Waves to Pixels: A Study of Neural Networks in Relation to Audio Encoding for Classification and Diagnosis

Joshua M. Daugherty

University of Alabama at Birmingham

The recent growth of artificial intelligence has led to significant advances in Neural Networks, especially in image recognition and medical diagnostics. Convolutional Neural Networks (CNNs) have become a cornerstone in the analysis of medical images for conditions such as pneumonia, cancer, and other various pulmonary diseases. In parallel, Vision Transformers (ViTs), which were originally developed for natural language processing, are gaining traction in computer vision due to their ability to capture relationships within images through tokenization and self-attention mechanisms. The purpose of this paper is to show understanding of CNNs, ViTs, and OPERA, a recently published framework for developing and benchmarking models with their potential use for classification and diagnosis in mind.

Keywords: Neural network, vision, convolution, transformer, audio encoding, OPERA, classification, medical imaging, diagnosis

1 A Review of Neural Networks

Neural Networks (NNs) take their name and structural inspiration from biological neurons in the human brain, which are responsible for processing and communicating information [1, 2]. Much like their biological counterparts, artificial neural networks consist of interconnected layers of nodes, or **neurons**, which pass data between layers to model complex patterns and relationships.

1.1 Early Neural Networks

One of the earliest forms of a neural network is the **perceptron** [1], as introduced by Frank Rosenblatt in 1958. The perceptron models a single neuron, taking multiple weighted inputs and producing a binary output. If the sum of the weighted inputs exceeds a predetermined threshold, the perceptron 'fires,' outputting a '1'; otherwise, it outputs a '0.' The bottleneck of the perceptron is that it can only solve linearly separable problems, which lead to further innovations, such as multi-layer perceptrons (MLPs) to handle more complex, non-linear data.

1.2 Shallow vs. Deep Networks

The **depth** of a neural network refers to the number of hidden layers between the input layer and the output layer. In shallow networks, there are typically one or two hidden layers, while deep neural networks (DNNs) contain many hidden layers - sometimes even hundreds or thousands. As depth increases, the model is allowed to learn and model more complex and abstract representations of the input data.

The concept of deep learning networks dates back to the first deep MLP, introduced in 1967 by Shun'ichi Amari[3]. In experiments conducted by Amari's graduate student, H. Saito, a five-layer MLP with two modifiable layers was able to learn representations for classifying non-linearly separable patterns [4].

1.3 Backpropagation

Backpropagation is the key technique in training these neural networks - done by adjusting weight vectors to minimize errors in predictions. The two steps of backpropagation are the **forward pass**, where data is fed forward through the network to generate an output, and the **backward pass**, where the error is propagated through the network in reverse. This process is repeated over many iterations (epochs) until the error is sufficiently minimized.

The modern backpropagation algorithm was independently developed in the 1970s by Seppo Linnainmaa, Paul Werbos, and later popularized by David E. Rumelhart et al. In 1970, Linnainmaa introduced reverse-mode differentiation [5], followed by Werbos's contributions in 1971 [6], and Rumelhart's 1986 work which explicitly demonstrated its effectiveness in training multilayer networks [7].

Mathematically, backpropagation relies on the chain rule of calculus to compute the gradients of the loss function with respect to each weight in the network. This involves calculating partial derivatives of the loss function, typically using the chain rule to backtrack through each layer. Specifically, for a given layer l, the weight update is computed as:

$$\Delta w_l = -\eta \frac{\partial L}{\partial w_l},$$

where L is the loss function, w_l are the weights in the current layer, and η is the learning rate. The gradient $\frac{\partial L}{\partial w_l}$ is computed by applying the chain rule iteratively, as the error is propagated backward through the layers of the network.

1.4 Activation Functions

Activation functions are what introduce the ability for models to represent non-linearity and learn complex patterns. There exists a wide variety of activation functions, each with unique benefits and downfalls in terms of computation, stability, and performance.

The sigmoid function ($\sigma(x)$) was first used by Rosenblatt in his perceptron model [1]. The tanh function, introduced as an alternative to the sigmoid, has roots in early network studies, and is commonly used to avoid the vanishing gradient problem [8]. The ReLU function, introduced by Nair and Hinton [9], restructured deep learning

Activation Function	Formula	Advantages and Disadvantages
Sigmoid (Logistic)	$\sigma(x) = \frac{1}{1+e^{-x}}$	**+ :** Smooth gradient, good for binary classification. **- :** Vanishing gradients, slow for deep networks.
Tanh (Hyperbolic Tangent)	$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	**+ :** Zero-centered output **- :** Still suffers from vanishing gradients.
ReLU (Rectified Linear Unit)	$\text{ReLU}(x) = \max(0, x)$	**+ :** Efficient, solves vanishing gradient, fast training. **- :** "Dying ReLU" problem (neurons stuck at zero).
Leaky ReLU	$\text{Leaky ReLU}(x) = \max(\alpha x, x)$	**+ :** Solves the "dying ReLU" problem, simple to implement. **- :** Small slope for negative values, inefficient training.
Softmax	$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$	**+ :** Outputs are probability distribution (multi-class). **- :** Computationally expensive for large output layers.

Table 1: Comparison of Commonly Used Activation Functions

with its simplicity and efficiency. To address the "dying neuron" problem in ReLU, Leaky ReLU was introduced by Maas et al. in 2013 [10], which allows small negative values for negative inputs. Finally, the Softmax function, commonly used for multi-class classification, was introduced by Johnson in 1986 [11]. As shown in the Table 1 formula, the softmax function makes use of the exponential to ensure that the outputs sum to one, allowing them to be interpreted as probabilities, which is key for its use in tasks like image classification and language modeling [11, 12].

1.5 Supervised, Unsupervised, and Self-Supervised Learning

Neural networks can be trained under various learning paradigms: **Supervised learning**, where models are trained with labeled datasets (input-output pairs) [13], **Unsupervised learning**, which discovers patterns in data without labels [14], and **Self-Supervised learning**, a hybrid approach where the model generates its own labels from raw data [15]. Self-supervised learning has gained importance lately, especially in areas like audio classification, where labeled data is scarce.

2 Convolutional Neural Networks

Building upon the progress made in NNs, **Convolutional Neural Networks** (CNNs) emerged as a special class of deep learning models that were well-suited for image processing tasks. Introduced by Yann LeCun in the late 1980s [13], CNNs take advantage of the spatial structure within an image by using convolution layers to extract features. These convolutional models have now been adopted in applications from object detection to medical imaging.

2.1 Components of CNNs

A CNN is composed of several distinct layers that work together to process and learn from image data. The key components of a CNN include:

- **Convolutional Layer:** Performs the convolution operation to extract features from the input image, using filters or kernels to detect edges, textures, and more.
- **Pooling Layer:** Reduces the dimensionality of the feature maps, usually through max-pooling, helping to retain important features while reducing computational

complexity.

- **Activation Function:** Applies a non-linear function, such as ReLU (Rectified Linear Unit), to introduce non-linearity to the network and help the model learn more complex patterns.
- **Fully Connected Layer:** Flattens the input and connects it to output neurons, allowing the network to make predictions based on the extracted features.
- **Normalization Layer:** Standardizes the inputs to help stabilize and speed up the learning process, often using techniques like Batch Normalization.
- **Dropout Layer:** Randomly removes a subset of neurons during training to prevent over-fitting and improve generalization.

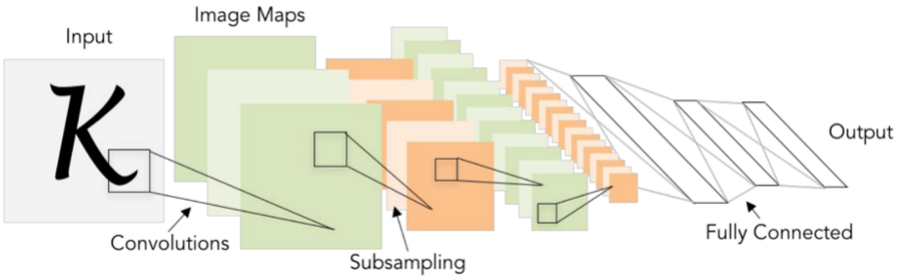


Figure 1: LeNet-5, a CNN introduced by LeCun, which uses convolutional and subsampling (pooling) layers process an input, represent the data, and finally return an output (classification) [13].

2.2 Modern Architectures & Frameworks

Mostly within the past decade or so, numerous deep learning architectures and frameworks have been introduced, and have continued pushing the boundaries of image and audio classification tasks, notably in the medical field.

2.2.1 AlexNet (2012):

Introduced by Alex Krizhevsky, AlexNet was a breakthrough CNN architecture that won the ImageNet competition. It demonstrated the power of deep convolutional networks, employing ReLU activations and dropout to reduce overfitting [16].

2.2.2 VGG (2014):

The VGG architecture, developed by the Visual Geometry Group at Oxford, uses a deep stack of small convolutional filters (3x3) to capture image features. It showed that increasing depth (around 16-19 layers) could lead to better performance, although at the cost of higher computational requirements [17].

2.2.3 ResNet (2015):

ResNet, or Residual Networks, introduced by He et al., tackled the problem of vanishing gradients by utilizing skip connections. This architecture allows for deeper networks without performance degradation, leading to state-of-the-art results in many computer vision tasks [18].

2.2.4 EfficientNet (2019):

EfficientNet, proposed by Tan and Le, introduced a scaling method that adjusts the width, depth, and resolution of the network simultaneously, leading to more efficient models that outperform many large architectures with fewer parameters [19].

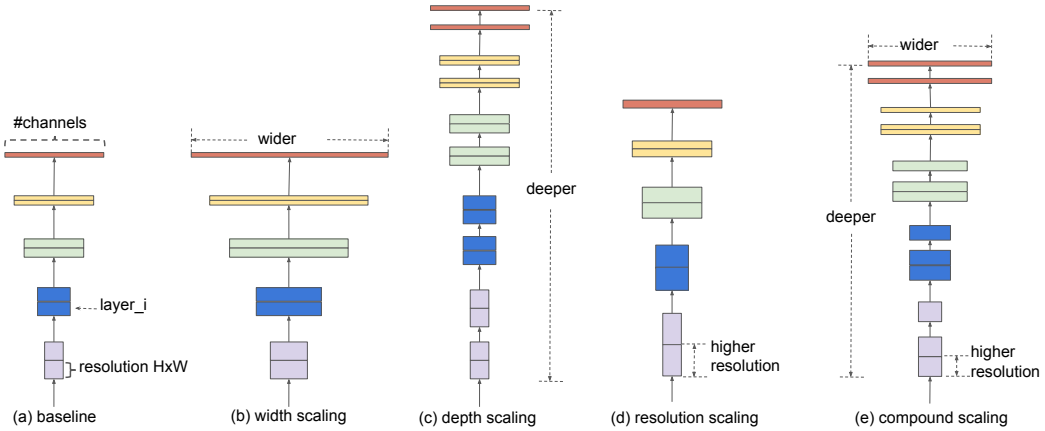


Figure 2: **Model Scaling;** (b)-(d) are standard model scaling techniques that increase one dimension, while (e) is the proposed compound scaling of all dimensions with a fixed ratio [19].

2.2.5 OPERA (2023):

The OPERA (**OPE**n **R**espiratory **A**coustic) framework is a model pretraining and benchmarking system, designed specifically for respiratory sound analysis [20]. It provides a standardized method for pretraining and benchmarking models for classifying acoustics signals, which has great potential in a healthcare context.[1]

2.3 Current Applications

CNNs have revolutionized the field of medical imaging, especially in diagnosing and classifying diseases. One key area is pulmonary disease diagnosis. CNNs are employed in processing medical images such as X-rays and CT scans to identify conditions like pneumonia, lung cancer, and Chronic Obstructive Pulmonary Disease (COPD). Studies have shown that CNNs can outperform traditional diagnostic methods, offering more accurate and faster detection by extracting features that may not be visible to the human eye [21, 22].

In addition to image-based diagnostics, CNNs are increasingly being applied to respiratory sound classification. Models developed under frameworks like OPERA classify acoustic signals, identifying respiratory conditions from coughing, breathing, and other sounds. This application is especially valuable in low-resource settings where imaging equipment may not be available, but audio recording devices can be used to capture useful diagnostic data. These models demonstrate the potential for expanding the reach of healthcare through simple, accessible tools.

[1] This framework is going to be revisited in depth after the ViT discussion, as it is a key component of thesis research for both this fall semester and the coming spring semester.

CNNs are also capable of transforming audio data such as heart and lung sounds into spectrograms for classification, detecting conditions like arrhythmia and lung abnormalities [23, 24]. This technique provides non-invasive, cost-effective diagnostic options. Beyond audio classification, CNNs are also being implemented in automated radiology diagnostics, such as tumor detection and the analysis of abnormalities in MRI, ultrasound, and retinal imaging, offering high precision and faster, more reliable medical decisions [25, 26].

3 Vision Transformers

Vision Transformers (ViTs) have more recently emerged as a powerful alternative to CNNs in tasks like image classification, object detection, and more. Introduced by Dosovitskiy et al. [27], ViTs split images into smaller patches, treat each patch as a unique token, and leverage the transformer architecture that was originally developed for NLP. This allows ViTs to model global relationships in the data more effectively compared to CNNs, which primarily focus on local patterns.

Several notable advancements have built on the initial ViT framework, such as the **Swin Transformer** [28], which introduces a hierarchical architecture for better handling of high-resolution images, and **HTS-AT** [29], which adapts the ViT for audio classification tasks by transforming audio signals into image-like representations.

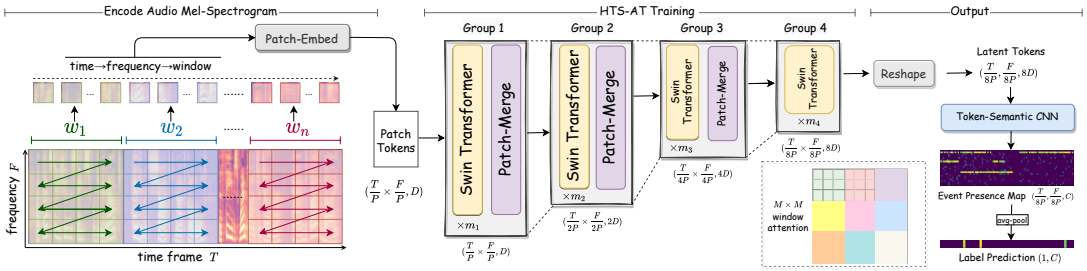


Figure 3: Model architecture of HTS-AT.

3.1 Patch Embeddings and Tokenization

ViTs split the input image into smaller patches, usually of equal size (i.e., 16×16), and each patch is then flattened into a 1-D vector. These vectors are linearly embedded into a higher-dimensional space, creating a sequence of image tokens. Since transformers lack a built-in notion of spatial structure, positional encodings are added to these tokens to preserve the spatial information in the input image [27].

3.2 Self-Attention Mechanism

Likely the most important component of ViTs is the self-attention mechanism, which allows the model to weigh the importance of different patches relative to each other. Each token (patch embedding) is transformed into three vectors: a query (Q), a key (K), and a value (V). The attention mechanism computes the relative effect of tokens

by calculating the scaled dot-product between their queries and keys:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where d_k is the dimensionality of the key vectors, and the softmax function, also discussed in the activation function section, ensures that the attention weights sum to one. This allows the model to attend to different parts of the image simultaneously.

3.3 Layer Normalization and Feed-Forward Networks

After applying self-attention, ViTs often include a normalization step and a feed-forward network to further process the token representations. Each token passes through two linear layers with a ReLU activation in between, allowing for non-linear transformations. This architecture helps refine the token embeddings before they are passed on to other layers.

3.4 Classification Token (CLS) and Output

For tasks like image classification, ViTs introduce a special classification token (denoted as [CLS]) that is prepended to the sequence of image tokens. After passing through all transformer layers, the representation of the [CLS] token is used as the final feature vector for classification tasks. Other adaptations of ViTs, such as the aforementioned **HTS-AT** [29], have repurposed this architecture for non-visual tasks like audio classification.

4 OPERA

The OPERA foundation model pretraining and benchmarking system was created to address the challenges in respiratory sound analysis, particularly the scarcity of large, labeled datasets for training models. Respiratory audio, such as coughing and breathing, contains rich physiological data that can serve as a foundation for various healthcare applications. By analyzing these audio signals, it is possible to predict conditions related to respiratory rate or lung function, and detect health issues like sleep apnea, flu, asthma, and the effects of smoking [20].

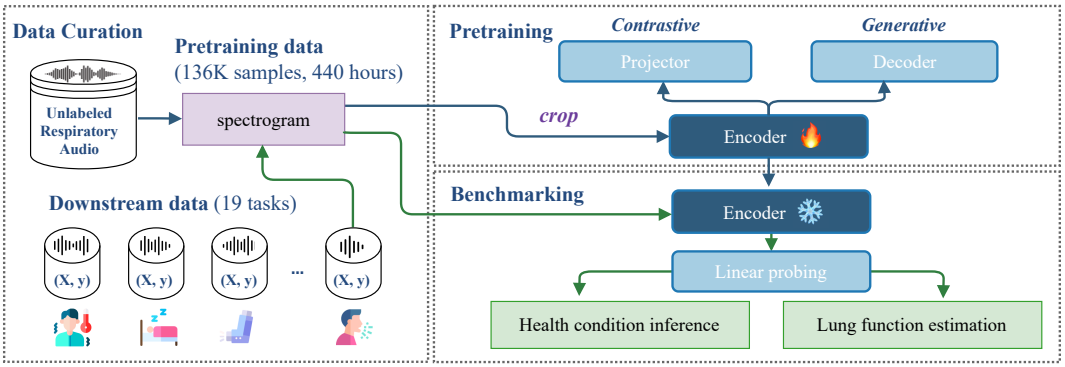


Figure 4: **System Overview;** After data curation, audio encoders are pretrained and then evaluated on various downstream health tasks - including both binary and multi-class output tasks.

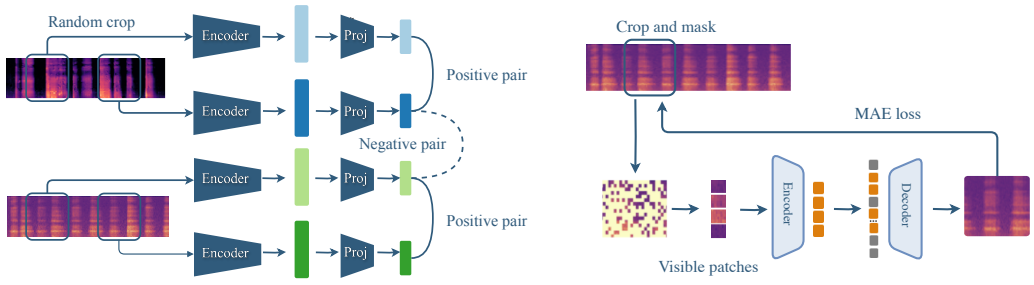
While there is great potential, the area is under-explored due to the difficulty in obtaining labeled, task-specific data. As shown in Figure 4, OPERA is composed of three main components: data curation of both unlabeled data for pretraining and labeled data for evaluation, general-purpose pretraining to develop generalizable acoustic models (encoding), and a benchmark comparing the pretrained models on various downstream tasks. While the OPERA data curation is impressive, this paper will focus on the last two components - **pretraining** and **benchmarking** of models.

4.1 Dataset and Model Pretraining

Large, unlabeled respiratory audio datasets were utilized for pretraining.[2] Self-supervised learning techniques, namely contrastive learning and generative modeling, are used in the models.

Dataset Preprocessing: Audio is trimmed to remove silence, all recordings are resampled to 16 kHz, and then merged into mono channels. Using 64 Mel filter banks with a 64 ms Hann window and 32 ms shift, the audio is converted into spectrograms (i.e., a 4-second recording becomes a 1×126×64 spectrogram). **Spectrograms** are a visual representation of the frequency spectrum of audio over time, and they capture both temporal and spectral features, which make them essential for understanding speech and sound patterns. These processed spectrograms are then used to pretrain the foundational models. This preprocessing enables the models to learn relevant acoustic features which improves performance in downstream tasks such as classification and regression.

Model Pretraining: The models were pretrained using a variety of respiratory audio datasets, which were divided into equally sized batches to ensure consistent processing. To accommodate the varying lengths of many audio samples, random cropping of spectrograms was applied. Pretraining was carried out using contrastive learning-based and generative pretraining-based methods. These approaches allowed the contrastive models to distinguish between audio segments, and allowed the generative model to reconstruct masked segments of the spectrogram.



(a) Contrastive learning (OPERA-CT, OPERA-CE) (b) Generative modeling (OPERA-GT)

Figure 5: **Pretraining Methods;** The CT and CE models rely on a contrastive learning architecture [29], while the GT model uses a generative ViT encoder and decoder [30].

[2] Datasets used in pretraining: COVID-19 Sounds, UK COVID-19, COUGHVID, ICBHI, and HF LUNG. See [20] for model dataset statistics.

4.1.1 OPERA-CT: Contrastive Transformer

The OPERA-CT model employs a contrastive learning approach based on transformer architecture. As shown in Figure 5(a), a transformer-based encoder extracts features from the audio segments, and a projector network (in their implementation, an MLP) maps these features to a low-dimensional space for similarity calculations. This model contains 31 million trainable parameters.

4.1.2 OPERA-CE: Contrastive EfficientNet

While OPERA-CE is another contrastive learning model, it uses the EfficientNet-B0 architecture as its encoder, which makes it more lightweight and efficient compared to OPERA-CT. This encoder outputs a feature dimension of 1280 and has approximately 4 million trainable parameters (compared to CT's 31M), which allow it to differentiate between spectrogram segments.

4.1.3 OPERA-GT: Generative Transformer

OPERA-GT is a generative model pre-trained with a masked spectrogram reconstruction task. It uses an encoder to extract features and a decoder to reconstruct the masked spectrogram patches. During training, 70% of the spectrogram patches are masked, and the model learns to reconstruct these missing sections. The encoder contains 21 million trainable parameters, and the decoder has 12 million, making it a powerful system for spectrogram reconstruction.

4.2 Contrastive and Generative Learning

Contrastive learning is a method for learning representations by comparing pairs of data samples. The core idea is to move similar (positive) pairs closer in the embedding space while pushing dissimilar (negative) pairs apart, therein simplifying the classification boundaries.

4.2.1 Contrastive Learning for OPERA-CT

In OPERA-CT, contrastive learning is used to differentiate between similar and dissimilar data samples by comparing an anchor x with a positive sample x^+ and negative samples x^-. The model aims to maximize the similarity between x and x^+ while minimizing similarities in representation for x^-.

The contrastive loss is formulated as:

$$L = -\log \frac{\exp(s(x, x^+))}{\sum_{x^- \in \{x^+, x^-\}} \exp(s(x, x^-))}$$

where $s(x, x')$ is the bilinear similarity score, calculated as:

$$s(x, x') = g(f(x))^T W g(f(x'))$$

where $f(x)$ and $f(x')$ are the encoded representations of the data points, $g(\cdot)$ is a projection function (again, an MLP as shown in [20]), and W is a learned weight matrix. By optimizing this loss function, the model learns to map positive data points close together and negative points farther apart in the feature space, which allows for a more distinguishable classification boundary.

4.2.2 Generative Learning in OPERA-GT

Generative learning in OPERA-GT is driven by a reconstruction task where a portion of the input spectrogram is masked, and the model must predict the missing regions. This encourages the model to capture the underlying data structure, which results in more generalizable representations that are robust enough for different tasks.

The GT model, composed of a ViT encoder and a lightweight *Swin Transformer* [28] decoder, minimizes the Mean Squared Error (MSE) loss:

$$L_{\text{gen}} = \frac{1}{N}\sum_{i=1}^{N}(S_i - \hat{S}_i)^2$$

where S_i and $\hat{S}_i$ represent the true and predicted values for pixel i, respectively. By reconstructing the masked sections accurately, the model learns to encode vital information, benefiting downstream tasks like classification or anomaly detection.

4.3 Benchmarking

To evaluate the pretrained models and provide a standardized comparison for future respiratory acoustic models, a comprehensive benchmark has been established. This benchmark incorporates 10 labeled respiratory audio datasets that span 6 different audio modalities. Notably, 6 of these datasets were unseen during the pretraining phase, ensuring an unbiased evaluation for model generalization.

4.3.1 Dataset and Task Setup

The 19 downstream tasks derived from these datasets include a mix of classification and regression challenges. These tasks are grouped into two categories:

- **Health condition inference:** Covering 12 tasks related to disease detection (such as COVID-19 and COPD), smoker and gender identification, disease severity, and sleep body position monitoring. Tasks 1-10 involve binary classification, while Tasks 11-12 span five classes.
- **Lung function estimation:** The remaining 7 tasks focus on predicting spirometry (a pulmonary function test) results and respiratory rate, which are framed as regression tasks to predict continuous values.

Where possible, the official train-test splits are used for Tasks 1-4 and 12-18, while the other tasks adopt a random participant-independent split to ensure realistic evaluation. Due to the limited number of participants in Tasks 13-19, leave-one-subject-out evaluation is employed, while a fixed random train-validation-test split is used for all other tasks [20].

4.3.2 Baselines

To establish a point of comparison, the benchmark includes several commonly used acoustic feature sets and pretrained acoustic models. These baselines consist of:

- **OpenSMILE** [31]: A widely-used acoustic feature set from the Emobase toolkit. It provides robust extraction of audio features for emotion recognition, speech processing, and general audio analysis.

- **VGGish** [32]: A pretrained model using supervised learning on large-scale audio datasets. VGGish leverages the architecture of VGG networks [17], originally designed for image classification, and adapts it to audio data, offering feature extraction for audio classification tasks.
- **AudioMAE** [30]: A self-supervised model focused on extracting general audio representations. Uses a masked auto-encoder to learn from large amounts of unlabeled audio data, improving the model's ability to generalize across different audio tasks (i.e., classification, detection, and regression).
- **CLAP** [33]: A language-supervised pretrained model that integrates audio and text data. CLAP learns joint embeddings of audio and text, enabling tasks like audio retrieval based on text queries. It connects different modalities and enhances applications in sound event detection and classification.

These methods serve as baselines and enable fair comparison with the OPERA pretrained models. In addition, these baseline architectures are pretrained using the same OPERA-trained dataset.

4.3.3 Evaluation

The performance of the OPERA models was evaluated across several health-related tasks, comparing them against popular baselines of OpenSMILE, VGGish, AudioMAE, and CLAP. For the health inference tasks (Tasks 1–12), the OPERA-CT model generally outperformed other models, achieving the highest AUROC in most cases. OPERA-CT excelled in tasks related to COVID, gender, and COPD severity detection. In contrast, for lung function estimation (Tasks 13–19), OPERA-GT demonstrated better results, outperforming other models in tasks related to obstructive lung conditions.

The OPERA models consistently outperformed other models on classification tasks, as measured by AUROC, while also achieving strong performance on regression tasks with Mean Absolute Error (MAE) loss. By integrating both contrastive and generative learning strategies, these models achieved impressive results. In summary, OPERA-CT's strong performance in health condition inference and OPERA-GT's accuracy in lung function estimation highlight the vast potential of these models, making them competitive in a variety of healthcare applications. Their success in both pretraining methods demonstrates a promising future in helping to advance medical diagnostics.

5 Conclusion

5.1 Review

This review presents an understanding of standard NNs, CNNs, and ViTs, along with methods of training them, the mathematical functions hidden within them, and frameworks and applications in which they may be used and explored. These models have *transformed* (pun intended) the way we approach various tasks, such as image recognition, speech processing, and, more recently, respiratory sound analysis. Techniques such as supervised, self-supervised, contrastive, and generative learning have further enhanced their capabilities, enabling them to generalize better across diverse tasks and datasets.

Among the applications of these models, respiratory acoustic modeling has gained considerable attention for its potential in disease detection and health monitoring. This paper covers the OPERA framework system, which demonstrates the power of applying CNNs and ViTs to respiratory acoustics - outperforming traditional methods in various downstream tasks related to health condition inference and lung function estimation.

5.2 Future Research

My future thesis research aims to explore the development of models using the OPERA framework to assist in diagnosing pulmonary diseases like COPD. By leveraging respiratory sound data and pretrained models, the goal is to enhance diagnostic tools for early detection and monitoring of such conditions. These advancements could help address the growing need for non-invasive and efficient diagnostic methods in pulmonary medicine.

6 Acknowledgment

I would like to thank my PI, Dr. Baocheng Geng, for his support and guidance throughout this study. His advice and feedback have been very helpful in making this study on neural networks and deep learning more manageable and enjoyable. I look forward to continuing my work under his guidance in the upcoming semester.

References

[1] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[2] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[3] Shun'ichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307, 1967.

[4] S. Amari and H. Saito. *Information Theory—Geometric Theory of Information*. Kyoritsu Publ., 1968. OCR-based PDF scan of pages 94-135, contains computer simulation results for a five-layer network with 2 modifiable layers which learns internal representations to classify non-linearly separable pattern classes. (in Japanese).

[5] Seppo Linnainmaa. On the computation of definite integrals. Master's thesis, University of Helsinki, 1970.

[6] Paul J. Werbos. A manual for the second-order method of backpropagation. *Harvard University*, 1971.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[8] Paul J. Werbos. *Beyond the Hebbian Synapse: The Training of a Multilayer Perceptron*. PhD thesis, Harvard University, Cambridge, MA, 1974.

[9] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (*ICML-10*), pages 807–814, 2010.

[10] Andrew L. Maas et al. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th International Conference on Machine Learning*, pages 1035–1043, 2013.

[11] M. Johnson. Softmax and its applications in neural networks. *Neural Networks and Learning Systems*, 4(2):54–67, 1986.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (*NeurIPS*), pages 10203–10213, 2018.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pages 770–778, 2016.

[19] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[20] Yuwei Zhang, Tong Xia, Jing Han, Yu Wu, Georgios Rizos, Yang Liu, Mohammed Mosuily, Jagmohan Chauhan, and Cecilia Mascolo. Towards open respiratory acoustic foundation models: Pretraining and benchmarking, 2024.

[21] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, December 2017.

[22] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Yi Ding, Aarti Bagul, Curtis P. Langlotz, Katie S. Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *CoRR*, abs/1711.05225, 2017.

[23] Qiyu Chen, Weibin Zhang, Xiang Tian, Xiaoxue Zhang, Shaoqiong Chen, and Wenkang Lei. Automatic heart and lung sounds classification using convolutional neural networks. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–4, 2016.

[24] Rexy Purnomo Wulan Pramono, Stuart Bowyer, and Esther Rodriguez-Villegas. Automatic adventitious respiratory sound analysis: A systematic review. *PloS one*, 14(5):e0218671, 2019.

[25] Gabriel Gonzalez, Javier Carretero, Laura Remón, and et al. Recent advances in deep learning for medical image analysis. *Annual review of biomedical engineering*, 20:163–188, 2018.

[26] Varun Gulshan, Lily Peng, Marc Coram, and et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[29] Jianyuan Chen, Qiuqiang Kong, Yong Xu, Hongji Sun, Yin Cao, and Wenwu Wang. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. *arXiv preprint arXiv:2202.00874*, 2022.

[30] Ziyi Huang, Yunchao Gong, Yiwen Li, Shiyu Liu, Heng Wu, Linli Xie, Xingxing Zhang, Xinwang Wang, et al. Audiomae: Self-supervised audio representation learning with masked autoencoders. *arXiv preprint arXiv:2207.06405*, 2022.

[31] Florian Eyben, Martin Wöllmer, and Björn W. Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In Alberto Del Bimbo, Shih-Fu Chang, and Arnold W. M. Smeulders, editors, *ACM Multimedia*, pages 1459–1462. ACM, 2010.

[32] Shawn Hershey, Sourish Chaudhuri, Daniel P.W. Ellis, Jort F. Gemmeke, Aren Jansen, Rachel Moore, Manoj Plakal, Dan Platt, Rif A. Saurous, Brion Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. A large-scale audio-visual dataset for understanding human actions in context. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 788–792, 2017.

[33] Yuanchao Wu, Meni Shtok, Ami Salomon, Tali Dekel, Ohad Ginat, et al. Clap: Learning audio-text joint embeddings from noisy correspondences. *arXiv preprint arXiv:2302.03954*, 2023.