Assessing the Viability of Lightweight Deep Learning Models Through Audio Gender Classification

Lingyuan Yan[1], **Mariel Werner**[1]

Received February 4, 2026
Accepted July 20, 2026
Electronic access September 15, 2026

Large-scale deep learning models have become the dominant approach in audio NLP, yet their computational demands make deployment impractical in resource-constrained environments. This study evaluates five lightweight architectures (SVM, MLP, BiLSTM, MobileNet, CRNN) against a transformer baseline on an audio gender classification task. Each model was trained and fine-tuned on a 14,461-sample dataset of participants reading from a uniform phrase. Despite having significantly fewer parameters, the MobileNet model achieved 96.9% accuracy, surpassing the transformer baseline, which achieved 95.3%. The MobileNet model was also better when compared in noisy audio environments. Training time was also reduced by a factor of 2.5. Notably, the SVM classifier achieved 90% accuracy on the clean dataset with a latency of less than ten milliseconds. These results challenge the idea that transformer-based or large-scale models are always the best option in audio classification problems.

Introduction

Recent years have seen explosive growth in the AI industry, fueled by heavy investment from large tech companies. A large contributing factor was the discovery of transformer models, a class of large, computationally expensive deep learning architectures that have driven much of the recent progress in AI[1]. However, these transformers, and other complex deep learning models like them, require a large quantity of training time, data, and resources to create, giving existing large companies a decisive edge over startup companies or independent researchers. This leads to a vital question: are lightweight deep learning models able to achieve competitive performance with large deep learning models such as transformers? If not, what is the potential trade-off between performance and training resources?

To investigate this problem, we can compare lightweight and large-scale models specifically in the case of audio gender classification. Audio gender classification is a problem with real world applications and possesses enough complexity to be loosely generalized to other deep-learning and natural language processing problems involving sequential language data and binary classification. Audio natural language processing also has the benefit of being less explored than text natural language processing problems. Another reason audio gender classification was selected is because as artificial intelligence is integrated into our society, more humans will interact directly with computers, with speech being one of the most common and meaningful mediums of communication.

Throughout this paper, we use the term 'gender classification' to remain consistent with prior literature, though we acknowledge that the acoustic features detected by these models more precisely reflect biological sex than gender identity.

Literature Review

Audio feature extraction methods

When working with audio samples in machine learning, it is important to think about choosing the best method to extract the relevant information into a format that a model can easily process. This is because raw audio signals are complex and multidimensional, comprising temporal, amplitude, and frequency components which makes feature extraction difficult. The most common method is mel spectrograms: a 2-dimensional representation of the amount of energy at each time and frequency[2]. Many also use log power mel spectrograms which scale the representation in a way that better fits human hearing.

Next, Mel-Frequency Cepstral Coefficients apply a Discrete Cosine Transformation to the log mel filter bank energies at each time frame, decorrelating the frequency bins and compressing the representation[3]. This is a more compact representation that captures the general shape of the audio waves for less memory. Besides these two extraction methods, there are other features such as chroma STFTs (which capture pitch class information), zero-crossing rate (which measures how often the signal crosses zero) and RMS energy (which cap-

[1] *The Hill School, 860 Beech St, Pottstown, Pennsylvania, 19464, United States*

tures loudness over time)[4]. These other features are sometimes used in addition to MFCCs or spectrograms in combined feature sets, a common technique in speech analysis.

Gender classification methods

The field of audio gender classification already has a lot of prior research. Over the years people have developed numerous different approaches to audio gender classification. Early research found that pitch was a somewhat reliable indicator of gender, with males speaking at an average of 120 Hz and females speaking at an average of $\sim$210 Hz[5]. This approach is unreliable as there are many cases of overlap. The earliest machine learning approaches included applying statistical models such as GMMs and SVMs to MFCCs[6]. These present a simple but effective, lightweight approach. However, they are less effective with noisy samples[7]. Researchers have tested the effectiveness of several other models and approaches. For example, Kushwah et al. (2019) implemented CART, XGBoost, SVM and Random Forest models with an approximate accuracy of 89%[8]. A more advanced but complex approach is applying image classification models to spectrograms. By converting audio to a grayscale image, this approach could reuse the extensive existing research in image classification with models such as VGGs, ResNets, or Mobilenets. Alnuaim et al. (2022) explored audio gender classification using pretrained fine-tuned ResNet34 and ResNet50 architecture specifically, achieving 97.94% and 98.57% accuracy respectively[9]. They also tested countless other models. Notable results include a MLP classifier at 95.81% accuracy, a k-nearest neighbor classifier at 95.10% accuracy, and a deep neural network they designed which achieved 95.97%. Notably, this study used MFCC, mel spectrograms, chroma STFT, and two other features as inputs. Gong et al. (2021) explored a new technique, adapting the preexisting Vision Transformer (ViT) for audio, applying it to mel spectrograms without CNN components[10]. They achieved outstanding results on audio classification benchmarks including 95.6% accuracy on ESC-50 and 98.1% on Speech Commands V2. Transformer-based models have been applied to audio gender classification tasks specifically with strong results. For example, Burkhardt et al. (2023) achieved 91.1% accuracy using a fine-tuned Wav2Vec 2.0 model[11]. Studying audio gender classification in noisy environments is also a common approach in research as it better resembles some real-life situations. While achieving good performance in a clean audio environment is relatively easy and has been done multiple times, doing so in a noisy environment is much harder.

Lightweight vs. large-scale model comparisons

From simple statistical models to audio transformers, there is a wide spectrum of machine learning approaches to audio gender classification. These methods vary in both complexity and sophistication. With smaller models, they come with the benefit of requiring less memory and time to train, as well as less time to process an input. However, these models often perform worse than large-scale models.

To quantify memory consumption, we will consider the number of parameters in a model which dictates the amount of memory needed to fully load a model. This also affects the amount of VRAM needed if using a GPU, but that can more easily be remedied by lowering batch size. The most important benefit of lightweight models is lower inference latency, or the time delay between when a model receives an input and when it produces the corresponding output. This is important for real-time applications with tighter time constraints like live translation or sound detection.

There are some techniques that exist to close the gap between lightweight and large-scale models in hope of achieving the performance of large-scale models and the benefits of lightweight models. One such technique is transfer learning which involves using pretrained weights from existing well-performing models when beginning to train new models, reducing training cost and time[12].

Other techniques aim to compress the large models without losing performance. These include pruning (removing less relevant weights to reduce parameter count), quantization (reducing the numerical precision of model weights to reduce memory usage and speed up inference), and knowledge distillation (training a new, smaller model to mimic the behavior of a larger model[13–15]. However, this paper will only focus on traditional model training approaches without using these techniques.

For this paper, a lightweight model will be defined as one with less than 100,000 parameters. While this is very small compared to most modern standards, during initial tests, we found that small models of this size were more than capable of achieving decent results.

Methodology

Dataset

This study employs the Speech Accent Archive, a publicly available dataset with 3,038 labeled audio samples collected from speakers of diverse linguistic and geographic backgrounds[16]. Of these samples, we removed the two American sign language samples and the singular home sign sample because the manner of speaking was very different from the other samples, with the home sign sample being nearly unintelligible.

Each recording contains the same standardized passage:

"Please call Stella. Ask her to bring these things with her from the store: six spoons of fresh snow peas, five thick slabs

of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station."

This passage was deliberately constructed to encompass nearly all phonetic components of the English language. This makes this dataset very suitable for training audio classification models. Additionally, each entry in the dataset is labeled with information including the speaker's age, gender, birthplace, country of origin, age of English onset, length of English learning to name a few. This allows this dataset to be applied in a wide variety of audio natural language processing problems.

For the purposes of this study, each audio sample was first broken down into five-second clips which are used as the inputs for the models. Not only does this create shorter and incomplete samples that are better aligned with how gender classification would be applied in a real-life scenario with real-time classification, this also creates a uniform sample size that can be fed more easily into models. Additionally, the first and last seconds of the clip are ignored to avoid clips with large segments of white noise before or after the speaker has begun talking. In doing so, 14,461 clips were generated from the 3,038 original audio samples.

Finally, each clip is converted into mel spectrograms to use as inputs to our models. This is because each spectrogram is a two-dimensional image that can be treated as a greyscale image. To be more precise, we used a sample rate of 16000Hz, a STFT window length of 400 samples, a hop length of 100 samples, a FFT size of 512 points that are reduced to 128 bins. This converts each 5000-millisecond clip into uniform, 128x800 two-dimensional arrays that can be easily used as machine learning data (Figure 1). This also addresses the problem varying sample lengths if we used the original samples (Figure 2). Lastly, we used log amplitude scaling and normalized the result, meaning that the values of all inputs are readjusted so that they only range from 0 to 1, which allows for faster and more efficient model training.

Other features such as MFCCs were not used considering that several of the models in our study rely on matrices as model inputs. Incorporating other inputs would mean fundamentally changing the architecture of these models.

With this dataset of five-second clips, we created two more copies of the dataset by adding a randomly selected clip from the MUSAN dataset to each clip in the dataset at five and ten decibels[17]. The random clip added is kept consistent across both datasets. This creates three total datasets with three different amounts of interference: zero decibels, five decibels, and ten decibels. These ranges are selected to match prior work in the closely related field of speech emotion recognition where MUSAN noise augmentation was used extensively[18]. These ranges are realistic for real-world applications and still provide a big challenge to the models without being completely unintelligible.

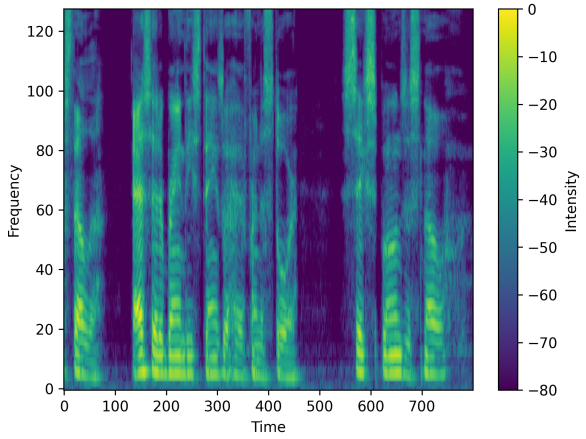

Figure. 1 The mel spectrogram of a five-second clip taken from english639. It is a 128x800 matrix.

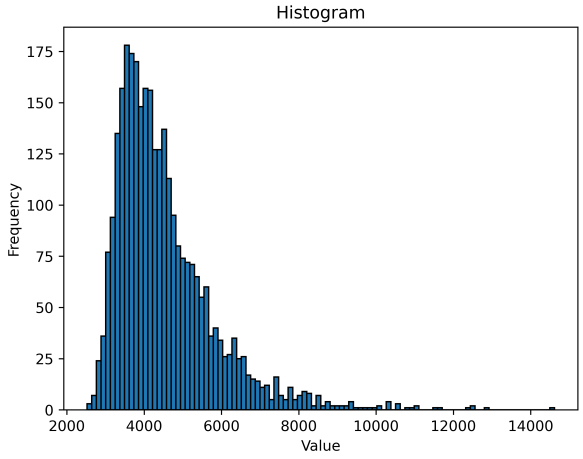

Figure. 2 A distribution of the length of full-length samples after being converted to spectrograms.

5-Fold Validation

This study also employs 5-fold cross-validation[19]. This involves splitting the dataset into 5 folds to train the model five times using the same dataset. In each fold, 10% of the dataset is held out as the test set. From the remaining 90%, approximately 80% of the total dataset is used for training and 10% for validation. The exact sizes vary slightly due to rounding, but the training set contains approximately 11,570 clips, while the validation and test sets each contain approximately 1,446 clips.

Each fold uses a different test partition, and there is no over-

lap between the five test sets. As a result, every sample is evaluated exactly once as test data across the five folds. Averaging the results across all folds provides a more reliable estimate of model performance than a single train-test split.

Splitting data at the clip level introduces a risk of data leakage, as clips originating from the same speaker could be assigned to different datasets. To prevent this, the dataset is first shuffled at the speaker level. When a partition would separate clips from the same speaker, all clips from that speaker are assigned to a single dataset. This ensures that no speaker appears in more than one of the training, validation, or test sets within a fold.

A 10% test split was used instead of the more common 20% split to maximize the amount of data available for training. Ten-fold cross-validation was also considered but was not used because it would approximately double the computational cost and training time compared to 5-fold cross-validation.

Models

This section provides descriptions of each of the six classification models evaluated in this work. All models share the same input format, a single-channel feature map of shape $(B, 1, 128, 800)$, and produce a single raw logit $(B, 1)$ used with binary cross-entropy loss. The architecture spans a spectrum from simple linear projections to a full transformer encoder, enabling ablation of inductive biases across spatial, temporal, and attention-based axes.

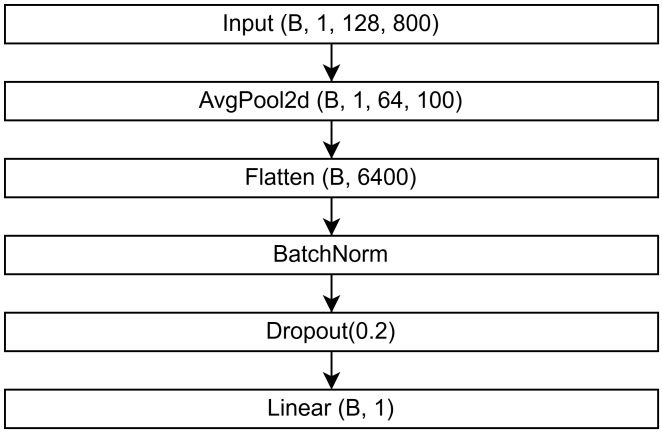

Figure. 3 A graph showing the architecture of the SVM model. The shape of each hidden layer is included to increase readability.

Support Vector Machine Classifier (SVM)

The SVM Classifier is the simplest model in the suite, included as a baseline to establish what a purely linear approach can achieve[20]. The spectrogram is first compressed using average pooling — which takes the mean value across small regions, reducing the input while preserving the overall energy distribution of each frequency band — then flattened into a single 6,400-dimensional vector[21]. A BatchNorm layer standardizes this vector before the final linear projection to a single logit, preventing any one frequency band with unusually high variance from dominating the gradient during training[22]. If more complex models do not substantially outperform this one, it suggests the classification boundary is linearly accessible in the pooled frequency-time representation and does not require deep feature learning.

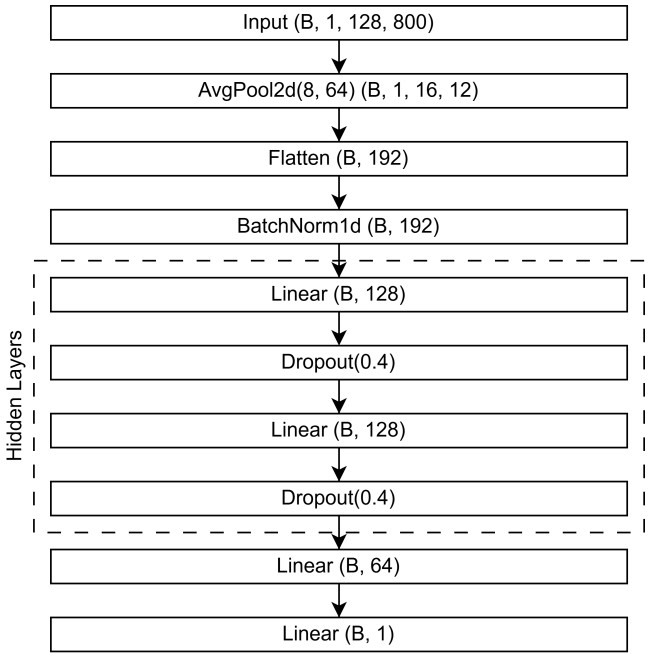

Figure. 4 A graph showing the architecture of the MLP model. The shape of each hidden layer is included to increase readability.

Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron extends the linear baseline by stacking three fully connected hidden layers between the pooled representation and the output, following the universal approximation framework that established the expressive power of such networks[23,24]. Each hidden layer applies a learned linear transformation followed by a ReLU activation — which sets negative values to zero, introducing the nonlinearity that allows the model to learn curved decision boundaries rather than a straight line. Dropout is applied after each hidden layer at $p = 0.4$, randomly disabling 40% of the units during training to discourage the network from relying on any single feature[25]. Like the SVM Classifier, the MLP operates on a pooled and flattened spectrogram with no awareness of

spatial or temporal structure, making it a useful baseline between the linear baseline and the models that follow.

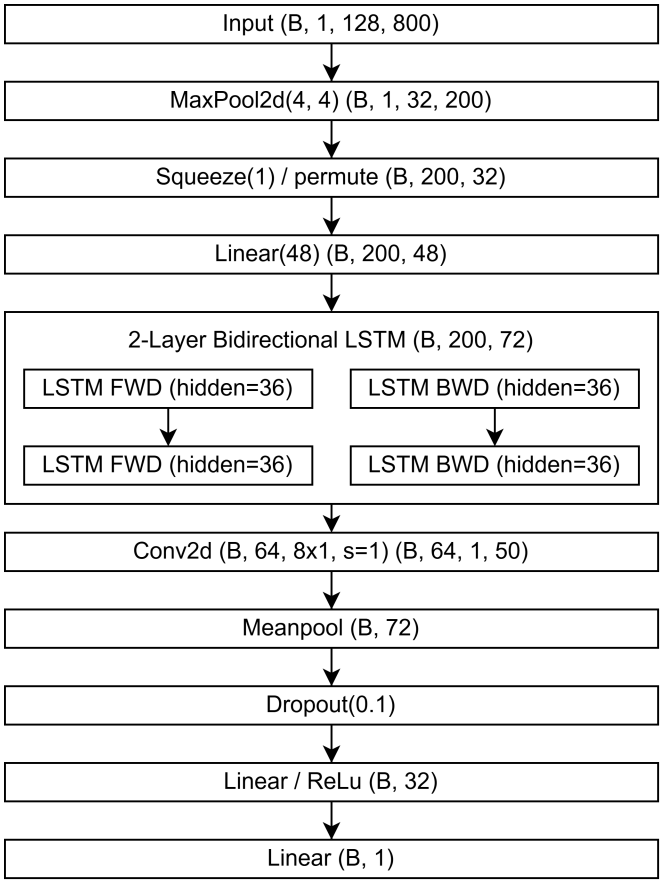

Figure. 5 A graph showing the architecture of the BiLSTM model. The shape of each hidden layer is included to increase readability.

Bidirectional Long Short-Term Memory (BiLSTM).

The BiLSTM Classifier takes a different approach from the CRNN: rather than using a CNN to extract features first, the spectrogram is compressed with a simple parameter-free pooling step and a single learned frequency projection before being handed directly to the recurrent model. The core is a 2-layer bidirectional LSTM — an extension of standard recurrent networks that reads the sequence both forwards and backwards, so each time-step is informed by both past and future context simultaneously [26,27]. Rather than using only the network's final summary state, all 200 time-step outputs are averaged together — an aggregation strategy that has been shown to outperform last-state approaches for classification tasks without a natural sequence endpoint, and one that is particularly natural here since neither the forward nor backward pass produces a privileged "final" summary [28]. A lightweight dropout rate of $p = 0.1$ is used throughout; at $\tilde{6}$0k parameters, aggressive regularization risks underfitting rather than preventing overfitting [25]. Comparing this model against the CRNN provides a direct ablation of the value of convolutional feature extraction before the recurrent stage.

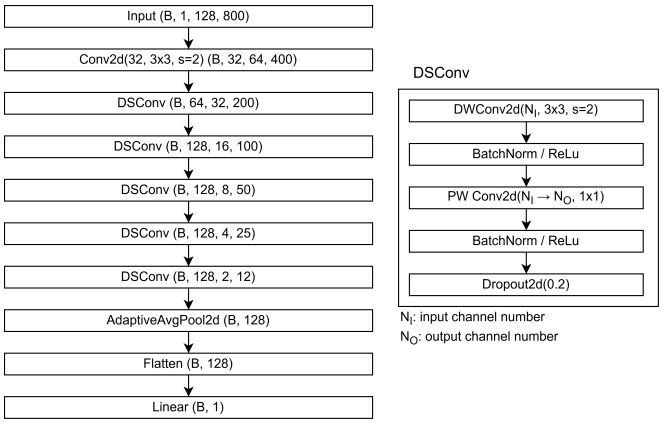

Figure. 6 A graph showing the architecture of the MobileNet model. The shape of each hidden layer is included to increase readability.

MobileNet

MobileNetV1 is a convolutional neural network designed for efficiency [29]. Convolutional layers scan the input with small, learned filters to detect local patterns — edges, textures, frequency contours — at every position in the spectrogram. The key innovation is the depthwise separable convolution, which splits a standard convolution into two cheaper steps: a depthwise convolution that detects spatial patterns within each channel independently, followed by a pointwise 1×1 convolution that combines information across channels. This achieves similar representational power at roughly 8–9$\times$ lower computational cost. The network processes the spectrogram through a stem convolution and five such blocks, each halving the spatial resolution via stride-2 convolution while expanding the channel count, before a global average pool collapses the spatial dimensions into a fixed-length vector for classification. Dropout2d — which drops entire feature channels rather than individual values — is applied inside every block as a regularization strategy across the full feature hierarchy [30]. In our implementation of the model, we reduced the model complexity even more from nine million parameters to $\sim$64k by cutting down on layers and channels. However, depthwise separable convolution layers are still a large part of the design, making it an even more lightweight version of the original MobileNetV1 model.

Convolutional Recurrent Neural Network (CRNN)

The CRNN combines two complementary ideas: a CNN to extract local spectral features, and a GRU to model how those features evolve over time, a hybrid approach that has shown

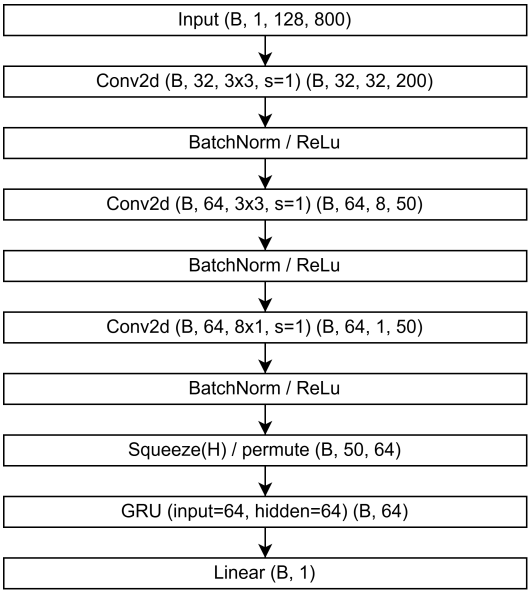

Figure. 7 A graph showing the architecture of the CRNN model. The shape of each hidden layer is included to increase readability.

strong results for audio classification tasks[31]. The CNN processes the spectrogram through three blocks, progressively reducing the spatial dimensions until the frequency axis is fully collapsed by a learned convolution, leaving a sequence of 50 feature vectors along the time axis. This learned frequency collapse, using a convolution whose kernel exactly spans the remaining height, is preferable to fixed pooling because the network can learn which frequency combinations are most discriminative rather than averaging or taking the maximum blindly. The GRU, a recurrent network that maintains a running memory through learned gating mechanisms, then reads through this sequence step by step, accumulating temporal context before its final hidden state is passed to the classification head[32]. The motivation is that some patterns may only be meaningful in the context of what came before, which a purely spatial model cannot capture.

Audio Spectrogram Transformer (AST)

The AST adapts the Vision Transformer for audio classification and is the most architecturally distinct model in the suite[10,33]. Rather than convolutions or recurrence, it uses self-attention — a mechanism that allows every part of the input to directly compare itself with every other part in a single operation, capturing long-range relationships that local or sequential models can only reach by propagating information through many steps[1]. The spectrogram is first divided into non-overlapping 16×16 patches and each is projected into a 192-dimensional vector, giving 400 tokens in total. A learnable CLS token is prepended and learned positional embeddings are added to all tokens to inject spatial ordering, since self-attention is inherently indifferent to the position of its inputs. The sequence then passes through six transformer blocks, each of which refines the representation using multi-head attention and a small feed-forward network, with pre-norm LayerNorm applied before each sub-layer for training stability[34]. After the final block, the CLS token's state is used for classification, having accumulated information from all 400 patch tokens through the attention mechanism. The AST is substantially larger than the other five models ($\sim$2.7M parameters), making it the most expressive but also the most data-hungry architecture in this study.

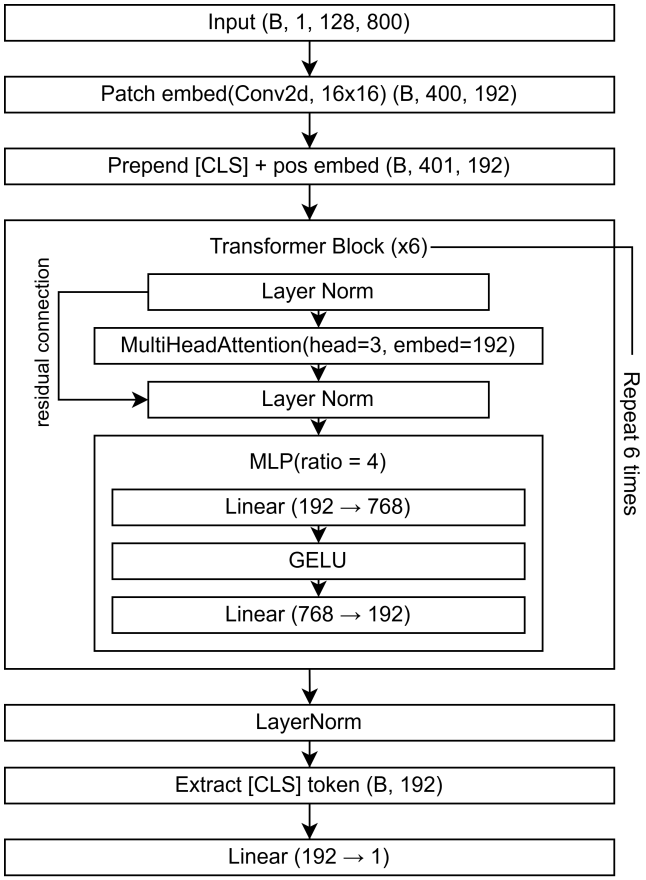

Figure. 8 A graph showing the architecture of the AST model. The shape of each hidden layer is included to increase readability.

Hyperparameters

For the sake of reproducibility, we have included the learning rate, weight decay, batch size, and epochs for each model. All models used the Adam optimizer and a learning rate scheduler[35]. All models have a final sigmoid activation layer and use Binary Cross Entropy Loss function. All models are trained for 10 epochs with a batch size of 128 samples.

Table 1 The list of hyperparameter values used by each model to achieve the stated output.

Model Name	Learning rate	Weight decay	Factor	Patience
SVM	1e−4	1e−4	0.5	2
MLP	1e−3	1e−3	0.5	2
BiLSTM	1e−3	1e−4	0.1	2
MobileNet	1e−3	1e−3	0.1	2
CRNN	1e−3	1e−2	0.1	2
AST	1e−4	1e−4	0.1	2

Table 2 Mean test accuracy and standard deviation of test accuracy results for each model type with no added noise. The highest performing models are in bold.

Model Type	Mean Test Accuracy	Test Accuracy Standard Deviation
SVM	90.4551%	0.0204918
MLP	91.8145%	0.0097196
BiLSTM	95.8718%	0.013224
MobileNet	**96.9755%**	**0.00513657**
CRNN	**97.0510%**	**0.0124657**
AST	95.3207%	0.012056

Table 3 Mean test accuracy and standard deviation of test accuracy results for each model type with 5 decibels of added noise. The highest performing models are bolded.

Model Type	Mean Test Accuracy	Test Accuracy Standard Deviation
SVM	77.7680%	0.00224572
MLP	79.4722%	0.0157028
BiLSTM	91.0598%	0.011074
MobileNet	**93.4919%**	**0.0161355**
CRNN	**94.1568%**	**0.010624**
AST	92.7524%	0.014565

Table 4 Mean test accuracy and standard deviation of test accuracy results for each model type with 10 decibels of added noise. The highest performing models are bolded.

Model Type	Mean Test Accuracy	Test Accuracy Standard Deviation
SVM	73.2922%	0.0133382
MLP	75.2707%	0.00761601
BiLSTM	74.0317%	0.088100
MobileNet	**91.5813%**	**0.0140017**
CRNN	**91.8443%**	**0.0113606**
AST	89.4923%	0.015336

Table 5 Mean Training time, inference latency and parameter count for each model.

Model Type	Mean Training Time	Inference latency	Parameter Count
SVM	14.4743667 seconds	9.5247 milliseconds	19,201
MLP	18.5767333 seconds	15.4152 milliseconds	49,921
BiLSTM	70.207354 seconds	50.2360 milliseconds	60,401
MobileNet	59.4242333 seconds	34.9541 milliseconds	48,257
CRNN	130.715667 seconds	47.3236 milliseconds	61,409
AST	152.973667 seconds	82.6870 milliseconds	2,796,289

Discussion

Despite both having far fewer parameters, both the MobileNet and CRNN models surpass the AST approach, regardless of noise level (Table 2-4). Both models also surpassed results reported in Alnuaim et al. (2022) by models with more features and much greater computational resources[9]. This clearly shows that for some problems, lightweight deep learning approaches are competitive with, and can even surpass, the larger-scale models including attention-based approaches. This pattern has been observed elsewhere in the literature. Anidjar et al. (2025) found that a traditional spectrogram-based method outperformed a Wav2Vec 2.0 transformer on a gender classification task in the Russian language, suggesting that architectural complexity does not necessarily yield better performance in audio classification[36].

In addition to better performance, both models also had significantly lower inference latency. However, the CRNN model only shows a slight improvement in training time compared to the AST while the MobileNet model was more than 2.5 times faster. With lower inference latency, training time, and parameter count, and performing barely lower than the CRNN model, MobileNet is the best performing model overall (Table 5).

The SVM and MLP classifiers are also of note, as they performed well on the clean training data, achieving a performance of 90% and 91% accuracy respectively (Table 2). However, their accuracy drops sharply when noise is introduced, meaning that they are only viable in certain conditions (Table 3-4). While the SVM classifier has a far lower latency compared to other models, CRNN and MobileNet classifiers are likely preferable in nearly all real-life applications unless there are very tight inference latency constraints.

The data also has a very low standard deviation of approximately 0.01 across all models and noise levels, showing our findings to be very reliable (Table 2-4). The sole exception is the BiLSTM model with a standard deviation of 0.08 (Table 4). Notably, the model's performance varied greatly across folds with a high of 81% accuracy and a low of 67% accuracy. Since this only occurred on the dataset with the highest noise

level, one explanation could be the BiLSTM's lack of a convolutional front-end, meaning that the model has no mechanism for local noise filtering. This would make the model much less robust in noisy environments.

Limitations

There were several limitations to this study. The most important is the effect of hyperparameter tuning on model performance. Small changes in hyperparameter values can drastically change the model's performance. Slightly different hyperparameters could have improved the performance of all models. However, the selected hyperparameters represent a practical optimum under realistic resource constraints. Regardless, the results should be treated as a lower bound on what is achievable.

Additionally, other transformer models are often significantly larger but infeasible for this paper due to hardware constraints. To address this, we implemented a larger AST classifier at 9.2 million parameters which did not perform better than the model used in our paper (Table 6). This shows that simply scaling the model did not improve the performance, further demonstrating that a bigger model does not mean better performance.

Table 6 The accuracy of the 9.2 million parameter AST model on three different levels of noise.

Model	Clean	5 decibels	10 decibels
AST (9.2 m parameters)	95.8051%	88.3629%	74.7087%

While these results are specific to audio binary classification NLP tasks, they are sufficient to show that there exist problems where lightweight architecture is competitive with transformer-based approaches, particularly in other domains with similar characteristics.

Conclusion

Having tested five different lightweight deep learning models against a transformer baseline, all architectures performed well despite strict parameter constraints. While conventional wisdom favors larger architectures such as transformers, the results from this study challenge this idea[37]. Additionally, this shows that task-specific lightweight architecture such as CRNNs and MobileNets remain competitive alternatives to transformer models. Even in noisy audio environments, the transformer baseline did not surpass these two lightweight approaches, demonstrating their robustness. While thorough, many avenues for future research remain such as incorporating comparisons with transfer learning, testing on more diverse datasets, or performing a similar analysis in a different domain such as computer vision.

Acknowledgements

I would like to acknowledge my parents for supporting me throughout my life and my education, and my teachers for teaching me curiosity and an appreciation for the pursuit of knowledge.

References

1 *A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems. Vol. 30, pg. 5998–6008, 2017,* https://doi.org/10.48550/arXiv.1706.03762.

2 *S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, K. Wilson. CNN architectures for large-scale audio classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. pg. 131–135, 2017,* https://doi.org/10.1109/ICASSP.2017.7952132.

3 *B. Logan. Mel frequency cepstral coefficients for music modeling. Proceedings of the International Symposium on Music Information Retrieval, 2000.*

4 *I. Ridae. Feature extraction methods for temporal signal recognition: An overview. arXiv preprint arXiv:1812.01780, 2018.* https://doi.org/10.48550/arXiv.1812.01780.

5 *D. H. Klatt, L. C. Klatt. Analysis, synthesis, and perception of voice quality variations among female and male talkers. Journal of the Acoustical Society of America. Vol. 87, pg. 820–857, 1990,* https://doi.org/10.1121/1.398894.

6 *H. Harb, L. Chen. Voice-based gender identification in multimedia applications. Journal of Intelligent Information Systems. Vol. 24, pg. 179–198, 2005,* https://doi.org/10.1007/s10844-005-0322-8.

7 *R. Togneri, D. Pullella. An overview of speaker identification: Accuracy and robustness issues. IEEE Circuits and Systems Magazine. Vol. 11, pg. 23–61, 2011,* https://doi.org/10.1109/MCAS.2011.941079.

8 *S. Kushwah, S. Singh, K. Vats, V. Nemade. Gender identification via voice analysis. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. Vol. 5, pg. 746–753, 2019,* https://doi.org/10.32628/CSEIT1952188.

9 *A. A. Alnuaim, M. Zakariah, C. Shashidhar, W. A. Hatamleh, H. Tarazi, P. K. Shukla, R. Ratna. Speaker gender recognition based on deep neural networks and ResNet50. Wireless Communications and Mobile Computing. Vol. 2022, pg. 4444388, 2022,* https://doi.org/10.1155/2022/4444388.

10 *Y. Gong, Y.-A. Chung, J. Glass. AST: Audio spectrogram transformer. Interspeech. pg. 571–575, 2021,* https://doi.org/10.21437/Interspeech.2021-698.

11 *F. Burkhardt, J. Wagner, H. Wierstorf, F. Eyben, B. Schuller. Speech-based age and gender prediction with transformers. arXiv preprint arXiv:2306.16962, 2023.* https://doi.org/10.48550/arXiv.2306.16962.

12 *S. J. Pan, Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering. Vol. 22, pg. 1345–1359, 2010,* https://doi.org/10.1109/TKDE.2009.191.

13 *Y. LeCun, J. S. Denker, S. A. Solla. Optimal brain damage. Advances in Neural Information Processing Systems. Vol. 2, pg. 598–605, 1990.*

14 *B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko. Quantization and training of neural networks for efficient*

integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pg. 2704–2713, 2018, [https://arxiv.org/abs/1712.05877.

15 *G. Hinton, O. Vinyals, J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.* [https://doi.org/10.48550/arXiv.1503.02531.

16 *S. Weinberger. Speech accent archive.* [https://accent.gmu.edu/browse_language.php*](https://accent.gmu.edu/browse_language.php).*

17 D. Snyder, G. Chen, D. Povey. MUSAN: A music, speech, and noise corpus. arXiv preprint arXiv:1510.08484, 2015. [https://doi.org/10.48550/arXiv.1510.08484.

18 R. Pappagari, J. Villalba, P. Zelasko, L. Moro-Velazquez, N. Dehak. An augmentation method for speech emotion recognition. arXiv preprint arXiv:2010.14602, 2020. [https://doi.org/10.48550/arXiv.2010.14602.

19 R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Vol. 2, pg. 1137–1143, 1995.

20 C. Cortes, V. Vapnik. Support-vector networks. Machine Learning. Vol. 20, pg. 273–297, 1995, [https://doi.org/10.1007/BF00994018.

21 Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Y. E. Rachmad. Gradient-based learning applied to document recognition. Proceedings of the IEEE. Vol. 86, pg. 2278–2324, 1998, [https://doi.org/10.1109/5.726791.

22 S. Ioffe, C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the International Conference on Machine Learning. Vol. 37, pg. 448–456, 2015, [https://arxiv.org/abs/1502.03167.

23 D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors. Nature. Vol. 323, pg. 533–536, 1986, [https://doi.org/10.1038/323533a0.

24 K. Hornik, M. Stinchcombe, H. White. Multilayer feedforward networks are universal approximators. Neural Networks. Vol. 2, pg. 359–366, 1989, [https://doi.org/10.1016/0893-6080(89)90020-8.

25 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. Vol. 15, pg. 1929–1958, 2014.

26 M. Schuster, K. K. Paliwal. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. Vol. 45, pg. 2673–2681, 1997, [https://doi.org/10.1109/78.650093.

27 A. Graves, J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Proceedings of the International Joint Conference on Neural Networks. Vol. 18, pg. 602–610, 2005, [https://doi.org/10.1016/j.neunet.2005.06.042.

28 T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, C. Zhang. DiSAN: Directional self-attention network for RNN/CNN-free language understanding. Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32, 2018, [https://doi.org/10.1609/aaai.v32i1.11941.

29 A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. [https://doi.org/10.48550/arXiv.1704.04861.

30 J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler. Efficient object localization using convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. arXiv preprint [https://arxiv.org/abs/1411.4280v3](https://arxiv.org/abs/1411.4280v3.

31 K. Choi, G. Fazekas, M. Sandler, K. Cho. Convolutional recurrent neural networks for music classification. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. pg. 2392–2396, 2017, [https://arxiv.org/abs/1609.04243.

32 K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1406.1259, 2014. [https://doi.org/10.48550/arXiv.1406.1259.

33 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations, 2021, [https://arxiv.org/abs/2010.11929.

34 R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, T.-Y. Liu. On layer normalization in the transformer architecture. Proceedings of the International Conference on Machine Learning. Vol. 119, pg. 10524–10533, 2020, [https://arxiv.org/abs/2002.04745.

35 D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations, 2015, [https://arxiv.org/abs/1412.6980.

36 O. H. Anidjar, R. Yozevitch. Transformer-based language-independent gender recognition in noisy audio environments. Scientific Reports. Vol. 15, pg. 14421, 2025, [https://doi.org/10.1038/s41598-025-99011-x.

37 K. Zaman, M. Sah, C. Direkoglu, M. Unoki. A survey of audio classification using deep learning. IEEE Access. Vol. 11, pg. 106620-106649, 2023, [https://doi.org/10.1109/ACCESS.2023.3318015.