Facial Liveness Testing: For The Web

Student Name: Ryan Collins
Supervisor Name: Prof A. Krokhin

Submitted as part of the degree of MEng Computer Science to the

Board of Examiners in the Department of Computer Sciences, Durham University
April 4, 2019

Abstract —

Context TODO

Aims

- Verify the results of the Image Quality Assessment test.
- Assess the outcome Convolutional Neural Networks on classifying real/spoofed images.
- Design and implement a new 3D based liveness test, aimed to prevent mask attacks.
- Determine the outcome of fusing the three above methods together, and how successful this is.

Method

- The image quality assessment test was implemented in Python
- A CNN based

Results TODO

Conclusions TODO

Keywords — Facial liveness, convolutional neural networks, image quality metrics

I INTRODUCTION

Currently, username and password authentication is commonplace throughout the web. However, username and password based authentication systems have a number of problems. Some common passwords can be broken using dictionary attacks, especially if they consist partially or entirely of a word in a standard dictionary. Furthermore, the process of shoulder surfing is possible (watching out for someone's password, and how they type it).

While there are different measures of detecting liveness, each method is specialised 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 encorporating 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 also confirm the success of the Image Quality Assessment method for Facial Liveness, and provide an improve

II RELATED WORK

As defined in [7], the types of face spoofing attacks can be described under three sections: Photo Attack, Video Attack and Mask Attack.

A 2D Spoofing Attacks

Photo and Video Attacks are both 2D spoofing attacks, which involve using a previously retrieved photo/video, and holding it in front of a camera. In the case of photo attacks, a single photo is used, where in video, some video would be played back on a screen. [7].

With video-based facial recognition systems, motions of some form can be used to determine whether the person is real or spoofed, such as blinking, head movement and others. In the method defined in [2], structure from motion was used on the video to produce a 3D model of a user, with the depth channel being used to determine whether a person is real, or whether it's simply an image. They also extended this by fusing this method with audio verification. The fusion of multiple methods provides greater reliablity. However, while SFM works with video, it doesn't work with a single image, and it also doesn't work if a video with little motion is provided. This fusion was completed using a Bayesian Network

While motion based methods are video-only, quality based methods are useful for both videos and images (either by extracting key video frames or using all video frames and combining the results).

While there are various quality metrics that have been used, combining a large number of them can yield some increased accuracy. By combining 25 different metrics, , yielding the resulting metric values into a large vector, and using that as input to a classifier (an LDA), this yields fairly high accuracy. [4]. This is an example of combining many items to yield better results. While each metric on its own isn't that great, using them all together yields better results.

Recently, deep learning based approaches have been applied to facial liveness (both video and image based).

In particular, Convolutional Neural Networks are a key approach to this to learn features (e.g. texture based methods). Due to the existing datasets available, training CNNs has been difficult due to lack of data where overfitting has been common. The method proposed in [8] uses CaffeNet, inputting both the full image along with the isolated face. The output yielded general texture differences, as well as specific facial texture differences. Another interesting idea proposed in this paper is the fusion of two algorithms together to produce an outcome, therefore reducing the false reject rate.

A.1 General 2D image classification models

Outside of the facial liveness field, image classification on the imagenet dataset has proven popular and yielded some fairly good results.

AlexNet One of the initial models was AlexNet, which has 5 convolutional layers (with some max pooling layers), and two globally connected layers. This model was used to classify 1.3 million high resolution images into 1000 classes. [6] Since the AlexNet paper was written, newer methods have been built that performed better.

VGG16 Network The VGG16 model improved AlexNet by replacing larger filters by more smaller filters one after another. However, VGG requires a high amount of computational power, something that's not easily deployable on a real-life system due to high time and space requirements.

Cite VGG lack of deployability

GoogLeNet Inception GoogLeNet Inception is an improved module that approximates a space Convolutional Network with a normal dense construction. Since a small number of neurons is effective, computational requirements are kept small.

Cite GoogLeNet Inception

Residual Networks Another key problem with deep convolutional networks on Image Classification is the vanishing gradient problem, where early layers have very small gradients during the training process and are therefore much more difficult to train. Residual Networks avoid this problem by allowing a direct path in links between the input and output of a building block. . The overall outcome is far better accuracy than VGG and GoogLeNet while being more efficent than VGG in terms of computational power needed. While these aren't directly associated with facial liveness, the nature of image classification is fairly similar to facial liveness (since the image is simply a classifier with two outputs instead of 1000).

Explain Resnets better

Cite Residual Networks

A.2 Datasets

While models exist, in order to test these models data is needed. One of the most common and earliest dataset for facial liveness is the NUAA dataset, which consists of photos of 15 subjects, with faked photos (both flat and warped) being placed in front of the camera. [9]

In 2012, the Replay-Attack dataset was first released , which consists of 1,300 video clips of both photo and video attacks. Each set of videos/images are taken under different illumination conditions, and various different attack methods were collected: printed photo, low resolution and high resolution screens with both photos and videos being displayed. [1]

A.3 Temporal-based Liveness Tests

These liveness tests require a video input, rather than an image. Rather than looking directly at an image, they mostly look at the differences between the images in a video.

Add Eye tracking source

Add face movement source

However, one drawback of temporal based liveness tests is that they require video input, which is often more computationally intensive than standard image input. Furthermore, over a network video input would require far greater network bandwidth.

B 3D Spoofing Attacks

Mask Attacks are a 3D spoofing attack, which involve creating a 3D mask of someone and wearing it. [7] These are much less prevalent, but with 3D printing becoming more mainstream, this could potentially get more prevalent in the future.

In 2013, the Mask Attack Dataset (MAD dataset) was released. [3]

Improve this section, explaining 3D based measures, more about the MAD dataset and attacks, etc.

III SOLUTION

A Image Quality Assessment based liveness test

For 2D spoofing attacks, spoofed images are typically lower quality than the real images, and thus by measuring the image quality one can train a classifier to detect real and spoofed images respectively.

The method used, based on the work of **(author?)** [4], implements 24 different metrics with varying differences, and produces a vector for each image. Initially, classification was done using a Support Vector Machine (SVM), but after experimentation this proved to be fairly unreliable (yielding 70% accuracy on the test set). The classifier was later changed to use Linear Discriminant Analysis (LDA) which yielded a much improved accuracy (96% accuracy on the test set).

TODO: give more accuracy figures of accuracy here, I can't remember the exact numbers

. A visual explanation of the method can be seen in Figure A.

Add more information here about each metric, the PyVideoQuality manual work that needed doing, and any custom code

B Residual Network based 2D liveness test

Recently, 2D convolutional neural networks have had great success in image classification tasks. Therefore, it might be possible to train a residual neural network (resnet) to classify for facial liveness tasks.

In order to simplify the process of training, an existing resnet model (ResNet50) was used, with only the final convolutional layer being set to trainable. This is because the initial convolutional layers contain the standard features contained within images, while the final one learns bundles of features. Internal feed forward activations use relu, while the external output uses the sigmoid activation function

While the Adam optimiser was initially used, this was later changed to Standard Gradient Descent, with a learning rate of 0.0001 and momentum of 0.9. While using Adam, the outcome of the results was a model that didn't generalise that well, and research found in [10] stated that SGD would perform better. This in fact was true, and yielded an improvement.

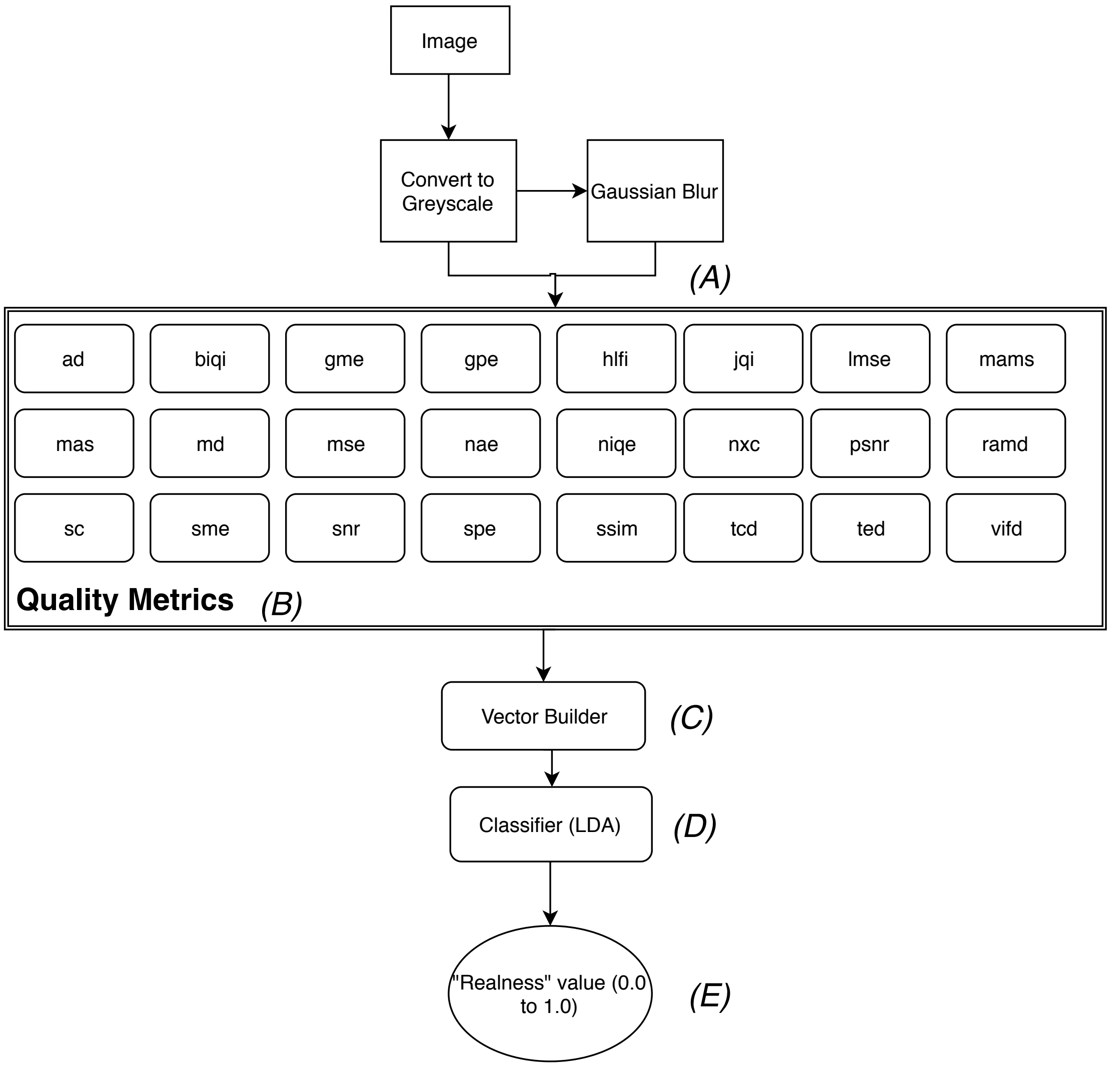


Figure 1: The architecture of the image quality liveness test. (A) The greyscale copy of the image, and a blurred copy of the image are input into each of the metrics. (B) The metrics are individually calculated, and a single value output from them. (C) These values build a 1D vector. (D) They are classified using an LDA classifier. (E) The realness value is 1.0 for real, and 0.0 for fake, or in between.

Initially, the entire image was fed directly into the Residual Network, but this yielded fairly poor performance and generalisation. As a result, a HoG based face detector was used to find the largest bounding box in an image (of a person's face), and crop the image around this face structure. The HoG detector was initially used due to performance benefits, since a neural network based face detector would require slightly more processing power, and therefore time, to both train and predict with our model.

The image is then resized to the expected input size (required for the Keras image data generator), before again being resized to an image of shape (224, 224). While this worked, the bounding box width and height ratio did differ a large amount, which could potentially have yielded slightly poorer performance. Given a bounding box $B = (top, bottom, left, right)$, we can create a new bounding box B' which retains square dimensions, by first finding the square side width s. Mathematically, this is defined as:

$$s = Max(bottom - top, right - left)$$

Now we create a new bounding box, defined as:

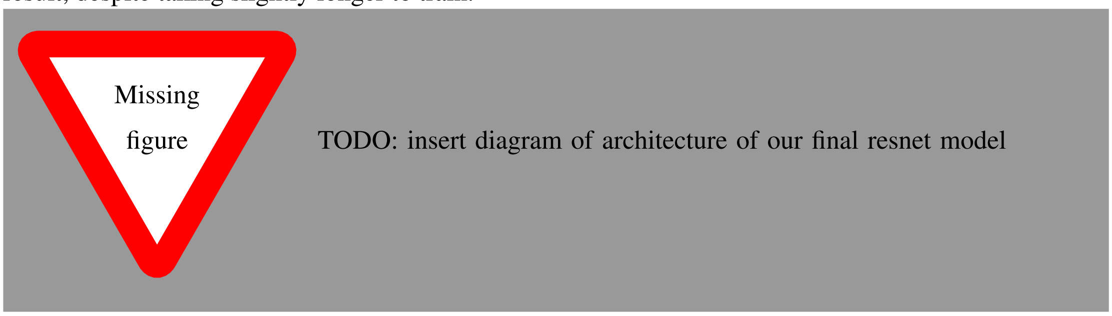
$$B' = (top, top + s, left, right + s)$$

.

By following this method, the model appeared to perform better overall, as rather than focusing on the overall image quality (which the previous model did), it would focus on the facial region.

During the process of training, it was noted that the HoG detector was missing approximately one eighth of the faces, therefore outputting the entire image, which could potentially have an impact on training and the accuracy of the model. Changing this to the CNN based classifier had surprising results: the computational performance increased, due to the underlying GPU acceleration, and the accuracy also improved due to the lack of random noise in the training set (through the generators). One thing to note with the CNN based face detector was to ensure upsampling was set to zero, otherwise memory issues would result (since the model and face detector model all need to be stored in GPU memory).

All of this massively improved the result, but generalisation was still a concern. Batch Normalization was therefore added to help make the make the model generalise further. This yielded a much better result, despite taking slightly longer to train.



TODO: insert citation for trying pretrained imagenet

C A system for preventing 3D spoofing attacks

While the systems before might go partially towards preventing 3D spoofing attacks, though primarily considering the 2D image, we now propose a method that is designed for classifying facial liveness based on a 3D point cloud.

C.1 2D to 3D Conversion

In order to classify an image/video, the 2D image needs to be converted to a 3D representation of a user's face. While 3d reconstruction is easier with videos (using structure from motion or other multiview based methods), there also exist image-based reconstruction methods such as vrn (**(author?)** 5) which are more specific and designed for reconstructing faces based on images. This also has benefits, as structure from motion is unable to reconstruct 3D from a single image, or from videos with very little motion.

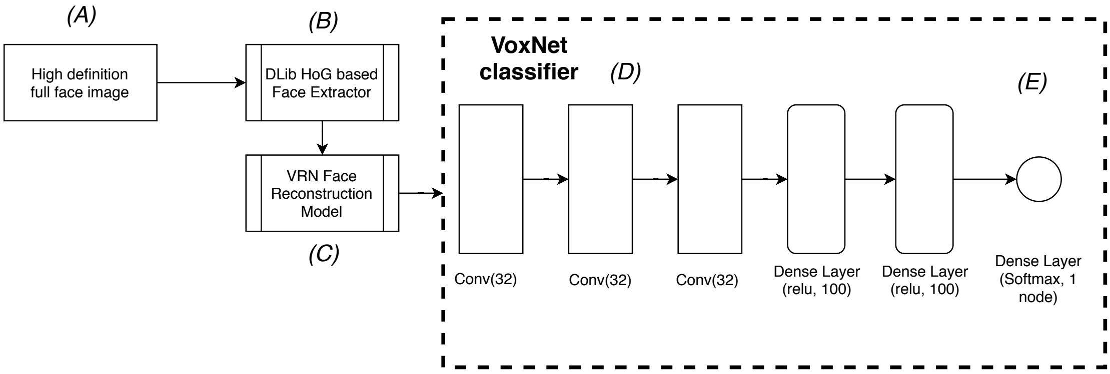


Figure 2: This is an overview of the 3D classifier. **(A)** a high resolution image is input into the classifier. **(B)** The image goes through a HoG based face detector. The bounding box of the face is extracted, and the image is cropped. The image is then resized to be 192x192 pixels, which is what's required by the VRN process. **(C)** The pretrained VRN face reconstruction model takes an image input, and outputs a voxel representation. Some postprocessing from the VRN model is necessary to convert an occupancy grid into a voxel representation (this is done here rather than in the VoxNet model). **(D)** The VoxNet classifier uses several 3D convolutional layers, along with a couple of Dense layers to classify. **(E)** The output of the last dense layer is simply a single number defined as the certainty of realness. 1.0 implies the model is certain that the input is real, while 0.0 implies the model is certain that the input is faked.

The image was converted by first applying a facial detection algorithm on the image, and cropping the image down to provide only the face. This cropped image was then resized to be of size $(192, 192)$, still in colour. This cropped and resized image was then fed through the VRN network. After this, the network output was filtered and stacked to provide the voxel input required.

The code to operate this can be found under the *liveness.vox.reconstruction* namespace within the code.

C.2 3D point cloud classification

Once the 3D reconstruction is obtained, one can then classify this using some model to produce the fake/real metric.

VoxNet takes in a point cloud and converts this to an occupancy grid. This is then fed through two convolutional layers, pooled, and then goes through a dense layer before reaching the classifier output (a dense layer with the k outcomes).

As a pretrained version of VoxNet wasn't readily available, the whole system was trained together from scratch.

C.3 Linking everything together

While each system is self-contained, linking them together took a little bit more work than expected. The models themselves couldn't be directly joined together, as VRN required extra postprocessing steps which couldn't be implemented using tensors within tensorflow. As such, the initial 2D to 3D conversion was required to be run as a preprocessing step.

To assist in the training phase, a generator was written in Python to conduct the postprocessing on the fly for each batch, which didn't require the entire preprocessing step to be done before training, thus reducing the peak memory usage problems. While an ImageDataGenerator was used previously, this isn't compatible with 3D, and therefore a custom module needed to be written.

Once the preprocessing had been completed, the preprocessed image was fed to the VoxNet.

D Visualisation and Demonstration

In order to visualise the overall outcome of facial liveness, a generic model

IV RESULTS

TODO results

V EVALUATION

TODO evaluation

VI CONCLUSIONS

References

[1] Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. 2012.

[2] Tanzeem Choudhury, Brian Clarkson, Tony Jebara, and Alex Pentland. Multimodal person recognition using unconstrained audio and video. In *Proceedings, International Conference on Audio-and Video-Based Person Authentication*, pages 176–181. Citeseer, 1999.

[3] Nesli Erdogmus and Sbastien Marcel. Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect. 2013.

[4] J. Galbally, S. Marcel, and J. Fierrez. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE Transactions on Image Processing*, 23(2):710–724, Feb 2014.

[5] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. *International Conference on Computer Vision*, 2017.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] Sandeep Kumar, Sukhwinder Singh, and Jagdish Kumar. A comparative study on face spoofing attacks. 05 2017.

[8] Keyurkumar Patel, Hu Han, and Anil K. Jain. Cross-database face antispoofing with robust feature representation. In *CCBR*, 2016.

[9] Xiaoyang Tan, Yi Li, Jun Liu, and Lin Jiang. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *ECCV*, 2010.

[10] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4148–4158. Curran Associates, Inc., 2017.