Facial Liveness Testing For The Web

Ryan Collins

Department of Computer Science, Durham University

Introduction

Currently, username and password authentication is commonplace throughout the web. However, this method has serious security concerns, being easily broken through dictionary attacks, or shoulder surfing. Facial recognition authentication doesn't have either of these problems, but introduces the risk of stolen identity through face spoofing, more specifically expressed in a quotation from Adam Schwartz, a lawyer with the Electronic Frontier Foundation speaking to NPR: "We can change our bank account numbers, we can even change our names, but we cannot change our faces. And once the information is out there, it could be misused". In order to reduce the risk of stolen identities, a system to detect potential spoofing attempts is needed. Providing facial liveness systems work correctly, the ability for criminals to steal people's faces would be reduced, making facial recognition more ideal for authentication on the web. The next step to making facial recognition more secure and mainstream, the implementation of a facial liveness service is necessary: that way developers won't need to worry about implementing their own liveness tests and can instead focus on the process of building their software. While there are different measures of detecting liveness, each method is specialized towards defending against a given attack. The aim of this project is to understand the existing liveness detection methods, which type of attack they aim to prevent, and how effective they are. Once this has been achieved, the aim shall be to bring each of these methods together, hopefully improving the effectiveness of such a system by incorporating multiple methods.
In this context, we propose a novel new 3D-based liveness test, based on a two part approach: (i) VRN based 3D reconstruction (ii) VoxNet based 3D classification. We provide a promising new liveness test using ResNet based networks. We also adapt an existing Image Quality Assessment test for use with whole image classification, rather than a face based method.

Image Quality Assessment

One method of testing liveness is to analyse the quality of the image. A spoofed image has noticable quality differences compared to a real image of a person's face, and therefore quantifying these differences is the first step towards testing liveness. The paper [**?**] proposed a method specifically for testing quality on a user's face, but this can be extended to work with any image.
24 image quality metrics were calculated per image. Some metrics compared pixelwise error, signal to noise ratio, correlation and frequency analysis between an image I, and the Gaussian Blurred image $\boldsymbol{I'}$. With 24 values classifying the quality of a given image, these were then fed into a classifier. The classifier of choice here is Linear Discriminant Analysis (LDA). Some of the metrics utilised were fairly rare, and as such a custom library was created from existing Python 2 scripts. The original scripts can be found in [**?**]. These pieces of Python 2 code were converted to Python 3, with optimisations being created to be run in library form, and then a Python module was created to allow interfacing to the library wtih ease. This library is available on GitHub. [**?**] Other metrics were implemented using a mixture of OpenCV, Numpy and Scikit image, most of them with custom code.
The accuracy of this liveness test was fairly high, yielding an 89% top-1 accuracy score on the Replay-Attack test dataset. When the confusion matrix was analysed, a high number of true positives and true negatives were yielded. However, when the liveness test incorrectly predicted the realness value, often it would yield false positives which is problematic in a security conscious system, but this could be improved in further training, or sensor fusion.

Residual Networks for Facial Liveness

The above metric analyses the overall image quality, but another method of understanding liveness is to consider facial structure. A facial extractor can be used to produce a 224x224 image of the isolated face, by using either a pretrained CNN or a HoG based method. In testing, the CNN method produced far greater accuracy and also yielded speed benefits. Given this 224x224 image, a pretrained ResNet50 model was used, with all but the last layer being frozen to allow for training. This minimized the number of parameters, and allowed the final component: the dense layers, to classify the liveness. Between the pretrained ResNet50 model and the Dense layers was a flatten layer. Batch Normalization was used to improve network performance. The accuracy of this metric was adequate, yielding a 71% top-1 accuracy score on the Replay-Attack test dataset. This shows that the model itself has promise, but it can certainly be improved. When looking at the confusion matrix, the model was excellent at identifying spoofing attacks, but it would be cautious about classifying a specific face as real. The common errors yielded would be false negatives, which while potentially annoying for an end user, yields a better security system. This accuracy figure could be improved by

Evaluation

Quality Test

The Quality Test performed well, and is suitable for a web based liveness service due to the high 87% accuracy achieved. True positives and true negatives (that is, correct predictions for both real and fake) were high which means that the classifier is classifying correctly. However, the high false positive rate shows a slight cause for concern, since the model in 12.5% of cases classified a fake image as real, which isn't ideal for our security focused solution. This could potentially be solved by adjusting output variable of the classifier (using a figure representing 'fakeness' rather than realness). This would hopefully lead to more false negatives, which are inconvenient but more secure for the system. In terms of computational performance, a 1.40 second time for classifying a single image is within the limits of the expected 2 seconds, and could be improved further with parallel based methods, and not relying on libsvm for the BIQI metric.

2D CNN

The 2D CNN Test performed adequately. While the accuracy was lower than expected (at 71%), the classifier itself still performed better than random. Furthermore, the model was very good at classifying true negatives, with a total percentage of 71.5% being true negatives. While the model itself had a higher than expected false negative percentage, this is just inconvenient for the user rather than a security problem. The model showed it could classify true positives, but this figure was fairly low. This could be improved by improving the training process: ensuring each input image has a correctly identified face (as some images without a detectable face would have been left to classify the entire image, just resized to the input size). The less noise in the input data, the better the potential results in the future. Furthermore, using a larger ResNet model, such as ResNet-101 could lead to better results. In terms of performance, classifying an image is very quick, but the time taken to load the model is what took the most time (due to memory needing to be allocated and written to). In production, providing a model was preloaded and ready to accept input, the computation time of this metric would be very fast, and therefore be ideal for inclusion in a liveness web service.

3D VoxNet Liveness Test

As seen from the results, this method of liveness test isn't feasible. With extra training, while the accuracy of the model could be improved, the real time memory requirements don't seem feasible. Based on the VoxNet paper, the accuracy achieved with a 32x32 VoxNet classifier on the SUOD dataset (a 3D dataset of objects) is 69%. While this accuracy could be justified with reasonable computational and memory characteristics, this further proves that this method isn't feasible in the current state. In the future, a new approach of detecting 3D attacks is necessary.

Conclusion

This project showed that creating a facial liveness service for the web is a feasible idea, and performs fairly well for 2D attacks, with adequate accuracy and computational requirements. The image quality liveness test is accurate and fast, while the ResNet based method is a feasible idea and performs adequately, and could be improved further to improve the accuracy. For 3D attacks however, the proposed VoxNet based model performed badly and is not recommended for inclusion in a liveness test web service.