[Samsung PRISM] Final Review Report

Real vs Recorded/Artificial Sound Classification Engine

College Professor:
Dr.G. Maragatham

Students:

- Prashant Kumar
- Ruchira Ray
- Sanka Karthik
- Vinayak Mathur

Department:
Computer Science

Worklet Area – AI, ML | Real vs Recorded/Artificial Sound Classification Engine

SAMSUNG

Problem Statement

- Voice assistant's are able to perform all kind of operations such as financial, personal etc. for a user. Security and authentication is thus a very critical requirement. In an IoT environment, multiple voice assistants work in parallel. Reducing the false trigger and device to device voice cancellation is very important.
- Goal of this project is to design and develop a sound classification engine which can :
  - Differentiate between human sound and recorded human sound. As example, the voice command is recorded from user and played to trigger voice assistant to get private data.
  - Differentiate between human sound and generated human sound. As example, using GAN human sound is generated and used as replay attack to trigger voice commands.
  - Differentiate between non speech sound events and artificial sound events. Such as baby cry is real, or a baby cry sound is from TV/Mobile.

Rashmi T Shankarappa
Associate Architect
rashmi.ts@samsung.com
+91-9845573133

Sourabh Tiwari
Chief Engineer
sourabh.t@samsung.com
+91-7406096314

Saksham Goyal
Senior Engineer
s.goyal@samsung.com
+91-9779964247

Additional Documentation:

- https://www.asvspoof.org/
- https://datashare.is.ed.ac.uk/handle/10283/3336
- https://research.google.com/audioset/dataset/index.html

Expectations

Work-let expected duration – 6 months

4

Students

- Feature engineering of sound such using MFCC, Log-mel, mel energy etc.
- Labelled sound data collection from available sources.
- Design and develop Deep learning model and sample application that can be run on PC/mobile to accept audio input and accurately classify the audio as real or artificial.
- API for training and testing with new sound dataset.
- Look for possible state of the art solutions and develop results better than the prior art.

Training/ Prerequisites

- Knowledge of Speech and Non Speech processing techniques and Machine learning concepts.
- Hands on in Deep learning development frameworks like Tensorflow, Keras etc.
- Model development, training and inference on CPU and GPU.

Kick Off < 1st Month >

- Understanding Speech & Non speech recognition concepts.
- Studying about features and characteristics of sound signals.
- Getting proficient with ML and DL algorithms and pre-processing techniques using Kaldi, librosa etc.

Milestone 1 < 2nd Month >

- Collecting dataset and pre-processing it.
- Architecture diagram showing training and inference pipeline and how model will be trained.
- Feature extraction from sound data.

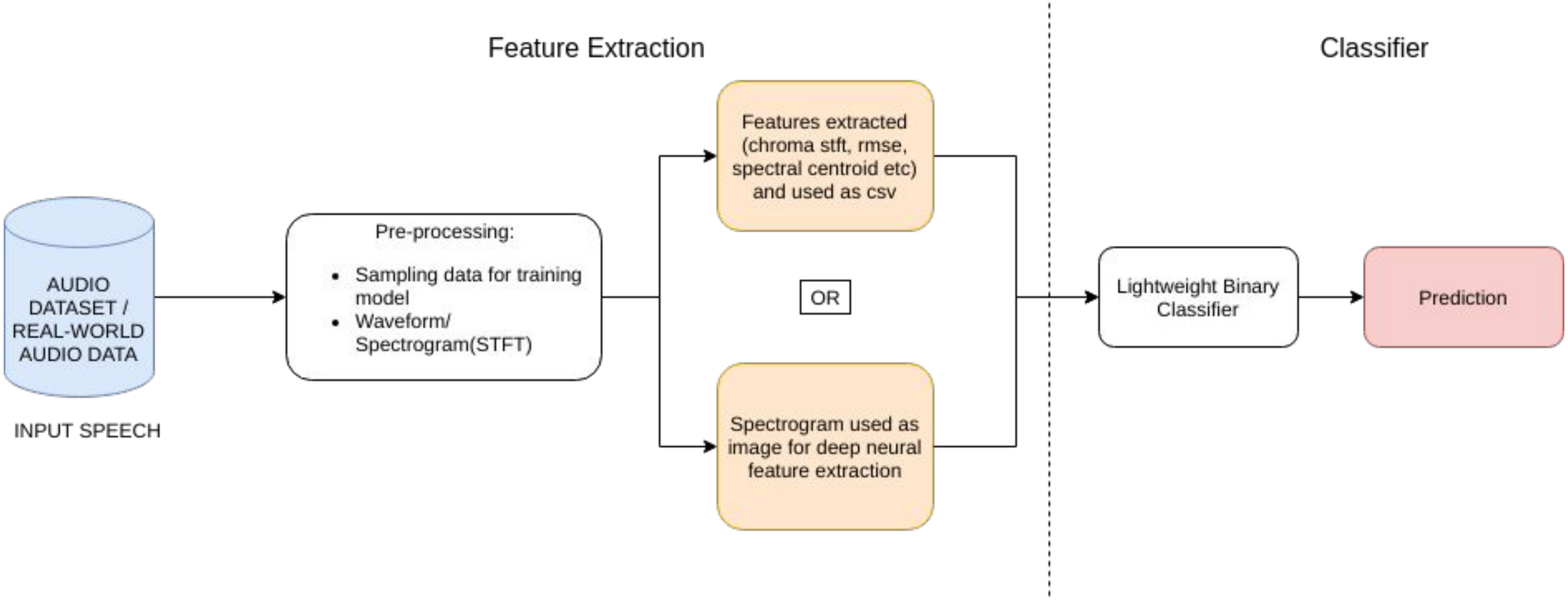
Milestone 2 < 4th Month >

- Developing a baseline model based on knowledge that can be improved further.
- Evaluating the model and tune hyper-parameters to increase performance.
- End to end testing application for PC/mobile with simulation.

Closure < 6th Month >

- Finalizing with the model that provides maximum accuracy.
- Final application which can be tested with live speech recording on PC/mobile
- Report on accuracy and comparison with standard approaches.

Approach / Solution



Feature Extraction

Classifier

AUDIO DATASET / REAL-WORLD AUDIO DATA

INPUT SPEECH

Pre-processing:

- Sampling data for training model
- Waveform/ Spectrogram(STFT)

Features extracted (chroma stft, rmse, spectral centroid etc) and used as csv

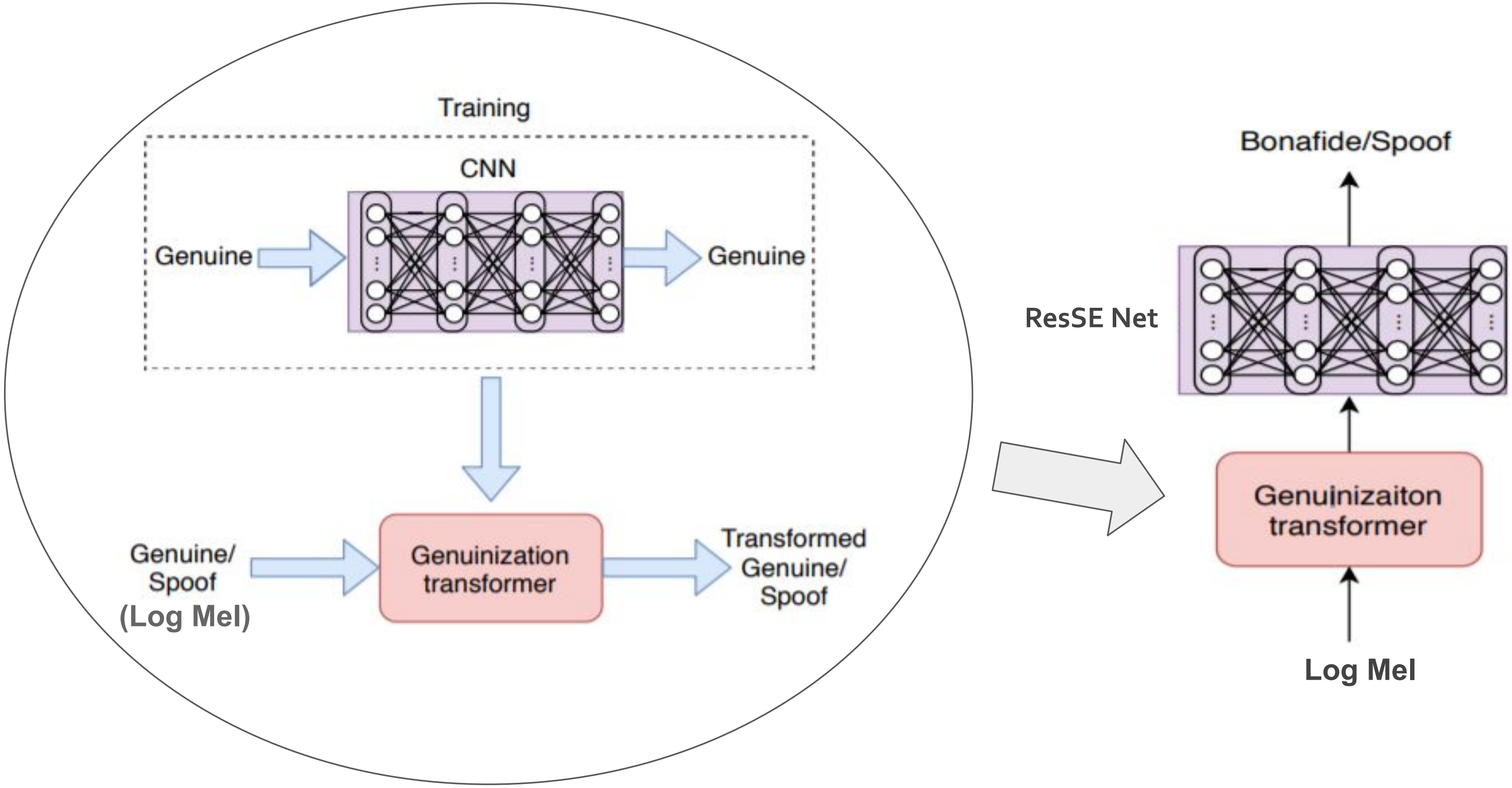
OR

Spectrogram used as image for deep neural feature extraction

Lightweight Binary Classifier

Prediction

Approach / Solution



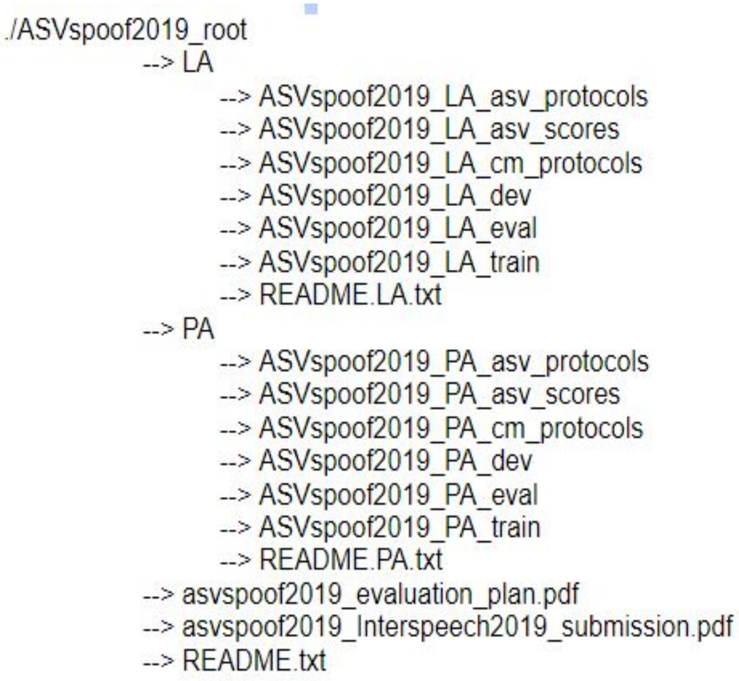
Dataset Analysis / Description

ASVSpoof 2019 Dataset

It is publicly available on https://datashare.ed.ac.uk/handle/10283/3336

The ASVSpoof 2019 Dataset provides thousands of Examples of spoofed and bonafide speech. The spoofed Data is obtained using different spoofing techniques.

Each containing designated files for Development, Evaluation and Training.



```
./ASVspoof2019_root
        --> LA
              --> ASVspoof2019_LA_asv_protocols
              --> ASVspoof2019_LA_asv_scores
              --> ASVspoof2019_LA_cm_protocols
              --> ASVspoof2019_LA_dev
              --> ASVspoof2019_LA_eval
              --> ASVspoof2019_LA_train
              --> README.LA.txt
        --> PA
              --> ASVspoof2019_PA_asv_protocols
              --> ASVspoof2019_PA_asv_scores
              --> ASVspoof2019_PA_cm_protocols
              --> ASVspoof2019_PA_dev
              --> ASVspoof2019_PA_eval
              --> ASVspoof2019_PA_train
              --> README.PA.txt
        --> asvspoof2019_evaluation_plan.pdf
        --> asvspoof2019_Interspeech2019_submission.pdf
        --> README.txt
```

Directory Structure

Dataset Analysis / Description

Logical Access:

Spoofing Techniques used for the LA dataset are Text-To-Speech(TTS) and Voice Conversion(VC) .

The Logical Access Dataset contains:

Development:

Bonafide:2548 Files

Spoof:22296 Files

Training:

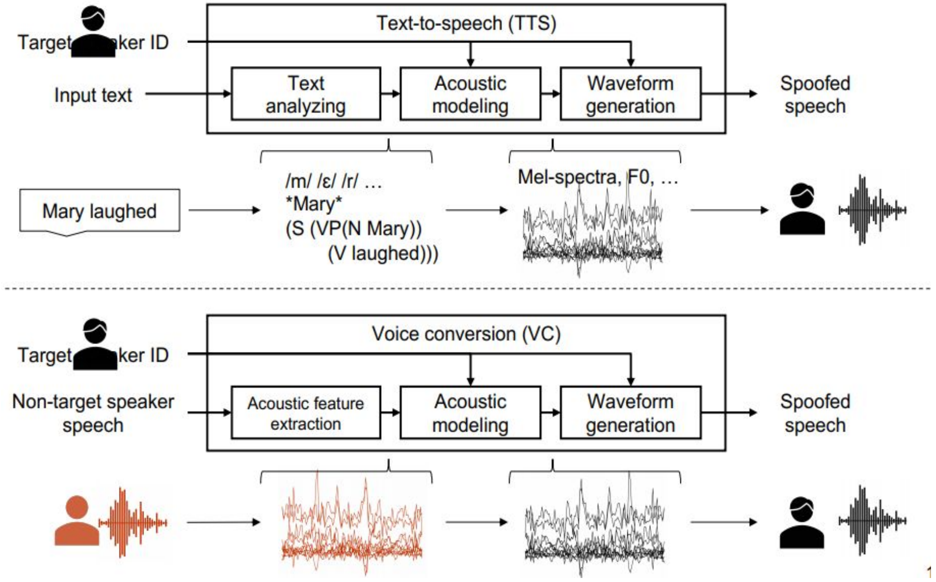
Bonafide:2580 Files

Spoof:22800 Files

Evaluation:

Bonafide:7355 Files

Spoof:63882 Files



Dataset Analysis / Description

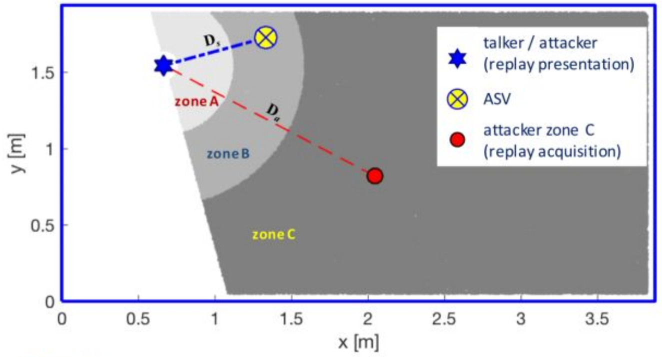
Physical Access:

Spoofing Techniques used for the PA dataset is Replay attack in different environments, device quality and distance.

The Physical Access Dataset contains:

Development:
Bonafide:5400 Files
Spoof:24300 Files
Training:
Bonafide:5400 Files
Spoof:48600 Files

Evaluation:
Bonafide:18090 Files
Spoof:116640 Files



D_s
zone A
D_a
zone B
zone C
1.5
1
0.5
0
y [m]
0
0.5
1
1.5
2
2.5
3
3.5
x [m]
talker / attacker (replay presentation)
ASV
attacker zone C (replay acquisition)

- **environment definition**
  - defined as a triplet (S, R, D_s)
  - the set (a, b, c) → categorical value
- **attack definition**
  - defined as duple (D_a, Q)
  - the set (A, B, C) → categorical value
- device quality (Q)
  - occupied bandwidth (OB) [kHz]
  - lower bound of OB (minF) [Hz]
  - linear/nonlinear OB power difference (linearity) [dB]

Environment definition	labels		
	a	b	c
S: Room size (square meters) R: T60 (ms) D_s: Talker-to-ASV distance (cm)	2-5 50-200 10-50	5-10 200-600 50-100	10-20 600-1000 100-150

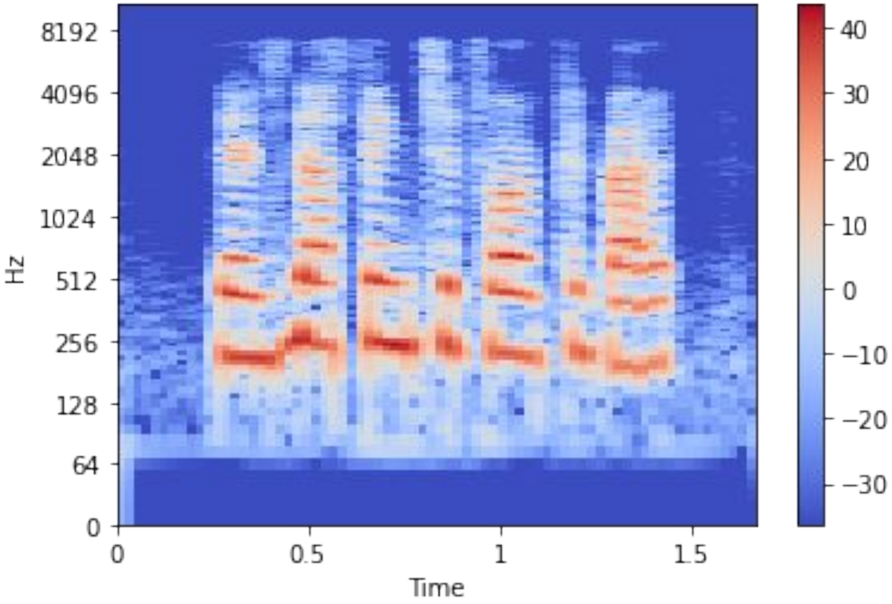
Attack definition	labels		
	A	B	C
D_a: Attacker-to-talker distance (cm) Q: Replay device quality	10-50 perfect	50-100 high	> 100 low

Replay device quality	OB (kHZ)	minF (Hz)	linearity (dB)
Perfect High Low	inf > 10 < 10	0 < 600 > 600	inf > 100 < 100

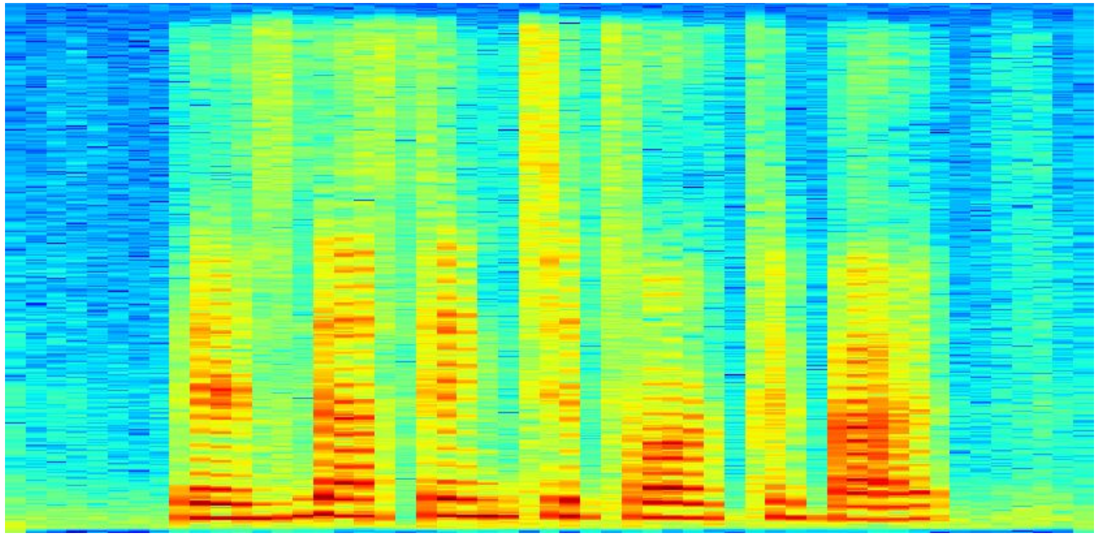
Data Preprocessing

- Audio data was converted into many fixed length windows to be fed to CNN. **Log-mel** was calculated for the fixed window of audio and concatenated along the required axis.
- Data(.flac) is converted into spectrograms of set framesize= 2**10.
- Each stft plotted cmap('inferno') (Figure size 1080x540 with 0 Axes) and saved in a new directory as images of dimension (407, 837, 4).
- We resize the image to (32 x 32) for ease of computation and normalise it with mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225].
- We load it to trainloadeder with a batch size =2 as the notebook keeps shutting down at anything higher than it.

Feature Extraction



Short-time Fourier transform (STFT)



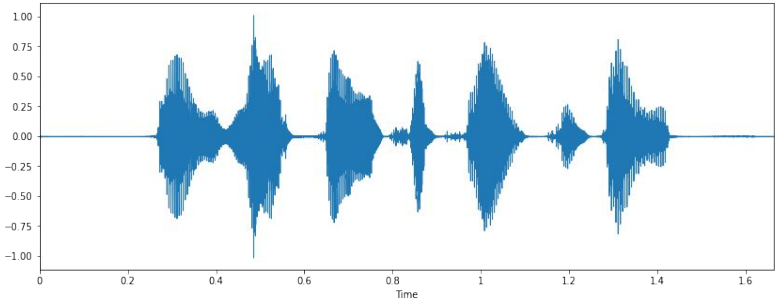
Spectrogram

Utilised spectrograms as image input to SE and resSE Net models.

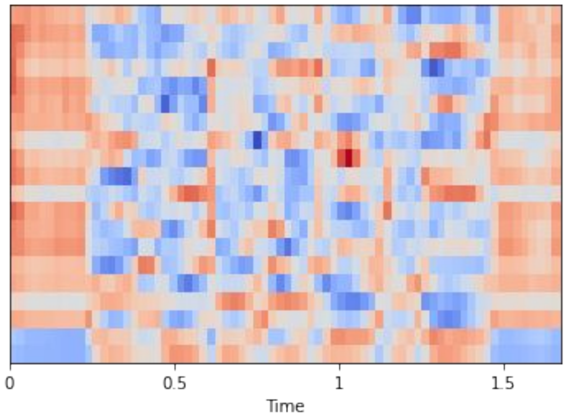
Feature Extraction

For ResNet model preprocessing is same as above with some additional changes are:-

- Used librosa to extract image form of mfcc feature of sound with first 13 number of mfcc, with 512 hope_length.
- We resize the image to (64 x 64) for easy to computation with 2 Classes spoof and bonafide.



Waveplot

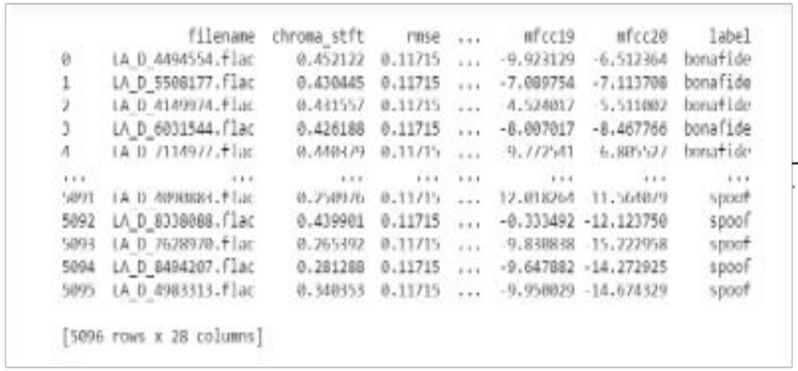


MFCC

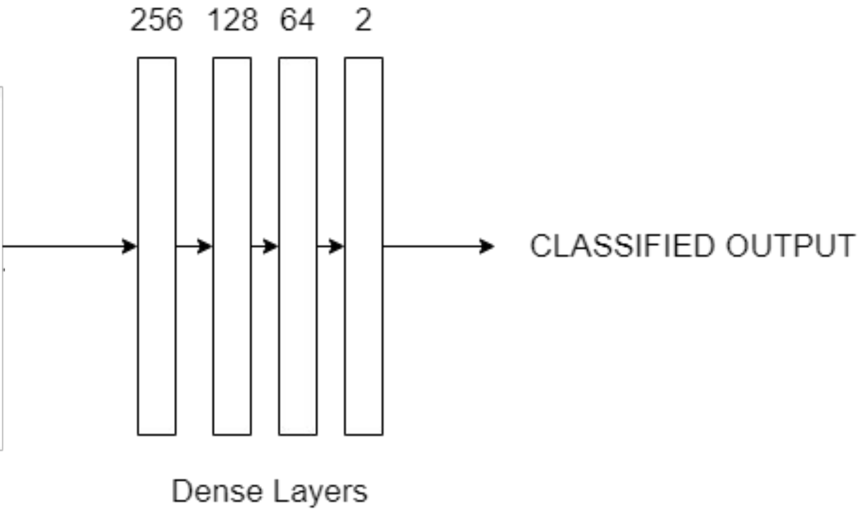
Vanilla Neural Net

Dense model:

loss binary_crossentropy, epochs 30, batch size 32.



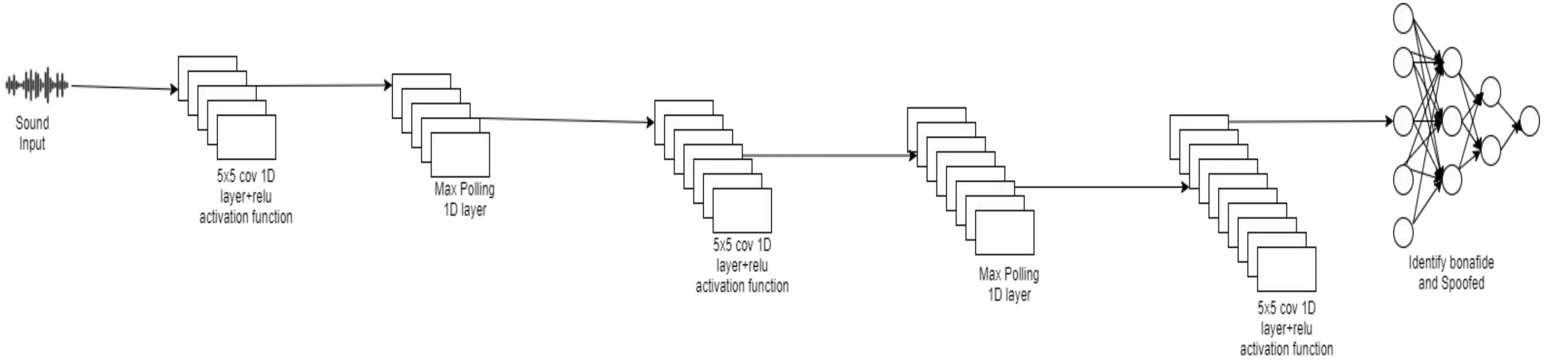
```
               filename  chroma_stft     rmse  ...     mfcc19     mfcc20     label
0     LA_D_4494554.flac     0.452122  0.11715  ...  -9.923129  -6.512364  bonafide
1     LA_D_5508177.flac     0.430445  0.11715  ...  -7.089754  -7.113708  bonafide
2     LA_D_4149974.flac     0.431557  0.11715  ...   4.524017  -5.511002  bonafide
3     LA_D_6031544.flac     0.426188  0.11715  ...  -8.007017  -8.467766  bonafide
4     LA_D_7114977.flac     0.440379  0.11715  ...  -9.772541  -6.885527  bonafide
...                 ...          ...      ...  ...        ...        ...       ...
5091  LA_D_4090083.flac     0.250076  0.11715  ...  12.018264  11.564079     spoof
5092  LA_D_8330088.flac     0.439901  0.11715  ...  -0.333492 -12.123750     spoof
5093  LA_D_7628970.flac     0.265392  0.11715  ...  -9.830838 -15.222958     spoof
5094  LA_D_8494207.flac     0.281288  0.11715  ...  -9.647882 -14.272925     spoof
5095  LA_D_4983313.flac     0.340353  0.11715  ...  -9.950029 -14.674329     spoof

[5096 rows x 28 columns]
```



CNN model:

→ 3 cnn layers and 30% dropout to overcome with Overfitting.

→ 25 epochs with 32 batch size.


Sound Input
5x5 cov 1D layer+relu activation function
Max Polling 1D layer
5x5 cov 1D layer+relu activation function
Max Polling 1D layer
5x5 cov 1D layer+relu activation function
Identify bonafide and Spoofed

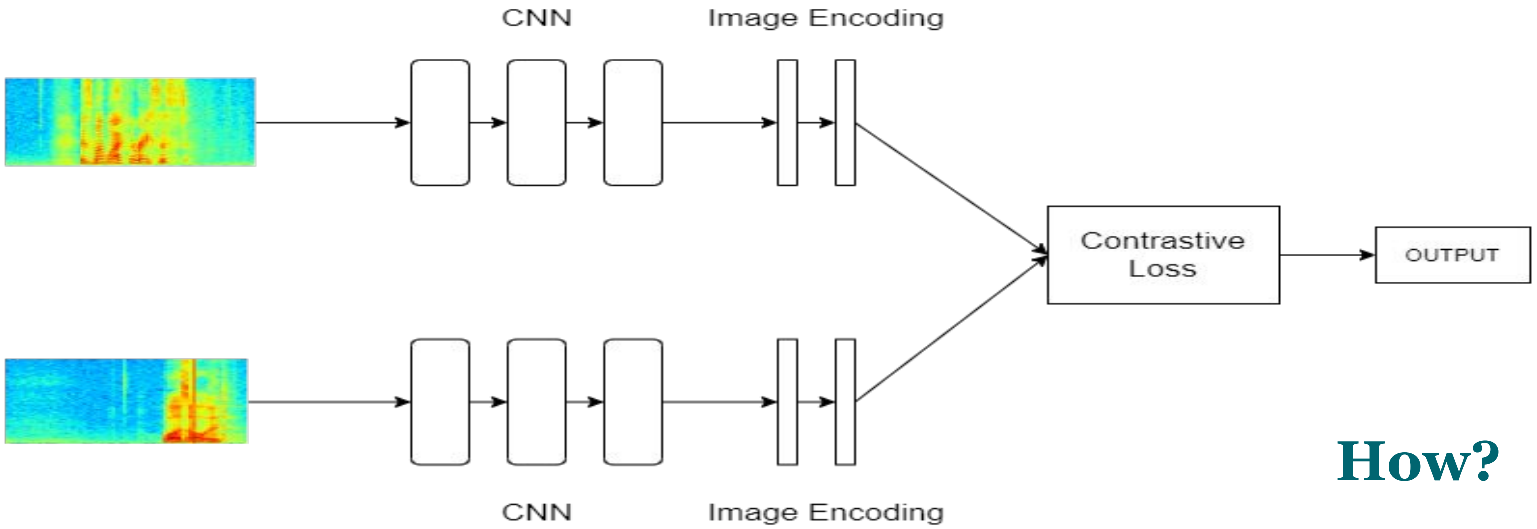
Siamese Net (One Shot Classification)

What?

Siamese network is a one-shot classification model and can perform prediction with just a single training example.

Why?

- it is learning a **similarity function**, which takes two images as input and expresses how similar they are.
- More **robust to class imbalance** as it requires very little information.

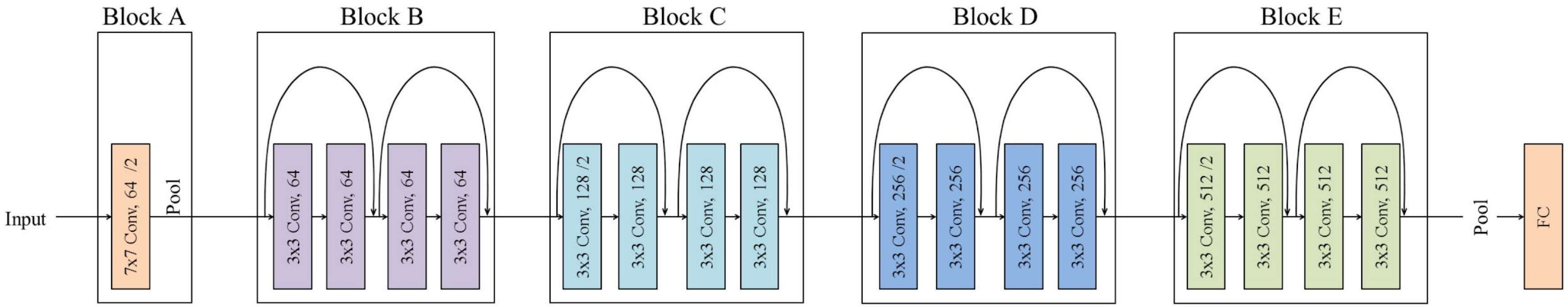


How?

CNN ResNet50

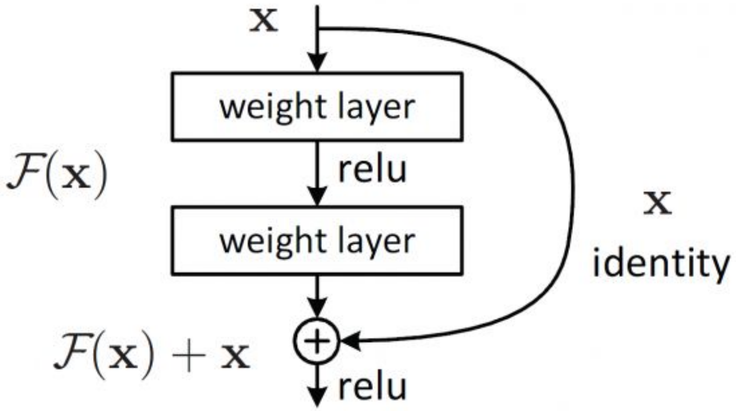
The convolutional layers mostly have 3×3 filters and have some rules:

- For the same output feature map, the layers have the same number of filters.
- If the size of the features map is halved, the number of filters is doubled to preserve the time complexity of each layer.

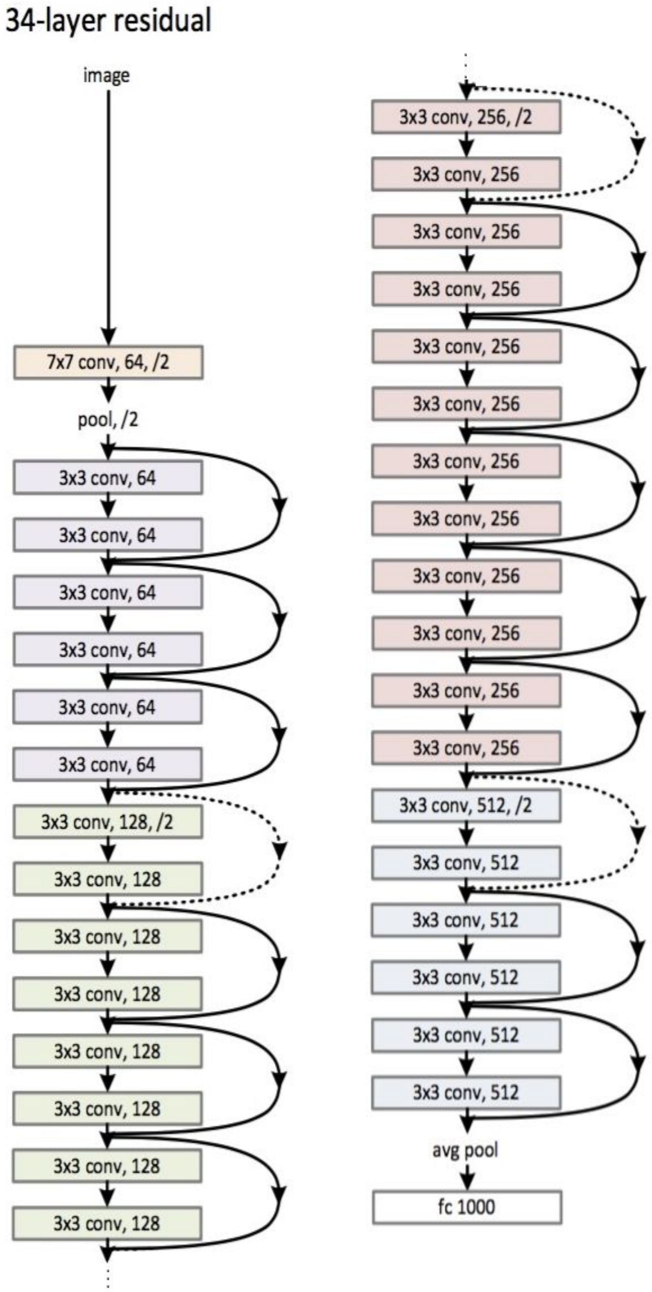


CNN ResNet50

- ○ EPOCHS = 11
- ○ BATCH_SIZE = 32
- ○ INPUT_SHAPE = 64 x 64 x 3
- ○ CLASSES = 2



x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu



Squeeze and excitation Network

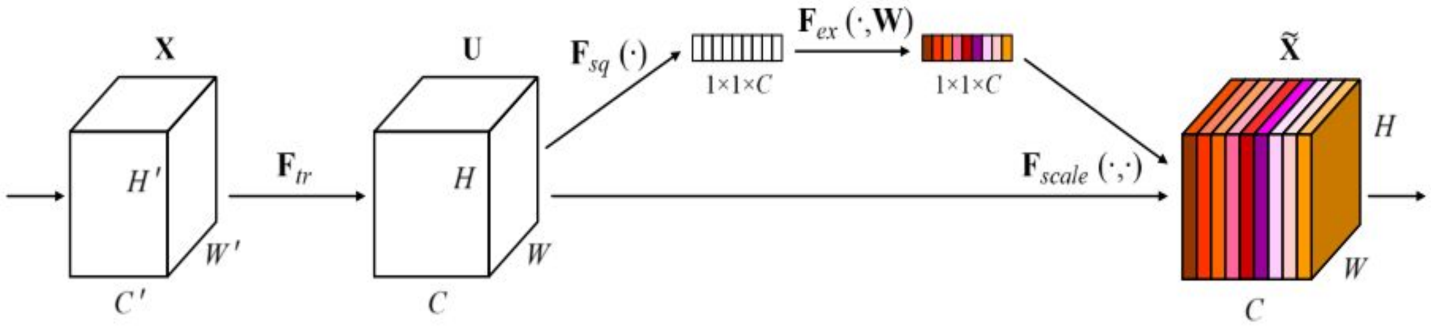
(Residual SE Net)

What?

Won the **first place in ILSVRC 2017 classification challenge** with 25% relative improvement.

Why?

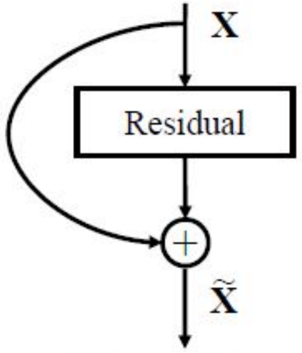
- Easily be added to existing architectures.
- This block helps dynamically "excite" feature maps that help classification and suppress feature maps that don't help based on the patterns of global averages of feature maps.



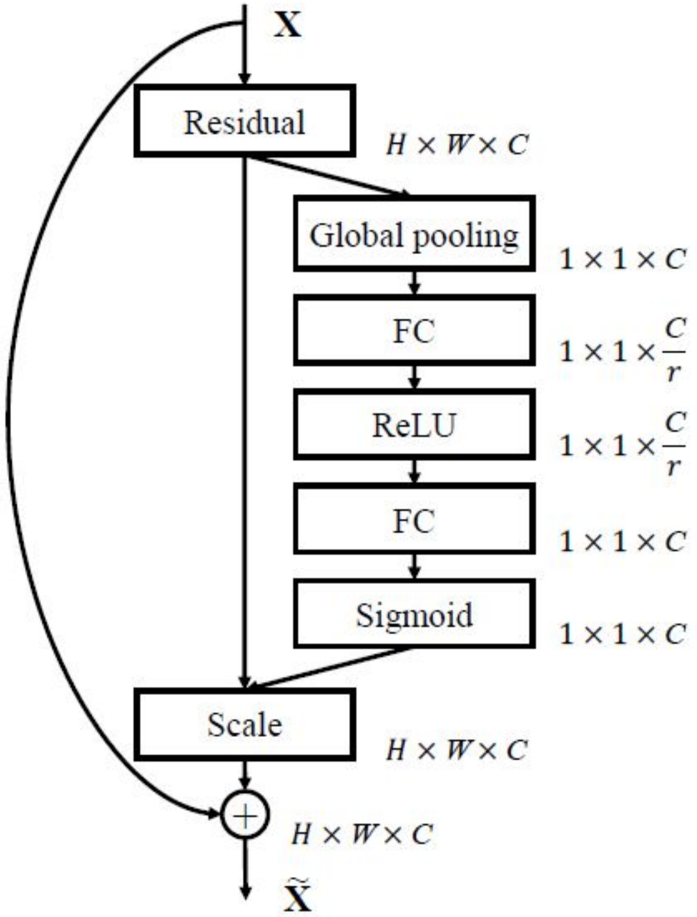
Squeeze and excitation Network (Residual SE Net)

How?

- EPOCHS = 40
- BATCH_SIZE = 32
- LEARNING_RATE = 1e-1
- WEIGHT_DECAY = 1e-4



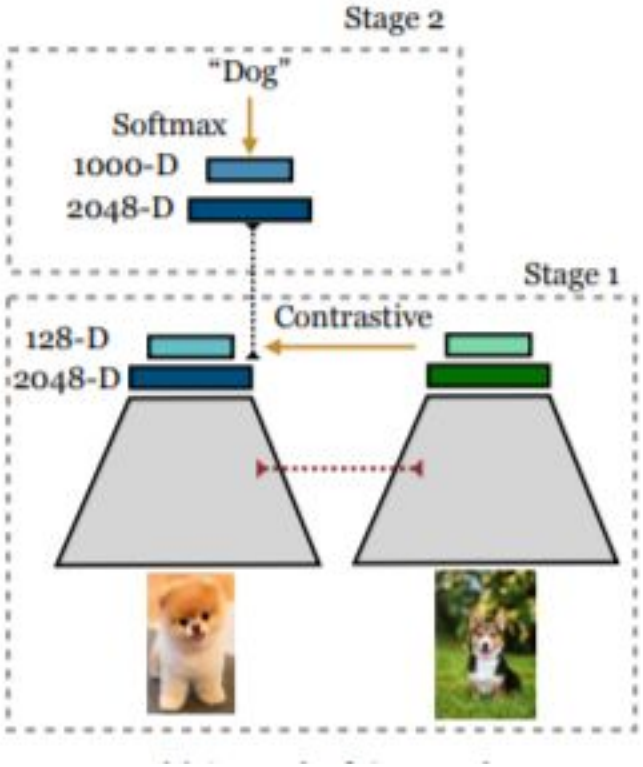
ResNet Module



SE-ResNet Module

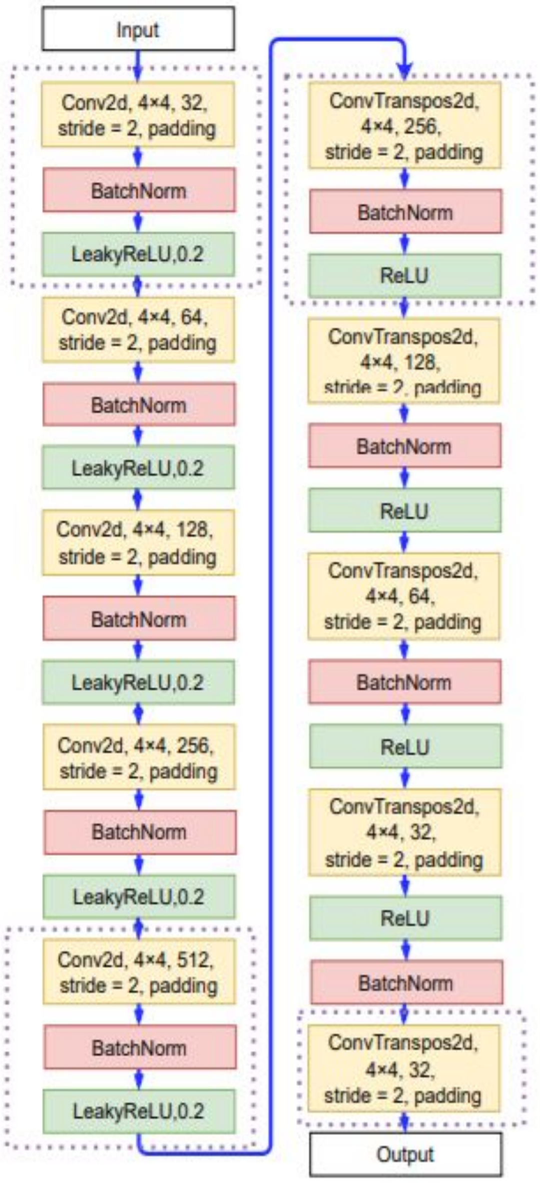
Supervised Contrastive Learning

- In this approach we learn a feature extraction bottle neck with contrastive loss function.
- Clusters of points belonging to the same class are pulled together in embedding space, while simultaneously pushing apart clusters of samples from different classes.
- Then we freeze the bottle neck and classify the last layer using another generic loss function such as cross entropy.

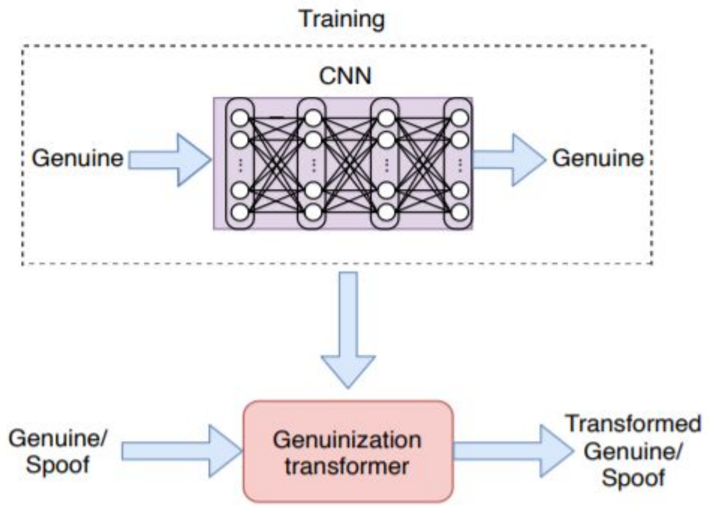


Genuinization Transformer

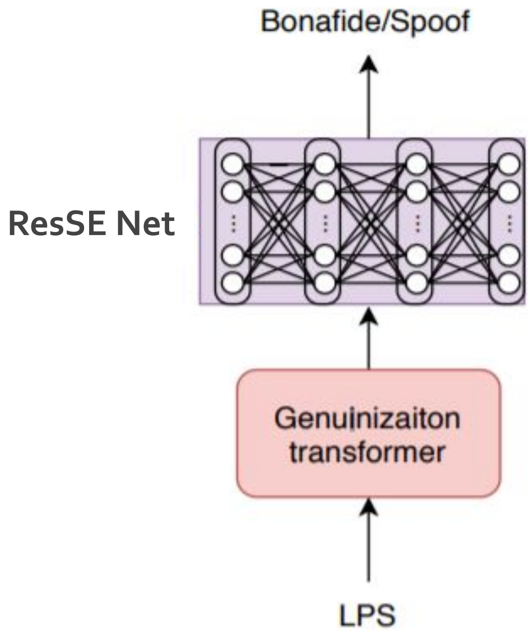
- This feature genuinization technique learns a
- Transformer with CNN the characteristics of only genuine speech.
- This transformer works based on the hypothesis that if we are able to derive a model that fits well the distribution of the genuine speech, such a model will take genuine speech as the input and generate genuine speech as the output following the same distribution of the genuine speech.
- However, when the model takes spoof speech as input, it will generate very different output, that amplifies the difference to genuine speech.

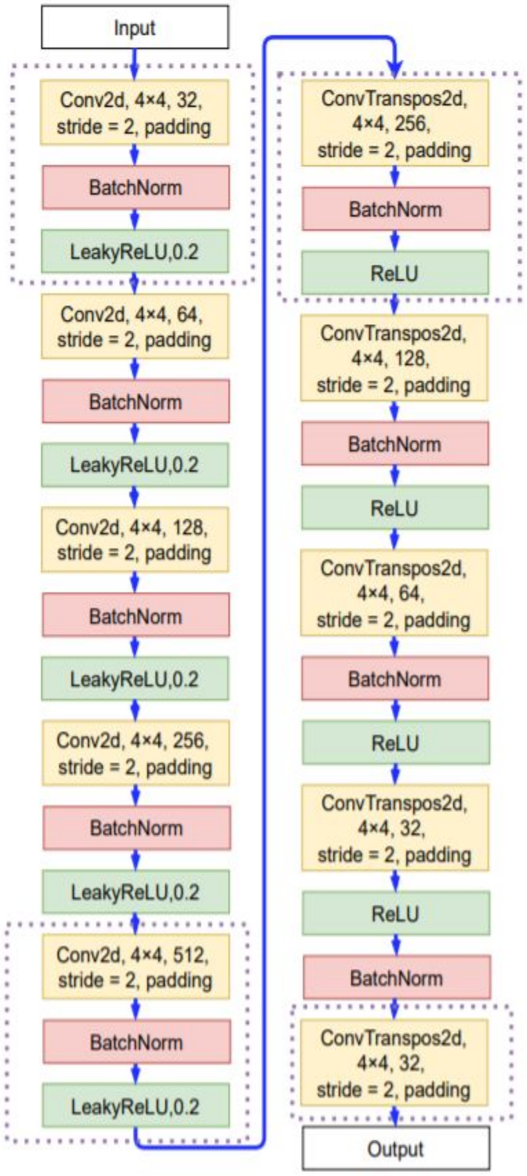

Input
Conv2d, 4×4, 32, stride = 2, padding
BatchNorm
LeakyReLU,0.2
Conv2d, 4×4, 64, stride = 2, padding
BatchNorm
LeakyReLU,0.2
Conv2d, 4×4, 128, stride = 2, padding
BatchNorm
LeakyReLU,0.2
Conv2d, 4×4, 256, stride = 2, padding
BatchNorm
LeakyReLU,0.2
Conv2d, 4×4, 512, stride = 2, padding
BatchNorm
LeakyReLU,0.2
ConvTranspos2d, 4×4, 256, stride = 2, padding
BatchNorm
ReLU
ConvTranspos2d, 4×4, 128, stride = 2, padding
BatchNorm
ReLU
ConvTranspos2d, 4×4, 64, stride = 2, padding
BatchNorm
ReLU
ConvTranspos2d, 4×4, 32, stride = 2, padding
ReLU
BatchNorm
ConvTranspos2d, 4×4, 32, stride = 2, padding
Output

Genuinization Transformer with Residual Squeeze and Excitation Net



Feature genuinization



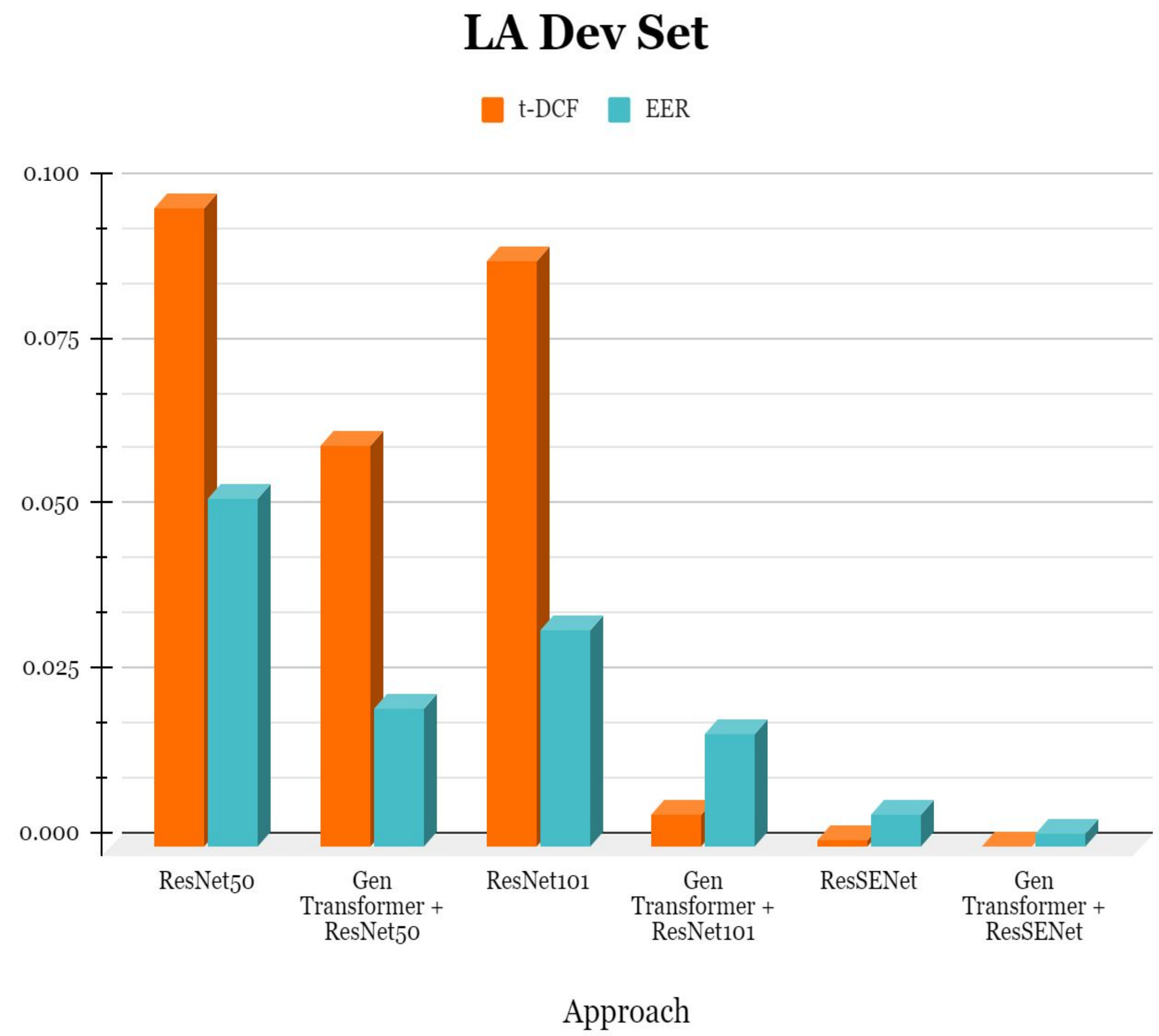


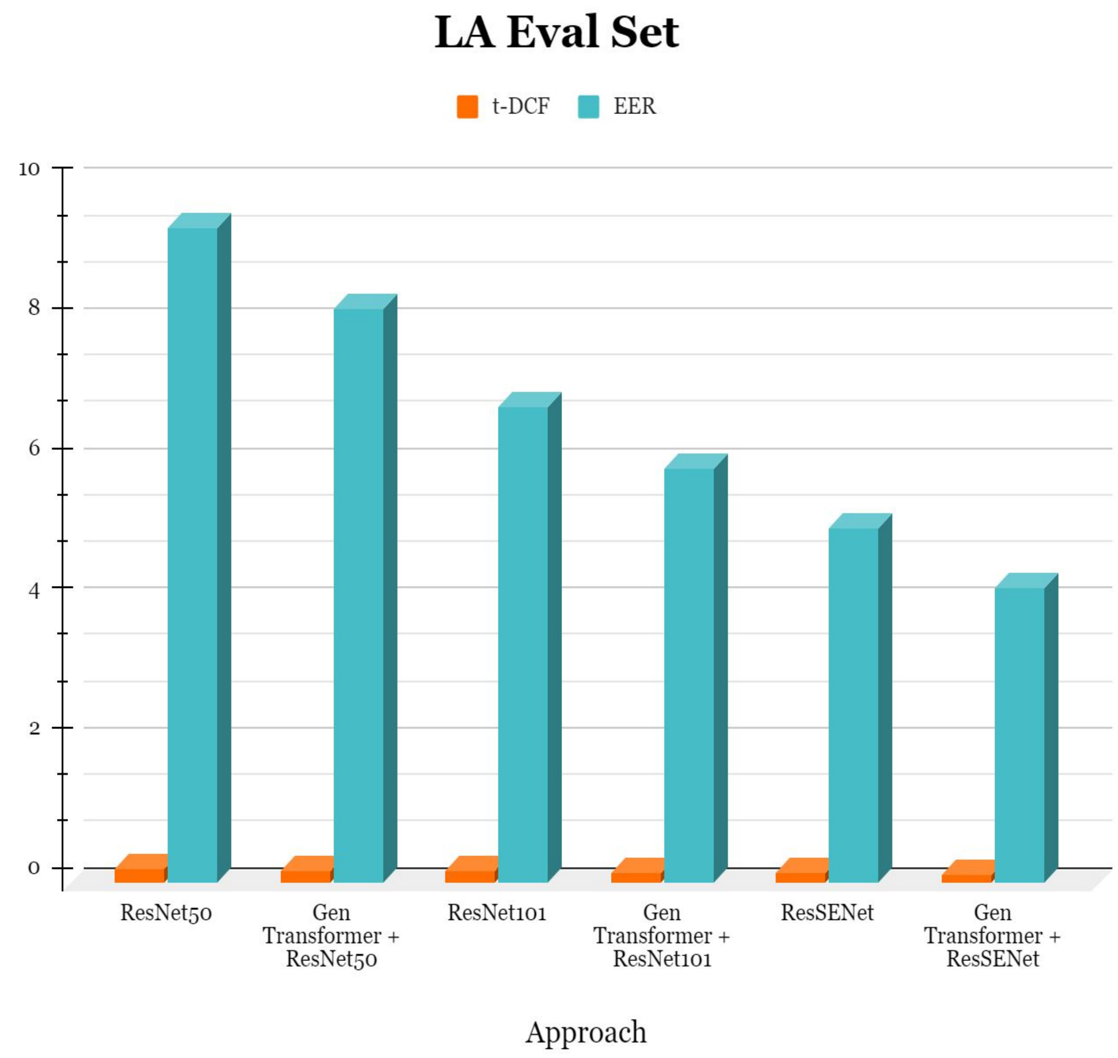
Genuinization transformer

Experimental Results / Observations

Performance level of Logical Access dataset

	LA Dev Set		**LA Eval Set**	
Approach	**t-DCF**	**EER**	**t-DCF**	**EER**
ResNet50	0.097	0.053	0.205	9.37
Gen Transformer + ResNet50	0.061	0.021	0.168	8.21
ResNet101	0.089	0.033	0.171	6.8
Gen Transformer + ResNet101	0.005	0.017	0.131	5.91
ResSENet	0.001	0.005	0.142	5.08
Gen Transformer + ResSENet	0	0.002	**0.109**	**4.22**


LA Dev Set
🟧 t-DCF
🟦 EER
0.100
0.075
0.050
0.025
0.000
ResNet50
Gen Transformer + ResNet50
ResNet101
Gen Transformer + ResNet101
ResSENet
Gen Transformer + ResSENet
Approach


LA Eval Set
🟧 t-DCF
🟦 EER
10
8
6
4
2
0
ResNet50
Gen Transformer + ResNet50
ResNet101
Gen Transformer + ResNet101
ResSENet
Gen Transformer + ResSENet
Approach

Deliverables

Final Deliverables:

- Deploy the model as an **API (FAST API)**.
- **Sample application** that can be run on PC/mobile to accept audio input and accurately classify the audio as real or artificial.
- Due to limited resources(**GPU issues**) we could only train the model for LA dataset.

Publication Plan:

We plan to submit our findings for the THE 12th INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND NETWORKING TECHNOLOGIES **(ICCCNT)**.

More details about the paper submission can be found on: https://www.12icccnt.com

Work-let Closure Details

KPIs delivered/Expectations Met:

- Feature engineering of sound such using MFCC, Log-mel, mel energy etc.
- Labelled sound data collection from available sources.
- Design and develop Deep learning model that can be run on to accept audio input and accurately classify the audio as real or artificial.
- Look for possible state of the art solutions and develop results better than the prior art.

GitHub Upload details

KLOC [Lines of Code in Thousands]	[7.7]
Model /Algorithm Details	[6 Models]
Details of Datasets uploaded [No of files – Images, Videos, etc.]	[Public Data, Link Mentioned]
List of Reports Uploaded [Name of All Documents uploaded	[README, Results, Pipeline]
Link to Github Repository	[https://github.ecodesamsung.com/SRIB-PRISM/SRMIST_VI31SRM_Real_vs_Recorded_Sound_Class_Engine]

Future Scope

In LA: A17, A18 type of attack performance was not good. A17 and A18 are voice conversion algorithms.

A17 uses waveform filtering and A18 uses vocoders.
All other attacks the performance is improved with Gen Transformer.

Supervised Contrastive pre training can be combined with Squeeze and Excitation network.

Possible use of attention heads

References

1. X. Tian, S. Du, X. Xiao, H. Xu, E. S. Chng and H. Li, "Detecting synthetic speech using long term magnitude and phase information," *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, Chengdu, 2015, pp. 611-615, doi: 10.1109/ChinaSIP.2015.7230476.

2. Khosla, Prannay, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, A. Maschinot, Ce Liu and Dilip Krishnan. “Supervised Contrastive Learning.” ArXiv abs/2004.11362 (2020): n. pag.

3. Lai, Jeff & Chen, Nanxin & Villalba, Jesús & Dehak, Najim. (2019). ASSERT: Anti-Spoofing with Squeeze-Excitation and Residual neTworks.

4. Nguyen, Thanh & Nguyen, Cuong M. & Nguyen, Tien & Nguyen, Duc & Nahavandi, Saeid. (2019). Deep Learning for Deepfakes Creation and Detection: A Survey.

5. Wu, Z., Das, R.K., Yang, J., Li, H. (2020) Light Convolutional Neural Network with Feature Genuinization for Detection of Synthetic Speech Attacks. Proc. Interspeech 2020, 1101-1105, DOI: 10.21437/Interspeech.2020-1810.

6. Ciftci, Umur & Demir, Ilke & Yin, Lijun. (2020). FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 1-1. 10.1109/TPAMI.2020.3009287.

7. Parasu, P., Epps, J., Sriskandaraja, K., Suthokumar, G. (2020) Investigating Light-ResNet Architecture for Spoofing Detection Under Mismatched Conditions. Proc. Interspeech 2020, 1111-1115, DOI: 10.21437/Interspeech.2020-2039.

8. Sahidullah, Md & Kinnunen, Tomi & Hanilçi, Cemal. (2015). A Comparison of Features for Synthetic Speech Detection.

9. Tomi Kinnunen, Kong Aik Lee, Hector Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, Douglas A. Reynolds: t-DCF: a Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification

References

1. Lei, Z., Yang, Y., Liu, C., Ye, J. (2020) Siamese Convolutional Neural Network Using Gaussian Probability Feature for Spoofing Speech Detection. Proc. Interspeech 2020, 1116-1120, DOI: 10.21437/Interspeech.2020-2723.

2. Platen, Patrick & Tao, Fei & Tur, Gokhan. (2020). Multi-Task Siamese Neural Network for Improving Replay Attack Detection.

3. David Kaspar, Alexander Bailey, Patrick Fuller, Librosa: A Python Audio Library (2019)

4. Adrian Yijie Xu, Urban Sound Classification using Convolutional Neural Networks with Keras: Theory and Implementation,

5. Admond Lee, How To Build A Speech Recognition Bot With Python (2019)

6. Rami S. Alkhawaldeh, DGR: Gender Recognition of Human Speech Using One-Dimensional Conventional Neural Network (2019)

7. Mike Smales, Sound Classification using Deep Learning, (2019)

THANK YOU