university of groningen

faculty of science and engineering

Predicting and Understanding Difficult Mask Ventilation:

Classification and Generative Networks

Mícaél McAuley

university of groningen

faculty of science and engineering

University of Groningen

Predicting and Understanding Difficult Mask Ventilation: Classification and Generative Networks

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Computational Cognitive Science
at University of Groningen under the supervision of
Marleen Schippers, PhD (Lecturer AI/CCS, University of Groningen)
and
Christopher Gundler (Institute for Applied Medical Informatics, Universitätsklinikum Hamburg-Eppendorf)

Mícaél McAuley (s2694956)

May 12, 2025

Contents

Acknowledgments

A huge thank you to my supervisors is necessary first and foremost, Marleen and Christopher both provided invaluable guidance while giving me the freedom to work very independently; hartelijk dank, vielen Dank, go raibh míle maith agaibh. I'd also like to acknowledge the rest of the AI team at the Institute for Applied Medical Informatics, for welcoming me in and creating a lovely working environment, thank you. To the long suffering medical students who recorded and annotated (by hand) the many hours of audio data, go my deepest appreciations. To my family for their unconditional love and support I am indebted, and finally to Kassandra, for your love, support, and gentle bullying when I tried to procrastinate, thank you.

Abstract

Mask ventilation is a vital aspect of airway management in clinical, emergency, and surgical settings, yet unexpected difficulty in mask ventilation remains a significant cause of morbidity and mortality. Current predictive methods for identifying at-risk patients are often insufficient, necessitating innovative approaches. Recent research suggests that vocal acoustics, influenced by upper airway anatomy, may offer a novel predictive tool.

This study explores human-centered artificial intelligence techniques, namely Convolutional Neural Networks and Variational Autoencoders, to improve the prediction and understanding of difficult mask ventilation. By applying AI-driven analysis to vocal biomarkers, this study aims to convert anecdotal intuition-based observations about voice and airway difficulty into an evidence-based predictive model. The first objective involves developing a predictive model using mel-frequency spectrograms, a human-interpretable audio representation, to classify patients based on vocal patterns. The second objective investigates the generation of synthetic voice samples simulating both high-risk and routine patients, enhancing model interpretability and serving as a potential educational tool.

Moderate success in classification was achieved, with a model obtaining a slight but meaningful ability to distinguish difficult mask ventilation from a patients voice alone. The generative model's computational requirements exceeded available resources, preventing successful synthesis of voice samples.

1 Introduction

Mask ventilation is a crucial component of airway management. It is a non-invasive technique whereby a mask is placed over the nose and mouth of a patient and connected to an oxygen supply. It allows healthy blood oxygen levels to be maintained when an individual cannot breathe for themselves. While it may occur in emergency situations, it is often necessary prior to endotracheal intubation (the insertion of a breathing tube). Mask ventilation acts as a bridge between spontaneous breathing (naturally) and the advanced airway management found in clinical practice and surgery (Gupta & Raina, 2024).

Failure to maintain proper ventilation can rapidly lead to serious problems, such as hypoxemia (abnormally low blood oxygen) or hypercarbia (abnormally high carbon dioxide in the blood), and subsequent complications like hemodynamic instability (unstable blood pressure) and hypoxic brain injury (where the brain is starved of oxygen) (Cao et al., 2021). This makes the efficacy of mask ventilation vital for patient safety. As noted by Cao et al. (2021), difficult mask ventilation is a significant contributor to neurological damage and death.

Despite its routine use, unexpected difficulty in mask ventilation remains a significant challenge, contributing to morbidity and mortality. Identifying predictors of difficult mask ventilation is essential for improving patient safety, however traditional methods have proven insufficient in their predictions in a significant number of cases. This gap in predictive ability and the identification of difficult mask ventilation opens the floor to innovative approaches.

Recent attempts to address this challenge with novel applications have highlighted promising avenues. By utilising subtle acoustic features in a patient's voice, which are influenced by the anatomy of the mouth and throat, it may be possible to develop more accurate predictive models. Previous research, such as that by Xia et al. (2021), has laid a foundation for this approach by demonstrating the feasibility of voice-based prediction in a related clinical context (namely, difficult intubation). Building on these insights, this study explores its application to ventilation. Notably, some experienced clinicians also report, anecdotally, that certain voice characteristics can provide early indications of difficult mask ventilation. This research aims to use the ground work of Xia et al. (2021)'s study to transform these clinical intuitions into an evidence-based framework.

The first objective of this research is to build and test a predictive model of difficult mask ventilation using mel-frequency spectrograms. These are image representations of sounds that highlight the frequencies humans hear, allowing analysis to be done on the same perceptual data a physician would hear. By treating audio as images, Convolutional Neural Networks can be employed to classify patients based on their future risk of difficult mask ventilation. Secondly, the use of Variational Autoencoders to generate synthetic voice samples, simulating those of both at risk and routine patients, is investigated. This is done with the goal to improve trust in the previous models classifications, and to serve as a potential educational tool. If this intuition is correct but only obtainable through practical experience, being able to generate examples of high-risk and routine voices could expedite this process.

1.1 Research Questions

To summarise, this thesis focuses on the following problems:

Q1. Can subtle but nevertheless perceptual acoustic features in the human voice be used to predict future difficult mask ventilation by artificial intelligence models?

Q2. Can generative artificial intelligence models be used to create new examples that capture these acoustic features?

1.2 Thesis Outline

This paper first presents the background literature on mask ventilation, human-centered audio representations, artificial intelligence in healthcare, and the theoretical framework of Convolutional Neural Networks and Variational Autoencoders. It then details the methodologies employed, beginning with data preprocessing, moving through the specifics of the model architectures, and finishing with the optimisation of the hyperparameters. Each models performance is then discussed, and finally a summary of the findings and future directions of research are given.

2 Background Literature

This section defines, and reviews prior studies on, each of the key components of the research. It covers mask ventilation, human centered audio representations, and artificial intelligence (AI) in healthcare, as well as the two AI models used and their theoretical and mathematical foundations.

2.1 Ventilation

While mask ventilation is often routine, difficulty can arise due to a number of factors. Identifying these factors in advance can allow for the prediction of difficult mask ventilation (DMV) and its mitigation. El-Orbany and Woehlck (2009) provide several predictors, such as presence of a beard, high body-mass-index, missing teeth, age over 55, or a history of snoring. They do however note that in previous studies, predictions based on physical and patient history findings failed to foresee DMV in 57% of patients who were difficult to ventilate. Thus while these factors are important, additional predictive techniques would be advantageous, for example using ultrasounds of the mouth and throat to predict DMV (Lin, Tzeng, Hsieh, Kao, & Huang, 2021).

Xia et al. (2021) suggest a different approach. They attempted to use acoustic features, namely the patients voice, to predict difficult intubation using logistic regression models. While an ultrasound provides a visual representation of the oropharyngeal region, that information is also held in the voice. Minute variations in pitch, timbre, strain, vibrato, and resonance across individuals depend on the anatomy of the mouth and throat (Sataloff, Heman-Ackah, & Hawkshaw, 2007). Xia et al. (2021) achieved moderately successful results, obtaining a model with a sensitivity of 86.7% and a specificity of 63%, quite accurate at predicting difficulty when it was present but with a relatively high false positive rate. Listening to a patient speak can, anecdotally, give physicians an impression of how difficult they may be to ventilate or intubate. This gut feeling is an example of '*clinical intuition*', a type of non-analytic perception. Research suggests it is an important part of how experienced physicians process information, and can guide their decision making both consciously and subconsciously (Vanstone et al., 2019; Woolley & Kostopoulou, 2013; Greenhalgh, 2002).

If subtle acoustic features in a patient's voice can serve as predictors of DMV, then human-centered AI techniques may offer a way to validate and enhance detection in the future. By using machine learning approaches that compliment human perception, it may be possible to leverage these vocal markers more systematically. In doing so, prediction models could be refined, clinical decision making aided, and patient safety enhanced.

2.2 Human-Centered Audio Representations

To effectively capture and analyse these subtle acoustic features that physicians may be attuned to, a human-centered audio representation is essential. A well established option is the mel-frequency spectrogram (or mel spectrogram). A mel spectrogram is a visual representation of the frequency content of a sound signal, specified to human perception of sound. Mel spectrograms are well suited for machine learning models for several reasons. They reduce the dimensionality of audio data by compressing high-frequencies, reducing the resources required for training. They also allow for the treatment of audio as images, thus allowing for the use of the well established methodologies of computer vision. Lastly, of particular relevance to this research, they highlight the perceptually relevant

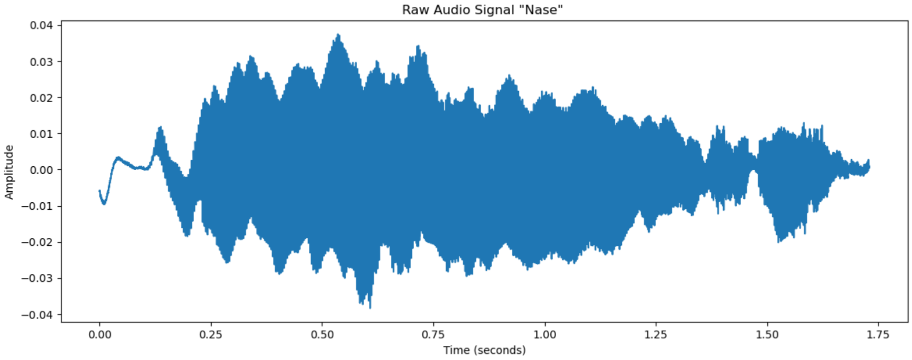


Figure 1: Waveform of audio of the word ”Nase”

features of the data, as they aim to represent how humans perceive sound (Gold, Morgan, & Ellis, 2011).

Raw audio signals represent amplitude over time, but don’t show information on the frequencies present in the sound. Spectrograms on the other hand plot frequencies over time. A raw audio signal can be converted to a spectrogram by taking small time windows of the signal and applying a Fourier transform to each window. This results in a 2D representation with time on the x-axis, frequency on the y-axis, and intensity represented by the value.

As the brain doesn’t perceive frequencies in sound linearly, being more sensitive to differences in lower frequencies than higher frequencies, a spectrogram doesn’t quite capture the information as we humans would experience it. The mel scale aims to approximate how humans hear pitch by compressing higher frequencies and emphasising lower frequencies, leading to levels that are interpreted as being uniform steps in pitch (O’shaughnessy, 1987). Frequency (f) is converted from a linear scale to the mel scale (m) via the following:

$$m = 2595\log_{10}\left(1 + \frac{f}{700}\right)$$

Next a series of overlapping triangular filters, or mel filter banks, are placed across the mel-scaled frequency range. Each filter corresponds to a specific frequency band and has a peak response at its centre frequency, tapering off to zero at the edges of adjacent filters. By passing a frequency spectrum through these filters, each filter computes a weighted sum of the energy in its frequency band, binning the frequencies into mel-scaled bands. The result is a mel-scaled representation of the frequency spectrum, where the y-axis is in mel bands instead of raw frequencies.

While mel spectrograms map frequencies to a scale that aligns with human hearing, the amplitude values are still linear. Linear amplitude values can be dominated by high-energy components, making it hard to see the lower-energy features. Converting to decibels (dB) further aligns the amplitude representation with human perception. The dB scale compresses the audio range, making quieter sounds more visible and easier to analyse alongside louder sounds. Additionally, when plotting spectrograms, the dB scale provides a clearer visualisation, highlighting details across the entire dynamic range. Each amplitude value in the spectrogram (power) is converted to dB by the following:

$$\mathrm{dB} = 10 \cdot \log_{10}(\mathrm{power})$$

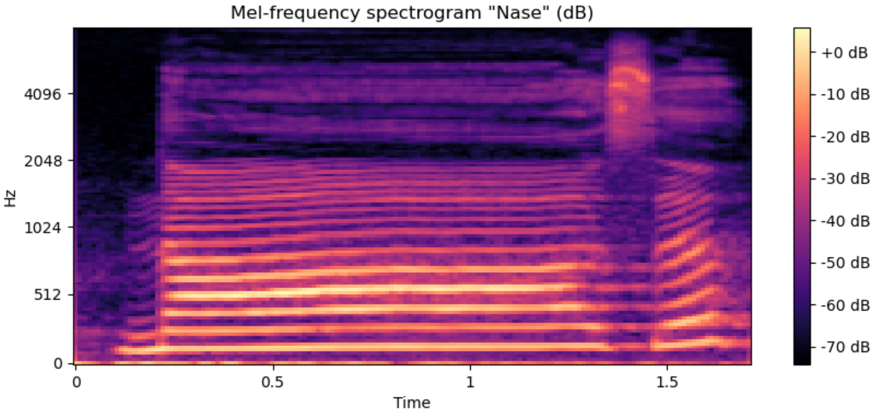


Figure 2: Mel-frequency spectrogram of the word ”Nase”. Time on the x-axis given in seconds, frequency on the y-axis in Hz and amplitude in decibels represented by colour intensity

Figure 1 shows the waveform of the raw audio of an individual saying the word *”Nase”*, while Figure 2 shows the same audio converted to a mel-frequency spectrogram in the dB scale. With audio in this human-centered visual format, we can apply machine-learning techniques as if analysing images, while hopefully capturing what a person would hear if they listened to the raw audio.

2.3 Convolutional Neural Networks

One such technique is the Convolutional Neural Network (CNN). A CNN is a class of deep learning model designed to process two dimensional (2D) data such as images. CNNs are foundational to fields like computer vision due to their ability to learn hierarchical features from data. An example of a CNN is seen in Figure 4. Unlike traditional fully connected neural networks, CNNs learn spatial hierarchies and local patterns through specialised convolutional layers, making them highly efficient for tasks involving structured data. They have proven to be effective in making predictions, particularly in tasks involving image and pattern recognition (Wu, 2017). This could be used to detect the subtle acoustic patterns that predict DMV if they are captured by a mel spectrogram.

The aforementioned convolutional layers consist of a set of filters called kernels, themselves a matrix of a defined size m and n. Multiple kernels are used in a convolutional layer to capture different features. Each kernel is passed over the input and performs a convolution operation, as seen in Figure 3. For a 2D input I and a kernel K, the convolution operation at position (i, j) of the input is given by:

$$(I * K)(i, j) = \sum m \sum n \, I(i+m, j+n) \cdot K(m,n)$$

This operation captures local patterns, such as edges, textures, or spectral features, depending on the input data. Additionally, A n by m kernel will only have $n * m$ weights associated with it, one for each element of the kernel. These weights are shared across the entire input. The use of shared weights in convolutional layers significantly reduces the number of parameters compared to fully connected layers, making CNNs computationally efficient.

After convolution, an activation function is applied to introduce non-linearity into the model. The Rectified Linear Unit (ReLU) is commonly used due to its simplicity and effectiveness in mitigating the vanishing gradient problem (Fukushima, 1969). It is defined as:

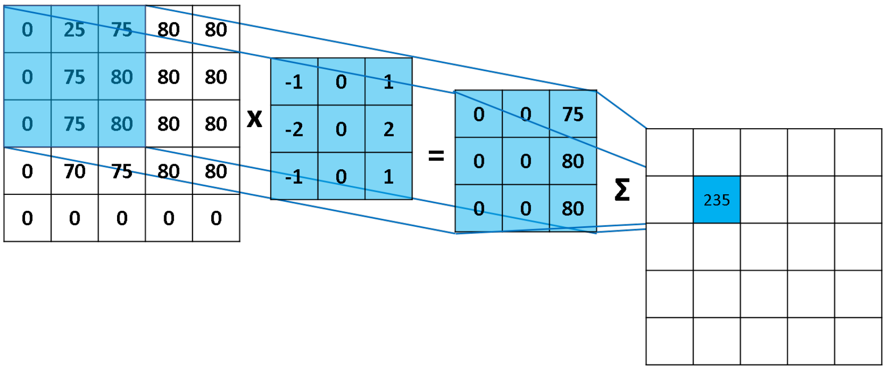


Figure 3: First convolution by a 3 by 3 kernel on a 5 by 5 input (Robinson, 2017)

$$ReLU(x) = max(0, x)$$

After the convolutional layers, the extracted features are passed through one or more fully connected layers. These layers combine the high-level features to produce the final output, such as class probabilities in a classification task.

A variety of additional layers and techniques may also be incorporated to improve performance. Following each convolutional layer a pooling layer may be added. These downsample the outputs of the convolutional layers with the aim of reducing computational complexity while maintaining the important feature information (Nagi et al., 2011). A common example is max pooling, which acts like a kernel, passing over the output and selecting the maximum value from a small window.

Before the activation function is applied, batch normalisation may be performed. During training, this technique standardises the outputs of a layer by subtracting the mean and dividing by the standard deviation of that layer. This helps mitigate issues like internal covariate shift, where the distribution of layer inputs changes during training, causing slower convergence (Ioffe & Szegedy, 2015).

Finally, a technique called dropout can be applied to prevent overfitting (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). A small fraction of neurons are deactivated randomly each training iteration, preventing the network from becoming overly reliant on any particular set of neurons encouraging more robust learning.

CNNs ability to automatically extract hierarchical features from data makes them well-suited for complex prediction tasks found in clinical settings, such as diagnosing medical conditions from imaging data like X-rays or MRIs (Kayalibay, Jensen, & van der Smagt, 2017). These hierarchical features range from simple shapes, edges, or textures in an image at the low level, to complex emergent patterns at the higher levels, like the presence of a lesion in an MRI, or a subtle frequency change in a spectrogram. This demonstrates strong potential for predicting DMV from a patients voice once it has been represented as an image via a mel spectrogram.

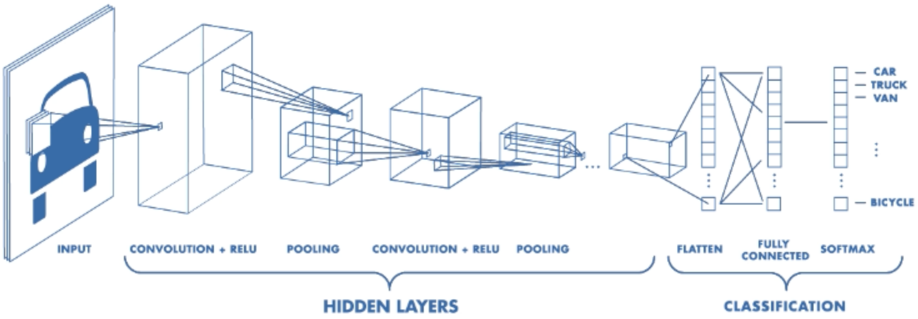


Figure 4: Example of a Convolutional Neural Network classifier for vehicles, with ReLU activation and pooling layers (MathWorks, 2017)

2.4 Applied AI in Healthcare

AI systems, including predictive models, must be designed to assist healthcare professionals rather than replace them. As noted in their roadmap to building safe, reliable, and effective AI systems in healthcare, Bajwa, Munir, Nori, and Williams (2021) espouse the view that AI should amplify and augment, rather than supplant human intelligence. By providing data-driven insights, AI can help doctors focus their expertise on critical aspects of patient care, however, the human element remains irreplaceable, as patient care involves empathy, ethical considerations, and complex decision-making that goes beyond algorithmic predictions. AI should therefore only ever be considered a tool to enhance the work of healthcare professionals.

AI systems face some major barriers to their adoption in healthcare, particularly their lack of interpretability (Frasca, La Torre, Pravettoni, & Cutica, 2024). While in some tasks, particularly analytical in nature, AI systems can outperform doctors, their ”black-box” aspect limits their use in actual medical contexts. Frasca et al. (2024) found in their review of AI in the medical field, that doctors and patients alike need to have some understanding of how and why a prediction was made in order to trust it. This poses a challenge to complex models like CNNs, whose complex deep learning designs are opaque in their classification.

An approach to improve interpretability in this domain is to use generative models. In this case these models can, potentially, create synthetic examples of voices that simulate what a patient at risk of DMV may sound like. This may help doctors to grasp the reasoning behind a model’s output. While a predictive model simply presents its result, a generative model can give concrete demonstrations of those results. This effectively allows one to hear what the model is hearing. By producing at risk and routine examples they, if accurate, can increase confidence in the black-box predictive models and grant better understanding of the acoustic features influencing the AI’s predictions. This would not only support model validation, but could also serve as an educational tool. If varied examples of comparable at risk and routine voices can be reliably generated, novices can have the opportunity to refine that intuitive sense of potential difficulty, that otherwise can only be gained through practical experience. A good generative model then can both increase trust in the predictive model, and help more junior practitioners gain the skills of said predictive model. Thus achieving the goal of aiding medical practitioners, rather than usurping them.

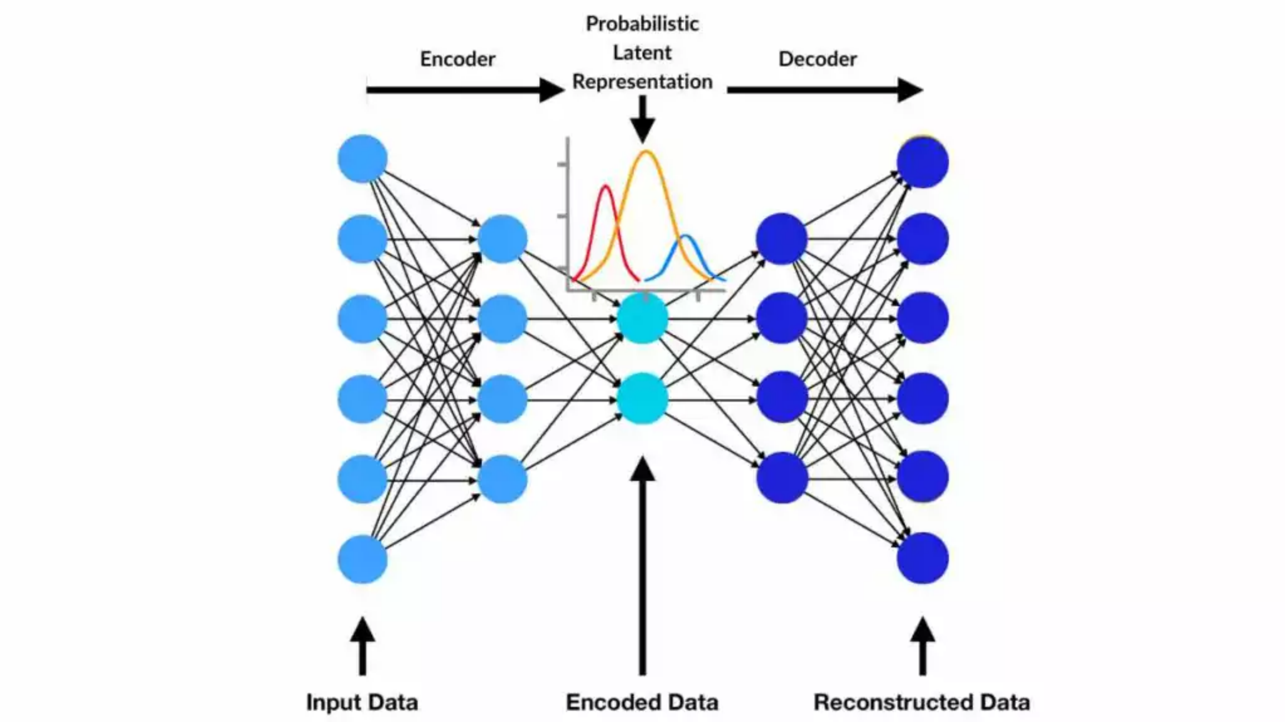


Figure 5: Structure of a Variational Autoencoder (Otten, 2023)

2.5 Variational Autoencoders

Variational Autoencoders (VAEs) are a type of generative model that are designed to learn a latent representation of data and generate new data samples that are similar to the training data. VAEs are particularly useful for tasks like image generation, data compression, and unsupervised learning (Kingma & Welling, 2022). In the context of DMV, they could be well suited to learn from the same mel spectrograms used to train a CNN classifier, and generate new examples that can be converted back into audible speech.

In an ordinary autoencoder model, input data is mapped to a latent space representation by an encoder. A decoder then maps the latent space representation back to the original data space. The goal is to minimise the reconstruction error, ensuring that the output is as close as possible to the input (Bank, Koenigstein, & Giryes, 2023).

In VAEs (Figure 5), the latent space is probabilistic. Instead of mapping an input to a single point in the latent space, the encoder maps it to a distribution over the latent space. This is typically a Gaussian distribution, characterised by a mean μ, and a variance σ.

VAEs use variational inference to approximate the posterior distribution of the latent variables given the input data. This variational inference is necessary because the posterior distribution of the latent variables is intractable, being computationally infeasible to compute directly, as explained mathematically below.

Let $\mathbf{x}$ be the observed data and $\mathbf{z}$ be the latent variables. The joint probability, the likelihood of observing data $\mathbf{x}$ and latent variables $\mathbf{z}$ together, is:

$$p(\mathbf{x},\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

Where $p(\mathbf{z})$ is the prior distribution over the latent variables $\mathbf{z}$, and $p(\mathbf{x}|\mathbf{z})$ is the likelihood, how the data $\mathbf{x}$ is generated from the latent variables $\mathbf{z}$. The goal is to then infer the posterior $p(\mathbf{z}|\mathbf{x})$, which is given by Bayes' theorem as:

$$p(\mathbf{z}|\mathbf{x}) = \tfrac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$$

Where $p(\mathbf{x})$ is the marginal likelihood of the observed data $\mathbf{x}$. It is obtained by integrating the joint probability $p(\mathbf{x},\mathbf{z})$ over all possible values of the latent variables $\mathbf{z}$. It represents the total probability of the data $\mathbf{x}$ under the model, considering all possible configurations of $\mathbf{z}$:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz$$

This integral is often intractable, as the dimensionality of the latent space may be large and the likelihood, being modelled by a neural network, may be complex and highly non-linear.

To get around this we introduce a variational distribution $q(\mathbf{z}|\mathbf{x})$ to approximate the true posterior $p(\mathbf{z})$. The distribution $q(\mathbf{z}|\mathbf{x})$ is chosen to be a Gaussian distribution with parameters (μ,σ) output by the encoder. The difference between these distributions is called the Kullback-Leibler (KL) divergence:

$$D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) = \mathbb{E}_{x\sim P}\left[\log \tfrac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right]$$

For a VAE with a Gaussian encoder $q(z|x) = \mathcal{N}(z;\mu(x),\sigma^2(x))$ and a standard normal prior $p(z) = \mathcal{N}(z;0,I)$, the KL divergence can be computed as:

$$D_{KL}(q(z|x)\|p(z)) = \tfrac{1}{2}\sum_{i=1}^{d}\left(\sigma_i^2 + \mu_i^2 - 1 - \log(\sigma_i^2)\right)$$

With d the dimensionality of the latent space, and μ_i and σ_i the mean and standard deviation of the encoder's output for the i-th latent dimension.

By combining the KL-divergence with the reconstruction loss (that is a measure of how closely the decoder recreates the input), we obtain a tractable objective that can be minimised to obtain a model that encodes latent variables into a normal distribution, creating a latent space where any random point can be interpolated into a new example of the input data.

The final issue is the aforementioned randomness. A neural network relying on random sampling cannot be optimised through backpropagation. A vector of random values has no derivative, and thus no gradient from which patterns in the models outputs can be gleaned. VAEs avoid this problem by using a reparameterisation trick. A random value ε is drawn from a standard normal distribution (between 0 and 1). The latent variable $\mathbf{z}$ is then redefined as

$$\mathbf{z} = \mu_x + \varepsilon * \sigma_x$$

Where μ_x and σ_x are outputs of the encoder for input $\mathbf{x}$. As ε is independent of the autoencoders parameters, it can be ignored during backpropagation, allowing the models parameters to be updated with gradient descent (Kingma & Welling, 2022).

In summary, VAEs provide a way to potentially generate new examples of voices that indicate a high risk of DMV. These example can be used as an educational aid, but also to build trust in the predictive ability of a classifier model like a CNN. With a trustworthy and accurate classification model, physicians would have another useful tool available to them in the perioperative period.

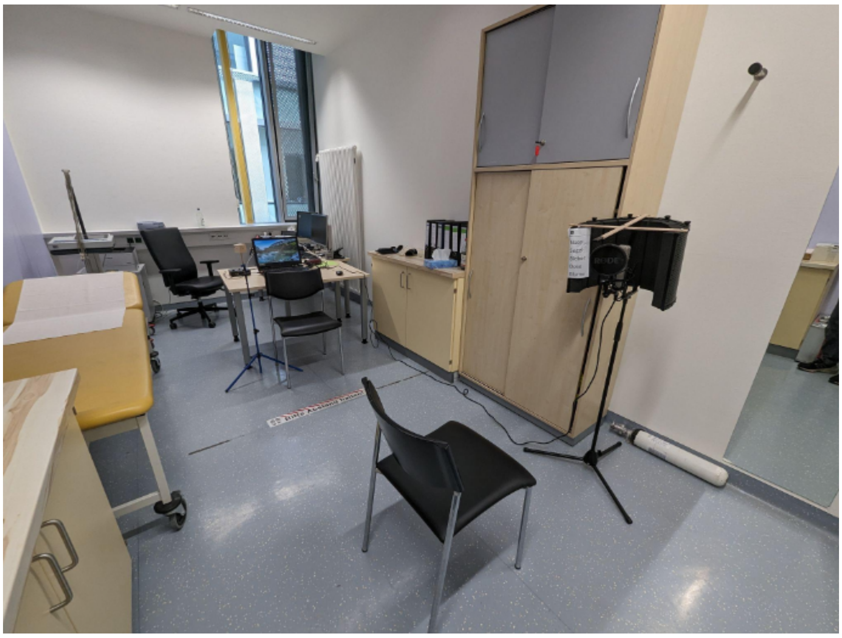


Figure 6: Experimental Setup

3 Methods

The following section is divided into three parts. The first covers the shared data preprocessing step both methodologies utilise, while the latter two cover said methodologies respectively.

3.1 Data

Data was recorded for 395 participants, prior to general anaesthetic for non throat related procedures, by medical students at the Universitätsklinikum Hamburg-Eppendorf. Participants ranged from 18 to 88 years old ($\mu = 52.6, \sigma = 20.25$), with 54% being male. Following their procedures, the anaesthesiologist filled in a survey on each participant including a *yes/no* question on whether or not they believed this patient would be difficult to ventilate in the future. This created two classes, *routine* and *DMV*, with 43 participants in the *DMV* class, just over 10% of the total.

3.1.1 Audio

Audio was recorded in a physician's office, using commercially available recording equipment, from a distance of approximately 50 cm (Fig 6). Each participant was recorded repeating a set of 5 words 3 times. The set consisted of the words “Nase”, “Segel”, “Biber”, “Dose”, and “Blume”. The first syllable of each of these words contains a vocal vowel (a, e, i, o, or u). The audio of each recording was then cut to obtain 15 recordings per participant, covering these vowel sounds, for a total of 5925 audio snippets. Vowels are used as they involve steady resonant frequencies of the vocal tract and mouth, and last longer than consonants, providing more useful data for analyses (Moon, Chung, Park, & Kim, 2012). All processing was performed with *Audacity* version 3.7.1, snippets were exported as .wav files (Audacity-Team, 2023).

3.1.2 Preprocessing

The .wav files were labelled based on the future prediction of difficult mask ventilation, 1 for *yes*, 0 for *no*. The dataset was then augmented using random oversampling of the minority group (in this case that was the *yes* group). For the generative model the vowel being demonstrated, and gender of speaker, was also added to the label.

The dataset was randomly split by participant into balanced training, testing, and validation sets with 20% of data in the testing set, 20% in validation set, and the remainder in training set. A standard duration was found based on the mean length of all audio snippets plus one standard deviation. Each audio snippet was padded with leading silence up to this ideal length, or trimmed down to it, to ensure consistent duration for training.

Each audio snippet was finally converted to a mel-frequency spectrogram. Fourier Transform windows were set to 2048 samples for higher quality, with the default 128 mel filter banks. All audio had a sample rate of 44100 Hz. Each mel spectrogram was then normalised based on the range of values in the training dataset. This was done to avoid exploding or vanishing gradients while also improving the gradient descent efficiency. Only the training data was used, to both prevent any information from the test or validation sets contaminating the training process, and to simulate real world conditions of unseen new data.

3.2 Convolutional Neural Network (CNN)

The model was configured via a *config* dictionary, allowing flexibility in, and optimisation of, the hyperparameters. These can be found in Table 1. Where multiple values are given, each value was possible and later refined during optimisation. The model was implemented using PyTorch version 2.2.2 (Ansel et al., 2024) and Pytorch Lightning version 2.2.3 (Falcon & The PyTorch Lightning team, 2019).

3.2.1 Architecture

An example diagram of the architecture of the CNN can be found in Figure 7. Each component is further explained below:

Input Layer: The input to the model is expected to be a 2D tensor representing mel spectrograms with a single channel (standard for black and white or intensity based images like mel spectrograms).

Convolutional Layers: The model consists of a configurable number of convolutional layers, `n_layers`, each with an adjustable kernel size of `kernel_size`, and followed by a ReLU activation function and max-pooling. The number of output channels (kernels) in the first convolutional layer is set by `n_kernels` and is doubled for each subsequent layer. This is done to capture more subtle features as the complexity of the learned representation increases with depth. It also preserves the information capacity of the model as each convolutional layer reduces the spatial dimensions of the input. Max-Pooling is performed with a kernel size and stride of two. This decreases the dimensions of the input to the next layer helping with computational complexity and the prevention of overfitting.

Parameter	Description	Value(s)
num_workers	*The number of subprocesses used for data loading. Higher values can speed up data processing but require more memory*	4
batch_size	*The number of training samples processed in one forward/backwards pass or gradient update. Higher values can increase stability during training but use more memory.*	64
criterion	*The loss function used to measure the difference between predicted and actual outputs*	Cross Entropy Loss
max_epochs	*The maximum number of times the model will iterate over the entire training dataset*	1000
n_classes	*The number of distinct labels in the classification task*	2
input_shape	*The dimensions of the input data, a mel spectrogram (channels, height, width)*	(1,128,260)
n_layers	*The number of convolutional layers in the network, determining its depth and complexity*	1, 2, 3, 4
n_kernels	*The number of kernels (feature detecting filters) in the first convolutional layer*	8, 16, 24, 32
kernel_size	*The size n of the kernels ($n \times n$) affecting the feature extraction and receptive field*	2, 3, 4, 5
dropout_rate	*The probability of randomly dropping neurons during training to prevent over fitting*	0 to 0.5 in steps of 0.1
learning_rate	*The step size at which the model updates its weights during optimisation*	0.0001 to 0.01
f_pos_penalty	*The multiplicative weight of the penalty applied to false positives in the loss function*	0 to 5 in steps of 0.1

Table 1: *config* dictionary of arguments and parameters passed to CNN model

Batch Normalisation: After the final convolutional layer, batch normalisation is applied to stabilise and accelerate training. Testing without this layer resulted in inconsistent results with the model occasionally failing to converge at all.

Dropout: A dropout layer is applied after batch normalisation to regularise the model and prevent overfitting, with the dropout rate configurable via `dropout_rate`.

Fully Connected Layer: The output of the former is flattened and passed through a fully connected dense layer. The size of the fully connected layer is dynamically calculated by passing a dummy tensor of `input_shape` through the model when initialised. This dense layer maps the features learned by the kernels to the number of `n_classes`, which for the binary classifier is two.

Output Layer: This final output corresponds to the raw predictions (logits) for each class. These

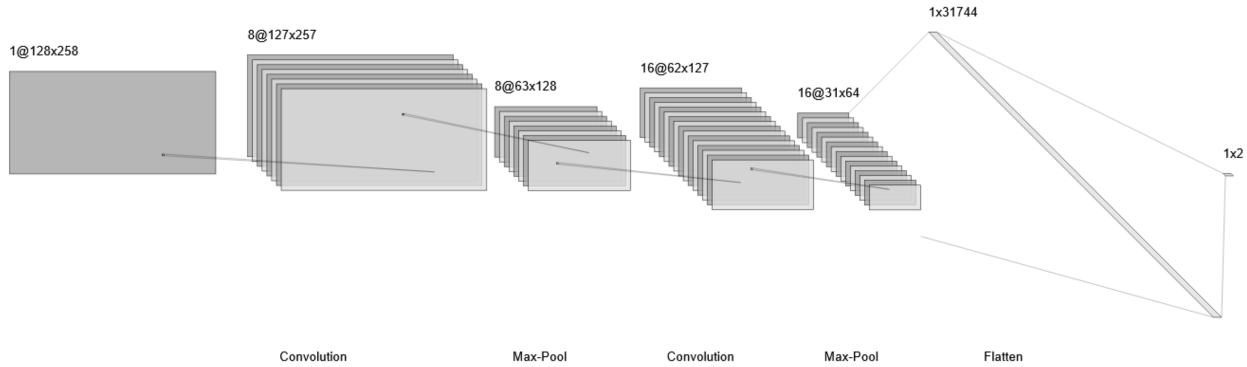


Figure 7: Simplified diagram of CNN with 2 convolutional layers, a kernel size of 2, 8 initial kernels, a single mel spectrogram as input, and a vector of raw class probabilities as output. Dimensions of elements of each layer are shown above, with operations labelled below in the order performed.

logits can be passed through a softmax function to then obtain class probabilities.

Loss Function: The model uses a custom loss function, combining the method supplied by `criterion`; Cross Entropy Loss, with a weighted penalty for false positives, given by `f_pos_penalty`. Cross Entropy Loss measures the difference between the predicted probability distribution, or the models output, *y*, and the true probability distribution, or ground truth labels, *p*. For binary classification it is calculated as follows:

$$Cross\ Entropy\ Loss = -(y\,log(p) + (1-y)log(1-p))$$

This function is commonly used in image classification tasks. It penalises confidently incorrect predictions heavily and encourages the model to output high probabilities for the correct class (Goodfellow, Bengio, & Courville, 2016b). In the case of a false positive a penalty is added to the loss. During the development of the model, much like with the model of Xia et al. (2021), false positive results were quite common in early tests. This additional hyperparameter was added to attempt to mitigate this by further penalising false positives.

Optimiser and Learning Rate Scheduler: The model uses a configurable learning rate (`learning_rate`) and the Adaptive Moment Estimation (Adam) optimiser. Adam adjusts the learning rate for each parameter individually using estimates of the mean and variance of the gradients, effectively combining two well established optimisation techniques, Momentum and RMSProp (Kingma & Ba, 2014). Additionally a learning rate scheduler is used to reduce the learning rate when the validation loss plateaus as this can avoid oscillations and divergence (Goodfellow, Bengio, & Courville, 2016c). Specifically the *ReduceLRonPlateau* scheduler with a patience of ten was implemented, reducing the learning rate by a factor of 0.1 if the validation loss hasn't changed by more than 0.0001 within the last ten epochs.

3.2.2 Hyperparameters

Hyperparameter optimisation was performed using *optuna* (Akiba, Sano, Yanase, Ohta, & Koyama, 2019). With *optuna* a range of values for specific hyperparameters can be defined, as seen in the *config* dictionary. An objective is then set, for example to minimise the validation loss, and *optuna* will sample the search space of all possible hyperparameter combinations training the model and evaluating it's performance by the objective for a set number of trials.

The objective for the CNN was defined as minimising the validation loss, that is the output of the loss function when the model after each epoch is passed over the validation set to simulate its current performance on unseen data.

The sampling method chosen was the Tree-structured Parzen Estimator Sampler (TPESampler). This is a Bayesian optimisation algorithm designed to efficiently explore the hyperparameter space (Watanabe, 2023). It begins by first randomly sampling the search space for a number of trials, in this case ten. It then takes the best performing trials and models the distribution of those hyperparameters $l(x)$ using a Gaussian Mixture Model (GMM). GMMs consist of several overlapping bell curve distributions, the hyperparameters are assumed to belong to these distributions, with the mean and variance of the well performing hyperparameters defining the parameters of each curve. The hyperparameters of the remaining trials are also modelled with a GMM $g(x)$. When selecting the next set of hyperparameters to test, rather than randomly sampling the entire search space, hyperparameters that maximise the ratio of $\frac{l(x)}{g(x)}$ are chosen, as based on the past trials these are more likely to do well.

In total 100 trials were performed to optimise the `n_layers`, `n_kernels`, `kernel_size`, `dropout_rate`, `learning_rate`, and `f_pos_penalty` hyperparameters. Early stopping with a patience of twenty was used, ending trials if the validation loss did not change for that many epochs, allowing an arbitrarily large `max_epochs`. Median Pruning was also used to end poorly performing trials early. This functions by tracking the objective functions result at each epoch of each trial. When this value for a trial is worse than the median value of the preceding trials at that epoch, that trial is cut short as it is unlikely to yield better than average results. Several trials are completed first to generate a median value in this case five. An example of this process is presented in Figure 8. The parameters `num_workers` and `batch_size` were chosen based on available hardware. The intermediate results of the optimisation process are shown in Figure 10.

The optimal set of hyperparameters were found on trial 31, the validation loss over this trial is shown in figure 9. These optimal hyperparameters are shown in Table 2.

Hyper Parameter	Tuned Value
`n_layers`	2
`n_kernels`	8
`kernel_size`	2
`dropout_rate`	0.1
`learning_rate`	0.00284
`f_pos_penalty`	0

Table 2: Optimised values of hyperparameters for CNN model

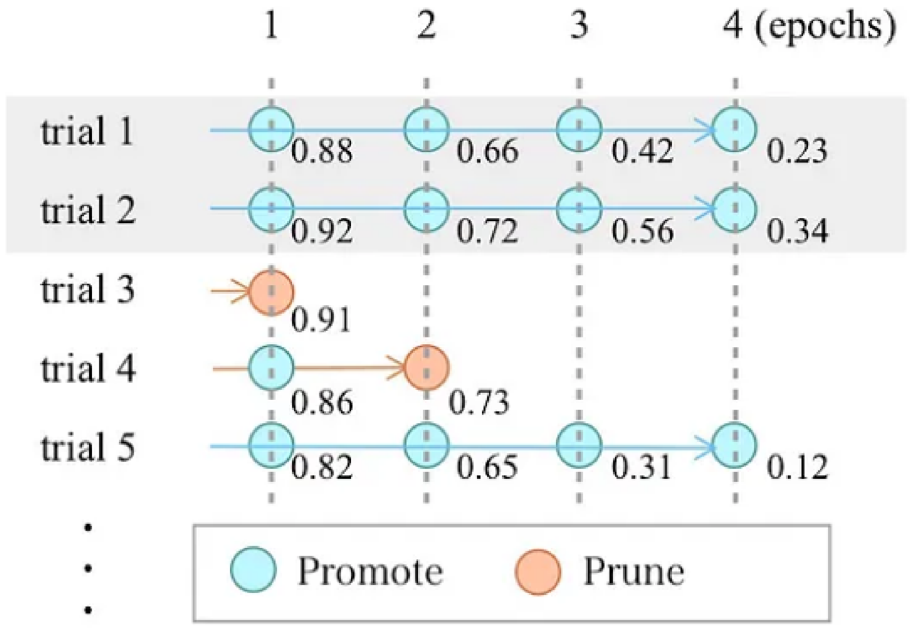


Figure 8: An illustration of Median Pruning for validation loss minimisation. Trials in grey are warm-up trials, values are the validation loss returned by the objective function. As the objective is to minimise this loss, trial 3 is pruned as a loss of 0.91 is worse than the median of the previous trials at that epoch ($median(0.88, 0.92) = 0.9 < 0.91$). (Medium, 2023)

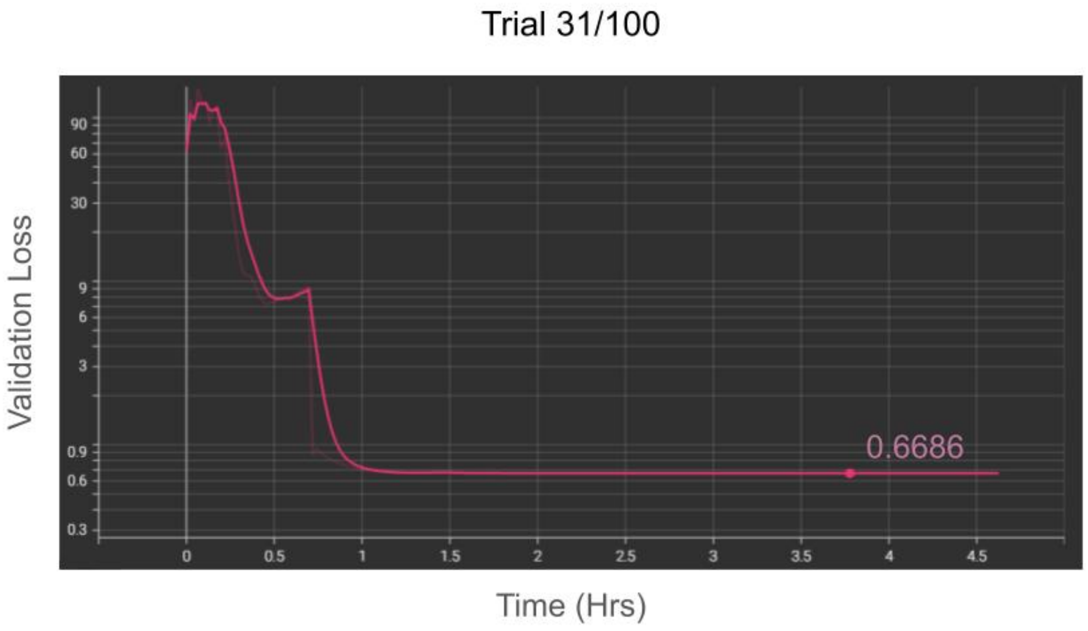


Figure 9: Validation loss of best performing trial over time. Minimum validation achieved before triggering early stopping indicated on graph with a value of 0.6686

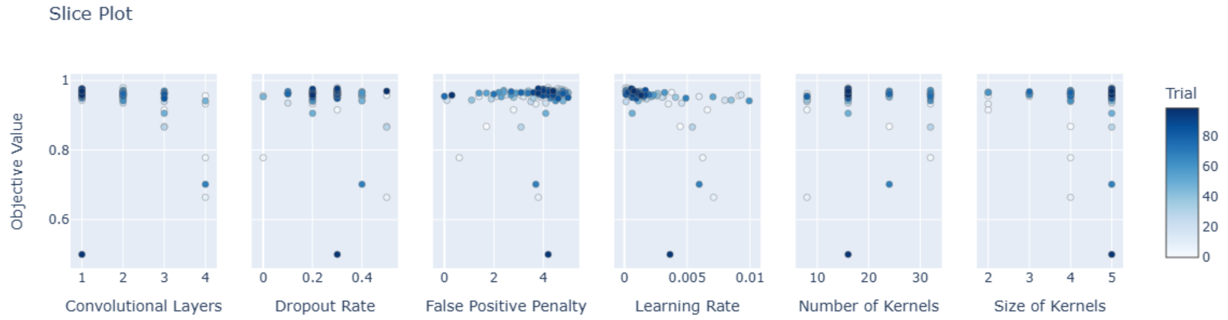


Figure 10: Values tested for optimisation of hyperparameters of CNN model over trials

3.3 Variational Autoencoder (VAE)

Similarly to the CNN, this model was configured with a *config* dictionary, found in Table 3. It was built using the same packages as the CNN, and its structure and hyperparameters are treated below:

3.3.1 Architecture

The components of the architecture unique to the VAE are treated below:

Input Layer: The input is the same as the CNN with the addition of label information. As this model is trying to construct a latent space rather than classify, the label information (the vowel being represented and the class to which the example belongs) is concatenated with the input. This will condition the VAE on these class labels, allowing a specific label to be requested when generating new samples.

Encoder: The encoder resembles the previously described CNN very similarly with a few changes detailed here. Batch normalisation is applied after each layer to aid stability. A LeakyReLU activation function is used instead of ReLU. This function acts the same as ReLU but applies a small multiplier to values the ReLU would set to 0. This allows for small gradients for negative inputs rather than shutting those neurons down. It can help mitigate the 'dying ReLU' problem where too many neurons are deactivated and the model fails to converge, an issue that occurred while building this model. Dropout is also performed after each convolutional layer rather than at the end as this was found to help with overfitting more in this model. Finally once the output is flattened, it is passed through two separate fully connected layers to produce the mean and log variance.

Latent Space: The latent space is defined by a Gaussian distribution with mean and log variance output by the encoder. The reparametrisation trick is used to sample from this distribution during training, allowing backpropagation through the stochastic sampling process.

Decoder: The decoder consists of a series of transposed convolutional layers that progressively upsample the latent representation to reconstruct a mel spectrogram. It mirrors the encoder but inversely with batch normalisation, LeakyReLU, and dropout being applied after each transposed convolutional layer. The final output is passed through a hyperbolic tangent (tanh) function. This ensures the output is in the same range [-1,1] as the normalised mel spectrograms used as inputs (Goodfellow, Bengio, & Courville, 2016a). It is defined as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Loss Function: The loss function consists of two components, the reconstruction loss and the KL-divergence. The reconstruction loss is computed using the supplied `criterion`, in this case Mean Squared Error (MSE) Loss. This measures the the average distance between the input mel spectrogram and the reconstructed one. It is well suited for continuous data and pushes the model to minimise pixel-wise differences, while also being differentiable and aligning with the Gaussian assumptions of VAEs (Goodfellow et al., 2016a). In this case the reduction parameter of the MSE loss is set to 'sum', providing the sum of squared differences rather than the mean. As the total loss is a combination of the reduction and KL-divergence, summing rather than taking the mean ensures both components are on similar scales, balancing their contributions (Kingma & Welling, 2022). MSE_{sum} is given by:

$$MSE_{sum}(x,\hat{x}) = \sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

where x_i is the true value, $\hat{x}_i$ is the predicted value, and N is the total number of elements.
The KL-divergence measures the difference between the learned latent distribution and a standard Gaussian prior. Its weight is controlled by the `beta` parameter. The sum of these two components is the total loss.

Optimiser and Learning Rate Scheduler: Once again the Adam optimiser and ReduceLROnPlateau scheduler are used.

3.3.2 Hyperparameters

Hyperparameter optimisation was again performed with *optuna*. The objective function was defined as minimising the validation loss. The sampling method was once again TPESample. 100 trials were attempted to optimise the `n_layers`, `n_kernels`, `kernel_size`, `dropout_rate`, `learning_rate`, and `relu_rate` hyperparameters. As the input data could be noisy, a `beta` of 0.1 was used to maximise reconstruction accuracy, for higher quality audio inputs a larger value may be more useful. Early stopping and median pruning are used again, this time with a patience of just 10, to minimise computational load at the cost of a higher risk of not finding the global minimum of validation loss. Additionally the `num_workers` is set to 0, slowing down training but once more reducing resource demands. Finally the `batch_size` is reduced to the minimum, risking poorer stability during training for less computational demands. These changes were implemented to optimise memory use as the more complex VAE is significantly more resource intensive.

Parameter	Description	Value(s)
`num_workers`	*The number of subprocesses used for data loading Higher values can speed up data processing but require more memory*	0
`batch_size`	*The number of training samples processed in one forward/backwards pass or gradient update. Higher values can increase stability during training but use more memory.*	1
`criterion`	*The loss function used to measure the difference between predicted and actual outputs*	Mean Squared Error Loss
`max_epochs`	*The maximum number of times the model will iterate over the entire training dataset*	1000
`n_classes`	*The number of distinct labels of the input data. Two classification labels, by each of the five vowel sounds, for both recorded sexes*	20
`example`	*A tuple containing a mel spectrogram and waveform, to access the dimensions of the input data, (channels, height, width)*	(mel,wave)
`n_layers`	*The number of convolutional layers in the network, determining its depth and complexity*	1, 2, 3, 4
`latent_dim`	*Size of the latent space learned by the model for generating new examples. Higher values may capture more subtle features but increases risks of overfitting*	16 to 400 in steps of 48
`n_kernels`	*The number of kernels (feature detecting filters) in the first convolutional layer*	8, 16, 24, 32
`kernel_size`	*The size n of the kernels ($n \times n$) affecting the feature extraction and receptive field*	2, 3, 4, 5
`dropout_rate`	*The probability of randomly dropping neurons during training to prevent over fitting*	0 to 0.5 in steps of 0.1
`learning_rate`	*The step size at which the model updates its weights during optimisation*	0.0001 to 0.01
`relu_rate`	*A multiplicative weight applied to negative inputs of the ReLU function allowing a small gradient rather than 0. Can address the problem of ReLU setting deactivating too many neurons*	0 to 0.1 in steps of 0.01
`beta`	*The multiplicative weight applied to the KL-divergence in the loss function. Higher values increase divergence giving a more structured latent space while lower values can improve reconstruction accuracy*	0.1 to 1 in steps of 0.1, then 1 to 10 in steps of 1

Table 3: *config* dictionary of arguments and parameters passed to VAE model

4 Results

With defined architecture and tuned hyperparameters, results can be generated. This section presents the results of the models, evaluating their performance through several metrics discussed below, first treating the CNN, then the VAE.

4.1 CNN

All results were obtained from the sequestered training dataset, using the optimal model found during training trials. The broader performance of the model is presented, as well as its results at a tuned classification threshold, and its comparison to arbitrary guesswork.

4.1.1 Precision and Recall Curve

The precision-recall curve plots the model's precision (the fraction of correct positive predictions) against its sensitivity or recall (the fraction of actual positives detected) across different classification thresholds (at what output value the model will label an input as belonging to the positive DMV class). It is presented in Fig 11. By analysing this curve, the threshold that maximises both precision and recall can be found (around the 'knee' of the curve, or the point closest to (1,1) on the graph). This is represented by an F1 score, the harmonic mean of precision and recall, which balances the two metrics for optimal decision-making. A high F1 score (closer to 1) indicates that the model achieves both high precision and high recall, while a low score (closer to 0) reveals a trade-off between the two, such as excessive false positives undermining precision or missed true positives reducing recall. It is given by:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall} + c}$$

where c is some small constant (1×10^{-9}) to prevent division by zero should the model completely fail in precision or recall. This is a useful metric as it penalises missed positives and false alarms (which are costly in a medical context) more harshly then just looking at the overall accuracy. The F1 score was found to be 0.695 (95% CI [0.676, 0.715]) with 0.355 as the associated classification threshold.

4.1.2 Threshold Dependent Metrics

With an ideal threshold found, more general metrics can be investigated, like accuracy. As previously stated, sensitivity (or recall) indicates how well the model identifies true positive cases, while specificity measures the model's ability to correctly identify negative cases. These are presented in Table 4. A confusion matrix visually presents these metrics together, showing the distribution of true positives, false positives, true negatives, and false negatives. It is presented in Figure 12.

4.1.3 Area Under the Receiver Operating Characteristic Curve

The receiver operating characteristic (ROC) curve is obtained by plotting the sensitivity of the model against the false positive rate (1 - specificity) for different thresholds of classification. The area under

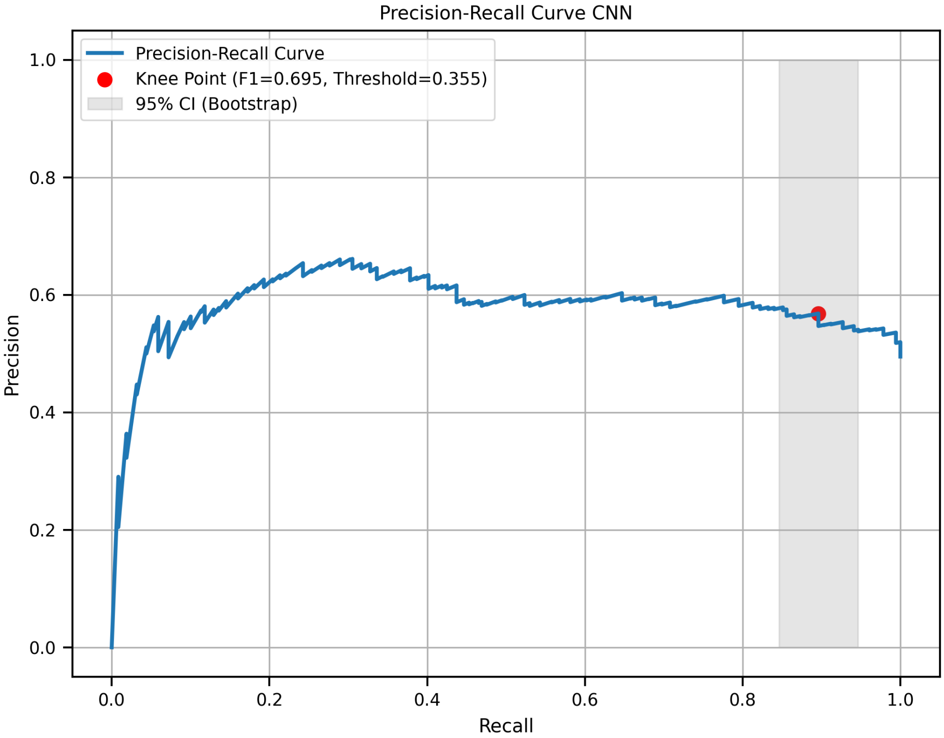


Figure 11: Precision Recall Curve for CNN. Knee point (red) best balance between precision and recall (F1 = 0.695, threshold 0.355), with shaded region showing 95% confidence interval across bootstrap resamples (N = 10^4)

Metric	Value
Accuracy	61.12%
Sensitivity	89.61%
Specificity	33.21%

Table 4: Threshold Dependent Metrics of CNN Model Performance

this curve (AUROC) represents the degree of separability, or how capable the model is of distinguishing between classes. This AUROC value ranges between 0 and 1 with 1 being a perfect model, 0.5 being equivalent to random guessing, and anything less being worse than random guessing. As the AUROC covers all possible classification thresholds, it provides a comprehensive measure of model performance. For this model an AUROC of 0.648 was obtained, the ROC curve is shown in Fig 13.

4.2 VAE

At the time of writing no VAE results are available, as no VAE model was able to successfully converge and produce an output of more than noise. This was due to the models computational requirements during training exceeding available resources. More details are provided in the following section.

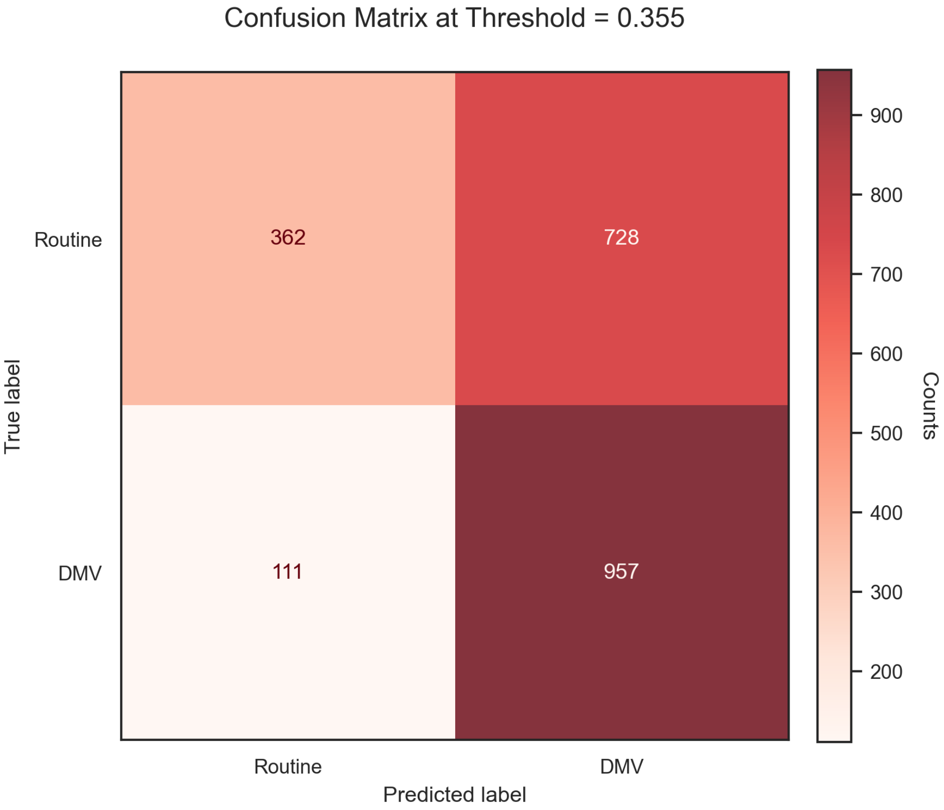


Figure 12: Confusion matrix for binary classification at threshold 0.355, showing model performance distinguishing Routine (negative class) and DMV (positive class) cases. Colour intensity represents count frequency in each cell

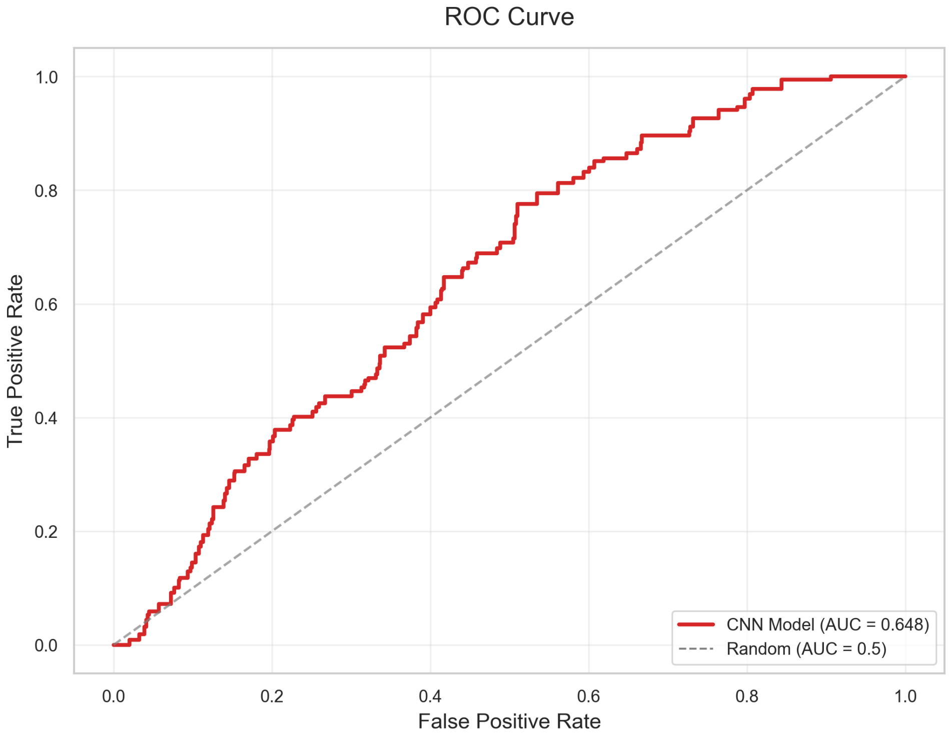


Figure 13: Receiver operating characteristic curve for CNN model showing classification performance. Red line shows models performance against grey-dashed line representing random guessing. Area under the curve (AUC) quantifies performance across all classification thresholds

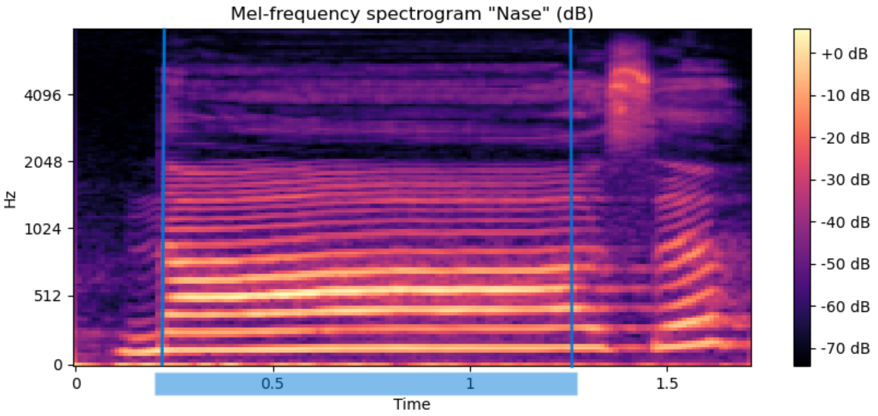


Figure 14: Mel-frequency spectrogram of the word ”Nase”. Time on the x-axis given in seconds, frequency on the y-axis in Hz and amplitude in decibels represented by colour intensity. Onset and offset of the ”a” in ”Nase” indicated by blue vertical lines, with its duration highlighted on the x-axis

5 Conclusion

With these results, this section presents the key findings, discussing their implications in the context of model performance. Limitations of the study, including noise in audio recordings and dataset constraints, are also explored, followed by an assessment of each model's results. Finally, directions for future research are proposed, emphasising improvements in audio representations and possible hybrid models.

5.1 Discussion

Before discussing the results of the individual models, the main limiting factor affecting both, the audio recordings themselves, needs to be addressed. The data is very noisy. While this can be useful for a classifier to help it distinguish noise from the start, it may take longer for a model to learn significant patterns in the data and usually requires larger training sets to achieve good results. The audio recordings were gathered in an uninsulated doctor's office. The distance to the microphone was not always consistent, and occasionally major background noise was picked up (examples include someone coughing in the next room, and on more than one occasion a nurse entering to speak to someone).

Furthermore, the segmentation of the data was not optimal. The audio was originally recorded in one continuous file for each patient, each about 2 minutes long, consisting of the vowel containing words used in this study and other audio for different applications. Each file was then annotated by hand by one of the medical students marking the start and end of each word on the recording. Looking back at Figure 2, the actual vowel in the word ”*Nase*” is captured from time 0.2 to 1.25, recognisable in the spectrogram by the long horizontal bands indicative of a vowel. Before this we see background noise followed by a vertical artifact of a consonant (”*N*”), and afterwards the high frequency of the sibilant ”*S*” and another shorter vowel sound (”*E*”). As it is not expected to obtain much useful anatomical information from the consonants, these areas are functionally additional noise (Moon et al., 2012). While all audio files were trimmed down to an optimal length that still adds a large amount of noise to the data. An example of the spectrogram with the useful vowel highlighted is provided in Figure 14.

Ideally the models would only be trained on the actual vowels extracted from the overall word. While this would be interesting for the CNN to compare its performance on noisy data, it would be very beneficial for the VAE. These models can be quite susceptible to noise, encoding it into the latent space and making it difficult for the decoder to recreate meaningful patterns. Extracting just the vowels from the recordings is practically feasible, however it would require either the precise and tedious work of noting the exact start and end point of each vowel in the thousands of spectrograms, or training an additional unsupervised model to detect and extract the vowels from the spectrograms, beyond the scope of this study.

Noise aside, the sample size was well constructed, with a representative distribution of ages and genders, and multiple recordings per participant strengthened the dataset's robustness. A larger test group would be preferable as only around 10% of the patients are in the DMV group, and while oversampling of the group was performed to balance the data a larger starting pool is always preferred. It is difficult to say what effect this had on performance.

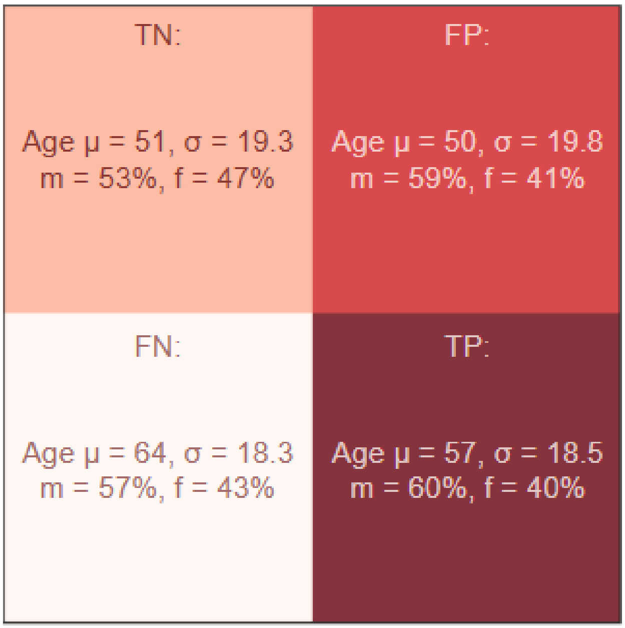
5.1.1 CNN

With an accuracy of 60%, the CNN exhibited suboptimal discriminatory power. That being said, it does perform better than guessing, suggesting there may in fact be something to the clinical intuition. Looking at the F1 score (0.695), the model achieves moderate balance between precision and recall, though not optimal. At the corresponding threshold, there is a clear trade off in favour of sensitivity. At 89.61% the model rarely misses a DMV case, outperforming the sensitivity of Xia et al. (2021)'s difficult intubation model in this metric. That is coming at the cost of a much lower specificity however. At 33.21% (almost half that of the aforementioned study) this model will generate a lot of false positives.

Turning to the confusion matrix in Figure 12, there is nothing immediately distinctive about the four groups in terms of constitution. All four consist of almost half male and female patients with roughly equivalent spreads of age. These can be seen in Figure 15. This is positive for the model as there is no one population it seems biased towards, but also doesn't reveal any useful avenues for improvement.

Investigating the proportion of vowels found in each group within the confusion matrix tells a slightly different story. As can be seen in Figure 16, the vowel 'i' is overrepresented in the false negative, and to a lesser extent, true negative groups. Furthermore the vowel 'a' is under-represented in the same groups, while the vowel 'e' is completely absent from the false negative group. What this tells us in practice though is that the model is simply biased in different directions towards different letters, leaning towards a negative prediction for 'i' and for 'a' and 'e' a positive one, stemming from the different resonant frequencies of the vowels. The 'i' for example is often characterised by higher frequencies than the other vowels (Moon et al., 2012).

Across thresholds the AUROC of 0.648 is suboptimal. As stated, this is better than guessing, suggesting the model is picking up on some quantifiable pattern in the voice that indicates DMV, but not effectively enough to confirm the validity of the clinical intuition, nor to be of use diagnostically, particularly with its propensity for false positives. In its current state, it is of questionable usefulness for clinical settings, and without a solid interpretable model (discussed shortly) to back up its classifications, its acceptance by physicians is doubtful.



TN: Age μ = 51, σ = 19.3 m = 53%, f = 47%	FP: Age μ = 50, σ = 19.8 m = 59%, f = 41%
FN: Age μ = 64, σ = 18.3 m = 57%, f = 43%	TP: Age μ = 57, σ = 18.5 m = 60%, f = 40%

Figure 15: Age and gender spreads for groups in confusion matrix at threshold 0.355, with mean and standard deviation of ages as well as percentages of male and female (m/f). Groups cover true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP)

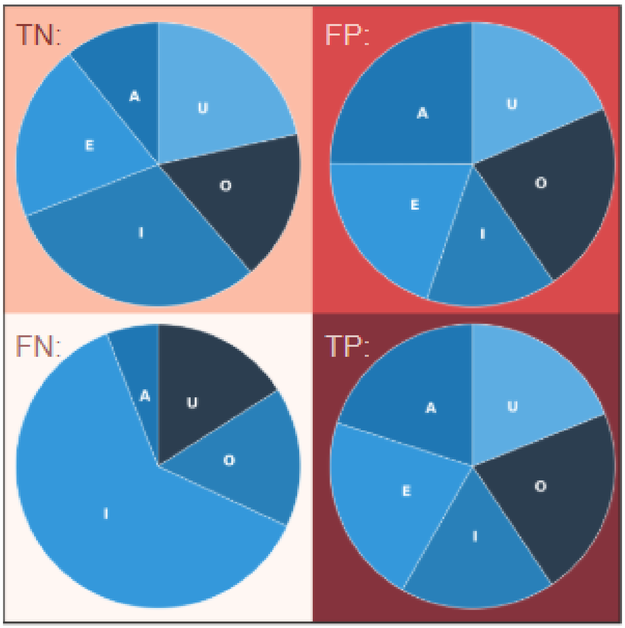


Figure 16: Proportion of vowels contained within each group of confusion matrix. Groups cover true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP)

That being said, the slight but meaningful ability to identify some DMV cases from voice alone is of note. Though difficult to make a direct comparison, El-Orbany and Woehlck (2009) state in their review that physical and patient history findings failed to predict unexpected DMV in 57% of patients in previous studies. That equates to a sensitivity of 43%. The sensitivity achieved here is more than double that, though with a high false positive rate that does not necessarily mean it is a superior model. With a specificity of 33.21% this model will give false alarms at a rate likely to trigger the 'cry wolf' effect, where systems who repeatedly give false alarms begin to be ignored (Breznitz, 2013).

While, as discussed shortly, further research is certainly necessary, this model adds evidence to the hypothesis that subtle acoustic features can be used diagnostically, and that there may be something to this clinical intuition.

5.1.2 VAE

As stated previously, a working VAE was not achieved. Despite measures to minimise the computational demands, the models requirements were beyond what was available for this project. The bigger a model, and the longer it takes to train, the more memory is required. No promising VAE was able to train long enough to start producing usable results. This is in part due to the general complexity of a VAE. Having twice the layers of the average CNNs tested here, they will simply require more resources to train. The VAE outputs an image which is compared pixel wise to the input image. The relative complexity of these outputs translates to a significant time and resource increase, as to effectively learn the latent space to then recreate an image takes many epochs, considerably more than the CNN required to output a classification.

The size of the mel-spectrograms themselves is also a contributing factor. In their paper, Ntalampiras and Potamitis (2021) were able to produce synthetic bird calls using a VAE with similar architecture and methodology as used here. An example of their input and output is shown in Figure 17, the x-axis is given in frames, but corresponds roughly to a fraction of a second. Comparing their input to the mel-spectrograms used here (such as in Figure 2), the discrepancy in size and complexity is immediately apparent.

The bird call spectrograms are an order of magnitude shorter, the audio for the mels here last around 3 seconds, while the bird calls last around 0.3 seconds. The number of mel-bands used (shown on the y-axis) also differs. Given in Hz in this paper, the bird calls use around half as many. This has an effect on reconstruction, when a mel-spectrogram is converted back to audio. A high number of mel-bands was necessary here as early testing found that fewer bands in the mel-spectrogram resulted in poorer, more robotic sounding audio when converted back into a playable sound wave. As the goal was to create usable examples a large number of bands is necessary, however once again this has a computational and resource cost associated with it. Of note is a comment in Ntalampiras and Potamitis (2021)'s paper that while visually their VAE was successful in generating new bird calls, the quality of the audio generated was less than desired, often sounding robotic, distorted, or unnatural.

Lastly, the somewhat underwhelming performance of the CNN will have also carried over into the VAE. As discussed, the initial encoder of a VAE is essentially a CNN itself. If the CNN is having trouble identifying clear patterns, the VAE may struggle to separate clear features in its compact latent space representation. This will lead to blurrier, less distinct images being generated, and a longer

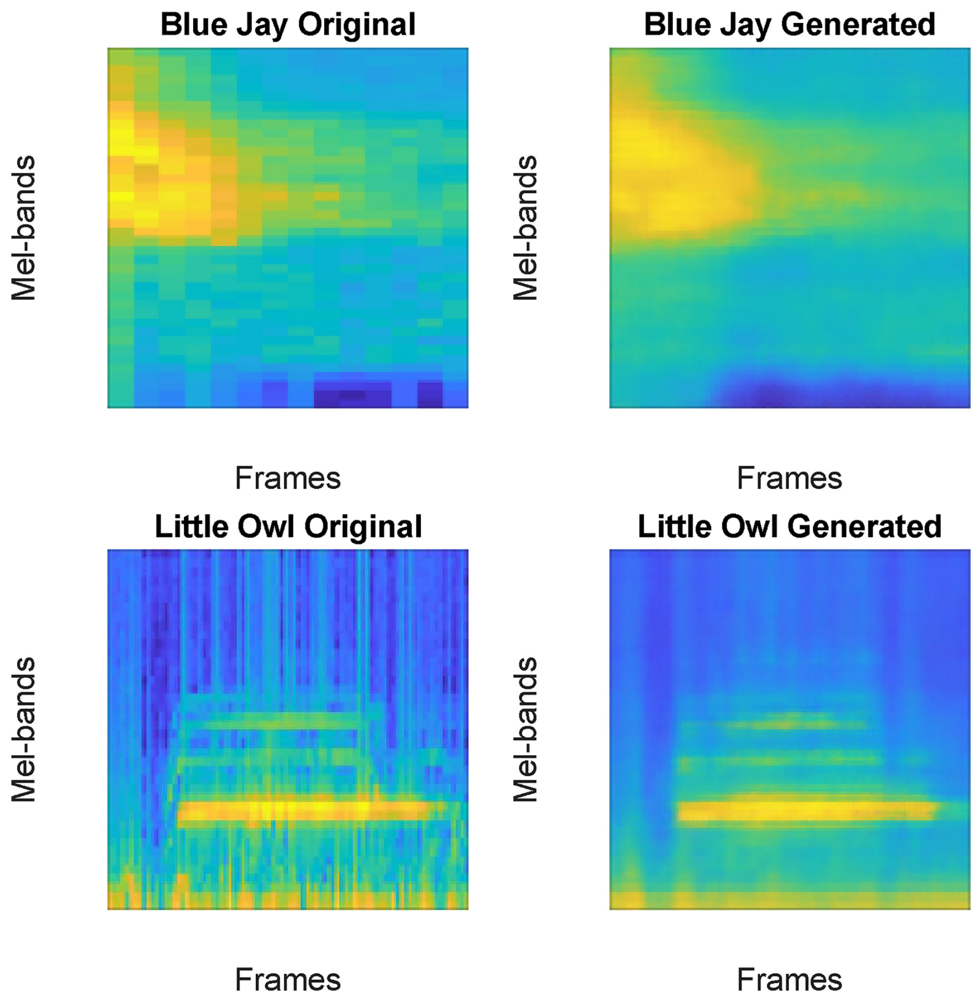


Figure 17: Example of mel spectrograms from Ntalampiras and Potamitis (2021), showing input spectrograms on the left, and those output by a VAE trained on bird calls on the right

training time as it requires more epochs to create a meaningful latent space.

In short, large highly detailed images, that have proven to be difficult to classify accurately, in combination with a complex model architecture, lead to a VAE model that required more computational resources than were available for this project.

5.2 Summary of Main Contributions

Despite lacklustre generative results, this research has found that DMV can be predicted from a patients voice alone at a rate better than guessing. It expands on and supports the findings of previous audio based predictions of difficult intubation, and contributes to the exploration of novel, low cost, and non-intrusive, diagnostic techniques. In its shortcomings, this study also highlights the challenges of interpretable AI within medical applications. These findings may serve as a starting point for future methodologies aimed at improving model transparency and reliability. The issues encountered here provide a case study demonstrating potential difficulties in applied AI in healthcare, hopefully offering valuable insights for the refinement of future approaches.

5.3 Future Work

To close this paper out, several potential next steps are discussed, including adjustments to the models, the audio itself and how it is represented, and finally some adjacent research that may shed more light on the results found here.

5.3.1 Augmented Models

As previously mentioned, there are some pre-existing predictors of DMV, such as presence of beard, high BMI, history of snoring (El-Orbany & Woehlck, 2009). These measurements could be combined with a patients audio data to train a hybrid model for classification. This could involve encoding this information into the mel-spectrograms, similar to how the VAE handles conditional label information, or with a multi branch approach, analysing the images with a CNN and using another architecture to make predictions from the additional information, with both branches being combined to produce a prediction. These predictors are recorded for all patients prior to mask ventilation, making the data readily available.

5.3.2 Audio Representations

It may be possible to acquire the useful physiological information from the voice without relying on vowel sounds. An electrolarynx is a handheld device often used in cases where an individual has lost their voice box (for example due to cancer of the larynx). Exhaling while holding the device against the throat creates vibrations that resonate through the upper airways, which can be shaped with the lips and tongue to create a robotic sounding voice. Simply exhaling with the mouth and jaw slack creates a monotone buzz that may also capture information about the anatomy of the mouth and throat. Using these buzzes rather than vowels could provide a more homogenous dataset, eliminating the variance of the words used here and reducing noise, as the monotone audio would be much easier to crop without losing information. An electrolarynx is also a simple battery operated device present already in many clinical settings, meaning if classification (and generation) is possible with this data, the barrier to clinical use is vastly smaller than with say ultrasounds. Electrolarynx audio was recorded for the patients of this study, making this the next logical step in future.

Regardless of the exact audio, this research focussed on human-centered audio representation, training models on what a human brain would detect when hearing the voice recordings. While this was done to explore the validity of clinical intuition, there may be acoustic features that predict DMV outside of human perception. To investigate this further, alternative representations of the audio could be experimented with, such as full spectrograms. This may require new recordings however, as low frequencies may be lost in the background noise of this slightly messy dataset. A model that is able to achieve success with more general audio representations will have to contend with its own set of interpretability issues, being further removed from human perception.

5.3.3 Clinical Intuition

Finally, on that human perception, this research was partly inspired by the anecdotal evidence of experienced physicians and their clinical intuition. To better understand the results of this paper, it would be beneficial to test this intuition. By having experienced physicians listen only to the same inputs that the models receive and make a prediction of DMV based on that, the models results can be put into a clearer context. The augmented hybrid model would also be interesting to compare to physicians predictions when they have access to the audio and this meta data. Lastly, this would also shed some light on the often contentious topic of clinical intuition, perhaps broadening our understanding of it, at least in the context of DMV.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining.*

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., ... Chintala, S. (2024, April). Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *29th acm international conference on architectural support for programming languages and operating systems, volume 2 (asplos '24).* ACM. Retrieved from `https://pytorch.org/assets/pytorch2-2.pdf` doi: 10.1145/3620665.3640366

Audacity-Team. (2023). *Audacity: Free audio editor and recorder.* The Audacity Team. Retrieved from `https://www.audacityteam.org/`

Bajwa, J., Munir, U., Nori, A., & Williams, B. (2021). Artificial intelligence in healthcare: transforming the practice of medicine. *Future healthcare journal*, *8*(2), e188–e194.

Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. In L. Rokach, O. Maimon, & E. Shmueli (Eds.), *Machine learning for data science handbook: Data mining and knowledge discovery handbook* (pp. 353–374). Springer International Publishing. Retrieved from `https://doi.org/10.1007/978-3-031-24628-9_16` doi: 10.1007/978-3-031-24628-9_16

Breznitz, S. (2013). *Cry wolf: The psychology of false alarms.* Psychology Press.

Cao, S., Xia, M., Zhou, R., Wang, J., Jin, C.-Y., Pei, B., ... Jiang, H. (2021, December). Voice parameters for difficult mask ventilation evaluation: an observational study. *Ann. Transl. Med.*, *9*(23), 1740.

El-Orbany, M., & Woehlck, H. J. (2009). Difficult mask ventilation. *Anesthesia & Analgesia*, *109*(6). Retrieved from `https://journals.lww.com/anesthesia-analgesia/fulltext/2009/12000/difficult_mask_ventilation.25.aspx`

Falcon, W., & The PyTorch Lightning team. (2019, March). *PyTorch Lightning.* Retrieved from `https://github.com/Lightning-AI/lightning` doi: 10.5281/zenodo.3828935

Frasca, M., La Torre, D., Pravettoni, G., & Cutica, I. (2024). Explainable and interpretable artificial intelligence in medicine: a systematic bibliometric review. *Discover Artificial Inelligence*, *4*(15).

Fukushima, K. (1969). Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, *5*(4), 322-333. doi: 10.1109/TSSC.1969.300225

Gold, B., Morgan, N., & Ellis, D. (2011). *Feature extraction for audio signals.* Wiley. Retrieved from `https://www.wiley.com/en-us/Speech+and+Audio+Signal+Processing%3A+Processing+and+Perception+of+Speech+and+Music-p-9780470649427`

Goodfellow, I., Bengio, Y., & Courville, A. (2016a). *Autoencoders.* MIT Press. Retrieved from `https://www.deeplearningbook.org`

Goodfellow, I., Bengio, Y., & Courville, A. (2016b). *Deep feedforward networks.* MIT Press. Retrieved from `https://www.deeplearningbook.org`

Goodfellow, I., Bengio, Y., & Courville, A. (2016c). *Optimization for training deep models.* MIT Press. Retrieved from `https://www.deeplearningbook.org`

Greenhalgh, T. (2002). Intuition and evidence–uneasy bedfellows? *British Journal of General Practice*, *52*(478), 395–400. Retrieved from `https://bjgp.org/content/52/478/395`

Gupta, A., & Raina, P. (2024). Mask ventilation. In N. Gupta, R. S. Ubaradka, A. Gupta, & D. K. Tripathy (Eds.), *Techniques in anesthesia, intensive care and emergency medicine* (pp.

19–24). Singapore: Springer Nature Singapore. Retrieved from https://doi.org/10.1007/978-981-96-1202-4_3 doi: 10.1007/978-981-96-1202-4_3

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* Retrieved from https://arxiv.org/abs/1502.03167

Kayalibay, B., Jensen, G., & van der Smagt, P. (2017). Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Retrieved from https://arxiv.org/abs/1412.6980

Kingma, D. P., & Welling, M. (2022). *Auto-encoding variational bayes.* Retrieved from https://arxiv.org/abs/1312.6114

Lin, H.-Y., Tzeng, I.-S., Hsieh, Y.-L., Kao, M.-C., & Huang, Y.-C. (2021). Submental ultrasound is effective in predicting difficult mask ventilation but not in difficult laryngoscopy. *Ultrasound in Medicine Biology*, *47*(8), 2243-2249. Retrieved from https://www.sciencedirect.com/science/article/pii/S0301562921001848 doi: https://doi.org/10.1016/j.ultrasmedbio.2021.04.004

MathWorks. (2017). *Introduction to deep learning: What are convolutional neural networks?* Video. Retrieved from https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

Medium. (2023). *An introduction to the implementation of optuna: A hyperparameter optimization framework.* https://medium.com/optuna/an-introduction-to-the-implementation-of-optuna-a-hyperparameter-optimization-framework-33995d9ec354. ([Image from ”An Introduction to the Implementation of Optuna: A Hyperparameter Optimization Framework”])

Moon, K. R., Chung, S. M., Park, H. S., & Kim, H. S. (2012). Materials of acoustic analysis: Sustained vowel versus sentence. *Journal of Voice*, *26*(5), 563-565. Retrieved from https://www.sciencedirect.com/science/article/pii/S0892199711001639 doi: https://doi.org/10.1016/j.jvoice.2011.09.007

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., ... Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 ieee international conference on signal and image processing applications (icsipa)* (pp. 342–347).

Ntalampiras, S., & Potamitis, I. (2021). Acoustic detection of unknown bird species and individuals. *CAAI Transactions on Intelligence Technology*, *6*(3), 291–300.

O’shaughnessy, D. (1987). *Speech communications: Human and machine (ieee).* Universities press.

Otten, N. V. (2023). *Variational autoencoders (vaes) made simple & how to tensorflow tutorial.* Retrieved from https://spotintelligence.com/2023/12/27/variational-autoencoders-vae/

Robinson, R. (2017). *Convolutional neural networks (cnn) tutorial.* Retrieved from https://mlnotebook.github.io/post/CNN1/ (Accessed: [Insert Date Accessed])

Sataloff, R. T., Heman-Ackah, Y. D., & Hawkshaw, M. J. (2007). Clinical anatomy and physiology of the voice. *Otolaryngologic Clinics of North America*, *40*(5), 909-929. Retrieved from https://www.sciencedirect.com/science/article/pii/S0030666507000874 (The Professional Voice) doi: https://doi.org/10.1016/j.otc.2007.05.002

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Vanstone, M., Monteiro, S., Colvin, E., Norman, G., Sherbino, J., Sibbald, M., ... Peters, A. (2019).

Experienced physician descriptions of intuition in clinical reasoning: a typology. *Diagnosis*, *6*(3), 259–268. Retrieved 2025-02-18, from `https://doi.org/10.1515/dx-2018-0069` doi: doi:10.1515/dx-2018-0069

Watanabe, S. (2023). *Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance.* Retrieved from `https://arxiv.org/abs/2304.11127`

Woolley, A., & Kostopoulou, O. (2013). Clinical intuition in family medicine: More than first impressions. *The Annals of Family Medicine*, *11*(1), 60–66. Retrieved from `https://www.annfammed.org/content/11/1/60` doi: 10.1370/afm.1433

Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, *5*(23), 495.

Xia, M., Cao, S., Zhou, R., Wang, J.-Y., Xu, T.-Y., Zhou, Z.-K., ... Jiang, H. (2021, September). Acoustic features as novel predictors of difficult laryngoscopy in orthognathic surgery: an observational study. *Ann. Transl. Med.*, *9*(18), 1466.