DoorLock Audio-Visual Authentication

Carlos Sancini
Madia Taher
Matt Brimmer
Xander Hathaway

April 15, 2020

Abstract

This paper applies a “VGGFace” model architecture based on RESNET50 for facial classification in conjunction with an audio classification model to biometrically authenticate a user for the purposes of unlocking a door. We use one-shot learning to learn how to authenticate users with limited onboarding data. Using two siamese neural network models to authenticate a user’s face and their voice, we are able to create a robust security system mimicking a door for home entry using a powerful edge device. Ultimately we achieve out of sample validation accuracy scores above 90% and F1 scores above 0.8 indicating the models are performing at a satisfactory level for our minimum viable product.

Introduction / Background

With the introduction of bigger data the privacy of the consumer has become a more important concern. As companies collect vast quantities about individuals, their interests and habits are stored in company databases.

A critical area of privacy lies with user authentication. A common method for authentication is a username / password combination, however, passwords can be made too simple and/or forgotten. Someone losing and/or forgetting an authentication method to their house raises all sorts of potential safety concerns.

Biometric authentication can serve as a more robust alternative to traditional lock and key mechanisms. Most biometric authentication services require two critical things. The first is enough computational power to run the necessary algorithms. The second is storage of very private information in remote cloud services where more powerful compute exists.

With the introduction of edge computing devices, like the NVIDIA Jetson TX2, we can move necessary compute power to the device, and keep private information secure at the edge without the need to send it to the cloud and ever put the information in jeopardy. In conjunction

with the edge capabilities, we use neural networks with a siamese architecture to train facial authentication and vocal authentication algorithms to authenticate a given user with very limited training data for a newly onboarded user. Typically classification neural network algorithms take extraordinary amounts of training data to learn classes, but with a siamese network we can learn to detect similarities from large training corpuses and apply the learnings to users with a short audio clip and face recording. Combining audio and visual authentication for a user grants a higher level of security

Method - Design and Implementation

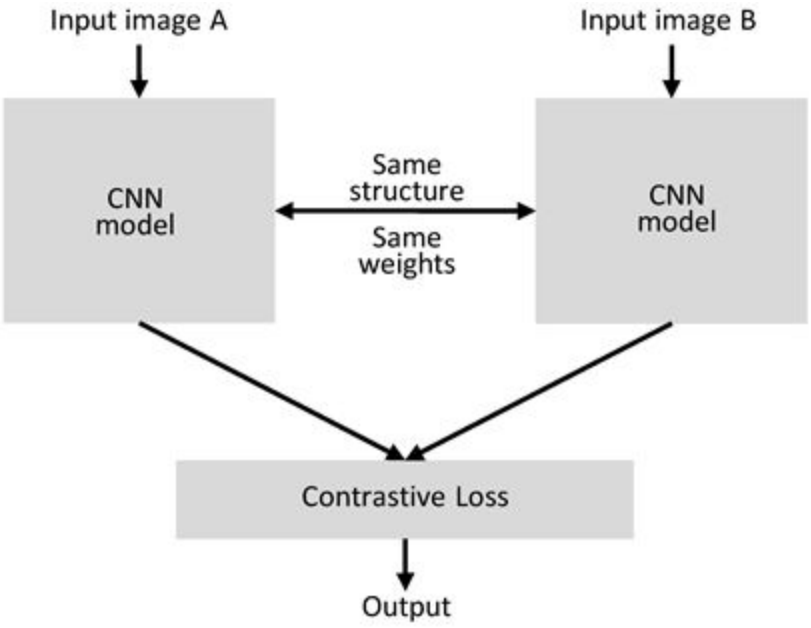
One Shot Learning Background

A typical image or speech recognition task using deep learning typically relies on large datasets and intensive computational power. This is also true for other supervised machine learning tasks that have been successfully used to achieve state-of-the-art performance in a variety of problems. However, these tasks are challenging to approach when little data is available as this fact poses a difficulty to the feature learning process of machine learning models.

One particularly interesting task is classification under the restriction that we may only observe a single example of each possible class before making a prediction about a test instance. This is called one-shot learning1[1]. The door unlocking project relied on this one-shot learning setting, in which the system must correctly make predictions given only a few examples of video and speech of a user.

The Siamese model is trained with input data composed of images and speeches of thousands of people. The input data is arranged in pairs such that positive pairs represent voices and images of the same person and negative pairs represent a mismatched. These pairs are fed to the model which produces a similarity score denoting the chances that the two input images belong to the same person. The similarity score is squished between 0 and 1 using a sigmoid function, wherein 0 denotes no similarity and 1 denotes full similarity. Any number between 0 and 1 is interpreted accordingly. The inference process is then a simple assessment of a reference image or speech, against a new acquired input.

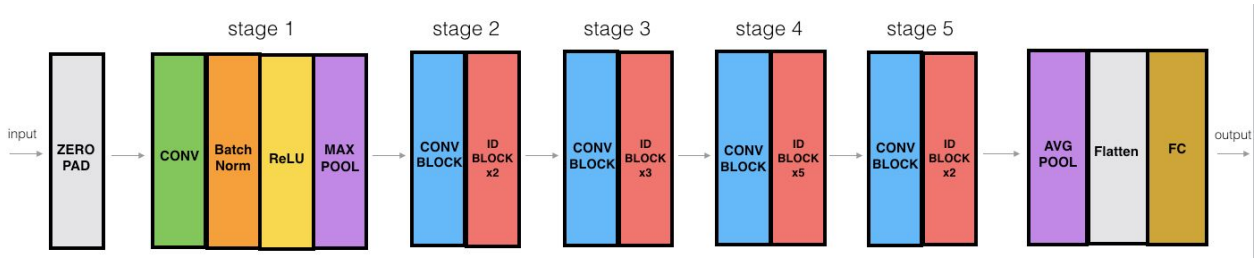
The Figure 1[2] contains a schematic view of the Siamese model architecture for the image task (the speech task works in the exact same way):



In the figure, the CNN model can be an implementation of any state-of-the-art compute vision architecture. An interesting fact of the Siamese architecture is that each input is fed to the same CNN model. In other terms, the weights are shared between the two inputs. The output of each CNN side is a multidimensional vector containing the important features of each input – in practice, the multidimensional vector is the typical last layer of a computer vision architecture that contains flattened convolutions. The multidimensional vectors, instead of being fed to a classifier layer (e.g. Softmax), are fed to a custom contrastive loss layer that calculates the distance (e.g. cosine or absolute distance) between the two multidimensional vectors generated by each input side. Therefore, the objective function minimizes the distance between the two multidimensional vectors, which causes the model to learn the content of the input A and B that are similar to each other. In other words, the model learns to extract the features that uniquely identifies a face (or voice) of a particular person.

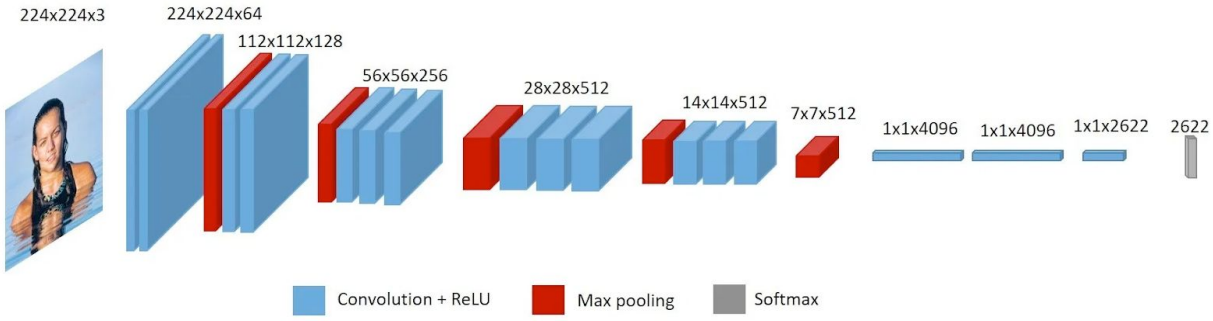
Face Detection

We used a model architecture called “VGGFace”, a Keras library. The model is based on RESNET50 architecture, which is popular in processing image data.



RESNET50

VGGFace was developed by researchers at the Visual Geometry Group at Oxford. The VGGFace model "encodes" a face into a representation of a number.




```
self.model = VGGFace(model='resnet50',
                     include_top=False,
                     input_shape=(224, 224, 3),
                     pooling='avg')
```

In order to add a new user to the model we use one-shot learning, similar to the audio model mentioned above. In one-shot learning, the model has to learn from only one example of data in our case an image of a person. That image might be noisy or the pose of the face can be not properly aligned for the model. In order to address this challenge, we captured a short video of a person and then extracted the face frame by frame. We then calculated the "mean features" by averaging all computed features for each image. We store the precomputed features as a pickle file for the next step.

For the face detection step, we calculated the Euclidean distance between two "encoded" faces. If the distance value is low, then we concluded that they are the same person. We can specify a threshold for the model to predict if two images are the same person.

Voice detection model

The dataset used to train the Siamese model for speech was obtained from Mozilla's Common Voice Project which contains 264,037 recordings from 23,858 English speakers with a diversity of accents, ages and genders. The recordings were arranged in 134,349 positive pairs (20%) and 537,396 (80%) pairs for a total dataset size of 671,745 observations. Empirically we found that raising the size of the dataset with negative pairs increased the performance of the model, even though this caused a skew in the distribution of labels. We had a limitation to increase the number of positive pairs since some individuals in the dataset had only a single recording, but, in contrast, we could create millions of negative pairs.

The MP3 files were converted to Mel-Frequency Cepstral Coefficients (MFCCs)[3] which are the most commonly used feature extraction method of audio signals. Intuitively, a MFCC is a representation of the frequency and amplitude content of audio signals. Each MP3 file, containing 5s to 10s of audio, is equivalent to a MFCC of 20x400 half precision floating point array, sampled at a rate of 22kHz (the limit of human perception). The MFCC conversion process was done in 10 hours with a script that parallelized the audio files in 16 threads (in a 16 CPU machine). Each file conversion took approximately 2 seconds, and, without the multithreading approach, the process would have been completed in 150 hours.

The choice of half precision floating points was due to memory issues, since the whole prepared dataset had a size of 60GB (float32 numpy arrays). This decision was helpful to fit the entire dataset in memory, which allowed the test of several models faster. We provisioned a powerful machine with 100GB memory, 16 CPUs and a V100 GPU. Our final model was trained in 200 epochs in approximately 10h.

The validation set results were quite satisfactory, with an accuracy of 98.72%. Figures 2 and 3 contain ROC and Precision-Recall curves, respectively, which also show the excellent performance of the model. We also tested our model against a test set composed of 231 pairs of recordings created with regular computer microphones. The recordings contained voices of friends, family and of our own. We did not treat them as this would not be a feasible step in production, consequently the files had background noise, different levels of length and quality. The accuracy achieved in the test set was 90.48%. Figures 4 and 5 show ROC and Precision-Recall curves of the test set, respectively. During the test of the system we found that the precision of 0.85 provided a good recall compromise, with an overall F1 score of 0.8.

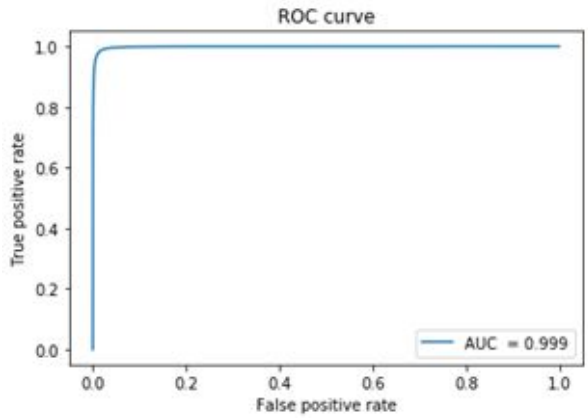


Figure 2: ROC curve for validation set

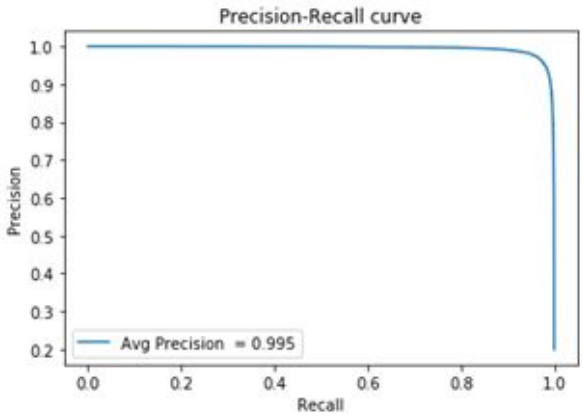


Figure 3: Precision-Recall curve for validation set

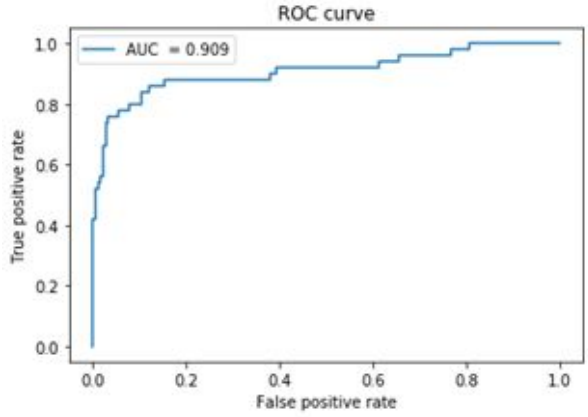


Figure 4: ROC curve for test set

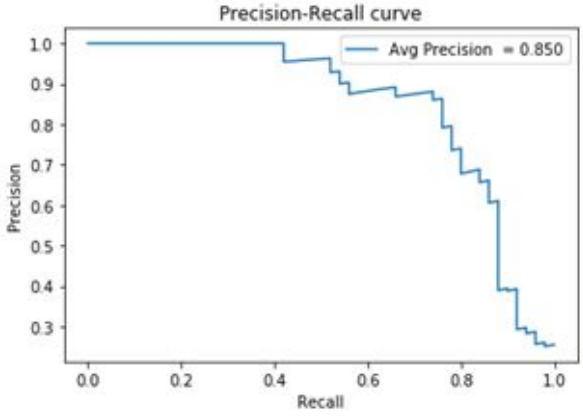


Figure 5: Precision-Recall curve for test set

Overall Authentication

The methodology we chose to implement for this minimum viable product is as follows: The visual model requires a 2-second clip and runs first to provide the visual authentication. After this authentication, we require a 6-second audio clip for audio inference. Upon audio authentication, the user is fully authenticated and the door unlocks.

Results / Deployment

The successful deployment of the inference onto the edge device (Jetson TX2) was a major milestone for our team. Although imperfect, the deployment demonstrated the ability to harness multiple models in collaboration to provide authentication at the edge.

From a practical standpoint, the TX2 comes very close to running out of memory when running both models. More work would need to be done to streamline the models and maybe reduce the framerate and computation being done by the processors to reduce the delays observed. Additionally, audio capture was imperfect. The face model could authenticate with every frame, and had the advantage of detecting the presence of a face before trying to run the inference model to authenticate the face. We didn't have the ability to intelligently detect that a voice was being captured in a similar way, and so we had to use blocking audio and command line queues to tell the user they needed to start speaking. Adding this functionality would allow the application to function in a natural state where it would be ready to accept user input at any moment.

Aside from these issues and those discussed above, the user testing was generally positive and authentication could be made by appropriate users.

Audio and Video Spoofing

A face spoof attack is an attempt to deceive a face detection system using a substitute for another's person's face – usually their photo, video recording or a 3D mask. Similar attacks can be made via audio by replaying a recording.

Active face liveness detection:
One of the techniques to avoid both audio and video attacks is challenge-response technique. The basic idea behind it is that the system challenges the user with some random instructions, for example asks the user to turn right or say a particular phrase, and then the response is checked to validate whether the instructions were followed.

Passive face liveness detection:
In passive techniques the user each time we detect a person, then detect an eye, we predict its status, and we keep track of the eyes status for a person.

Conclusion and Future Work / Next Steps

While the deployed model meets the goals set forth by the team, there are numerous ways in which the next iteration could build on the framework and improve. Below are the top additional features or improvements we would make in a subsequent release.

- User interface
    - In addition to just authenticating users, the user interface should be capable of adding new users. This would take the form of uploading audio and visual files to be used in the one-shot learning process.
- Audio / Visual simultaneously
    - Running both models simultaneously is a lot to ask of a relatively low power device like the Jetson, however, possibly if we use some non-blocking audio recording and reduced the frame rate, it's possible that it could be a more real-time system.
- Improve models
    - Although the models perform at a sufficient level, improvements in both precision and recall would likely be required to make this a viable product.
- Anti-spoofing capabilities
    - A challenge/response system for either the audio or visual component.

Conclusion

We were able to successfully train our bimodal system to detect and recognize the faces and audio of authenticated subjects and deploy the model for inference at the edge (in the Jetson TX2). We achieved F1 scores over 0.8 and accuracy rates exceeding 90%. This level of performance, while not appropriate for production, demonstrates the potential feasibility of the approach. We look forward to future iterations.

References

Mozilla's Common Voice Project: https://voice.mozilla.org/en/datasets

Siamese Neural Networks for One-shot Image Recognition: https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification: https://arxiv.org/abs/1707.02131

One-shot learning of generative speech concepts:
https://groups.csail.mit.edu/sls/publications/2014/lake-cogsci14.pdf

Few Shot Speaker Recognition using Deep Neural Networks:
https://arxiv.org/pdf/1904.08775.pdf

How to do Speech Recognition with Deep Learning (cool 101 guide):
https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a

One Shot Learning with Siamese Networks using Keras:
https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d

The Ultimate Guide To Speech Recognition With Python:
https://realpython.com/python-speech-recognition/

You Only Speak Once: https://github.com/Speaker-Identification/You-Only-Speak-Once

One Shot Audio: https://github.com/zdmc23/oneshot-audio/blob/master/OneShot.ipynb

Keras Face Identification in Real Time:
https://github.com/Tony607/Keras_face_identification_realtime

VGG Face: http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

VGG Face: https://github.com/ox-vgg/vgg_face2/

[1] 2017 Dattaraj J Rao, Shruti Mittal, S. Ritika - Siamese Neural Networks for One-shot Image Recognition

[2] Figure sourced from
https://medium.com/@prabhnoor0212/siamese-network-keras-31a3a8f37d04

[3] https://en.wikipedia.org/wiki/Mel-frequency_cepstrum