776.2441847.

Jain, V.; and Learned-Miller, E. 2010. Fddb: A benchmark for face detection in unconstrained settings. University of Massachusetts. *Amherst, Tech. Rep. UM-CS-2010-009* 2(7): 8.

Kaur, H.; Gordon, M.; Yang, Y.; Bigham, J. P.; Teevan, J.; Kamar, E.; and Lasecki, W. S. 2017. CrowdMask: Using Crowds to Preserve Privacy in Crowd-Powered Systems via Progressive Filtering. In *Fifth AAAI Conference on Human*

Computation and Crowdsourcing. URL https://www.aaai.org/ocs/index.php/HCOMP/HCOMP17/paper/view/15938.

Li, H.; Lin, Z.; Shen, X.; Brandt, J.; and Hua, G. 2015. A convolutional neural network cascade for face detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5325–5334. doi:10.1109/CVPR.2015.7299170.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, 21–37. Cham: Springer International Publishing. ISBN 9783319464480. doi:10.1007/978-3-319-46448-0_2.

Liu, Y.; Wang, F.; Sun, B.; and Li, H. 2021. MogFace: Rethinking Scale Augmentation on the Face Detector. *arXiv:2103.11139 [cs]* URL http://arxiv.org/abs/2103.11139. ArXiv: 2103.11139.

Marcus, A.; Karger, D.; Madden, S.; Miller, R.; and Oh, S. 2012. Counting with the crowd. *Proceedings of the VLDB Endowment* 6(2): 109–120. ISSN 2150-8097. doi:10.14778/2535568.2448944. URL https://doi.org/10.14778/2535568.2448944.

Microsoft. 2021. Facial Recognition | Microsoft Azure. https://azure.microsoft.com/en-us/services/cognitive-services/face/. Accessed: 2021-06-12.

Moreno-Barea, F. J.; Strazzera, F.; Jerez, J. M.; Urda, D.; and Franco, L. 2018. Forward Noise Adjustment Scheme for Data Augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 728–734. doi:10.1109/SSCI.2018.8628917.

Qian, N. 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks: The Official Journal of the International Neural Network Society* 12(1): 145–151. ISSN 1879-2782. doi:10.1016/s0893-6080(98)00116-6.

Qin, H.; Yan, J.; Li, X.; and Hu, X. 2016. Joint Training of Cascaded CNN for Face Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3456–3465. doi:10.1109/CVPR.2016.376. ISSN: 1063-6919.

Redmon, J.; and Farhadi, A. 2017. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525. doi:10.1109/CVPR.2017.690. ISSN: 1063-6919.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6): 1137–1149. ISSN 1939-3539. doi:10.1109/TPAMI.2016.2577031.

Rothe, R.; Timofte, R.; and Van Gool, L. 2015. DEX: Deep EXpectation of Apparent Age from a Single Image. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 252–257. doi:10.1109/ICCVW.2015.41.

Seo, J.; and Ko, H. 2004. Face detection using support vector domain description in color images. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, V–729. doi:10.1109/ICASSP.2004.1327214. ISSN: 1520-6149.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6(1): 60. ISSN 2196-1115. doi:10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* URL http://arxiv.org/abs/1409.1556. ArXiv: 1409.1556.

Sweeney, L. 2002. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05): 557–570. ISSN 0218-4885. doi:10.1142/S0218488502001648. URL https://www.worldscientific.com/doi/abs/10.1142/S0218488502001648.

Taylor, L.; and Nitschke, G. 2017. Improving Deep Learning using Generic Data Augmentation. *arXiv:1708.06020 [cs, stat]* URL http://arxiv.org/abs/1708.06020. ArXiv: 1708.06020.

Yadav, R.; and Priyanka. 2021. An Overview of Recent Developments in Convolutional Neural Network (CNN) Based Face Detector. In Singh, V.; Asari, V. K.; Kumar, S.; and Patel, R. B., eds., *Computational Methods and Data Engineering*, Advances in Intelligent Systems and Computing, 243–258. Singapore: Springer. ISBN 9789811579073. doi:10.1007/978-981-15-7907-3_19.

Yang, S.; Luo, P.; Loy, C. C.; and Tang, X. 2016. WIDER FACE: A Face Detection Benchmark. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5525–5533. doi:10.1109/CVPR.2016.596. ISSN: 1063-6919.

Zhang, K.; Zhang, Z.; Li, Z.; and Qiao, Y. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23(10): 1499–1503. ISSN 1558-2361. doi:10.1109/LSP.2016.2603342.

Zhu, X.; and Ramanan, D. 2012. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2879–2886. doi:10.1109/CVPR.2012.6248014. ISSN: 1063-6919.