Face detection and blurring

Tomislav Prhat
FER
Zagreb, Croatia
tomislav.prhat@fer.hr

Ana Čačić
FER
Zagreb, Croatia
ana.cacic@fer.hr

Dora Tomić
FER
Zagreb, Croatia
dora.tomic@fer.hr

Bruno Torbarina
FER
Zagreb, Croatia
bruno.torbarina@fer.hr

Vedran Zoričić
FER
Zagreb, Croatia
vedran.zoricic@fer.hr

***Abstract*—Computer vision can lead to privacy issues such as unauthorized disclosure of private information and identity theft, but it may also be used to preserve user privacy. For example, using computer vision, we may be able to identify sensitive elements of an image and obfuscate those elements thereby protecting private information or identity. However, there is a lack of research investigating the effectiveness of applying obfuscation techniques to parts of images as a privacy-enhancing technology. The objective of this work is to explore this issue by using a YOLO model for detecting people's faces and pixelization as a blurring method.**

***Index Terms*—computer vision, YOLO, face detection, blurring, privacy protection**

I. Introduction

In today's digital world, it is essential to keep personal privacy up to the highest standards and ensure to minimize one's digital footprint. Since most smartphones have some sort of facial recognition to unlock the phone, having your facial features possibly leaking all over the Internet is not a good thing. Even posting pictures with privacy protection on social media platforms is not completely safe and that is especially dangerous for underage children. These photos could be used to engineer anyone's identity and thus gain access to sensitive personal data. These are only a few examples of why there needs to be a complete solution to keep people's data private. The goal of our project is to combat these problems by finding faces in an image using deep neural networks and blurring them out using image processing.

II. Related work

Face detection is a trifling task for humans, which we can perform naturally with almost no effort. However, the task is complicated to perform via machines and requires many computationally complex steps to be undertaken. Through the years, face detection has been widely studied in computer vision. Some approaches focus on developing efficient and accurate machine learning algorithms. The general practice in these approaches is to collect a large set of faces and non-faces as training examples and apply certain machine learning algorithms to perform the classification of human faces [1]. Subsequent research has been focused on the improvement of the algorithm to tackle factors such as light conditions and multi-view face detection. Other approaches concentrate on optimization of face detection through non-learning approaches such as template matching and skin color detection. Blurring is the most commonly used and widely studied approach to controlling photo content disclosure [1]. De-identification effectiveness of blurring has been studied both in photo and video-based media. There are different levels of blurring obfuscation. Increasing the blur level generally reduces the accuracy of identification by both machines and human viewers [2].

One of the first applications of face blurring was with Google Street View in 2008 when testing on the platform started. Since then, many companies slowly started using this procedure, one of which is YouTube with its face blurring tool designed to protect activists which launched in 2012.

III. Dataset

A. Overview

The dataset used in this project is the famous COCO (Common Objects in Context) dataset [3]. The COCO dataset has over 330k images (of which more than 220k are labeled) and 80 object categories. However, since the focus of this project is only on images containing faces, not all images in the dataset were usable. A reduction had to be made and the result of this action is a dataset containing 1247 training images, 270 validation images, and 270 testing images.

B. Images

The images inside the dataset (mostly) contain people in everyday situations. The images vary in size and color models. Most of them are RBG images, with only a few grayscale ones.

C. Labels

The images are labeled using the following format:

$$class\ x\_center\ y\_center\ width\ height$$

The variable $class$ denotes the class to which the image belongs.

(x_center, y_center) represent the coordinates of the center of the bounding box.

The variables $width$ and $height$ are the dimensions of the bounding box.

IV. Model

Object detection is a well-studied task in computer vision and image processing. Nowadays, multiple efficient models perform this task. Some of the most popular models are:

- R-CNN (Region Based Convolutional Neural Networks)
- HOG (Histogram of Oriented Gradients)

- SSD (Single Shot Detector)
- YOLO (You Only Look Once)

The model used in this project is the last one mentioned. To be precise, the used model is the fifth version of the YOLO model known as YOLOv5.

YOLO uses an end-to-end neural network that predicts the bounding boxes and class probabilities. The algorithm divides the image into N equal grids and each of them is responsible for predicting the object inside it [4].

V. Results

The results of our face detection model are shown in the table below. Since a lot of the pictures in the dataset contain slightly covered faces or faces that are turned to the side our model achieved 86.8% precision, 75% recall and 82.9% mean average precision. For comparison, bigger and more sophisticated models can achieve a precision score of over 99%. Perhaps using a larger dataset and stronger GPU-s with longer training times would improve our results.

YOLOv5	
Metric	Value
Precision	86.8%
Recall	75%
mAP	82.9%

Training metrics are shown in Fig. 1 and Fig. 2

The metrics depict the loss through epoch on the train and validation set, as well as the previously mentioned metrics. *box_loss* represents the bounding box regression loss (MSE). *obj_loss* is the confidence of object presence is the objectness loss (binary cross entropy). *cls_loss* denotes the classification loss (cross entropy).

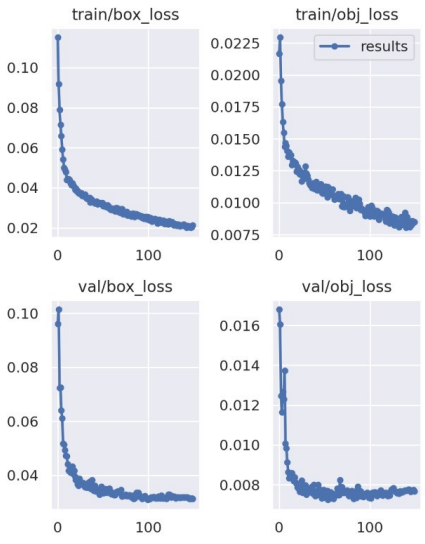

Fig. 1. Results of feature extraction training - loss through epochs

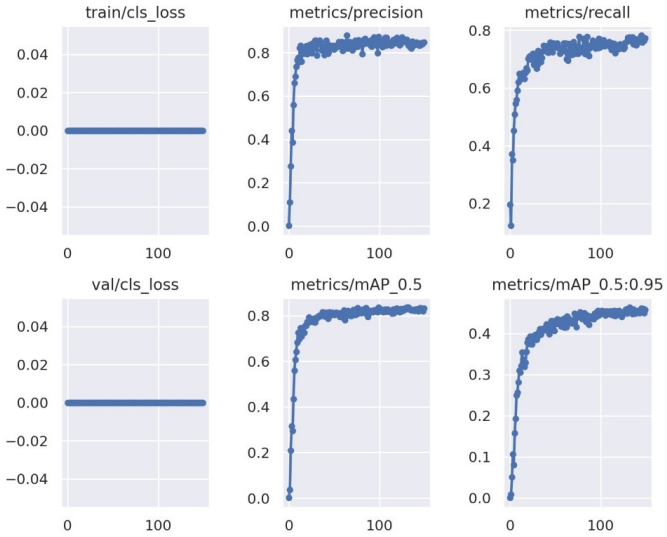

Fig. 2. Results of feature extraction training

An example of how the model performs can be seen on Figure 3.

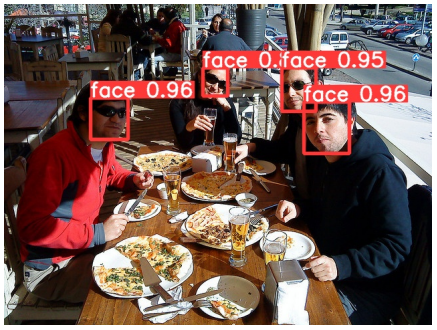

Fig. 3. Example of face detection

VI. Blurring

There are multiple ways to blur an image. One of these is the use of a mask. A mask is essentially a filter. Masking is known as filtering which is a process of convolving a mask with an image. We can use masks to blur an image or a part of an image. In addition to blurring, there are numerous other uses of masks, one of which is edge detection.

When we look at an image, we can perceive it and the objects in the image by the edges of the objects in the image. Blurring works by reducing the edges in an image and smoothing the transitions between colors. The most common filters (masks) used for blurring are the average filter and the Gaussian filter.

A. Average filter

This filter, also commonly known as a mean filter, works so that it replaces each pixel value with the average value of its neighboring pixels and itself. This filter is based on a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 kernel

is used, but we can also use larger kernels for more severe blurring (e.g. 5×5). We carry out the filtering process by computing the convolution of the image and the average filter. An example of a 3×3 kernel of this filter is as follows.

$$K = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

B. Gaussian blur

This filter gives a different weight to each entry in the kernel based on the distance of a pixel to the selected pixel. The closer the pixel is to the selected filter, the greater its weight, the further away it is, the lower the weight. The formula for this filter is

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

where x and y correspond to the horizontal and vertical distance and σ to the standard deviation. The greater the value of σ, the greater the blur. We estimate the Gaussian function by an odd-sized kernel whose entries are the estimations of the Gaussian function in that pixel. An example of a 3×3 Gaussian kernel is as follows.

$$K = \frac{1}{9} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

C. Pixelization

Through our testing, we have concluded that we hadn't gotten a satisfactory result using the aforementioned filters so we went with a simpler approach. We used the technique of pixelization where we lowered the size of the image in the part that we wanted to blur and then increased it. This gave us a "chessboard effect" that has worked really well at censoring the faces in the images.

The result of this blurring process can be seen on Figure 4.

Fig. 4. Pixelization performed on example image

VII. Security issues

From a privacy-enhancement perspective, while face blurring may be useful in some situations, such as those where the person in the photo is unknown to the viewer, in other cases, such as in online social networks where the person in the image is known to the viewer, face blurring does not provide privacy protection. Three experiments were conducted exploring the effect of blurring on unfamiliar faces and identification results indicated that even in their highest blurring degree condition, the identification rate was quite successful (around 40%) [1]. Moreover, identification accuracy increases when identifying familiar faces.

VIII. Discussion and conclusion

To increase personal privacy, this work analyzes a method that can detect people's faces and uses masks to blur their faces effectively hiding their identity for security purposes. Different types of blurring methods and models can be used to best fit the data that is being processed. Although the model used in this work is not perfect, it shows that there is a lot of potential for this type of approach. For future research, the effectiveness of this model could be tested by training a model that tries to remove blurring, and by combining both models a new and improved model could be developed.

References

[1] Liu, Z., Hao, M., Hu, Y. Visual Anonymity: Automated Human Face Blurring for Privacy, University of Wisconsin at Madison

[2] Li, Y., Vishwamitra, N., Knijnenburg, B. P., Hu, H., Caine, K., Blur vs Block: Investigating the Effectiveness of Privacy-Enhancing Obfuscation for Images, Clemson University, 2017.

[3] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).