Coty Ma
261118266
MAIS 202: Accelerated Introduction to ML
Deliverable 3

1. Final Training Results

Due to the fact that I did not produce results before the last deliverable, this is the first result of the project. Instead of creating a custom CNN, I ended up downloading a pre-made model from pytorch and retraining it on my own dataset. This is because the CNN I tried to code myself ended up producing negative coordinates. As such, I took the pre-made Faster RCNN Mobilenet V3 from pytorch. This is because it takes up less memory from the GPU, thus allowing me to use the 15 GB RAM GPU from Google Colab. It was already trained to identify 80 different classes of objects, but I only needed to detect 2: faces and the background. As such, I changed the number of classes and number of in_features into the box predictor of the model. Finally, I trained it on 19 epochs in batches of 32 - also due to the limited memory of the GPU.

Finally, due to the fact that my project is only composed of object detection, I measured the accuracy of my model with the IoU (intersection over Union). The only downside to this measure is that if the model blurs the entire image, for example, then it would obtain a perfect score since it would necessarily cover the area of the actual boundary boxes. To mitigate this, I added filter conditions to my model as well as another variable: $\frac{Total\ Area\ of\ Actual\ Boxes}{Total\ Area\ of\ Predicted\ \ Boxes}$. The lower this variable is, the more it'll show that the model is covering far more area than what is needed. As for the filtering conditions, I removed boxes that cover more than 60% of the image. I also removed boxes that, statistically, would definitely not cover faces such as boxes that cover less than 20% of the image (note that I trained my model to ignore small faces), as well as boxes with an abnormally large/small width to height ratio ($0.5 < w/h < 5$), as those thin boxes also do not properly cover faces. The boxes are allowed to be wider rather than longer since the model combines the boxes of faces that are close to each other into a single large rectangle. Note that both the variable and the IoU take into account that multiple predicted boxes may overlap, and I accounted for that by not including overlap areas between those. In addition, if the IoU is low, I can use the new variable to tell if it's because the model does not intersect with the target boxes or if the predicted boxes have an abnormally large area.

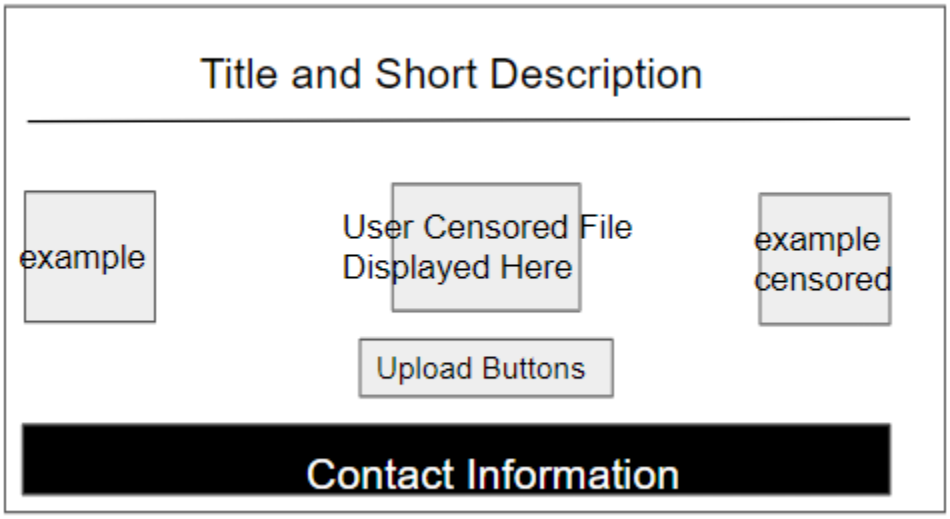
Ultimately, the average IoU of the model was 0.15, and the average value of the new variable was 0.21. Nevertheless, the goal of my project was strictly to censor faces. So if we disregard the fact that the model overshoots the size of the box, the percent of times the model censors all the faces in images is 0.65. I would like to add that I would have further trained the model if Google did not recognize my IP address and limit GPU usage to below 6 hours per new account.

2. Final Demonstration Proposal

For the final demonstration, I have already prepared a Web-App with Flask that introduces the project, how it works (in terms of functionality), and a basic UI that allows users to upload an image or a video file.

The layout is very simple. The title of the project and a brief description is on top. Two demonstration images are on the sides of the page, and the bottom contains advice on using the app and contact information. But most importantly, there's a black box in the center of the page, and when the user uploads a file, the black box becomes the edited, i.e. censored, version of the image or video.

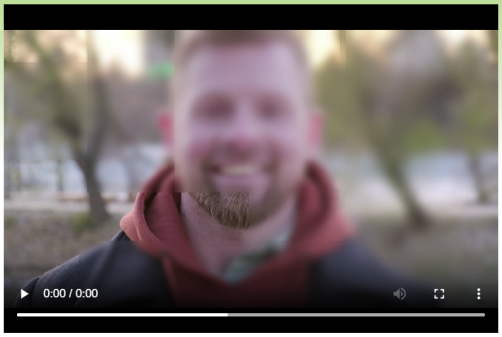
The complete web app should already be on Github. Also, since my computer does not have a GPU built-in, the processing time for videos lasts a few minutes due to using a CPU. I believe the same problem would not occur if I was using a GPU instead.

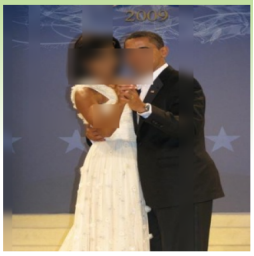


>

Censor Cam

Upload an Image or Video, and you'll receive it back, but with your face censored!





Choose File No file chosen Upload

Due to production limitations, the accuracy may vary based on the positioning of the faces.

Contact Info:
Coty Ma
cotyli00@gmail.com
Github Link: https://github.com/Student0544/AI-Face-Blurring

Inspiration:
Characterized Crowd Instance-level Human Parsing (CCIHP)
https://kalisteo.cea.fr/wp-content/uploads/2021/09/README.ht