Person Identification Using Image and Voice

Sun Yitong[1], Ang Marcelo H. Jr.[2]

[1]SRP student, National Junior College

[2]Department of Mechanical Engineering, National University of Singapore

Abstract

Identification of people is woven into our daily life. However, extracting facial features and identifying people remains a challenge when it is happening under some conditions such as when the environment is dark or when the face is covered by cloths. Identification of peoples under these poor conditions are extremely difficult and even with the newest technology, the probability of recognising a man with face partially covered is low and not even mention the situation when the image is blurry. This project explores the possibility of applying deep learning methods combining both image and voice recognition systems to identify a person when one of them is not working well. The results have shown that when both facial features and vocal features are used, accuracy of identifying the person improves significantly. Possible sources of error and inherent challenges of identifying a person are discussed to provide potential directions for further research.

Introduction

In philosophy, the matter of personal identity deals with such questions as, "What makes it true that a person at one time is the same thing as a person at another time?" or "What kinds of things are we persons?" Generally, personal identity is the unique numerical identity of a person in the course of time. That is the necessary and sufficient conditions under which a person at one time and a person at another time can be said to be the same person, persisting through time [1]. The technology of identification of people is used in almost every aspect of our life, including daily phone unlocking, drawing cash from ATM, security checking, etc. As the correct identification of people would ensure the security given that there are a huge number of potential risks in life, accurate and effective methods of identifying a person is highly demanded. Normal citizens need the identification system to protect their privacy and property. Business companies need the system to keep commercial secrets and manage their finance. Police agencies also need them to detect criminals and keep the public safe. Moreover, even countries need the system to make sure their military force is under control in case someone pretends to be the commander and gives fake orders.

Since human civilization first started, we use facial features to recognise our mates and friends. Till 1964, work has been done to try to extract biological features from people and use them to recognise people facially with the help of computer [2]. As time passes, the recognition system keeps developing and the accuracy became higher and higher. Though 3-D face scans have shown the advantages, the usage of normal cameras to read the 2-D images are still the trend today due to the limitation of the hardware. As a result, in the paper, the traditional 2-D camera will still be used.

This project explores the possibility of applying deep learning methods combining both image and voice recognition to identify a person when one of them is not working well. Facial images and voice clips from 200 people will be used, including faces that are clearly shown and partially covered. The results have shown that when both facial features and speaker features are used, the accuracy of identifying the person improves significantly. Possible sources of error and inherent challenges of identifying a person are discussed to provide potential directions for further research.

Materials and Methods

As shown in Figure 1, the faces in the training set will first be cropped out and the important features will be extracted to form an array containing the RBG values. The hyperparameters of the facial model will then be tuned and suit the feature arrays to minimize the loss and improve the accuracy.

As illustrated in Figure 2, when there is a person to be identified, the system will first attempt to detect the position of the face. If the face is detected no matter it is partially covered or not. The face model will predict and return thein confidence value. With the combination of the voice confidence value from the voice model, the final confidence value of each person will be returned and the identity of the person will be selected and outputted as the result. However, if no face is detected at all, the model will only rely on the voice model to return the confidence value of the voice clip of the person and return the person's identity with the highest confidence value.

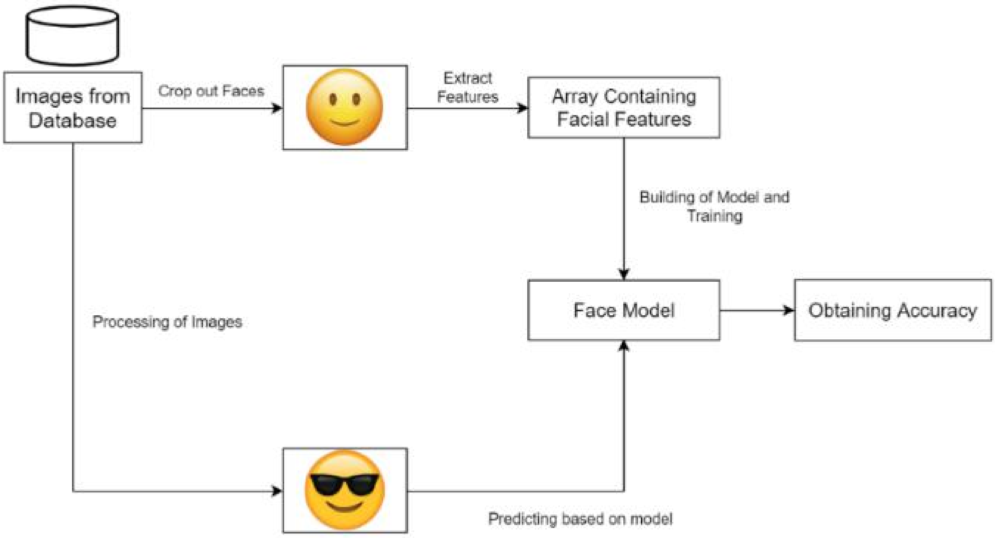


Figure 1.

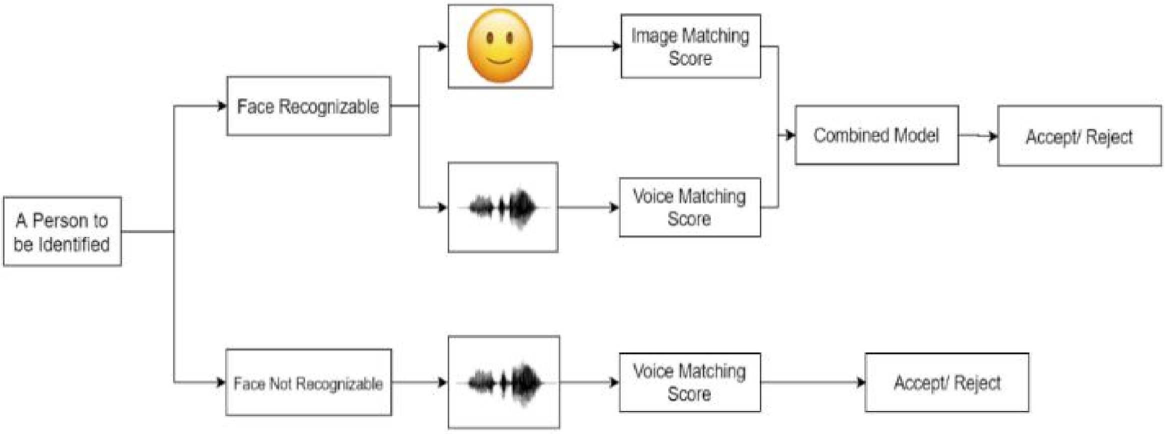


Figure 2.

DATABASE

The data used for the project is mainly from open-source websites, the images of faces are from VGGFace2 [3] where 200 people of different gender, races, and ages are selected in this project. The dataset of voice is mainly from AISHELL-1 [4], which includes 200 speakers from different accent areas in China. The length of each .wav file is about 3 to 10 seconds, which is the suitable length of identifying a person. In the process of experimenting, the images of each person have intentionally corresponded to the voices of each person to test the effectiveness of identification.

PROCESSING OF DATA

To simulate the situation when faces are partially covered and at a different light intensity. Mouth, eyes, and nose on each face are covered by mosaic as shown in Figure 3. Besides, the darkness of the images of faces has been adjusted to show various situations. The resultant images are all used to test the accuracy of the model.

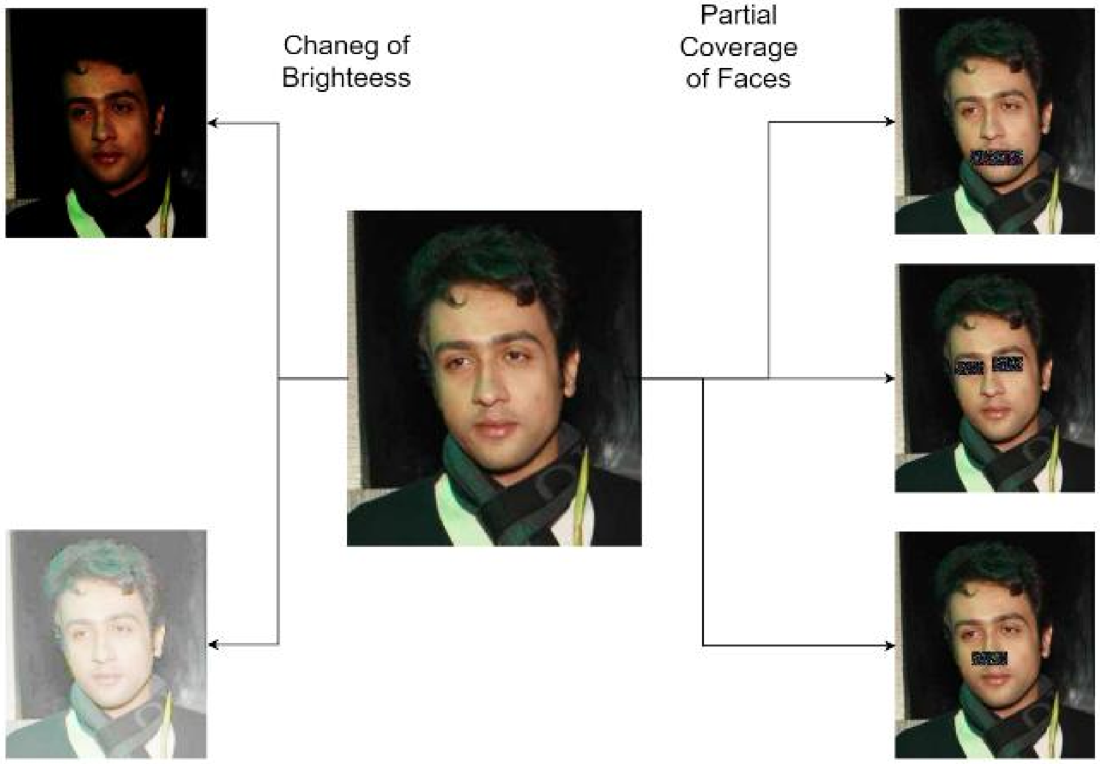


Figure 3.

FEATURE EXTRACTION

Open-cv is used in the stage of extracting facial features, all the faces from raw data are cropped out with using ‘haarcascade_frontalface_default.xml’ as the classifier. The faces are then transformed into RBG format as shown in Figure 4. The figures will be resized into images of resolution 200*200. Hence the numpy array of size (200, 200,3) representing the image will be returned as stored in the database.

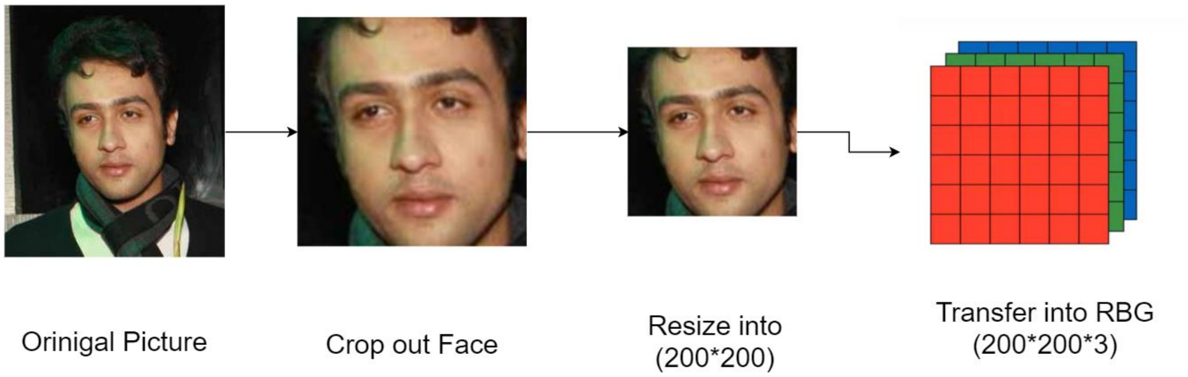


Figure 4.

For the features in the voice clips, modules of librosa and python_speech_features are used to extract the log Mel-filterbank energy features from each .wav file with 16000 as the sample rate. To extract the features, the voice clips will undergo pre-emphasis to balance the frequency spectrum, avoid numerical problems in Fourier transformation, and improve the Signal-to-Noise Ratio (SNR). The coefficient used in this case is α=0.97 with the use of the following equation applied to signal x [5].

$$\mathrm{y(t) = x(t) - \alpha x(t-1)}$$

After pre-emphasis, the signal is split into short-time frames with the frame size of 0.025 second and the frame step of 0.01 second. Then the window function is applied to each frame as shown.

$$w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1})$$

Then N-point FFT is done on each frame to calculate the frequency spectrum with the nfft=512 using the following equation where x_i is the i^{th}frame signal of x.

$$P = \frac{|FFT(x_i)|^2}{N}$$

After that, by applying 64 triangle filters in this case, the filterbank features are extracted and logged to simulate the non-linear human ear perception of sound. This returns a numpy array of size (2, 64, 299, 1) with 2 as the batch size and 3 seconds as the length.

The last step is to get the Delta features of the Mel-filterbank energy with the following formula where t is the frame number, N = 1 and N = 2 are both used in this case. By concatenating both Delta values of N = 1 and N=2, the final sound features will be extracted and represented by a numpy array of shape (2, 64, 299, 3) with 2 as the batch size and 3 seconds as the length [6].

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}$$

MODEL TRAINING AND TESTING

Deep neural networks are used in both the training of face model and voice model. After generating the models for the facial features and voice features, test sets are put into the models to evaluate the accuracy of the models. The accuracy is recorded in the results section. The structures of the models are shown in Appendix A.

Under the circumstances that the person's face can be seen, to combine both methods, the confidence value of the model can be calculated by multiplying the confidence value of both the face model and voice model. If the combined confidence level is below the threshold, the identification will be output as unknown; otherwise, the person with the highest identification confidence will be selected as the output. In order words, by the formula shown below, the matrix $P_{combines}$ with a shape of (200,1) will be generated and the label of the largest value in the matrix will be selected as the result.

$$P_{combined} = \frac{P_{face}}{P_{face_{max}}} + \frac{P_{voice}}{P_{voice_{max}}}$$

However, if the face of the person could be detected at all, the overall confidence value will be only dependent on the voice model.

$$P_{combined} = P_{voice}$$

Results

	Uncovered Faces under Normal Light Condition	Mouth Covered	Eyes Covered	Nose Covered	Overexposure	Underexposure	Totally Covered Faces
Face Model Accuracy	76.47%	60.52%	62.45%	65.36%	62.90%	49.60%	N/A
Voice Model Accuracy	70.36%	70.36%	70.36%	70.36%	70.36%	70.36%	70.36%
Combined Accuracy	84.62%	72.94%	73.25%	77.90%	74.17%	71.33%	70.36%

Discussion

Through the experimenting and testing of the models generated, it can be obviously seen that when the faces are partially covered, the accuracy of recognizing the identity will decrease rapidly from over 75% to below 65%. However, by combined both models, the accuracy of identifying a person in undesired environments will be improved and the limitation that facial recognition is totally ineffective when the face is not detected can be further addressed. In this way, the accuracy of identifying a person can increasing to over 70% even though it is under undesired conditions.

Due to the resources and time limit, the database may not be large enough to represent all the situations. There could be more situations where not only one part of the face is covered and different facial expression could also affect the results.

Besides, the cropping of faces using open cv may not be exactly accurate, this error in the training set may affect the model hence affect the result. One solution to this is that we could manually check the faces being cropped out to reduce the sources of errors.

Conclusion

This paper presents a person recognition system which uses images and voice clips. The methods are fusing both visual and vocal information to a classifier and return the confidences to recognize a person. The double-model system obviously outperforms other single-model systems.

Based on the limitations and challenges discussed in the Discussion section, the following main directions of research are proposed:

1. We can use a more advanced method on building the face model with focusing on main facial features

like the eyes, noses, and mouth instead of the whole face. In this way, even one or two facial features are covered, the system will still be able to identify the person accurately based on the other features. This will improve the accuracy of the model as the it will only target on one partial feature which is what the model is designed to fit.

2. We would like to add noise cancelling features on the voice recognition part which would increase the effectiveness of the model under noisy conditions.
3. Living body detection module such as blinking detection could be added to the system in order to eliminate the cases where the person's photo or voice recording is used to fool the system.

Acknowledgements

I would like to express my sincere gratitude to my project supervisor Professor Ang Marcelo H. Jr. for being patient and open-minded through the whole process of the research. His enlightening and insightful suggestions and guidance helped and supported me through many obstacles and difficulties.

References

[1] Carsten Korfmacher, Oxford University. Personal Identity

[2] De Leeuw, Karl; Bergstra, Jan (2007). The History of Information Security: A Comprehensive Handbook.

[3] Visual Geometry Group, (2014). VGGFace2. Retrieved from http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

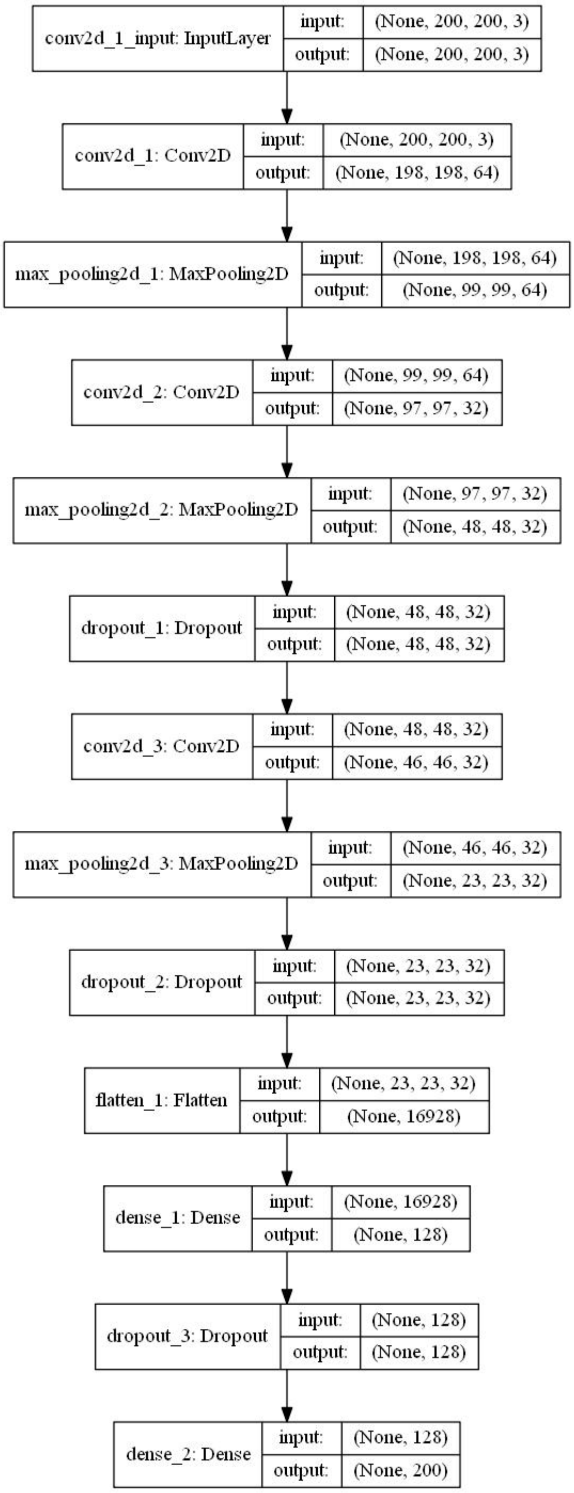
[4] AISHELL Tech Inc, (2019). Retrieved from http://www.aishelltech.com/kysjcp.

[5] Haytham Fayek, (2016). Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between

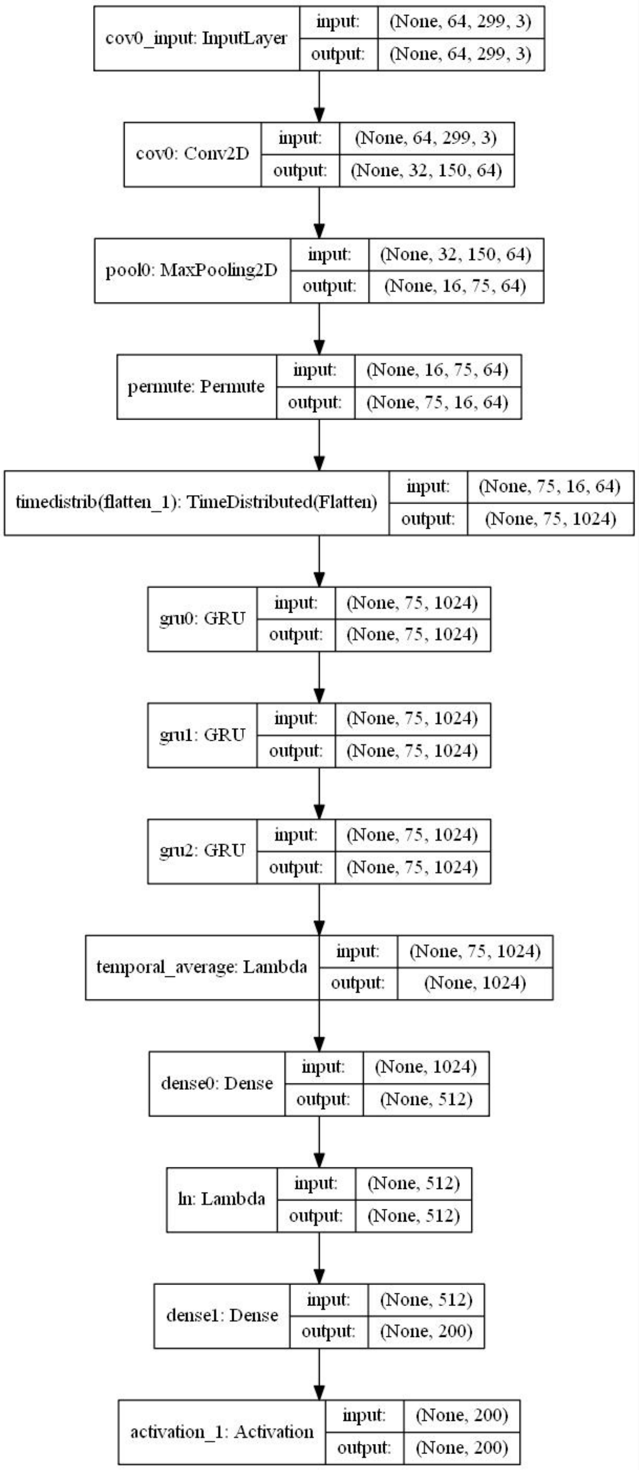
[6] James Lyons, (2013). Python_speech_features's Documentation

Appendix A

Structures of both face and voice deep learning models


conv2d_1_input: InputLayer | input: (None, 200, 200, 3) | output: (None, 200, 200, 3)
conv2d_1: Conv2D | input: (None, 200, 200, 3) | output: (None, 198, 198, 64)
max_pooling2d_1: MaxPooling2D | input: (None, 198, 198, 64) | output: (None, 99, 99, 64)
conv2d_2: Conv2D | input: (None, 99, 99, 64) | output: (None, 97, 97, 32)
max_pooling2d_2: MaxPooling2D | input: (None, 97, 97, 32) | output: (None, 48, 48, 32)
dropout_1: Dropout | input: (None, 48, 48, 32) | output: (None, 48, 48, 32)
conv2d_3: Conv2D | input: (None, 48, 48, 32) | output: (None, 46, 46, 32)
max_pooling2d_3: MaxPooling2D | input: (None, 46, 46, 32) | output: (None, 23, 23, 32)
dropout_2: Dropout | input: (None, 23, 23, 32) | output: (None, 23, 23, 32)
flatten_1: Flatten | input: (None, 23, 23, 32) | output: (None, 16928)
dense_1: Dense | input: (None, 16928) | output: (None, 128)
dropout_3: Dropout | input: (None, 128) | output: (None, 128)
dense_2: Dense | input: (None, 128) | output: (None, 200)

Face Model


cov0_input: InputLayer | input: (None, 64, 299, 3) | output: (None, 64, 299, 3)
cov0: Conv2D | input: (None, 64, 299, 3) | output: (None, 32, 150, 64)
pool0: MaxPooling2D | input: (None, 32, 150, 64) | output: (None, 16, 75, 64)
permute: Permute | input: (None, 16, 75, 64) | output: (None, 75, 16, 64)
timedistrib(flatten_1): TimeDistributed(Flatten) | input: (None, 75, 16, 64) | output: (None, 75, 1024)
gru0: GRU | input: (None, 75, 1024) | output: (None, 75, 1024)
gru1: GRU | input: (None, 75, 1024) | output: (None, 75, 1024)
gru2: GRU | input: (None, 75, 1024) | output: (None, 75, 1024)
temporal_average: Lambda | input: (None, 75, 1024) | output: (None, 1024)
dense0: Dense | input: (None, 1024) | output: (None, 512)
ln: Lambda | input: (None, 512) | output: (None, 512)
dense1: Dense | input: (None, 512) | output: (None, 200)
activation_1: Activation | input: (None, 200) | output: (None, 200)

Voice Model