STYF: Stop Touching Your Face, A Real-Time YOLOv3 Object Detection Model for Detecting Face Touches

Abhinav Gupta
IIIT, Hyderabad
Centre for Visual Information Technology
abhinav.g@students.iiit.ac.in

P. J. Narayanan
IIIT, Hyderabad
Centre for Visual Information Technology
pjn@iiit.ac.in

Abstract

The World Health Organisation (WHO) proposes many preventive techniques to stay safe from the ongoing coronavirus pandemic, especially to avoid touching our faces in order to decrease the probability of contracting the virus. However, touching one's face is a very involuntary and habitual action. We propose a 'Stop Touching Your Face' YOLOv3 object detection model trained on a custom dataset of 'Face-Hands' images that would detect a hand touching the face in real time and alert the user about the imminent danger.

1. Introduction

We are living in unprecedented times indeed. Deadly pandemics are often once-in-a-lifetime experiences, but when they do happen, they tend to cause havoc and kill thousands, even millions of people. A century ago, it was the 'Spanish flu' of 1918 and today, it is the COVID-19. Scientists are yet to discover an effective drug or medication against the virus.

The current pandemic has truly changed the face of this planet, infecting and disrupting millions of lives and causing us to reflect and contemplate about mankind's bad environmental practices. Cases of the virus have been growing exponentially and to 'flatten the curve', the countries are adopting the age-old teaching of 'prevention is better than cure', as there still is no 'cure' to the virus. Prevention is in terms of complete lockdowns, travel bans, social distancing and so on, causing everyone to stay at home, causing the daily life schedule to go astray. Because as of now, while there is no 'cure', the only cure is prevention itself.

The World Health Organisation has proposed some guidelines for people all over the world to stay safe from the deadly virus. Some of them are:

- To stay home as much as feasible.
- To keep a distance from anyone coughing/sneezing.
- To wash hands often with soap and water.
- To not touch one's eyes, nose or face!

There is an exigent need for all of mankind to follow these guidelines strictly in order to remain safe and protected. Of the above guidelines, one of the very crucial ones is to not touch one's face. Even if humans were to shake hands or touch contaminated surfaces, they would not fall sick unless they were to touch their faces with these contaminated hands. Unfortunately, touching our faces with our hands is something very involuntary and habitual. Be it scratching our eyes or nose, holding our chin while thinking or using our palms as support for our faces. We propose a YOLOv3 object detection model, trained on the 'Face-Hands' dataset which accurately detects when a person touches his or her face and issues a warning for the same.

2. Literature Review

There is currently a colossal amount of ongoing research in all the countries for tackling the various problems that have arisen due to the coronavirus, apart from the search of drugs and vaccines. The use of Computer Vision and Deep Learning is abundant, especially in medical vision where scientists are analysing images of lungs using CNN's for diagnosing the virus. Wearing masks has become absolutely mandatory in countries all over the world, to the extent that people are being fined if they are found without any mask. CCTV's are being installed at airports and other public places to detect people without masks using object detection and face recognition techniques.

The WHO urges people to avoid touching their faces as that is one of the major causes of infection. However, the psychological make-up of mankind is such that we are often tempted to touch our faces. It is very hard to refrain from touching one's face, and it is the pressing need of the day to

Figure 1. The Face-Hands Dataset, which comprises of 160 images of a human face. 70 of them are just faces in different orientations. 70 more images have the palm of the hand touching the face at different locations. 20 images are used exclusively for testing.

Figure 2. Manually labelling the images using Microsoft's Visual Object Tagging Tool (VoTT), an interactive and user-friendly tool for annotations. We use only two classes: 'face' and 'hand' to label the images.

avoid doing so. This problem can be solved with the nagging presence of something which would keep warning us if we were to touch our faces. This task can be accomplished using computer vision and machine learning, by using an object detection model which would be trained to identify faces and the palms touching them.

You only look once (YOLO) is a state-of-the-art real-time object detection system [4]. It uses a very different approach as compared to the other object detection networks. A single neural network is applied to the full image, which divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm "only looks once" at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes. YOLOv3, which is the third optimised version [5], is extremely fast and has caught the attention of computer vision researchers all over the world for object detection purposes.

3. Data Collection and Augmentation

3.1. Face-Hands: A Custom Dataset

We curated a custom dataset called 'Face-Hands' from scratch with 160 images of human faces using a computer webcam, as shown in Figure 1. About 70 images are 'face-only' images, looking around in different orientations. We are interested in the action of the hand touching the face, so it may seem unnecessary at first to collect only face images. However, this is a necessary step because the model should also be able to accurately detect the human face even when it is not being touched. The confidence score of the detection of a 'face' significantly improved after we did this, as evident from Table 1.

Method	Average Confident Score
Not using 'face-only' images	0.38
Using 'face-only' images	0.61
Augmenting the Data	0.74

Table 1. Results on improving and augmenting the dataset. Using face-only images improves the confidence score of the YOLOv3 model in detecting faces. Data Augmentation proves to be effective as it increases the confidence score of the network even more.

There are 70 images which include the palm of the hand touching the face in different ways, including covering one's mouth while yawning or coughing, scratching one's cheeks or eyes and touching the chin while thinking. The remaining 20 images are used for testing the network. Hence, we successfully curated the Face-Hands dataset which would help in feeding the YOLO network with good training examples.

3.2. Data Annotation

We use Microsoft's Visual Object Tagging Tool (VoTT) to manually label the images of the Face-Hands dataset. All the 140 images were annotated with two classes: 'face' and 'hand'. We manually drew bounding boxes around the faces and hands of the images and labelled them appropriately. The annotations were converted to the popular .h5py format for easier comprehension of the data.

3.3. Data Augmentation

Initially, we hadn't performed any data augmentation due to which our model was not performing very well on images that were not as bright and taken in dimly lit rooms. The goal of any machine learning model is to generalise, that is, perform well even in unseen situations. Data augmentation is one of the very popular methods of regularisation that helps in avoiding overfitting. We augmented our data in order to improve its performance and it worked well, as evident from Table 1.

Figure 3. Colour Jitter: One of the simplest and most effective methods of data augmentation. We randomise the contrast and brightness of the images, to ensure the model works well in dimly lit and brightly illuminated settings as well. The augmentation improves the confidence score of the network, as shown in Table 1.

4. Training the Network

We chose YOLOv3 for the task as it is extremely fast, accurate and can detect objects in real-time. It also makes predictions with a single network evaluation unlike systems like R-CNN [3] which require thousands for a single image. This makes it extremely fast, 100 times faster than Fast R-CNN [2].

4.1. Transfer Learning

Our Face-Hands Dataset comprises of only 140 images for training, which seem very less. It's often said that a lot more data is required (even after data augmentation) for training deep neural networks for images. But we are still able to obtain a decently high confidence score and accuracy by fine-tuning a pre-trained model of YOLOv3 on the ImageNet dataset [1]. ImageNet contains many images of people and faces, so those weights definitely come in handy for our purpose as well. We start off with the pre-trained ImageNet weights, apply transfer learning and fine-tune the network for our relatively smaller Face-Hands Dataset. This is a very common practice today, even Fast RCNN uses a CNN that is pretrained on ImageNet.

4.2. Hyperparameter Tuning

The model was trained for a total of 172 epochs after which the loss function successfully converged. YOLO uses the Adam optimisation algorithm for optimising over the loss function. The learning rate was initially set to 5e-4 and was reduced at particular intervals later on. We do this 'step decay' to ensure that our optimiser doesn't get stuck on a lesser optimum loss. By reducing the learning rate, we increase the probability of reaching a more optimum loss. This is very commonly followed in ResNets, where the learning rate is reduced at few fixed points. While training our model, we reduced the learning rate thrice - at epochs 121, 165 and 172. If we hadn't done so, the network would have stopped training at epoch 121 itself, thereby resulting in less optimum weights.

4.3. Testing

The YOLO network detects objects with a 'confidence score', which tells us how confident the network is about its prediction. After training and fine-tuning the model for the Face-Hands dataset, it was tested out on the testing data. It was able to successfully detect the faces and hands, albeit with a lower confidence score. But when we collected more 'face-only' images, that is, images with only the face and not the hand, the accuracy improved as explained earlier. This is essential because if we leave such data out, the network learns faces with the hands touching them. Hence, if we test on a 'face-only' image, it would certainly detect the face, but with a lower confidence score. We also performed data augmentation and our confidence scores improved, as shown in Table 1.

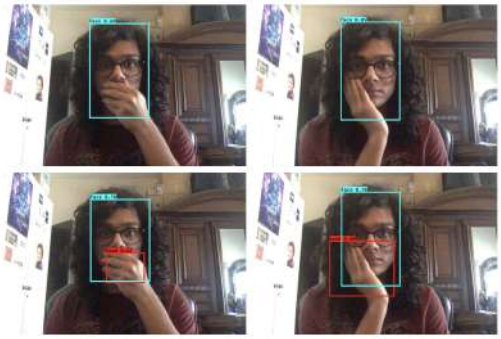

Figure 4. The top row shows the results of the model after training for 60 epochs on 60 images. The results were very poor and the network was unable to detect the hands in the images. The results massively improved when we augmented the data and used more 'face-only' images and trained it for more number of epochs. The second row shows the improved results, where the hand is now being successfully detected.

The network would also sometimes detects multiple instances of the same object, even though the max suppression algorithm is a key feature of YOLO. In such scenarios, we simply consider only one bounding box, the one with the highest confidence score.

Initially, we only trained for around 60 epochs, and only for around 60 images in the dataset. The results were very poor and the model was unable to even detect the hand and the face. So we increased the size of the dataset and collected more images (including more 'face-only' images), performed data augmentation and trained for 172 epochs due to which the results improved by a huge amount.

4.4. Bounding Box Collision

In order to check if the hand is touching the face, we simply check whether the two bounding boxes are overlapping with each other. If there is an overlap, it means that the hand is getting too close to the face and we must warn the user about the same. If the boxes collide, the code plays a small audio clip which screams 'stop touching your face!' and the user is warned about the danger.

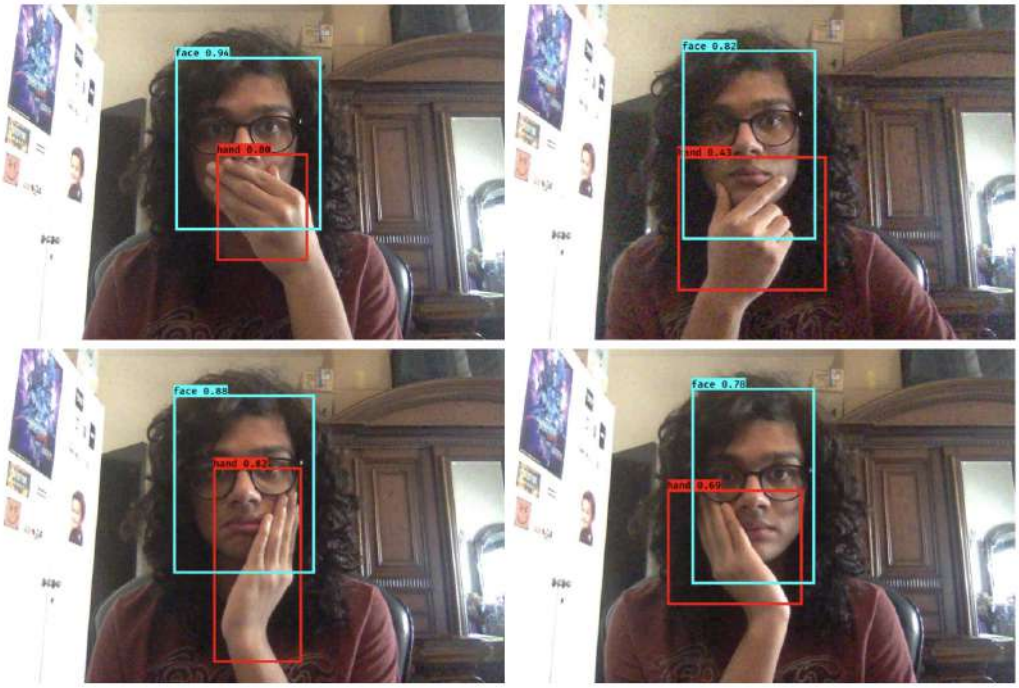

Figure 5. The results of our YOLOv3 model on the testing set, after training on the augmented and improved dataset. We notice that the face is being detected with a high confidence score of above 0.8. The hand, regardless of its orientation, is being detected as well, with a slightly lower confidence score. As soon as the bounding boxes collide, a verbal warning is thrown at the user to stop touching his or her face. Here is a video, showing our final results from a live webcam feed: link to video

4.5. Webcam Integration

The best part about YOLO is that it is real-time! It only takes around 0.7 seconds to test per image, that too on a CPU! In order for our system to work, it is essential for it to be live, because we would need to continuously monitor the user's actions and warn him or her immediately if the hand gets too close to the face. We wrote a python script that would capture images from the computer's webcam and would then use the trained model for inference. Hence, the network can be run in real time and can be tested on images from the webcam's live feed! Here is a video, showing our final results from a live webcam feed: link to video

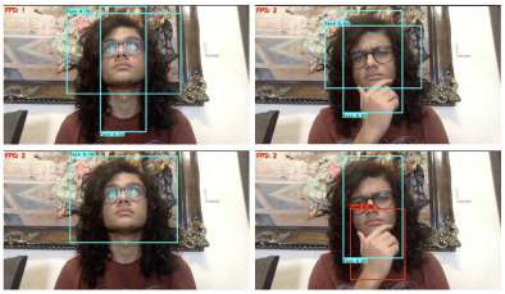

Figure 6. Initially, the network was detecting multiple instances of the same object, as shown in the top row. Our model considers only one bounding box, the one with the highest confidence score, as shown in the second row.

4.6. Performance on Other Humans

Our model performs decently well for the face it was trained for. But our idea would not be very effective if it didn't work for other human faces. Moreover, there are other factors that could affect performance like the presence of spectacles or the length of the hair. We tested out the model in such scenarios as well and it was able to detect and successfully warn the user, albeit with a slightly lower confidence score.

4.7. Conclusion and Future Work

Our YOLOv3 object detection model is thus able to successfully detect if the user touches his or her face through a live camera feed. The network was fine-tuned on the custom Face-Hands dataset using the pre-trained weights from ImageNet. The network checks for any overlap between the bounding boxes and verbally warns the user if the hand gets close to the face. Our model works well with data that is similar to the data in the Face-Hands dataset, but it still

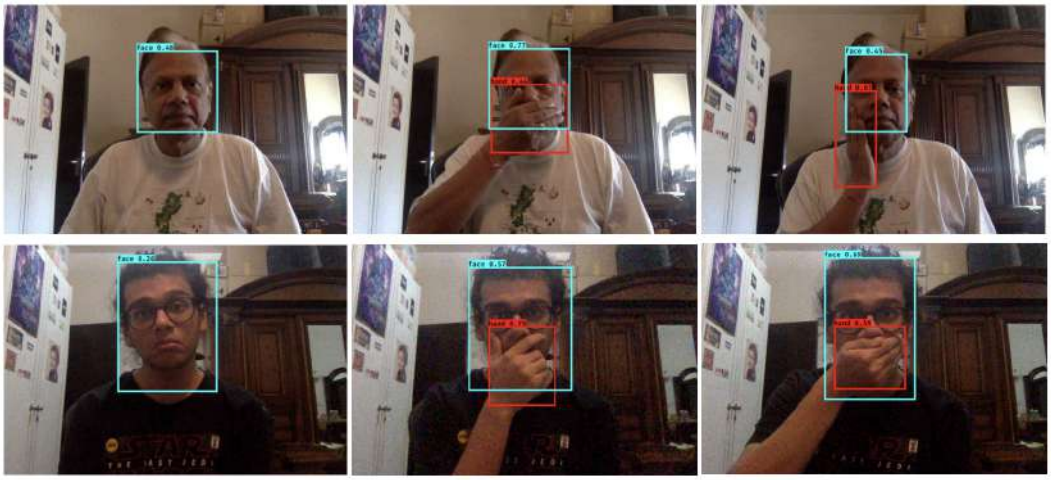

Figure 7. The first row shows the results of the model on another human face which it was not trained on. It is able to successfully detect the hand and the face and warn the user when the boxes overlap. The second row shows the result of the model on the same human it was trained for, but with a different hairstyle. The model is still able to accurately detect the face and the hand.

needs to be improved further to make it more versatile for different faces. We would need to train the network on atleast 20-30 more humans to be able to generalise for all of mankind.

Another problem with this network is that it would shoot out a warning even if the user just puts his or her hand in front of the camera. Even though the hand is not touching the face, the bounding boxes would still overlap causing it to scream 'stop touching your face'. One way to tackle such a problem would be to also compute the angle between the boxes. In order for the network to work flawlessly for any human, we would need to train it for more amount of data and for visibly different kind of people to account for different skin colours, facial features and hand movements.

Often, when we sit near our computers while doing an assignment or watching a Youtube tutorial, we tend to inadvertently touch our face. Some have a bad habit of touching their chin while trying to think, which is unsafe in these times. Our 'STYF' model is a really useful application of computer vision as even touching one's own face can prove to be very dangerous in these troubled times.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. F. Li. Imagenet: a large-scale hierarchical image database. pages 248–255, 06 2009. 3

[2] R. Girshick. Fast r-cnn, 2015. 3

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013. 3

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2015. 2

[5] J. Redmon and A. Farhadi. Yolov3: An incremental improvement, 2018. 2