# Memory-efficient model for Automatic Speaker Verification

**Tanay Narshana** [1] **Tushar Pandurang Kadam** [1]

## Abstract

Synthetic speech generated using state-of-the-art text-to-speech (TTS) and voice-conversion (VC) systems have led to an increase in their misuse with the increase in their accessibility. Thus, it is essential to develop countermeasures that detect synthetic speech of existing systems and reliably detect unseen attacks from systems that may arise in the future. In this project, we focus on developing accurate models that are parameter efficient in addition to being sufficiently capable of generalizing over synthetic speech generated by unseen systems.

## 1. Technical Details

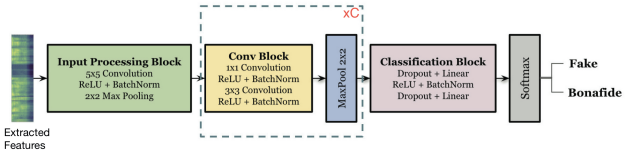


*Figure 1.* Generalized EfficientCNN – Contains an input processing block, $C$ convolution blocks, and a classification block

*Model 1*: As a part of ASVSpoof2019, Subramani & Rao use the EfficientCNN which is Generalized EfficientCNN (*Figure 1*) with $C = 4$ for the logical access (LA) task. We replicate their network except for taking 8 kernels (of respective sizes that they specify) for each convolutional layer and a dropout rate of $10\%$ for each dropout layer. In the classification block, the linear, reLU and the last linear layer are composed of 64, 64, and 32 nodes respectively.

*Model 2*: We select the CQCC features (Todisco et al., 2017) of an audio signal of length 4 seconds (truncate longer signals, repeat shorter signals) as an input to the Generalized Efficient CNN with $C = 2$. The CQCC features contained 96 bins per octave, $fs/2$ and $fs/1024$ as the highest and lowest frequency to be analyzed (where $fs$ is the sampling frequency of the audio file), 16 uniform samples in the first octave and 30 cepstral coefficients including the $0^{th}$ coefficient. We also use static, delta, and delta-delta coefficients. The resulting feature vector has size $90 \times 469$. The model has 4 kernels per convolutional layer and a dropout rate of $15\%$.

*Model 3*: Same as *Model 2* except for an additional Convolutional block (hence $C = 3$), no max-pooling layer in the Input Processing Block, and a dropout rate of $25\%$.

## 2. Results

The ASVSpoof2019 challenge gives *dev* and *train* data sets to train the model and *eval* data set(which contains synthetic and bonafide data from state-of-the-art systems whose samples are not in *dev* and *train*) to see how well a model responds to a new attack vector.

| Model | Hold-Out Set | Eval Data | # of Parameters |
|---|---|---|---|
| EfficientCNN | 97 | - | $\lesssim 50000$ |
| Model 1 | 99 | 91 | 32274 |
| Model 2 | 99.9 | 66.4 | 13962 |
| Model 3 | 99.9 | 70.15 | 21122 |

*Table 1.* F1 scores of models for Hold-Out Set and Eval Data along with the number of parameters in each model

Our CQCC based models don't adapt well, but on the brighter side, they classify exceptionally well for systems that they have been trained on.

## 3. Novel Contributions

Subramani & Rao supply the entire data set with associated weights (due to $\sim$ 90-10 imbalance) in one go for training. We lacked the computational power they had. Thus, we shuffle the data and use oversampling. In this way, we pass chunks of the data set with 50-50 samples from each class. We use the CQCC features which have been quite successfull in ASV related tasks as feature vectors. These features also help us reduce the number of Convolutional Blocks in the Generalized EfficientCNN.

*Individual Contributions*:

- *Tanay*: Setting up the training framework, preprocessing, model selection, feature selections, training and hyper-parameter tuning.
- *Tushar*: Quantitative and Qualitative evaluation of methods and models.

[1]Indian Institute of Science. Correspondence to: Tanay N <tanayn@iisc.ac.in>, Tushar K <tusharpk@iisc.ac.in>.

## 4. Tools Used

Keras(for the convolutional networks), Numpy, Librosa (for creating log of spectrogram features), MATLAB implementation of CQCC (courtesy (Todisco et al., 2017)) for generating CQCC features, Google Colab for training using GPU.

## References

Subramani, N. and Rao, D. Learning efficient representations for fake speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5859–5866, 2020.

Todisco, M., Delgado, H., and Evans, N. Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45: 516–535, 2017.