CS 230 Final Project Spring 2020

KiDINet: Face De-Identification of Minors

Isaac Hales, Laurel Hales, Akhil Jariwala

Abstract

Although the protection of children's online privacy is widely agreed to be sacrosanct, children everywhere have personally identifying photos uploaded to the internet every day by their parents and other sources. The researchers created KiDINet, an end-to-end pipeline that collects photos and returns images with the faces of all minors deidentified. The pipeline includes a face detection model based on faced*, an age detection model architected by the researchers based on VGG-Face, and a facial swapping stage using FSGAN. The age detection model performed well with minor identification on the validation set, reaching a 90% F1 score; however, end-to-end evaluation yielded a minor face deidentification success rate below 20%, due to a more diverse test set and error compounding. The project shows promise that deep learning models may be used to protect the privacy of children online without manual censoring one day.*

1. Introduction

Every day, tens of thousands of pictures of children are uploaded to the internet. Sharenting, when parents share information about or photographs of their children on their own personal social media accounts, causes many problems. Barclays has predicted that by 2030 sharenting will account for ⅔ of identity fraud [1]. Another issue is virtual kidnapping, where seemingly innocent photographs of children, posted on social media accounts are used to create pornography. An unofficial survey of an online child pornography site by the Australian government found that about half of the millions of images on the website were sourced directly from social media [2]. Sometimes posted photographs of children are used for bullying by peers. Many child advocacy groups have also raised the concern about preserving digital autonomy for children and the need to give them a say in what is posted [3].

Many parents attempt to solve these problems by manually editing images by placing an emoji over a child's face to protect their identity. However, this method is time consuming and creates an image that is unnatural looking and has clearly been edited. KiDINet is a proposed method to de-identify the faces of minors in photos using neural networks that replaces the faces of minors with a natural looking substitute. KiDINet works by connecting three models that each perform a specific task: 1. Face detection 2. Age and gender detection 3. Face de-identification. This is the first attempt to we have seen to create an end-to-end solution for this specific task and provide parents with a simple and safe way for parents to share pictures that include children while protecting their children.

1.1. Related Works

We have not found any other solutions to this particular issue, but there has been a lot published on each of the pieces of our solution. Some of the previously used methods for each of the sub-tasks are described below.

1.1.1 Face Detection

The history of facial detection is closely intertwined with the history of generic object detection. In 2004, Viola and Jones (VJ) built a model that used Haar-Like features and AdaBoost, creating the first practical face-detector. Later papers built on Viola and Jones using Histogram-Oriented Gradients and Scale-Invariant Feature Transform features for human detection. Real-time performance arrived with the use of deep convolutional neural networks, namely Regions with CNN features. Popular algorithms that employ this approach include the Single Shot Scale-invariant Face Detector proposed by Zhang et al. and Single Stage Headless Face Detector by Najibi et al. The latest advances include neural networks that utilize the YOLO model, an object detection algorithm that uses anchors and grids [4]. Even with YOLO, researchers are still grappling with the tradeoff between speed and accuracy, capturing faces at varying scales, and detecting faces with obscuring objects like sunglasses or scarves.

1.1.2 Age and Gender Detection

The first use of a CNN for age estimation and gender classification was performed by Yang et al. in 2011. Dong et al. overcame the problem of lacking labeled training images through Deep ConvNets, which used transfer learning to extract high-level age features. Later, Levi and Hassner used a shallow CNN to classify faces into 8 ages, reducing computational load and complexity with limited performance loss [5, 6]. However this network was trained on a small dataset, only about 2000 different people and it only had only an 80% accuracy. Smith et al. created a network with higher accuracy using transfer learning from VGG-19 [7]. We attempted to create a network that followed the same outline as the one described in Smith et al.

1.1.3 Face De-Identification

Early approaches to face de-identification included k-anonymity models, which replaced a face with a substitute that was the average of the closest k identities computed from the same set of images. Jourabloo et al. used Active Appearance Models to manipulate facial attributes until a facial verification classifier recognizes the two images as separate subjects. The application of generative neural networks to face de-identification is newer. In 2017, Meden et al. used GNNs to create a unique face using just a handful of appearance-related parameters like skin color and gender [8]. Modern techniques include facial attribute replacement (replacing some facial attributes, like a nose or mouth, with attributes from consenting donors), while retaining facial expression and realistic features [9], or using a GAN to generate full faces for replacement [8]. These models need to balance maintaining realism in the facial images while effectively de-identifying faces. As research in these areas continues, we will be able to update our pipeline to take advantage of state-of-the-art de-identification techniques.

1.2. Methods

Our pipeline takes in photos with multiple faces and returns a photo with the faces of minors in the photo de-identified. This is accomplished through connecting four separate models to perform the desired task shown in Figure 1. Model 1 detects the location of every face in a photo; Model 2 identifies the gender of each face; Model 3 identifies the age of the face; Model 4 uses the age and gender to select a donor face for each minor and swap the minor’s face with that of the donor.

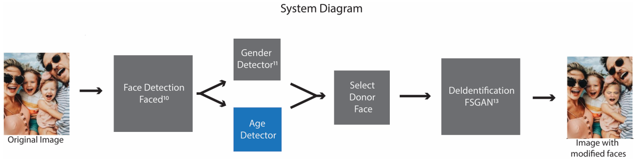


Figure 1: The entire KiDINet network, showing all of the steps of the pipeline

1.3. Models

1.3.1 Face Detection

Our facial detection algorithm was based on *faced[10]*, an open-sourced variant of YOLO that specializes in face detection, as opposed to multi-class detection. *Faced* is an open-source ensemble of two convolutional neural networks. The first neural network, called the Face Detector, breaks each image into discrete mini-boxes, predicting the probability of a face in each. The second neural network, called Face Corrector, is designed to take the output of the first network and return the face bounding box. The *faced* implementation is based on YOLO, but can achieve real-time performance on a CPU through model simplification

1.3.2 Gender Detection

For this project we have started with an implementation of the model designed by Tal Hassner and Gil Levi [11]. This model is comprised of 3 convolutional neural networks and 2 fully connected layers.

1.3.3 Age Detection

We designed our own age detector based on the work of Smith et al [7]. We started with the VGG-Face network trained on the VGG-Face database [12, 13] without the top layers. We replaced the original top layers with three fully connected layers of our own. In our final network the first two fully connected layers had 1024 hidden units each followed by a ReLu activation function. The final layer produced a single value and had no activation function as shown in Figure 2. The VGG-Face layers were frozen during training and so retained their original values.

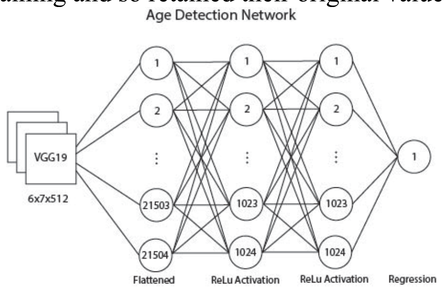


Figure 2: The layers we built that replace the top layer of the VGG-Face network.

1.3.4 Face De-identification

We implemented the FSGAN model to exchange the original face in the picture with a donor face [14]. FSGAN is a state-of-the art model that uses a series of networks (among other techniques) for face detection, face reenactment, face generation (for occluded images), and facial blending (to account for differences in lighting and skin tone). FSGAN can be applied to faces that it has not been trained on, which is important for our use case. While we originally planned to replace the true face of each minor with a generated face, recent thinking suggests that face generation is overkill for deidentification. Instead, we swapped these faces with those of existing child faces from our age-detection dataset. Our decision dramatically reduced the time to implementation, since training a GAN to generate low-resolution child faces would likely take weeks or months [15].

1.4. Hyperparameter tuning

We tested three different types of loss functions for our age detector. We tested a binary model (two possible outcomes of minor/not minor) with a cross-entropy loss function, a model that placed the ages into 4 different bins ((0,8), (9-15), (16,25), and (26, 100)) again using a cross-entropy loss function. And a final regression model that used a mean squared error loss to return the age as a single number.

We could calculate the minor accuracy during training but the F1 score had to be calculated afterwards. After initial training both the binary and the continuous versions were able to attain 97% accuracy on detecting minors so we chose to use the continuous model with the mean squared error loss since it also provided us with an approximate age that we used to identify the most appropriate donor image.

The final loss function that we used in the final model was a mean squared error loss function:

$$MSE = \frac{1}{m}\sum_{i=0}^{m}\left(y_{pred} - y_{true}\right)^2.$$

However, we evaluated our performance on two different measures, the minor accuracy, or accuracy of predicting whether a face belonged to a minor and the F1 score. Due to the facts that we were using a regression model, and that TensorFlow does not allow the use of conditional statements in calculating metrics, calculating the minor accuracy was challenging. We calculated it in the following way:

$$true\ positive = \frac{1}{m}\sum_{i=1}^{m}\left(1 - \frac{\max\left(0, y_{pred} - 15\right)}{y_{pred} - 15}\right)\left(1 - \frac{\max\left(0, y_{true} - 15\right)}{y_{true} - 15}\right)$$

$$true\ negative = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\max\left(0, y_{pred} - 15\right)}{y_{pred} - 15}\right)\left(\frac{\max\left(0, y_{true} - 15\right)}{y_{true} - 15}\right)$$

$$false\ positives = \frac{1}{m}\sum_{i=1}^{m}\left(1 - \frac{\max\left(0, y_{pred} - 15\right)}{y_{pred} - 15}\right)\left(\frac{\max\left(0, y_{true} - 15\right)}{y_{true} - 15}\right)$$

$$false\ negatives = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\max\left(0, y_{pred} - 15\right)}{y_{pred} - 15}\right)\left(1 - \frac{\max\left(0, y_{true} - 15\right)}{y_{true} - 15}\right)$$

$$minor\ accuracy = true\ positive + true\ negative$$

We calculated the F1 score after training using the following formula:

$$F1 = \frac{2 \cdot true\ positve}{2 \cdot true\ positive + false\ positive + false\ negative}.$$

We chose to measure the success of the model using the minor accuracy and F1 score because the most important goal of this model is to correctly identify models. We compared the F1 score and the mean squared error after each epoch during training for each experiment as shown in Figure 3. We believe that this figure justifies our use of the F1 score in place of the MSE because especially for high F1 score and low MSE, the values are clearly correlated.

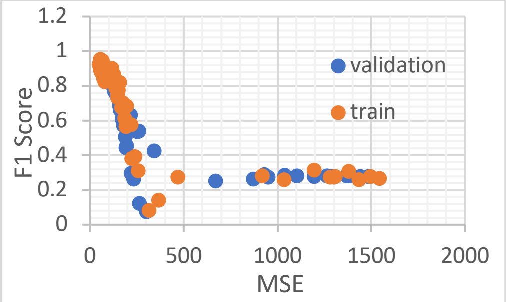


Figure 3: The relationship between the loss function (MSE) and the F1 score for minor identification

In order to identify the optimal values for other hyperparameters, we performed a simple grid search. We tested four different values for the learning rate (1e-2, 1e-3, 1e-4, 1e-5) and the size of the two dense layers (32, 64, 1024, 2048) For each experiment we trained the model for 5 epochs. The results, as shown in Figure 4, show that a learning rate of 1e-3 with a hidden layer size of 1024 produce the highest F1 score. These results are similar to those seen by Smith et al. which we expected since our models are very similar.

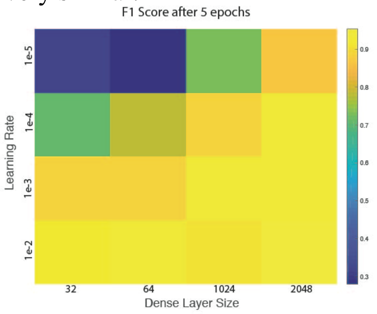


Figure 4: Heatmap showing the results of the hyperparameter experiments. The highest value corresponds to a dense layer size of 1024 and a learning rate of 1e-3

While we recognize that a more comprehensive search, using a random search instead of a grid search, and experimenting with different hyperparameters like dropout rate, and batch size, might have yielded even better results, but we were limited by time constraints because each experiment took approximately 20 minutes to train. After choosing the final parameters we trained the final model for 10 epochs.

1.5. Datasets

Each of the three parts was trained independently on a different dataset. This is largely unavoidable since each of the steps consists of a professionally designed pre-trained model. Faced was trained on the WIDER FACE dataset of nearly 400,000 faces across 32,000 images. This dataset includes a diversity of face scales, poses, occlusions, expressions, and makeup [16]. The gender detector was trained on the Adience dataset which contains 26,580 photos of 2,284 subjects collected from Flickr albums, the photos are labeled in age ranges listed above. Roughly 46% of the images were of males and 52% were of females (2% of the images did not specify gender) [17]. Some data augmentation was done to the original Adience dataset including rotation of the images, lowering the resolution of the image, and darkening and lightening the image. FSGAN was trained on multiple datasets including IDB-C, VGGFace2, CelebA, LFW Parts Labels, Figaro, and FaceForensics++ datasets. We trained our age detector on the cropped and aligned UTKFace dataset [18]. This dataset is comprised of over 23,000 facial photos with 15.4% being below the age of 15 (minors), 52% being male, and a racial distribution of 42.5% white, 19.1% black, 16.8% Indian, 14.5% Asian, and 7.1% other races. We used testing and validation sets of 1,000 images each with the remaining images in the training dataset. Donor faces used for deidentification were also chosen from the cropped and aligned UTKFace dataset [18].

Since all three of the models were trained separately, we created a separate dataset for testing our complete model. Since our model is designed to be used by parents posting pictures on their social media page our dataset is a collection of images, all available on the internet, comprising family pictures, Instagram posts from public accounts, and pictures from news articles. There is a mix of profession and amateur photography.

1.6. End-to-End Testing

Upon completion of our end-to-end system, we performed a test against 100 unlabeled family photos curated from Pinterest and Google Images, comparing the output of our system against a human grader's. For every test photo, we assessed the undetected minor faces, miscategorized minor faces that were labeled as adult faces, and the quality of the swapped face. First, we investigated the accuracy of our age detection model within the overall system, assessing the share of detected faces that were accurately classified as minor faces. Second, we evaluated the end-to-end success rate of the system, which we defined to be the percent of minor faces that were swapped with another face that was both realistic and anonymized well. We evaluated authenticity on a scale from 1 to 3, ranging from faces that were (1) very unrealistic (e.g. cartoonish, missing entire facial features, visible stitching borders) to (3) believable at first glance. We also assessed the anonymization of the photos, ranging from (1) immediately identifiable to (3) unrecognizable. We defined the success rate as the % of minor faces that were detected, classified as minor faces, and swapped with a face that was both realistic and anonymized.

2. Results

Because much of our work was focused specifically on the age detection portion of our network, we evaluated that portion of the network separately from the full end-to-end pipeline. After testing both a binary (actually a two-class softmax) and a continuous output, we decided to use a continuous output for age estimation. Because of this choice we used Mean Squared Error as the loss function of choice for our training. Evaluating on our test dataset we found the Mean Squared Error to be 48.89.

In the output, however, we were most interested in the detection of minors (not true age estimates), so we tuned hyperparameters and ultimately evaluated our model based on the binary decision of whether a face belongs to a minor. Our results based on binary minor selection are shown in Table 1:

Table 1: Results of Age Detector on test dataset based on binary minor selection

Precision:	94.7%
Recall:	86.8%
F1 Score:	90.5%
Accuracy:	97.4%

Overall, we were fairly satisfied with our results, with the accuracy of minor detection reaching 97% and an F1 Score of over 90%.

One interesting phenomenon we observed wile training was that the validation loss was consistently lower than the training loss as shown in figure 5. This is the opposite of what we would expect. We would have to do further analysis on the errors to understand exactly what caused this, but we assume that the validation set contained images that were easier for some reason for the model to interpret.

In order to determine the most important areas of improvement of our system, we evaluated the both the age detection model and the overall system performance against 100 unlabeled test photos (see Appendix 1). The human grader identified 328 total faces, and 190 minor faces in the dataset, whereas the system detected 303 faces and 77 minor faces. Of the 113 missed minor faces, 27 were undetected by the face detection module and 86 were miscategorized as adult faces. The F1 Score for minor detection was 88%. Surprisingly, the precision of minor classification was 100%, meaning that there were no adult faces that were categorized as children's faces. Out of 190 minor faces, only 15 were successfully deidentified with a realistic, anonymized face, meaning an end-to-end success rate of 7.9%.

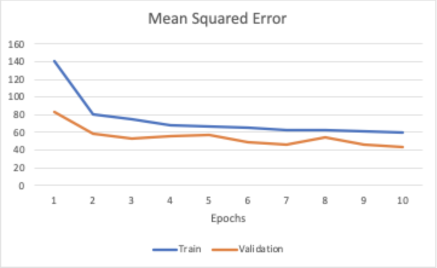


Figure 5: MSE for the training and validation sets shown for each epoch of the final training. Note that the validation dataset has a consistently lower error than the training dataset.

3. Discussion

The end-to-end evaluation of KiDINet suggested that all 3 individual stages stand to improve. Our face detection module in particular tends to miss baby faces. Future work should use transfer learning with more labeled baby face data in order to improve the detection accuracy of these baby faces. This will likely be challenging, given the limited availability of labeled baby face data in public sources. Data augmentation may help with this limitation.

The age detection stage needs the most improvement. 76% of the minor faces missed by the system were misclassified as adults. Given that the system failed to identify a single adult face as a child's face, we believe that the model biases older in its age determination. We believe that this could be improved by further oversampling minor faces in the dataset, or adjusting our loss function during training to track minor detection accuracy instead of the mean squared error of age estimate. We also built our age detection model based on centered and aligned examples, but the real-world inputs fed to our model were not properly centered or aligned. Improving this in the pipeline will help to improve outcomes. Manual error analysis revealed that there were specific circumstances in particular that the model struggled with. These include highly angled faces, faces partially occluded by sunglasses or other objects, and crowded photos with several smaller faces (see Appendix 2). Future work should dig deeper into the model engineering to determine why the ages of these faces are systematically overestimated. Human-grading also revealed to us that using image data alone to evaluate a person's age has a maximum possible performance. Our human grader struggled on several occasions to determine whether or not an individual's face belonged to a minor or a young adult, a phenomenon well-documented in the literature [19]. To achieve optimal performance, the system would need to secure information from sources other than the image data, such as the subject's birthday from social media accounts.

Manual inspection of the final output images of face deidentification yielded disappointing results. Of the 77 detected minor faces, the model only returned 15 faces (19.5%) that were determined to be realistic and anonymized according to the human grader. The most common issues were deep facial discoloration and unusually beady eyes. While these issues could be mitigated by incorporating race into the donor face determination and enhancing the FSGAN model, we posit that face generation might simply be surfeit engineering in this case, when simply blurring a face or swapping with a cartoon baby face or an emoji would do. Future work should collect feedback from prospective parent users to determine if this additional engineering work could be bypassed in an initial product release.

Given the issues highlighted above, our system may not be ready for some time for family photos in which the faces of the subjects are at the center of attention for the image viewer. However, the system could be more effective in applications where individual faces are of lower visual importance, such as news photos and videos.

4. Conclusions

We have been able to demonstrate that deep learning technology makes it possible to protect minors' privacy on the internet without human involvement. KiDINet is the first end-to-end minor face detection and deidentification system, built by connecting together existing deep learning networks with our own age detection neural network. Although our current end-to-end success rate was low, by further tuning our age detection algorithm and making some modifications in our facial deidentification approach, we are confident that a commercially viable product is possible. We are hopeful that the availability of this technology for use in press photography and, social media sharenting will help usher in a new era of children's online privacy.

5. Contributions

We feel that the work was fairly and evenly divided among our group members.

Writing:

We worked together to outline the paper and create a storyboard for the video. While each of us contributed to all parts of the paper, each person focused on the portion of the paper corresponding to the part of the project they performed. With Akhil writing the Related works, the part of the Results detailing the complete KiDINet pipeline, Discussion, Conclusion and Abstract. Isaac wrote the portion of the Results relating to the Age detector, and description of FSGAN, and wrote and recorded the video. Laurel wrote the motivation and the methods sections and made the system diagrams.

Programming:

Akhil: Implemented FACED network, worked with Isaac to develop the completed pipeline, and helped Laurel develop the framework for testing hyperparameters for the age detector and performed the analysis of the completed KiDINet pipeline.

Isaac: Implemented the FSGAN network, worked with Akhil to connect all of the parts and complete the pipeline, created the transfer learning portion of the age detection network. Compiled the final test results for the age detector and added that network to the pipeline.

Laurel: Implemented the original age and gender detectors and implemented the binary and regression versions of the age detection network building on Isaac's transfer learning work. She worked with Akhil to develop the softmax classification version of the age detection network. Worked with Akhil to develop the framework for testing the hyperparameters, and performed the hyperparameter testing, and then compiled those results and performed the final training of the age detection network.

References

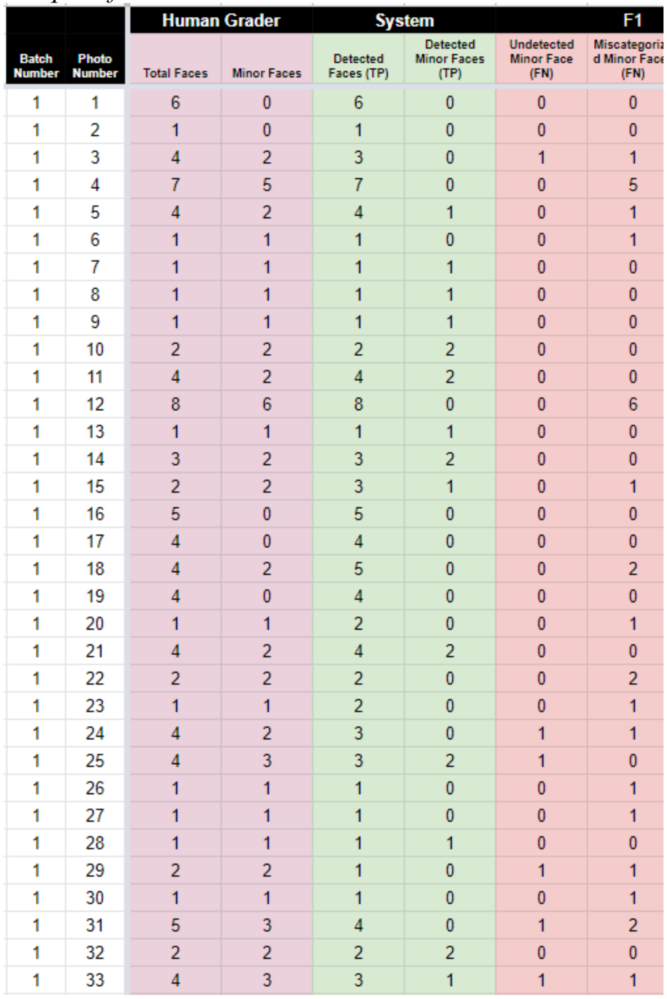
1. Children's Commissioner for England. (2018). Who knows what about me? A children's Commissioner report into the collection and sharing of children's data.
2. Haelle, T. (Writer). (2016). Do Parents Invade Children's Privacy When They Post Photos Online. *Your Health. NPR*
3. Steinberg, S. B. (2016). Sharenting: Children's privacy in the age of social media. *Emory LJ*, *66*, 839.
4. Dubois, A. (2018). Master thesis: Facial recognition using deep neural networks. http://hdl.handle.net/2268.2/4650
5. Al-Shannaq, A. S., & Elrefaei, L. A. (2019). Comprehensive Analysis of the Literature for Age Estimation From Facial Images. *IEEE Access*, *7*, 93229-93249.
6. Levi - Gil Levi and Tal Hassner, *Age and Gender Classification Using Convolutional Neural Networks,* IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015
7. Smith, P., & Chen, C. (2018, December). Transfer Learning with Deep CNNs for Gender Recognition and Age Estimation. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 2564-2571). IEEE.
8. Meden, B., Mallı, R. C., Fabijan, S., Ekenel, H. K., Štruc, V., & Peer, P. (2017). Face deidentification with generative deep neural networks. *IET Signal Processing*, *11*(9), 1046-1054.
9. Li, Y., & Lyu, S. (2019, July). De-identification without losing faces. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security* (pp. 83-88).
10. Itzcovich, I. (2018). FACED. Retrieved June 9, 2020, from https://github.com/iitzco/faced
11. Levi, G., & Hassner, T. (2015). Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 34-42).
12. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition.
13. Serengil, S. I. (2018, August 6). Deep Face Recognition with Keras [Web log post]. Retrieved May 29, 2020, from https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/
14. Nirkin, Y., Keller, Y., & Hassner, T. (2019). Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 7184-7193).
15. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2019). Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*.
16. Yang, S., Juo, P., Loy, C. C., & Tang, X. (2016). WIDER FACE: A Face Detection Benchmark. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
17. E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. Trans. on Inform. Forensics and Security, 9(12), 2014
18. Geralds, J. UTKFace Large Scale Face Dataset. *github.com*
19. Clifford, C. W., Watson, T. L., & White, D. (2018). Two sources of bias explain errors in facial age estimation. *Royal Society open science*, *5*(10), 180841.

Additional References not cited in text

1. Getting started with Tensor Board [Web log post]. (2020, May 20). Retrieved May 29, 2020, from https://www.tensorflow.org/tensorboard/get_started
2. Brownlee, J. (2020, May 29). How to Perform Face Recognition With VGGFace2 in Keras. Retrieved June 09, 2020, from https://machinelearningmastery.com/how-to-perform-face-recognition-with-vggface2-convolutional-neural-network-in-keras/
3. Transfer learning with a pretrained ConvNet: TensorFlow Core. (n.d.). Retrieved June 09, 2020, from https://www.tensorflow.org/tutorials/images/transfer_learning
4. Load images: TensorFlow Core. (n.d.). Retrieved June 09, 2020, from https://www.tensorflow.org/tutorials/load_data/images
5. Nirkin, Y. (2020, May 16). Fsgan github. Retrieved June 09, 2020, from https://github.com/YuvalNirkin/fsgan
6. DataFlair Team. (2020, January 07). Interesting Python Project of Gender and Age Detection with OpenCV. Retrieved June 09, 2020, from https://data-flair.training/blogs/python-project-gender-age-detection/

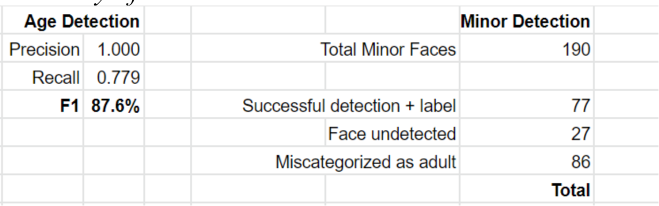
Appendix 1: End-To-End System Evaluation

Sample of Human Grader Evaluation



		Human Grader		System			F1
Batch Number	Photo Number	Total Faces	Minor Faces	Detected Faces (TP)	Detected Minor Faces (TP)	Undetected Minor Face (FN)	Miscategoriz d Minor Face (FN)
1	1	6	0	6	0	0	0
1	2	1	0	1	0	0	0
1	3	4	2	3	0	1	1
1	4	7	5	7	0	0	5
1	5	4	2	4	1	0	1
1	6	1	1	1	0	0	1
1	7	1	1	1	1	0	0
1	8	1	1	1	1	0	0
1	9	1	1	1	1	0	0
1	10	2	2	2	2	0	0
1	11	4	2	4	2	0	0
1	12	8	6	8	0	0	6
1	13	1	1	1	1	0	0
1	14	3	2	3	2	0	0
1	15	2	2	3	1	0	1
1	16	5	0	5	0	0	0
1	17	4	0	4	0	0	0
1	18	4	2	5	0	0	2
1	19	4	0	4	0	0	0
1	20	1	1	2	0	0	1
1	21	4	2	4	2	0	0
1	22	2	2	2	0	0	2
1	23	1	1	2	0	0	1
1	24	4	2	3	0	1	1
1	25	4	3	3	2	1	0
1	26	1	1	1	0	0	1
1	27	1	1	1	0	0	1
1	28	1	1	1	1	0	0
1	29	2	2	1	0	1	1
1	30	1	1	1	0	0	1
1	31	5	3	4	0	1	2
1	32	2	2	2	2	0	0
1	33	4	3	3	1	1	1

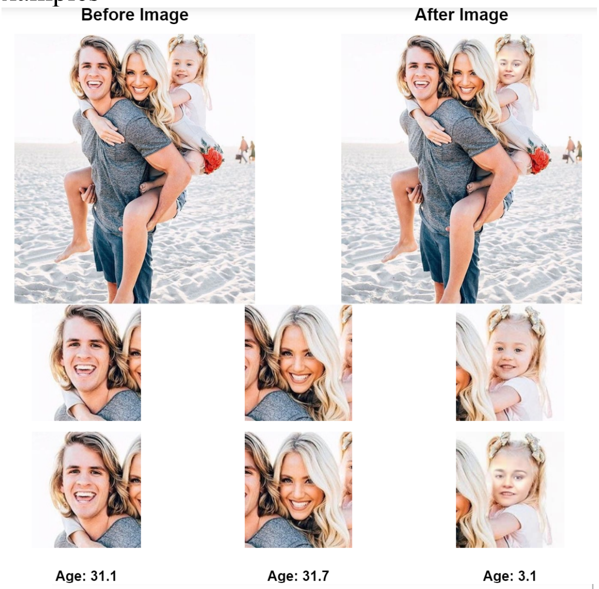
Summary of Results

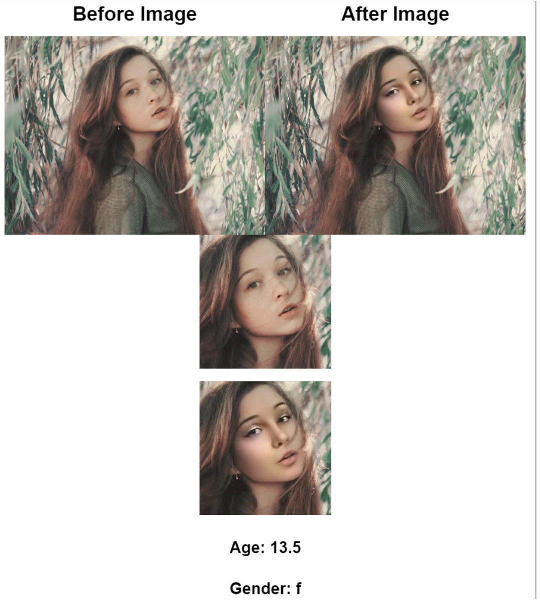


Age Detection			Minor Detection
Precision	1.000	Total Minor Faces	190
Recall	0.779		
F1	**87.6%**	Successful detection + label	77
		Face undetected	27
		Miscategorized as adult	86
			Total

Appendix 2: KiDi Output Examples

Successful Minor Face Detection and Deidentification Examples





Unrealistic Minor Face Deidentification Examples