UNIVERSITY OF AMSTERDAM

An exploration of the suitability of Fawkes for practical applications

Danny Janssen
djanssen@os3.nl
Security and Network Engineering

Simon Carton
scarton@os3.nl
Security and Network Engineering

January 2021

Abstract

Facial recognition has been around for a while and the costs of entry for building accurate models using machine learning have gone down considerably, which means that more players can enter the market. Companies like Clearview.ai can use the internet and especially the numerous social media networks as a means of collecting data for training, without the people knowing or consenting. As a response on unauthorized use, different adversarial attacks have been devised in order to thwart automated classification by facial recognition models. Fawkes is one such tool, which is stated to work against the Microsoft Azure Face, Amazon Rekognition and Face++ platforms. Developed by a team of the SAND lab at the University of Chicago, the researchers claim it achieves 100% effectiveness against identification by cloaking images that are used to train facial recognition models on those platforms. We aim to verify some of the results from and claims made by that original paper on the Microsoft Azure Face platform and make a proof of concept in order to analyze the suitability of the tool for use in a practical application like a social media network. We show that the current version (0.3.1) of the tool is ineffective against models trained by the most recent version of the Microsoft Azure Face platform and that the tool is currently not suitable for use in applications like social media networks.

1 Introduction

The world has seen a gradual rise of machine learning research and also a rise in the use and commercialization of machine learning, with uses like smart assistants such as Siri or Alexa or more accurate weather prediction. Another field of machine learning that has been pursued is computer vision for uses such as face detection, but also face recognition. Facial recognition models need to be trained on data that consists of human faces in order to recognize them. The prevalence of social media networks in current times means that gathering data for machine learning training has never been easier. This is, however, not without privacy related costs for the users of those social media networks. The amount of resources needed to construct these systems have been getting smaller year over year with researchers even developing efficient prediction algorithms on IoT devices that fit in 2KB of RAM [1]. The lowering costs of machine learning also lowers the cost of entry, enabling more entities to scour the internet for the data needed to train facial recognition models. Companies like ClearView.ai can harvest the pictures of millions of people [2], without needing their direct consent and even without their knowledge. This data is then used to train the models of these companies with the goal of economical gain, which will often be at the expense of individual privacy rights. Facial recognition models can also be used by oppressive regimes in order to identify and track targets [3]. The act of collecting these kinds of images can thus pose a threat for the personal privacy of millions of people.

Facial recognition models can be thwarted by adversarial evasion attacks, such as putting carefully computed stickers on hats, which reduces the likelihood for that user to be recognized. Specially crafted masks or glasses can also help in lowering the chances of being recognized, but these kinds of evasive attacks require the user to wear accessories and clothing that might be impractical for normal use. The makers of the accessories also need full access to the models used to track them and the accessories can lose their function when the model is updated.

Another class of evasion attacks works by poisoning the system to disrupt their training. One of these kinds of attacks is a "clean label poisoning" attack. A team from the SAND laboratory at the University of Chicago has researched such an adversarial attack [4] and released a python based tool called Fawkes [5]. This tool is built to disrupt facial recognition models using clean label poisoning attacks, which will be talked about later on in this paper, and can serve as a way to regain a bit of privacy.

The goal of this research is to verify the study of the SAND laboratory and to build a proof of concept website where uploaded face pictures can be cloaked using the Fawkes tool. We utilize this proof of concept as a tool to determine whether this can be used in other larger scale applications such as social media platforms or other platforms that host substantive amount of photos with faces.

The paper is organized as follows. In section 2 our research questions will be explicitly defined. Then, in section 3 the background information regarding facial recognition will be given, while in section 4 other related works will be discussed. Section 5 will then introduce a more detailed outline of our research by describing our methodology. This is then followed by the results in section 6. After this, we will discuss these results in section 7. A quick ethical consideration of this research will be outlined in section 8. And finally, we will give our conclusion in section 9 and indicate possible future work in section 10.

2 Research questions

How suitable is the Fawkes tool for protecting the privacy of individuals within practical applications?

Sub-questions:

- Does the Fawkes tool still work effectively against the Microsoft Azure Face API?
- How scalable is the Fawkes tool for use in social media that host public face images?

3 Background

As a means to clarify the subject matter for the uninitiated, a short description will be given in this section on several key elements of facial recognition and the adversarial attacks that are possible against it.

3.1 Facial Recognition

In the field of biometrics, a facial recognition system can be used as either or both a face verification system and a face identification (or recognition) system [6]. The first

one is used to authenticate subjects and involves one-to-one matching, the latter is used to recognize subjects and involves one-to-many matching. The Fawkes research focused on an adversarial attack on facial recognition systems. These systems work by building models that can distinguish between the users the model was trained with. The output space of these models is called the feature space, the model is trained on images with attached labels (oftentimes identities) in order to find decision boundaries between the feature vectors of the identities located in the training data. The ultimate accuracy of the model is dependent on the quality of the training data, factors such as facial pose, expression, facial wear and illumination play a role. Whether a users cooperates or not is also a factor, it is easier to get more qualitative data for training when a user cooperative.

3.2 Adversarial Attacks

Attacks against facial recognition is also achievable, with advanced facial recognition systems with a deep learning-based architecture exhibiting vulnerabilities against adversarial attacks [7]. This class of attacks focuses on attacking the facial recognition models by means of using adversarial inputs, images to which an amount of (nearly) imperceptible or perceptible but natural-looking noise is added in order to thwart the efforts of the models [7][8][9]. These attacks can be used to poison the machine learning process of facial recognition models in order to lower the accuracy of identification, or to render it completely ineffective [4], [10].

4 Related Work

The original research by a team of the SAND Lab at the University of Chicago produced the Fawkes tool [5], which can 'cloak' images in order to be used for clean-label poisoning attacks, without perturbing the images to a substantial degree [4]. Clean-label poisoning attacks inject poisoned images into the training data of the machine learning models with the ultimate goal of causing the trained model to misclassify non-poisoned images. Clean-label attacks are different from normal machine learning poisoning attacks, in that the image labels stay identical to the original image labels and only the content of the images is altered. For this, the tool creates a cloak by using a target image and applies this to the original images. The target image is another identity, the tool tries to minimize feature distance between the cloak and the target and tries to maximize the distance between the cloak and the original image. This means that the features of the users images trained on this data will change in a manner as seen in Figure 1. The researchers posed that the pixel-level changes applied to the images are imperceptible to human vision using DSSIM scores. Three facial recognition platforms were tested in the original research: Microsoft Azure Face, Amazon Rekognition and Face++. It was stated that 100% effectiveness against all three platforms was achieved.

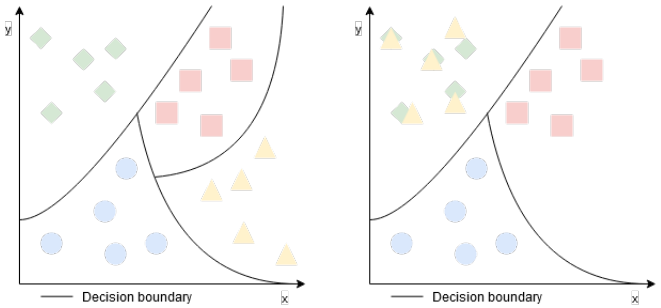

Figure 1: The cloaking process shifts the features of the users images (yellow triangles) towards that of the target (green diamonds)

Other research also looked into adversarial attacks, such as the research performed research on k-Randomized Transparent Image Overlays [11]. This is a reversible image perturbation technique that can thwart classification done by automated classifiers. They found that their method is 90% effective against state-of-the-art facial recognition systems and that the overlays are also computationally cheaper in comparison to learning-based methods.

Another example of adversarial attack mechanisms is TensorClog, which takes a more general approach towards creating adversarial perturbation that is not specifically focused on cloaking faces [10]. TensorClog was designed with the purpose of privacy protection by generating poisoned samples that hamper the transfer training process of Deep Neural Networks to result in worse test accuracy for the model as a whole. TensorClog maintains high degree of visual similarity with the original image, however the success rate in guarding users against facial recognition is limited at only 50%. To ensure the accuracy of facial recognition models there has also been done research on combating these adversarial attacks. One example of that is Faceguard devised by *Deb et al.* at the Michigan State University [12]. This generalized defensive technique

shows promising results in identifying perturbed images without explicitly being trained on the adversarial attack algorithms.

Further related research also includes the Structural Similarity Index Measure (SSIM) which is an algorithm proposed by *Wang et al.* [13]. SSIM is a method for comparing the difference of images based on several properties of the human visual system. The method has been used extensively since its release in 2004, with the paper having been cited over 20,000 times and has been integral to the field of image quality assessment as a whole. However, recent research by NVIDIA suggests that usage of the method should be carefully considered as it can produce unexpected results in both synthetic and realistic use cases [14]. Furthermore, they point out that in the context of deep learning, when using SSIM as a loss function, it can lead the training process astray. The Fawkes research uses a different score that they calculate using the SSIM value, this value is used in an adjusted form in the Fawkes research to calculate the score to specify the degree of perturbations permitted by their output model with regards to the cloak. The score they use is not the *similarity* index, but the Structural *Dissimilarity* Index Measure (DSSIM).

5 Methodology

In this section we will describe the tests that were executed and and the general layout of our research with respects to the experiments that were done.

The research consists of two parts: the verification and extension on some of the results published in the Fawkes paper and an analysis on the suitability for the software to be used in practical applications. For the first part we utilized one of the public cloud-based facial recognition software that the original Fawkes authors also used, namely: Microsoft Azure Face [15]. In order to analyze the effectiveness of the tool, several different configurations were tested with varying levels of cloaked images in the initial training set. Moreover, by testing the Fawkes software against Microsoft Azure Face several months later, it might be possible to identify if any optimizations that have been made to thwart this attack in the meantime.

Additionally, a proof of concept website was built that can cloak uploaded pictures using the Fawkes tool. The website was set up with the use of open source libraries and with adequate considerations for security and privacy in mind. To further aid this experiment, we also analyzed the source code of the Fawkes software to identify whether any obvious optimizations in terms of efficiency or scalability can be made, with a large focus on calling functions asynchronously. The implementation of these optimizations, however, are deemed out of scope. In the end, we will test our proof of concept based on the load it can handle and relate this to usage in other realistic scenarios such as social media platforms.

5.1 Verification of the Original Research

To test the Fawkes tool, multiple photos were taken of one of the authors in multiple kinds of lighting conditions and various kinds of poses. After having taken a larger amount of images than in the original paper, the photos were cloaked using the Fawkes tool version 0.3.1. The Fawkes tool has multiple different options that can be used while cloaking the images, like the batch size and the format of the cloaked output image. We varied our tests by cloaking images with and without separate targets, changes were made to the tool to use the same target for all the images cloaked without the separate target option, as not all images could be cloaked at once. Likewise, a number of the cloaking 'levels' found in the original tool were used, making multiple cloaked training image sets. We chose to use three cloaking levels, the 'min' level, the 'mid' level and the 'high' level, in order to see the effects on the amount of images detected and the obtained confidence levels with the respective perturbation budgets of the cloaking levels. A set of the cloaked photos, together with a face data set for more identities, was used in order to train a model on the Microsoft Azure Face platform. This was done using a python script, which also collected and stored the results we received from the Microsoft Azure Face platform. As we noticed that models trained with the same settings and the same data give the same results, each test was run once.

101 high-resolution photos of one of the authors were collected, 70 of which were randomly chosen and cloaked on various cloaking levels. Those were used to train Microsoft Azure Face models together with the IMM face data set [16], which contains 40 identities. The original images we used to train the model were not changed in between tests. One test was also done with the PubFig data set [17] as an additional source of identities, giving us 83 more. The remaining 31 photos from the set were used to test the trained model, running each photo through it

in order to identify the face contained in each one. Two kinds of data were collected: if the face in a test image was detected by the trained model or not, and the confidence the model had for all the test images. A lower confidence would suggest that the cloak was more effective. The confidence interval returned by the models is one of 0 to 1, which we directly translate to percentages. The tests and their results are described in the next sections.

5.1.1 No Cloaked Images

To get a baseline for the amount of photos identified and the confidence on those images, we trained a model using the photos before they were cloaked, so that we have something to compare to.

5.1.2 Min Cloaking Level

Multiple test were done on the 'min' cloaking level. This mode has a perturbation budget of 0.002 and takes 20 steps to cloak. The first test was done on a model trained using images cloaked on the same target, the 'Min' test. The second test was done on a model that trained on images that were cloaked using the separate target option in the Fawkes tool, the 'Min Separate' test. Additionally, a test was done using half of the photos cloaked with separate targets, with the other half being uncloaked images.

5.1.3 Mid Cloaking Level

The mid level cloak has a perturbation budget of 0.005 and takes 200 steps to cloak. Just as with the min level cloak; we ran tests with ('Mid') and without ('Mid Separate') using the separate targets option and with half of the training images cloaked and the other half uncloaked ('Mid Half').

5.1.4 High Cloaking Level

For the cloak on the high level we chose not to cloak the photos with separate targets; judging from test done before, using that option would only result in the confidence values being higher. For this same reasoning we also did not run a test with an even split of cloaked and uncloaked images. The high cloaking level in the Fawkes tool has a perturbation budget of 0.008 and takes 500 steps at most, costing a considerable amount of time to cloak.

5.1.5 More Identities

We wanted to see the effects of having more identities in the model than we were using for the other tests. The previous tests had 40 identities of the IMM data set, plus the identity of one of the authors. The original paper used the PubFig data set in some of their tests; we used this set for one test together with the IMM data set for a total of 124 identities. In total, one model was trained with the same target mid cloaked images, with all the sets in total resulting in over 14.000 images. This test was ran using the images cloaked on the mid level, without using separate targets.

5.2 Proof of Concept

To demonstrate the capabilities of the Fawkes tool within a more practical environment, we built a proof of concept website which can cloak uploaded pictures containing faces on-the-fly. This was used in combination with the source code analysis to determine what the status of the application is for usage in larger systems. Additionally, this helped us identify any potential optimizations that could be made.

Since the original tool was written as a Python program, we decided to use a Python web framework for a greater degree of possible integration. As we wanted to keep clear control over the implementation of our website, we selected the micro web framework Flask to serve as the back end. Flask was used in combination with uWSGI handling the reverse proxy requests coming in from the Nginx web server.

In order to evaluate the capabilities of our proof of concept, a test was set up that measures the resource usage of the web application in several different settings. To be specific, we measured the CPU performance, memory usage and duration of cloaking 1, 2, 5 and 10 images during a session. The images used were part of the IMM face data set. Also, the images are all of the resolution 640×480 and between the size of 114kB and 127kB. Furthermore, the tests were executed on Ubuntu 20.04 on a laptop with an Intel 4720HQ (2.60Ghz, 4 cores, 8 threads) and a NVIDIA 960m GPU.

6 Results

In this section we will display the results we collected from running our various tests and visualize these in a number of plots. We also evaluate the proof of concept and discuss our code analysis.

6.1 No Cloaked Images

The single baseline test ran as expected, with all the images being detected as the right person, with a high average confidence. The results found in Table 1 show that all of the 31 images were detected with the same identity and that the average confidence is high at 96,6%. The spread of all the confidence values is shown in Figure 2. The spread is tightly located around the average with only one outlier towards the bottom, suggesting that the model was overall very confident in the process of identifying.

	No cloak
Amount identified	31
Average confidence	96,6%

Table 1: Results from test without cloaked images, with the amount of correctly identified images and the average confidence of the identification

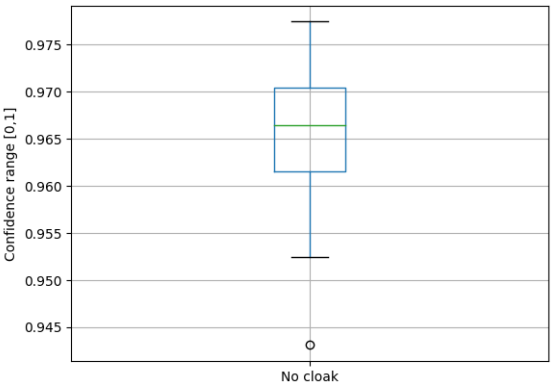

Figure 2: The box plot shows quite a tight spread around the average, with only one outlier

6.2 Min Cloaking Level

The test on the min level showed us the gains possible by changing the various options and the amount of cloaked images, with the same target cloaks giving us the lowest average confidence. Table 2 shows the test on the model trained using an even split of cloaked and uncloaked images has the highest average confidence of 96,1%, being very close to the baseline test with only half a percent of a difference. The model trained with images cloaked on the same target gives the lowest average confidence for this cloaking level at 93,1%, a 3,5% improvement. Just as with the model trained without using cloaked images, all the images were correctly identified. Figure 2 displays the spread of the confidence values for the three tests done with this cloaking level. All the tests have a consistent outlier at the bottom, suggesting that there could be a picture that profits more from the cloak with regards to lowering the confidence.

	Same	Separate	Half
Amount identified	31	31	31
Average confidence	93,1%	94,3%	96,1%

Table 2: Results from the tests with images cloaked on the min level, with the amount of correctly identified images and the average confidence of the identification

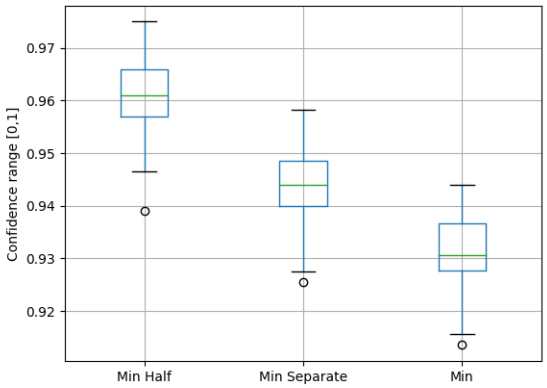

Figure 3: The box plot containing results from the three tests on the min cloaking level shows improvements with each change in the test

6.3 Mid Cloaking Level

As with the min level, each change in the test lowers the average confidence of the model, suggesting that the cloak works better in those circumstances. Table 3 shows a similar relative results as with the min cloak tests, with the model trained with the even split of cloak and no cloak having the highest confidence of 95,8% for the mid level.

Additionally, all 31 images were detected as the having a face with the right identity. Figure 4 shows the spread and outliers of the three tests. The same target test has more outliers than the other tests, which was not expected. This could mean that the cloak is more effective as more of the test image have lower confidence values.

	Same	Separate	Half
Amount identified	31	31	31
Average confidence	88,9%	92,6%	95,8%

Table 3: Results from the tests with images cloaked on the mid level, with the amount of correctly identified images and the average confidence of the identification

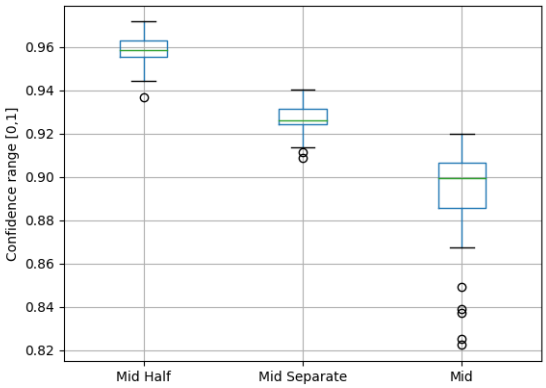

Figure 4: The box plot containing results from the three tests on the mid cloaking level shows improvements with each change in the test, with the same target test giving the most outliers as well

6.4 High Cloaking Level

As expected, the high cloaking level performed the best out of all our tests. The higher perturbation budget allows the distance between the cloaked image and the original image to be increased even further, with Table 4 showing the lowest average confidence we have obtained so far of 79,3% and the lowest confidence of all the images being 74,9%. Figure 5 shows the spread of the confidence values. There is an outlier that is relatively far removed from the average at 0.89 on the confidence interval, otherwise the spread is quite uniform.

	Same
Amount identified	31
Average confidence	79,3%

Table 4: Results from the test done on the high cloaking level shows that all images were correctly identified

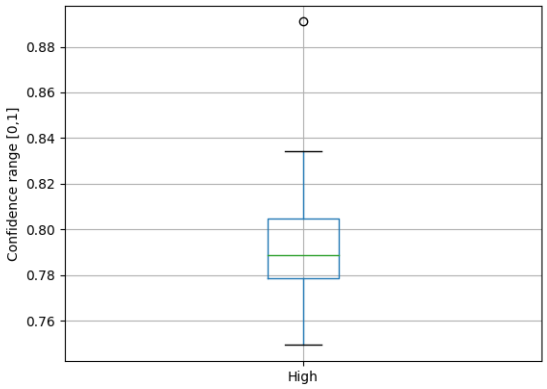

Figure 5: A box plot containing results from the one test on the high cloaking level shows a narrow spread for most of the images with one big outlier however

6.5 More Identities

This test performed exactly the same as the test on the mid level cloaked with the same target, with Table 5 showing that the average confidence value is 88,9%. This can be explained the images being identical between the mid level cloak test with the same target and this test. As the same images from that specific test were used, we argue that adding more identities than the 41 that were used in previous tests has no effect on the confidence values, until at least 124 identities. The spread of the confidence values can be found in Figure 6.

	Same
Amount identified	31
Average confidence	88,9%

Table 5: Results from test with more identities, with the amount of correctly identified images and the average confidence of the identification

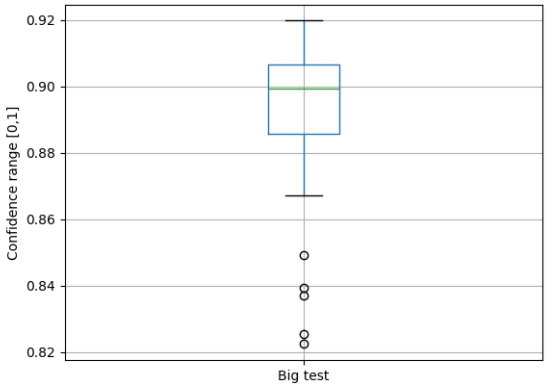

Figure 6: The spread of the results from the test using more identities. Just as the mid cloak test with the same target, the plot shows a number of outliers.

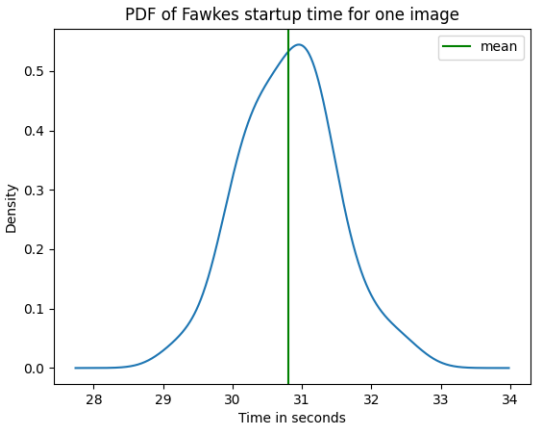

Figure 7: The startup time of the Fawkes tool when cloaking one image on the min level. This density graph is the result of 30 tests, stopping the tool when it finished preprocessing.

6.6 Code analysis

The small code analysis was done by hand, going through the code to identify areas where a speedup could possibly be achieved. The startup time of the tool for even one image is quite substantial, as can be found in the density function in Figure 7. Here we see that the startup of the tool is taking just under 31 seconds before it starts to actually cloak the image. We mainly identified areas where sequential loops could be executed in parallel, like loading the images. This specific example would not speedup the tool when running it with a single image, but could speedup running large batches at the same time. Other areas of the tool, like detecting faces in the images, would likely also be able to run in multiple threads, which could lead to a speedup of the tool.

6.7 Proof of Concept

We created a simple proof of concept website (Figure 8) to test the practical use of the Fawkes tool. The proof of concept website cloaks the images with the min setting of the Fawkes tool. However, even with the lowest setting being used, a considerable amount of time was needed to cloak images. As mentioned in the previous section, the startup time of the tool takes on average a little under 31 seconds; this highlights why the time spend per photo decreases as more images are uploaded at once.

Furthermore, the tests of the proof of concept the results of the analysis were further confirmed. During the cloaking process itself, the GPU is utilized well with its utilization exceeding 90% throughout the cloaking process. However, as can be seen in Table 6, the CPU is underutilized. Since the test machine has eight available threads, the theoretical CPU Load with no other programs running should be 800%. In a practical situation such as this (even with other unnecessary user programs having been killed) this is not possible to achieve. Nonetheless, the lower CPU load numbers highlight the under-utilization of parallel processing power.

Additionally, we found the memory footprint of the application to be relatively high irregardless of the amount of images being processed. This limits the amount of possible simultaneous sessions that can be handled if all processing is limited to one machine.

Amount Cloaked	1	3	5	10
Real Time	90	104	121	160
CPU Load	98.9%	102.9%	100.8%	104.4%
Peak Memory	2.2GB	2.2GB	2.3GB	2.3GB

Table 6: Results from the performance test of proof of concept website. The amount of images cloaked from the IMM data set can be seen on top with the average cloaking time, CPU load and peak physical memory usage in gigabytes. The cloaking time is given in seconds.

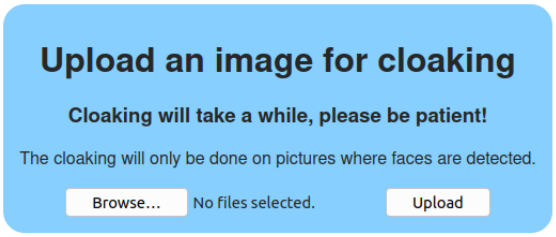

Figure 8: A screenshot of the main page of the proof of concept website.

7 Discussion

Research like the original paper are becoming more important as technology evolves. Privacy often takes the backseat when leaps are made in technology. This research verified part of the original paper and looked at the possibility of using the Fawkes tool in a setting like a social media network, one of the primary collection grounds for training data.

7.1 Facial recognition platform as a black box

The facial recognition platform that we used, Microsoft Azure Face, can be seen as a black box. We only have limited control over how it works and sometimes no satisfying answers were found in the documentation with regards to certain defaults, like the confidence threshold used by the models. This makes it harder to accurately talk about the implications of some of the results.

7.2 Fawkes for Social Media

With our proof of concept website we demonstrated how a high amount of resources are necessary for the calculation of a cloaked image pose challenges for smaller scale systems. To resolve the problem of limited amount of resources available on an individual machine, a distributed version where the cloaking process happens on separate nodes could allow for greater scaling capabilities. However, the question remains whether a social media platform would find it beneficial to integrate the Fawkes tool within their infrastructure.

During our testing we noticed that Fawkes alters the images in a way that is sometimes not favorable for the subject. Especially on settings with a higher perturbation budget, the features of the subjects face tend to be modified in such a manner that it could often be perceived as if the subject was unhealthy, with blue spots over the face. An example of this can be seen in Figure 9, this was one of the cloaks at this level with less artifacts. Furthermore, we observed that cloaking images taken in bad lightning conditions produced noticeable artifacts, as is visible in Figure 10. However, for both of these findings to be objectively determined, further research would need to empirically evaluate this. Nonetheless, this raises the question whether Fawkes would be suitable for social media as the arguably negative alteration of uploaded photos could reflect poorly on the social media companies themselves.

Figure 9: An image cloaked on the high setting with some artifacts in the face, mainly blue spots here and there that could appear to be bruises.

Furthermore, as we have shown, the Fawkes tool proves to currently be ineffective against Microsoft Azure Face. The evidence for possible circumvention of the advertised cloaking effects diminish the benefits social media companies could expect from integrating the tool.

7.3 Update Fawkes team

On the 28th of January this year, the team that worked on the Fawkes tool came out with a statement on their website [5], stating that it came to their attention that "*a significant change was made to the Microsoft Azure facial recognition platform in their backend model. Along with general improvements, our experiments seem to indicate that Azure has been trained to lower the efficacy of the*

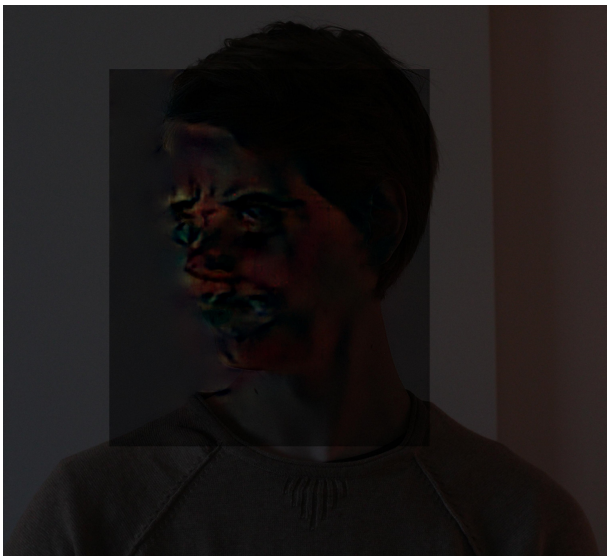

Figure 10: A low light image cloaked on the high setting with some noticeable artifacts, mainly the rectangular overlay that is visible.

specific version of Fawkes that has been released in the wild." They do not state how much the efficacy has been lowered, but this could at least partly confirm our results of all the images still being correctly identified.

7.4 API Limit

One of the constant bottlenecks while training and testing the images was that we were bound to a limit of 20 API calls per minute. With the big test having over 14.000 images, the time it took to run that test was over half a day, clocking in over 13.5 hours in all. The free tier is also capped at 30.000 API calls, which is easily reached when multiple big experiments are executed. In order to run more tests and to run them quicker, the paid tier is recommended.

7.5 Long Term Effectiveness

As was seen with the update from the Fawkes team, an adversarial attack that is effective today, might become ineffective tomorrow. This cat and mouse game is analogous to encrypting messages: the encryption might be impossible to crack with today's technology but in the future, when a method is found to feasibly crack that type of encryption, all the stored messages will be able to be decrypted, thus revealing the contents inside. The same is true for this attack. A company with an interest in defeating these attacks can focus their efforts on exactly that, after which all the cloaks that were made before are rendered ineffective, as we saw in the update of the Fawkes team. It could be argued that the cloaking of the images is only effective on the short term, having to re-cloak and re-upload the images when the previous method is defeated. This fact notwithstanding, the authors of this paper still believe research into regaining privacy is important in the modern era.

8 Ethical Issues

It could be argued that there are some ethical issues connected to the Fawkes tool. A user can cloak their pictures to a certain target, in theory this property can be used to give the target the classification of the user when they are being identified by the model. Say a known criminal user cloaks their photos using an innocent citizen target and those photos are used by law enforcement to create a model to search for criminals, the citizen can falsely trigger the model. Law enforcement could then suspect the citizen of being the criminal user, if the law enforcement were to be careless about verifying the classification.

9 Conclusion

Taking the results from section 6 into consideration, we conclude that the Fawkes tool that we used is not effective against the current Microsoft Azure facial recognition platform. In all the tests the model could recognize and correctly identify the face in the test images, with a high confidence. The lowest confidence value achieved of a single image was 74,9% on the high level, all the other images from all the tests were above 75%. Purely based on the results given by the Microsoft Azure facial recognition platform we reach that conclusion; if another confidence threshold were to be used the results could change. The original paper does not mention a threshold they considered and finding information about the confidence threshold used by the Microsoft Azure did not deliver a satisfying answer.

Furthermore, given the ineffectiveness of Fawkes against the Microsoft Azure, the practical use for the tool within the context social media networks is questionable. As the Fawkes team has indicated that they will try to counter the measures taken by Microsoft, a possible update of

the Fawkes tool could raise the effectiveness. However, currently we would not advise the usage of the tool for systems of a larger scale.

10 Future work

As the Fawkes team released a statement regarding the efficacy of the current version of the tool on the Microsoft Azure facial recognition platform, future research could focus on verifying the claims made in the original paper using the new version of the tool the team is working on.

As the original paper made claims about the efficacy of the tool on three facial recognition products, future research could also focus on verifying those claims on the Amazon Rekognition and the Face++ platforms.

Further research could look into using DSSIM as a measure for difference between images, we noticed that even with relatively low DSSIM budgets images could still become quite perturbed. For the cloaks to be useful, a high similarity with the original image is required when the cloak is to be uploaded to a social media network.

11 Acknowledgements

We thank our KPMG supervisors Aristide Bouix and Huub van Wieren for their invested time and their expertise during our research.

References

[1] Ashish Kumar, Saurabh Goyal, and Manik Varma. Resource-efficient Machine Learning in 2 KB RAM for the Internet of Things. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1935–1944, International Convention Centre, Sydney, Australia, August 2017. PMLR. URL `http://proceedings.mlr.press/v70/kumar17a.html`.

[2] Kashmir Hill. The Secretive Company That Might End Privacy as We Know It. January 2020. URL `https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html`.

[3] Angela Daly. Algorithmic oppression with Chinese characteristics : AI against Xinjiang's Uyghurs. In *Artificial intelligence: Human rights, social justice and development*, pages 108–112. APC, 2019. URL `https://strathprints.strath.ac.uk/71586/`.

[4] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. page 16, 2020.

[5] Shawn Shan. SAND Lab: Fawkes. URL `http://sandlab.cs.uchicago.edu/fawkes/`.

[6] Stan Z. Li and Anil K. Jain, editors. *Handbook of Face Recognition*. Springer London, London, 2011. ISBN 978-0-85729-931-4 978-0-85729-932-1. doi: 10.1007/978-0-85729-932-1. URL `http://link.springer.com/10.1007/978-0-85729-932-1`.

[7] Fatemeh Vakhshiteh, Raghavendra Ramachandra, and Ahmad Nickabadi. Threat of Adversarial Attacks on Face Recognition: A Comprehensive Survey. *arXiv:2007.11709 [cs, eess]*, January 2021. URL `http://arxiv.org/abs/2007.11709`. arXiv: 2007.11709.

[8] Fabio Valerio Massoli, Fabio Carrara, Giuseppe Amato, and Fabrizio Falchi. Detection of Face Recognition Adversarial Attacks. *Computer Vision and Image Understanding*, 202:103103, January 2021. ISSN 10773142. doi: 10.1016/j.cviu.2020.

103103. URL https://linkinghub.elsevier.com/retrieve/pii/S1077314220301296.

[9] Morgan Frearson and Kien Nguyen. Adversarial Attack on Facial Recognition using Visible Light. *arXiv:2011.12680 [cs]*, November 2020. URL http://arxiv.org/abs/2011.12680. arXiv: 2011.12680.

[10] Juncheng Shen, Xiaolei Zhu, and De Ma. TensorClog: An Imperceptible Poisoning Attack on Deep Neural Network Applications. *IEEE Access*, 7:41498–41506, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2905915. URL https://ieeexplore.ieee.org/document/8668758/.

[11] Arezoo Rajabi, Rakesh B. Bobba, Mike Rosulek, Charles V. Wright, and Wu-chi Feng. On the (Im)Practicality of Adversarial Perturbation for Image Privacy. *Proceedings on Privacy Enhancing Technologies*, 2021(1):85–106, January 2021. ISSN 2299-0984. doi: 10.2478/popets-2021-0006. URL https://content.sciendo.com/view/journals/popets/2021/1/article-p85.xml.

[12] Debayan Deb, Xiaoming Liu, and Anil K. Jain. FaceGuard: A Self-Supervised Defense Against Adversarial Face Images. *arXiv:2011.14218 [cs]*, November 2020. URL http://arxiv.org/abs/2011.14218. arXiv: 2011.14218.

[13] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861. URL http://ieeexplore.ieee.org/document/1284395/.

[14] Jim Nilsson and Tomas Akenine-Möller. Understanding SSIM. *arXiv:2006.13846 [cs, eess]*, June 2020. URL http://arxiv.org/abs/2006.13846. arXiv: 2006.13846.

[15] Microsoft. Microsoft Azure Facial Recognition. URL https://azure.microsoft.com/en-us/services/cognitive-services/face/.

[16] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM Face Database - An Annotated Dataset of 240 Face Images. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, May 2004. URL http://www2.compute.dtu.dk/pubdb/pubs/3160-full.html.

[17] Nicolas Pinto, Zak Stone, Todd Zickler, and David Cox. Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. In *CVPR 2011 WORKSHOPS*, pages 35–42, Colorado Springs, CO, USA, June 2011. IEEE. ISBN 978-1-4577-0529-8. doi: 10.1109/CVPRW.2011.5981788. URL http://ieeexplore.ieee.org/document/5981788/.