Memory-efficient model for Automatic Speaker Verification

Tanay Narshana, Tushar Pandurang Kadam

MLSP Final Project 2021

Problem Definition

- With the advancement of AI, attackers can replicate your voice to get into systems requiring audio verification.
- ASVSpoof2019 focuses on development of reliable and generalized countermeasures that can distinguish between bonafide and spoofed speech.
- Our work focuses on the logical access (LA) task where we develop countermeasures for attacks originating from S.O.T.A. Text-To-Speech (TTS) and voice-conversion (VC) systems.
- Additionally, we want to develop countermeasures that are robust enough to tackle new systems developed in the future.

- Data Set:
  - Training Data: 2.5k+ bonafide and 22k+ spoof (from 2 VC, 4TTS systems) audio files
  - Dev Data: 2.5k+ bonafide and 22k+ spoof (from one of the above systems) audio files
  - Eval Data: 70k+ bonafide and spoof samples from speakers and systems not used in Training and Test Data Set.

Technical Details

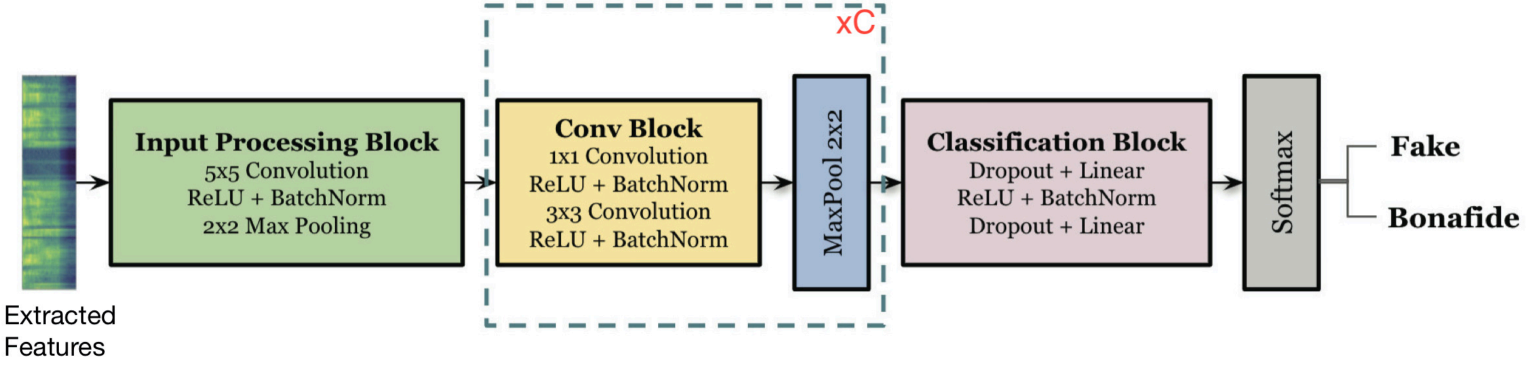


Figure 1: Generalized EfficientCNN

- Subramani et al.(2020) propose the EfficientCNN. It is basically Generalized EfficientCNN with C=4.
- Extracted Features: Z-normalized log of the spectrogram for audio files of 4 seconds (sampling rate of 16 GHz, 1728 FFTs, and a hamming window with a length of 108ms and a 10ms window shift)
- Dropout Rate = 0.2
- Weighted Cross-Entropy Loss for the imbalanced data set.

Novel Contributions

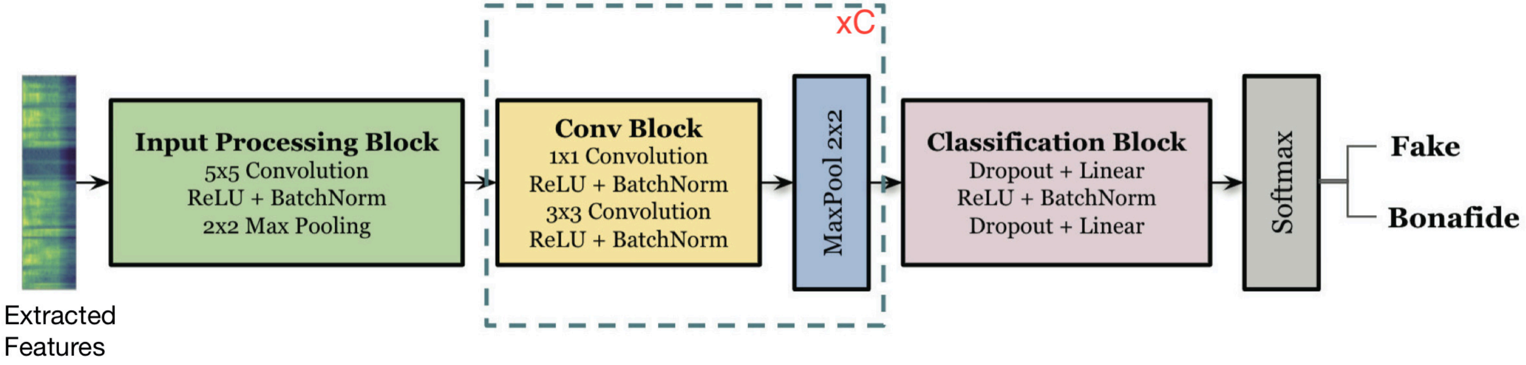


Figure 1: Generalized EfficientCNN

- The data set is sufficiently large (~64GB for the spectrogram features) to not fit in our RAM and hence we use oversampling
- We randomly select a subset of the dataset (with 50-50 class ratios) to train our model with normal cross-entropy error to not break the i.i.d. assumption of SGD.

- Model 1:
  - Trying to replicate, we found oscillating accuracies.
  - Increased kernels per convolutional layer to 4 or 8 (as a hyperparameter)
  - Vary drop out rate from 0.1 to 0.3 (increments of 0.05)
  - 64, 64, & 32 neurons in the linear, reLU, and linear layers of the classification block.

Novel Contributions

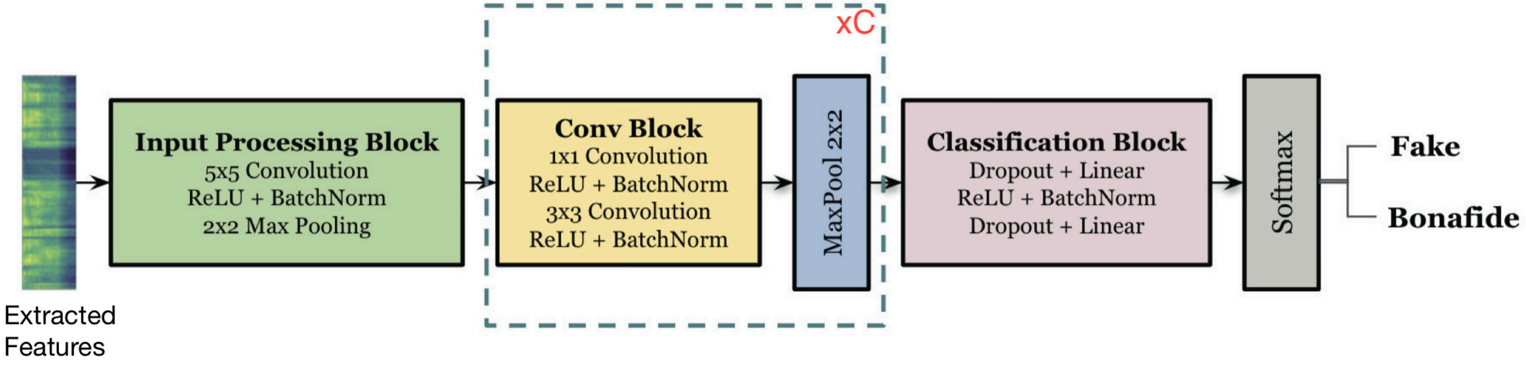


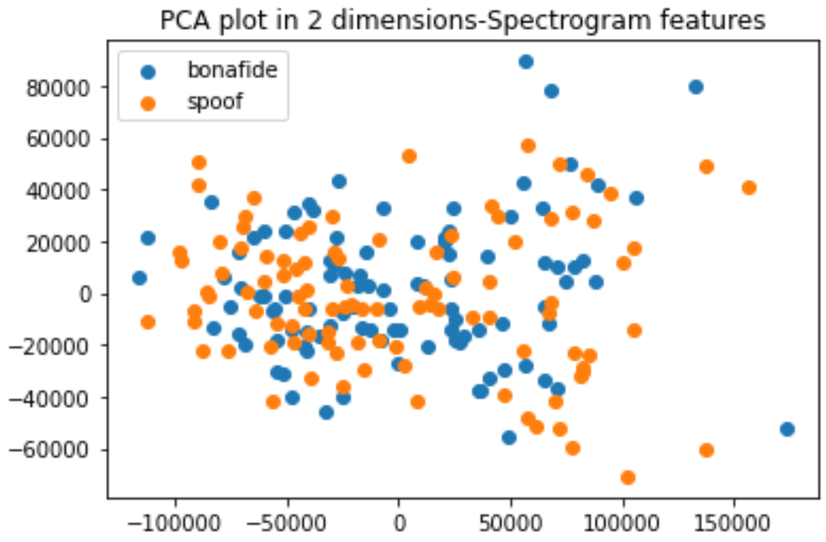
Figure 1: Generalized EfficientCNN

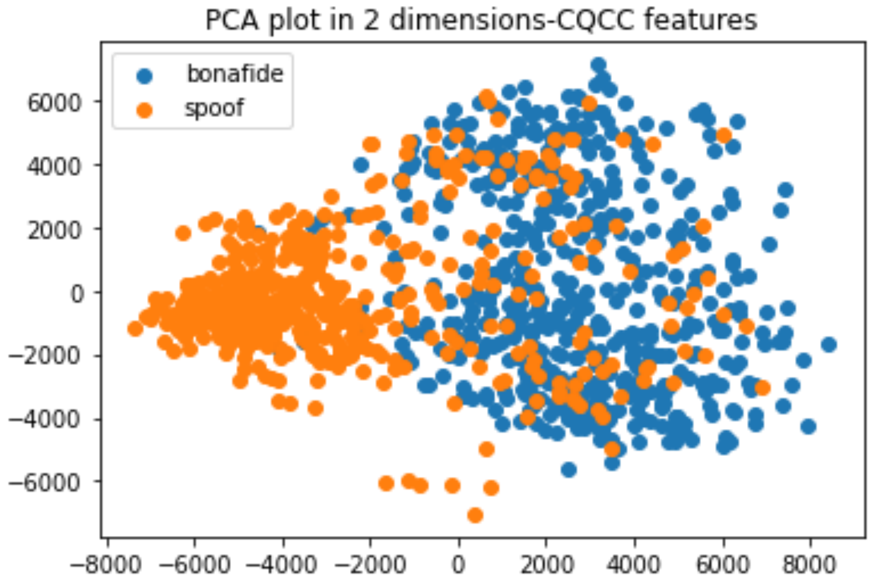
- We use the constant Q cepstral coefficients (CQCC) features that are considered effective for ASV tasks.
  - Specifications: 96 bins per octave, *fs/2* & *fs/1024* as the highest & lowest frequencies to be analysed, 16 uniform samples in the first octave and 30 cepstral coefficients(including the 0^{th} coefficient). We also use the static, delta, and delta-delta coefficients.
- Model 2:
  - Generalized EfficientCNN with C = 2, 4 kernels per convolutional layer, and dropout rate of 0.15
- Model 3:
  - Generalized EfficientCNN with C = 3, 4 kernels per convolutional layer, dropout rate 0.25, and no max pooling layer in the Input Processing Block.

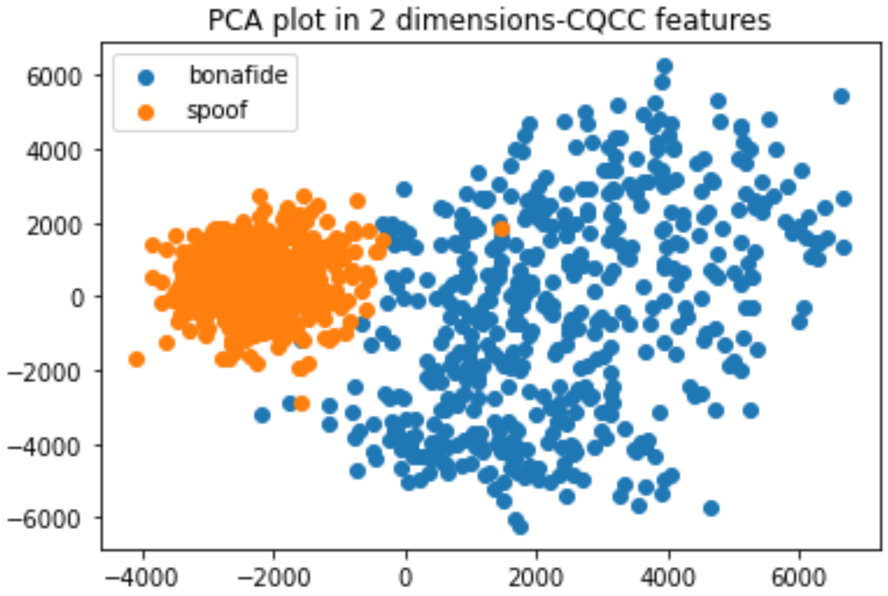
Results – Data Visualization

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







Results and Benchmarks

Metrics (Accuracy, Precision, & Recall) on Eval Data :

Type of model	Precision Score	Recall Score	Accuracy
Spectrogram (Model 1)	0.96	0.86	0.9528
CQCC features (Model 2)	0.75	0.62	0.77
CQCC features (Model 3)	0.80	0.61	0.80

Latency :

Model type	Feature generation	Prediction
Spectrogram (Model 1)	~4 µs	~50.3 ms
CQCC (Model 2)	~1.2 secs	~25.7 ms
CQCC (Model 3)	~1.2 secs	~26.5 ms