universität freiburg

Intelligent Embedded Systems Lab

Hahn Schickard

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

ChewFiit's

Chewing sound interpretation by Deep Learning

Ali Sarlak, Arpit Baranwal, Arindam Paul

Supervisors: Prof. Dr. Oliver Amft, Addythia Saphala

COSIMA´23

FAULHABER

infineon

BOSCH Technik fürs Leben

VDE

Contents

1 Introduction

Our research, which focuses on developing a machine-learning model to analyze masticatory patterns during food consumption, addresses a notable gap in the medical field. This gap is relevant to diagnosing potential anomalies and has broader implications, particularly in the context of obesity and related health issues.

Mastication, or chewing, plays a pivotal role in optimal digestive functioning and is closely linked to overall health. Inadequate chewing can lead to digestive system complications and related health issues, including obesity. Research has shown that proper chewing is essential for controlling food intake, as it allows the brain to receive timely signals of fullness from the digestive system. Thus, understanding and optimizing chewing patterns can be crucial in managing caloric intake and addressing obesity.

While previous research used Electromyography (EMG) sensors on participants to differentiate food types, the approach had clear limitations, as it used very sensitive and uncomfortable EMG sensors. Our current project aims to overcome these limitations by exclusively utilizing captured sound signals, eliminating the dependence on EMG data.

We will pursue food classification based on the signals derived from participants' chewing activities by leveraging deep learning and machine learning techniques, which have demonstrated promising results in various fields, including signal processing. This innovative approach not only addresses the identified research gap but also holds the potential to contribute to the broader effort to combat obesity and improve overall health outcomes.

2 Methods

In-Lab Investigation: We conducted an in-lab investigation to achieve our goal of creating an "Automated Dietary Monitoring" system. During this phase, we capture vibration signals while individuals consume foods with varying textures and hardness. The primary objective was to examine the distance differences during chewing and other activities to gather data for analysis.

Data Collection and Processing: Throughout the in-lab investigation, vibration data was collected from the integrated sensors. After the data collection phase, we subjected the raw vibration data to data processing to remove noise and irrelevant information, making the data suitable for utilization by deep learning models. Different datasets were prepared for various deep learning models, including tabular datasets, time series datasets, and spectrogram datasets, catering to the requirements of each model.

Machine Learning Model Development: We developed and tested multiple machine-learning models to identify patterns in chewing sequences and associate them with specific food types. The following models were considered for evaluation:

a. LSTM (Long Short Term Memory): A recurrent neural network architecture suitable for processing sequential data, such as time series data from the vibration sensors.

b. SVMs (State Vector Machines): A classic supervised learning algorithm for classification tasks. I suits various data types like text/images/tables.

c. CNN (Convolutional Neural Networks): A deep learning architecture known for its ability to extract features from images.

d. Transformer: A powerful attention-based architecture known for its ability to process sequential data efficiently.

Food Type Prediction: To achieve accurate food type prediction, we performed signal segmentation to isolate chewing sequences from other activities recorded by the vibration sensors. By analyzing the segmented signals, the deep learning models were trained to predict the type of food being consumed based on the characteristic chewing patterns exhibited in the data.

Evaluation and Results: After training the deep learning models on the segmented vibration data, we evaluated their performance and compared the results. The innovative method of detecting chewing sequences and eating events through vibration sensor signals showed promising results, indicating the potential for creating an efficient and automated dietary monitoring system.

3 Implementation

3.1 Data Analysis

Certainly, the quality and reliability of the data play a crucial role in determining the effectiveness of any model. Therefore, this section seeks to delve into the attributes of the previously recorded data and to validate the accuracy of the measurements and assertions made.

In order to provide the readers of this report with a comprehensive understanding of the data, it is imperative to elucidate the nature of the signals and the specific circumstances under which the data was collected.

The experimental cohort consisted of eleven participants who actively engaged in the study. Each participant wore the specialized glasses and consumed a set of five distinct food items, namely apple, beef jerky, baguette, cheddar, and chips. However, it is important to acknowledge that an uneven distribution of data was observed among the food types. Consequently, certain food items were not grasped by a few participants, resulting in a scarcity of data for those specific food categories in comparison to others.

Furthermore, separate signals corresponding to both the food types and individual participants were obtained and made available for analysis.

The data was acquired using a sampling rate of 24 kHz, resulting in the collection of 24,000 samples per second. However, it is important to note that the accuracy of the data acquisition was not consistent. As a result, the actual number of samples acquired per second ranged from 24,000 to 24,104, with slight variations observed in different cases. The following table provides an overview of the participants and the specific food items they consumed during the data collection process.

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
apple	beef jerky	beef jerky	chips	chips	apple	chips	apple	apple	baguette	apple
beef jerky	apple	chips	chips	chips	apple	chips	apple	chips	baguette	chips
apple		chips	chips	beef jerky	cheddar	chips		chips	baguette	beef jerky
		chips	chips	beef jerky	cheddar	chips		chips	apple	beef jerky
		beef jerky	chips	beef jerky		chips		chips	apple	apple
		beef jerky	chips	beef jerky		chips			apple	chips
						baguette			apple	chips
									chips	
									chips	
									chips	
									cheddar	
									cheddar	
									beef jerky	
									chips	
									chips	

Table 1: Overview of the participants and the specific food items they consumed

As evident from the aforementioned table, it is apparent that there is an uneven distribution of data among the participants. This uneven distribution poses a challenge for many machine learning and deep learning models, as they often require a balanced dataset for optimal performance. However, it is worth noting that various approaches exist to mitigate this issue.

One possible approach is data augmentation, which involves artificially increasing the size of the dataset by applying transformations such as phase shift, scaling, or adding noise to

the existing samples. This can help balance the dataset and provide additional variations for training the models.

Another approach is to employ techniques such as oversampling or undersampling. Oversampling involves replicating the minority class samples to match the number of majority class samples, while undersampling involves randomly removing samples from the majority class to match the number of minority class samples. Both techniques aim to achieve a more balanced dataset.

Additionally, ensemble methods, which combine predictions from multiple models, can be utilized to account for the class imbalance. By training several models on different subsets of the data or using different algorithms, ensemble methods can improve the overall performance and robustness of the models.

Overall, while dealing with an uneven distribution of data among participants presents a challenge, employing techniques such as data augmentation, oversampling, undersampling, or ensemble methods can help mitigate this problem and enhance the performance of machine learning and deep learning models.

3.1.1 Collected Signals

Prior to delving into other aspects, it is essential to focus on the characteristics of the signals themselves and the context in which the previous team collected the data. Each participant was seated in a comfortable chair and engaged in a series of activities, including responding to questions to capture muscle activity during speech, followed by the consumption of various food items. Subsequently, the recorded data was partitioned into multiple CSV files, with each file encompassing approximately 48 MB of data, corresponding to approximately 1,000,000 records. The number of CSV files ranged from 20 to 25 for each participant, depending on the number of foods they consumed. Within this extensive dataset, the region of interest lies in the portion pertaining to food chewing.

Additionally, another dataset was compiled specifically for food chewing, consisting of signals isolated for each food item and participant. This dataset serves as the training, validation, and testing sets for the subsequent models. Furthermore, a supplementary dataset comprising text files was included, containing annotations denoting the onset of chewing for each chewing sequence within the signal. Each line in the text files represents a time value in seconds, signifying the initiation of chewing as mentioned.

3.1.2 Verification

The initial task at hand involves confirming the presence of the chewing sequence for a specific participant within the complete signal. To accomplish this, the individual CSV files were combined to construct a cohesive and comprehensive dataset stored in memory. Subsequently, given that the recorded signals were susceptible to noise, a preprocessing technique was employed to enhance the signal quality and eliminate outliers. Among the various filters tested, the "hampel" filter was selected for its ability to effectively remove outliers and achieve signal smoothing. Notably, it is important to mention that certain filters, such as the median filter, proved unsuitable for this particular task as they exhibited a tendency to distort the input signal.

The Hampel filter is a statistical method commonly used for outlier detection and removal in signal processing. It operates by identifying data points that deviate significantly from the surrounding values and replaces them with more representative estimates. This filter considers the median absolute deviation (MAD) as a robust measure of dispersion to detect outliers. The MAD is calculated by taking the median of the absolute differences between each data point and the median of its neighboring values. The Hampel filter compares each data point to a threshold based on a specified multiple of the MAD. If the difference exceeds this threshold, the data point is considered an outlier and replaced with a more suitable estimate, such as the median of the surrounding values.

On the other hand, the median filter is a widely used technique for signal smoothing and noise reduction. It operates by replacing each data point with the median value within a defined neighborhood window. The median is less sensitive to extreme values, making it effective in mitigating the impact of outliers and preserving the overall shape of the signal. However, it is worth noting that the median filter can also introduce certain **distortions** in the signal, particularly when applied to signals with abrupt changes or sharp features. These distortions may be undesirable in certain applications, such as the analysis of chewing sequences, where preserving the fine details of the signal is crucial.

For a comprehensive understanding of the input signal and the impact of noise, Figure 1 displays the plotted signal in its raw form. This visualization provides insights into the inherent characteristics and fluctuations within the signal. However, to mitigate the effects of noise and enhance the signal quality, preprocessing techniques were applied. Figure 2 presents the signal after undergoing the designated preprocessing steps, showcasing the result of noise reduction and smoothing. By comparing these two figures, one can gain a better appreciation of the improvements achieved through the preprocessing stage.

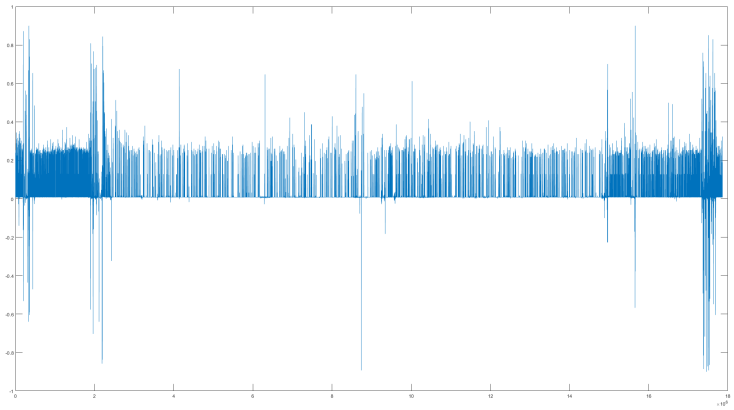

Figure 1: Raw signal

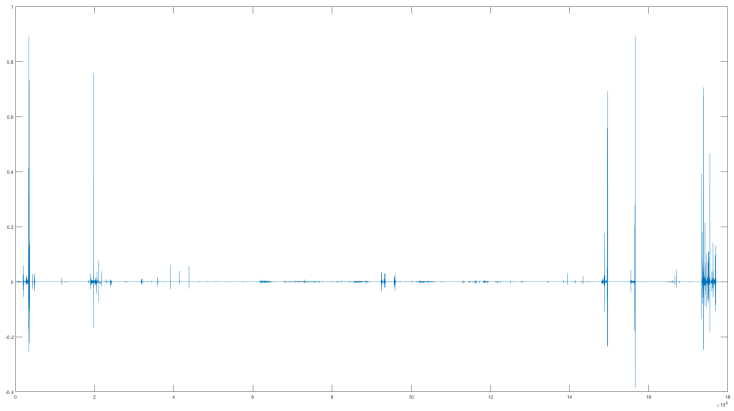

Figure 2: Filtered signal

Figure 3 showcases a zoomed-in portion of both the raw signal and the filtered signal. This magnified view provides a closer examination of the details within the signal, allowing for a more precise analysis of the impact of the filtering process. By visually comparing the raw and filtered versions in this smaller segment, one can observe the specific changes and improvements brought about by the applied filtering technique.

3.1.3 Finding masticate signals in entire signal

It is of utmost importance to validate the previous team's efforts in accurately determining the chewing sequences within the entire signal. This verification process serves not only to

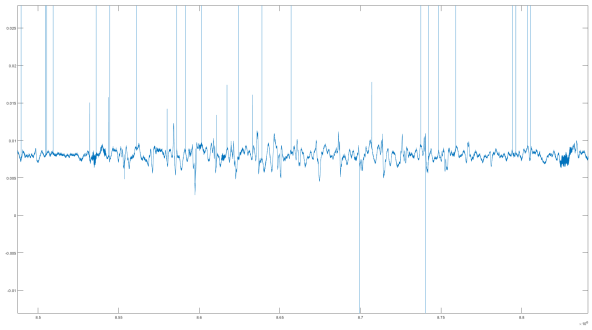

Figure 3: Filtered and raw signal comparison

ensure the correctness and accuracy of the data but also plays a critical role in establishing an approach or procedure for identifying such sequences in future work. By confirming the reliability of the chewing sequence determination, researchers can establish a robust foundation for subsequent analyses and investigations related to this specific aspect.

The subsequent section provides a detailed explanation of the chosen approach for identifying the mentioned sequences. Cross correlation, a widely accepted method for detecting a sequence in a signal, was initially considered. However, extensive research and experimentation revealed that cross correlation struggles to handle large sequences effectively, often resulting in confusion regarding the optimal match within the larger signal. Cross correlation may face challenges when dealing with large sequences due to several reasons. One of the main limitations is the computational complexity associated with cross correlation calculations for large data sets. As the sequence length increases, the number of comparisons required grows exponentially, resulting in a significant increase in computational resources and time.

Moreover, as the sequence size increases, the likelihood of finding multiple occurrences or similar patterns within the larger signal also increases. This can lead to ambiguity in determining the exact match or alignment between the two signals, making it difficult to identify the precise location of the desired sequence.

Furthermore, cross correlation assumes stationarity and linear relationships between the signals, which may not hold true in the case of large and complex data sets. Large sequences often exhibit nonlinear behavior and non-stationary characteristics, which can affect the accuracy of cross correlation-based matching.

To address these challenges, alternative approaches such as the multi-cross correlation method mentioned earlier can be employed. These modified techniques take into account

the specific limitations of cross correlation for large data sets and offer more efficient and effective solutions for sequence identification.

To address this limitation, a modified approach called "multi-cross correlation" was devised.

In the multi-cross correlation approach, a sliding window was employed to divide the larger signal into segments with 50% overlap, ensuring comprehensive coverage of the entire signal. Each window was twice the length of the small signal portion (the sequence of samples) under consideration, and regular cross correlation was applied within each window. Consequently, a match, representing the most probable match sequence, was identified within each window.

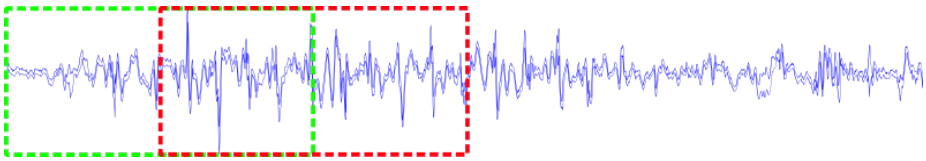

Figure 4: Sliding window with 50 per cent overlap

To determine the correct match, a norm 2 distance calculation and thresholding technique were utilized. The norm 2 distance measured the dissimilarity between non-matching signals, with a larger value indicating a greater deviation from a perfect match. Note that in the proposed approach, a Fast Fourier Transform (FFT) method has been utilized to decompose the matched portion of signals within each window. By applying the FFT, the frequency components of the matched portion can be analyzed and represented in the frequency domain. Subsequently, the norm 2 distance calculation is performed on the decomposed signals.

By computing the norm 2 of the decomposed signals, the dissimilarity between the matched portion and the larger signal can be quantified. The norm 2 distance serves as a measure of the overall difference between the frequency components of the matched portion and the signal itself. This enables a more comprehensive evaluation of the match and enhances the accuracy of sequence identification.

The inclusion of the FFT and subsequent norm 2 distance calculation in the approach provides a more refined and detailed analysis of the frequency characteristics of the matched portion within the larger signal. This additional step enhances the effectiveness and reliability of the sequence identification process. By applying a threshold, the approach accounted for inconsistencies between the two signals under investigation, allowing for flexibility in accommodating perturbations and imperfect matches.

The inclusion of a threshold was essential to accommodate variations in the signals and ensure robustness in identifying the correct match. This adaptive approach enabled accurate sequence detection despite potential deviations and inconsistencies between the signals.

Figure 6 clearly demonstrates the effectiveness of the proposed algorithm in accurately identifying the desired match within the entire signal. The high precision achieved by the algorithm is evident, as indicated by the close alignment between the identified match and the expected sequence. This successful outcome validates the robustness and reliability of the algorithm in accurately locating the desired sequence within the larger signal. The results depicted in Figure 6 provide strong evidence of the algorithm's ability to achieve precise and accurate match identification, further affirming its suitability for the intended purpose.

3.1.4 Start of chewing annotation

Although not directly impacting the modeling section, it is imperative to validate the information regarding the start of chewing for each specific type of food within the chewing sequence dataset. While this validation process may not have a direct bearing on the modeling aspect, it plays a crucial role in ensuring the accuracy and reliability of the data. By verifying the provided information, researchers can ascertain the correctness of the annotated start times for chewing events, which in turn enhances the overall quality and integrity of the dataset.

In the pursuit of identifying the start of chewing within a sequence of chewing food, a specific approach developed by a group of researchers was employed. This approach, implemented by the researchers, successfully enabled the identification of the precise initiation of chewing events.

The detection of the beginning of each chew was facilitated by a relatively simple algorithm, as outlined in the referenced paper. This algorithm involved evaluating the short-time signal energy within a 20 ms window and comparing it to a predetermined energy threshold. If the short-time signal energy exceeded the threshold, the resulting signal was set to 1; otherwise, it was set to 0. Subsequently, the squared signal was subjected to low-pass filtering using a 4th order Butterworth filter. The specific choice of a filter with a 3dB cut-off frequency of 4 to 5 Hz effectively responded to the pause in phase 4 while filtering out the shorter pause in phase 2. The hill climbing algorithm was then employed to accurately detect the beginning of each chew, as depicted in Figure 7.

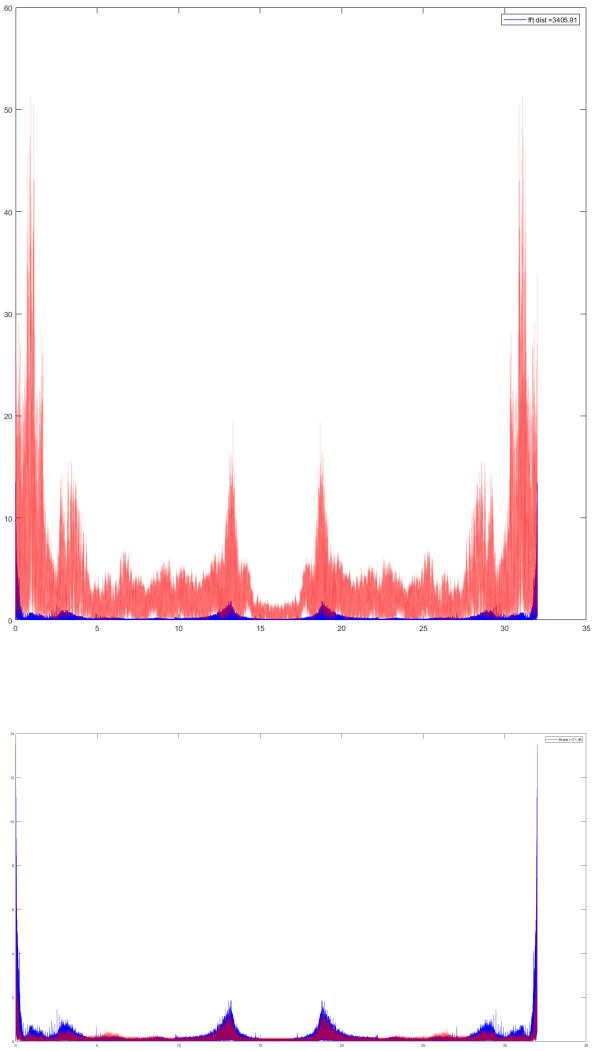

Figure 5: FFT matching comparison with norm2 distance

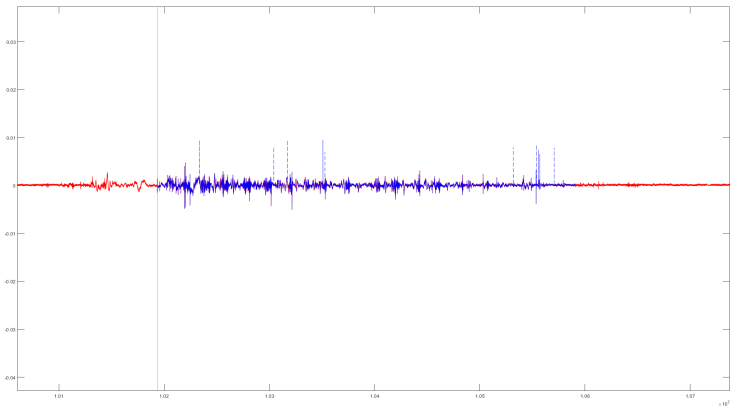

Figure 6: Start of match and matched signal in entire signal

According to the findings presented in the paper, this algorithm demonstrated a high success rate, accurately detecting the start point of approximately 90% of all chews. Importantly, the algorithm exhibited minimal false insertions, further affirming its reliability and effectiveness in accurately identifying the onset of chewing events. [1]

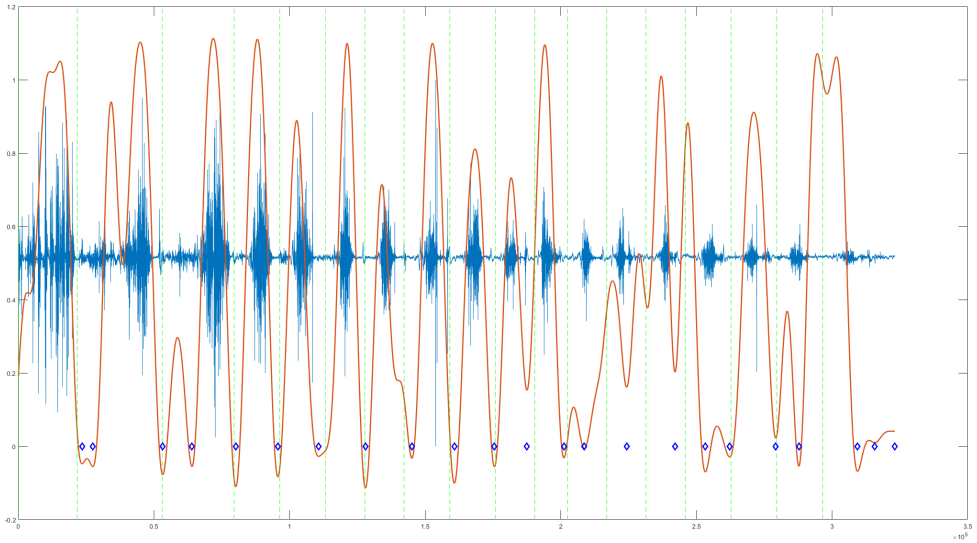

Figure 7: Start of chewing annotation

The observed differences between the annotated start of chewing and the start of chewing identified by the algorithm can be attributed to various factors, including inconsistencies

in the definition of the start of chewing and the manner in which the data was recorded.

The definition of the start of chewing may vary among researchers and experts, leading to variations in the identification of this specific event. Different criteria or thresholds may be applied to determine the precise moment when chewing begins, resulting in discrepancies between manual annotations and algorithmic detection.

Furthermore, the process of recording the data itself can introduce certain limitations and challenges. Factors such as the positioning of sensors , variations in sensor sensitivity, and noise interference can impact the accuracy of identifying the exact start of chewing. These factors can contribute to differences between the algorithm's marked start of chewing and the annotations provided.

It is important to recognize and acknowledge these inconsistencies and limitations when comparing the algorithm's results with manual annotations. Such discrepancies can provide insights into the complexities of accurately identifying the start of chewing and highlight areas for improvement in future research and algorithm development.

3.1.5 Data Preparation for modeling

Data preparation plays a pivotal role in the modeling process and significantly influences the ultimate outcomes obtained. In our study, we have identified several models, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN) (some variants including ResNet, EfficientNet, DenseNet), Recurrent Neural Networks (RNN), Transformers, and Vision Transformers (ViT), which we aim to implement. Each of these models has specific requirements and is suitable for different types of datasets. Consequently, we have formulated distinct datasets tailored to meet the specific needs of each model, taking into account their respective strengths. To cater to models that excel in sequence-based data analysis, such as RNNs, we have transformed our input signals into image representations known as spectrograms. By representing the signals as sequences in the time domain, we leverage the inherent sequential nature of the data, enabling effective utilization by these models. Furthermore, we have explored the extraction of relevant features from the signals, enabling us to create tabular datasets. This approach is particularly useful for models that operate on tabular data, facilitating their application and harnessing the potential insights derived from the transformed signal data.

In the subsequent sections, we will elaborate on the preprocessing techniques employed and discuss the specific conditions and considerations involved in preparing the datasets

tailored to each model's requirements.

3.2 Model Planning

3.2.1 Convolutional Neural Network (CNN)

Artificial Neural Networks (ANNs) are computational processing systems of which are heavily inspired by way biological nervous systems (such as the human brain) operate. ANNs are mainly comprised of a high number of interconnected computational nodes (referred to as neurons), of which work entwine in a distributed fashion to collectively learn from the input in order to optimise its final output [2]. Convolutional networks (LeCun, 1989), also known as convolutional neural networks, or CNNs, are a specialized kind of neural network for processing data that has a known grid-like topology [3]. Convolutional Neural Network has had ground breaking results over the past decade in a variety of fields related to pattern recognition; from image processing to voice recognition. The most beneficial aspect of CNNs is reducing the number of parameters in ANN [4].

In this section we discuss CNN architectures developed from the first successful to the current state-of-the-art architectures.

LeNet-5

LeCun et al. [5] constructed a CNN for handwritten zip code recognition and first used the term "convolution," which is the original version of LeNet [6]. The architecture of LeNet-5 is shown in Fig. 1 [7], including 3 convolutional layers, 2 sub-sampling layers and 2 fully connected layers.

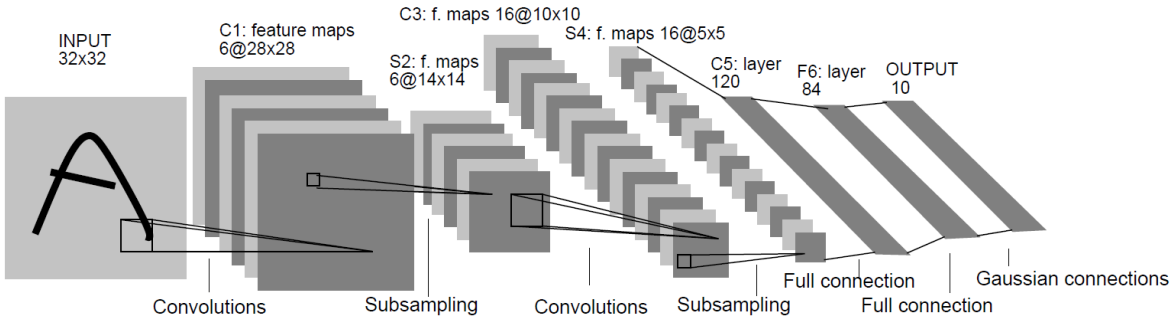

Figure 8: LeNet-5 architecture

AlexNet

As shown in Fig. 2, AlexNet has a deep convolutional neural network which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully connected layers with a final 1000-way softmax. Furthermore, this architecture made it possible to use two GPUs efficiently, where One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. In order to fight overfitting, they used dropout as a regularization technique [8].

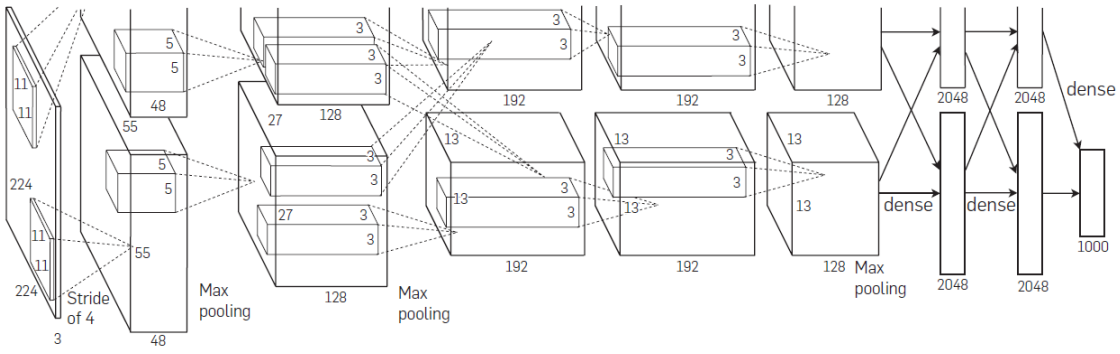

Figure 9: AlexNet architecture

VGGNet

Simonyan and Zisserman used deep networks with very small 3×3 convolution filters to construct VGGNet architecture. Their ConvNet configurations with 16 and 19 weight layers have 138 and 144 parameters respectively [9]. These many parameters make inference expensive.

ResNet

According to the difficulty of training deeper neural networks, ResNet was developed by employing residual learning framework to overcome this challenge. Fig. 3 shows shortcut connections that simply perform identity mapping where their outputs are added to the output of the stacked layers [10]. Fig. 4 shows architecture of VGG-19 model at the left, a plain network with 34 parameter layers at the middle and a residual network with 34 parameter layers at the right [10]. As illustrated in Fig. 4 [10], architecture of ResNet-34 consists of very deep network with skip connections and only a single fully connected layer at the end.

DenseNet

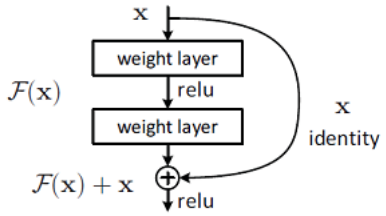

Figure 10: Residual learning

As CNNs become increasingly deep, a new research problem emerges: as information about the input or gradient passes through many layers, it can vanish and "wash out" by the time it reaches the end (or beginning) of the network. Dense Convolutional Network (DenseNet) leverages dense connectivity pattern that ensures maximum information flow between layers in the network by connecting all layers (with matching feature-map sizes) directly with each other. To preserve the feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. Fig. 5 shows a dense block where each layer takes all preceding feature-maps as input [11].

MobileNets

MobileNets are a class of efficient models for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. As shown in Fig. 6, they replaced standard convolutional filters by depthwise convolution and pointwise convolution layers to build a depthwise separable filter [12]. According to the lower number of parameters in MobileNets, they are renowned for their efficiency.

EfficientNet

Convolutional Neural Networks are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. Tan and Le proposed a new scaling method that uniformly scales all dimensions of depth, width and resolution using a simple highly effective compound coefficient. They demonstrated the effectiveness of this method on scaling up MobileNets and ResNet. Furthermore, they used neural architecture search to design a new baseline network and scaled it up to obtain EfficientNets, which achieved much better accuracy and efficiency than previous ConvNets at that time. Fig. 7 [13] illustrates EfficientNets superior performance in comparison to other ConvNets at that time, considering their accuracy and number of parameters.

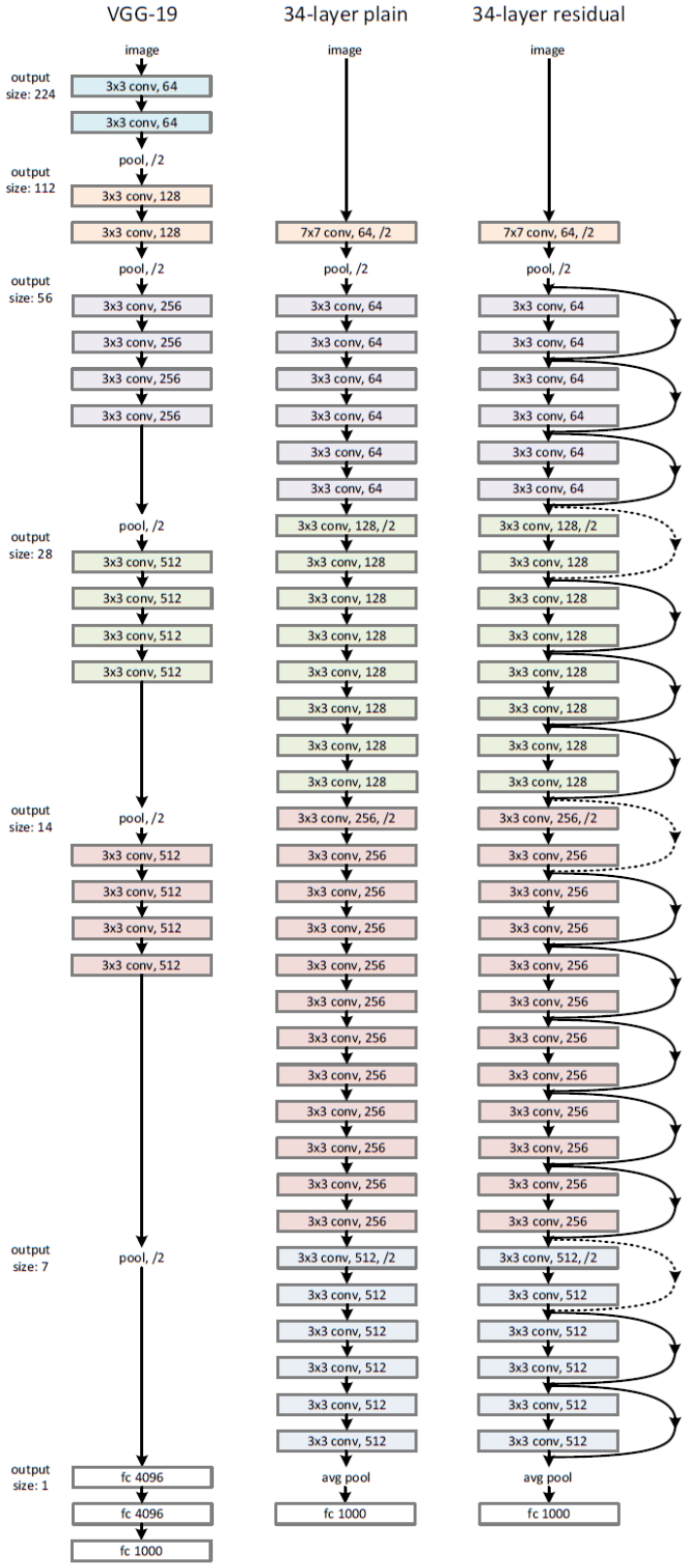

Figure 11: ResNet architecture

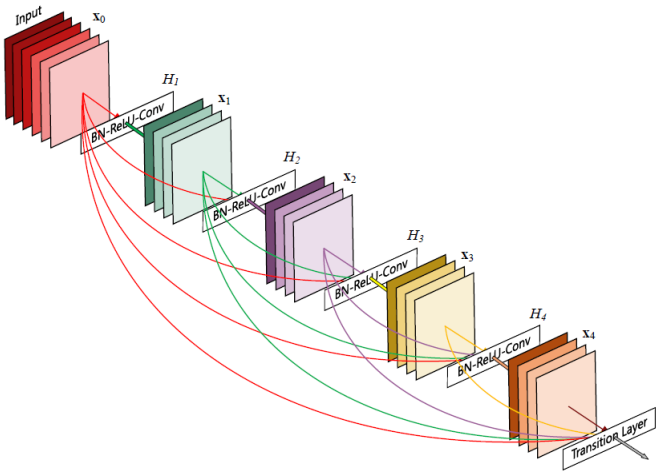

Figure 12: Residual learning

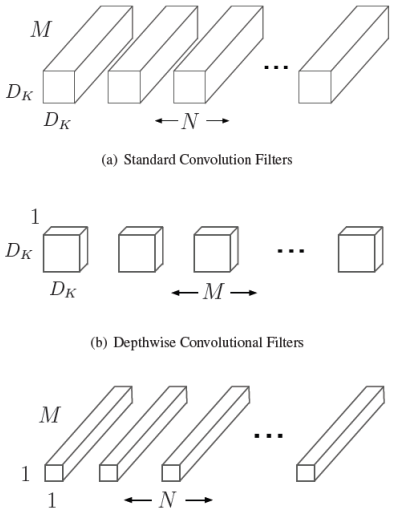

Figure 13: Residual learning

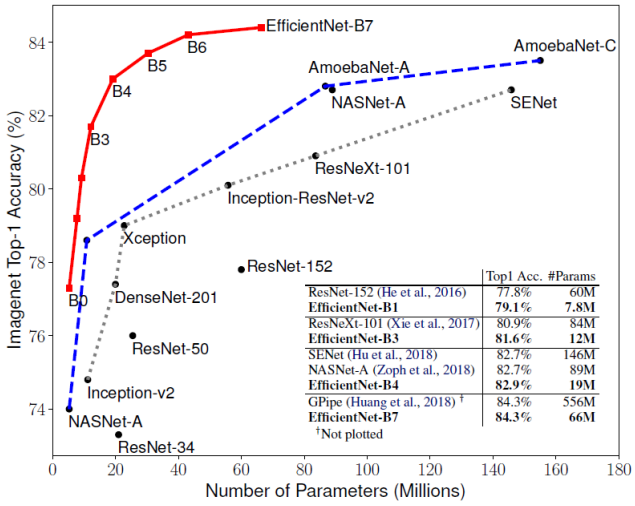

Figure 14: Residual learning

Fig.8 shows the difference between conventional scaling methods which only increase one dimension of the network width, depth or resolution and compound scaling method implemented in EfficientNets. This effective compound scaling method balances scaling of dimensions considering target resource constraints, while maintaining model efficiency. As a result, a mobile-size EfficientNet model can be scaled up very effectively [13].

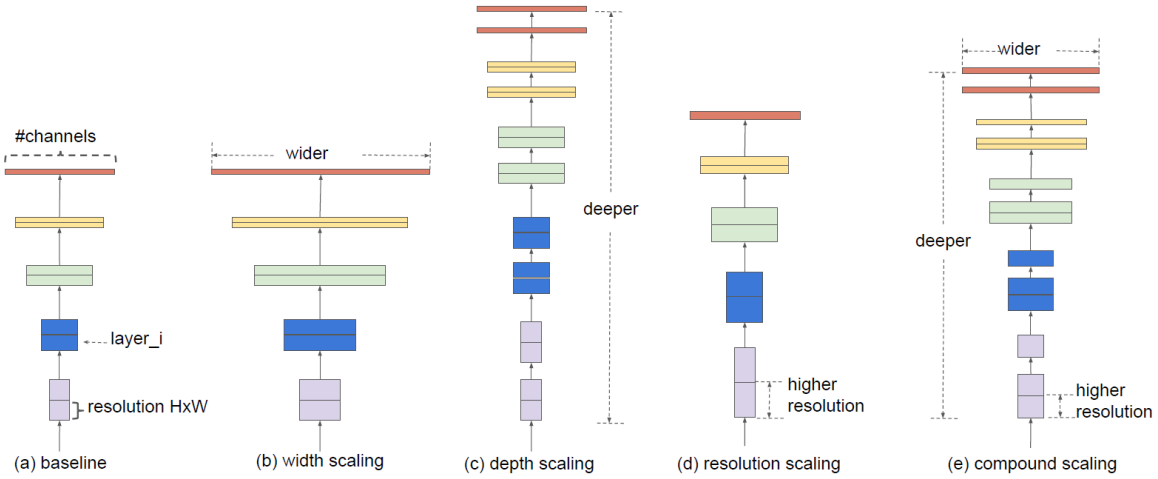

Figure 15: Residual learning

3.2.2 Support Vector Machines (SVMs)

Support Vector Machines (SVMs in short) are machine learning algorithms that are used for classification and regression purposes. SVMs are one of the powerful machine learning algorithms for classification, regression and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier.

SVMs can be used for linear classification purposes. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enables us to implicitly map the inputs into high dimensional feature spaces.

We should be familiar with some SVM terminology.

Hyperplane

A hyperplane is a decision boundary which separates between a given set of data points having different class labels. The SVM classifier separates data points using a hyperplane with the maximum amount of margin. This hyperplane is known as the maximum margin hyperplane and the linear classifier it defines is known as the maximum margin classifier.

Support Vectors

Support vectors are the sample data points, which are closest to the hyperplane. These data points will define the separating line or hyperplane better by calculating margins.

Margin

A margin is a separation gap between the two lines on the closest data points. It is calculated as the perpendicular distance from the line to support vectors or closest data points. In SVMs, we try to maximize this separation gap so that we get maximum margin.

The following diagram illustrates these concepts visually.

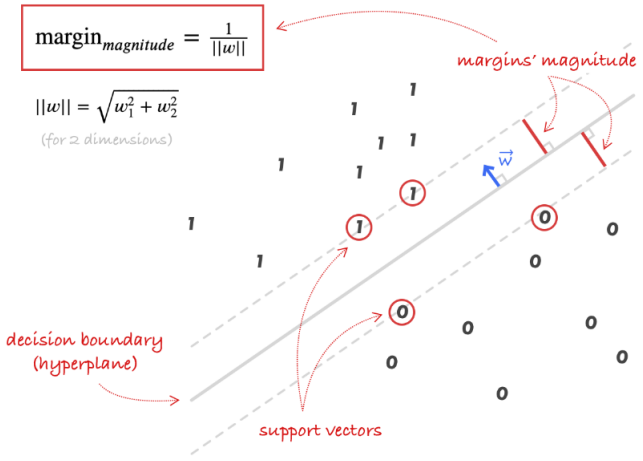

Figure 16: SVM classification illustrated

3.2.3 Long Short Term Memory Networks (LSTM)

A long short-term memory network, known as LSTM, belongs to the category of recurrent neural networks (RNNs). LSTMs find extensive application in learning, processing, and categorizing sequential data due to their capability to grasp long-term dependencies between different time steps of the data. These networks are widely used in various domains, including sentiment analysis, language modeling, speech recognition, and video analysis.

In the mid-90s, German researchers Sepp Hochreiter and Juergen Schmidhuber introduced a variation of recurrent neural networks known as Long Short-Term Memory units, or LSTMs, to address the vanishing gradient problem.

LSTMs play a crucial role in preserving error during backpropagation through time and layers. This capability enables recurrent nets to learn over extended time steps, even beyond 1000, allowing them to establish connections between distant causes and effects. This is a significant challenge in the fields of machine learning and AI, as these algorithms often encounter environments with sparse and delayed reward signals, similar to real-life scenarios where consequences may be remote and difficult to associate with actions (a concept that religious thinkers have also pondered with ideas like karma or divine reward).

LSTMs incorporate a gated cell that holds information independently of the normal flow in the recurrent network. Similar to a computer's memory, data can be stored, written, or read from the cell. The cell's gates, operating through element-wise multiplication by

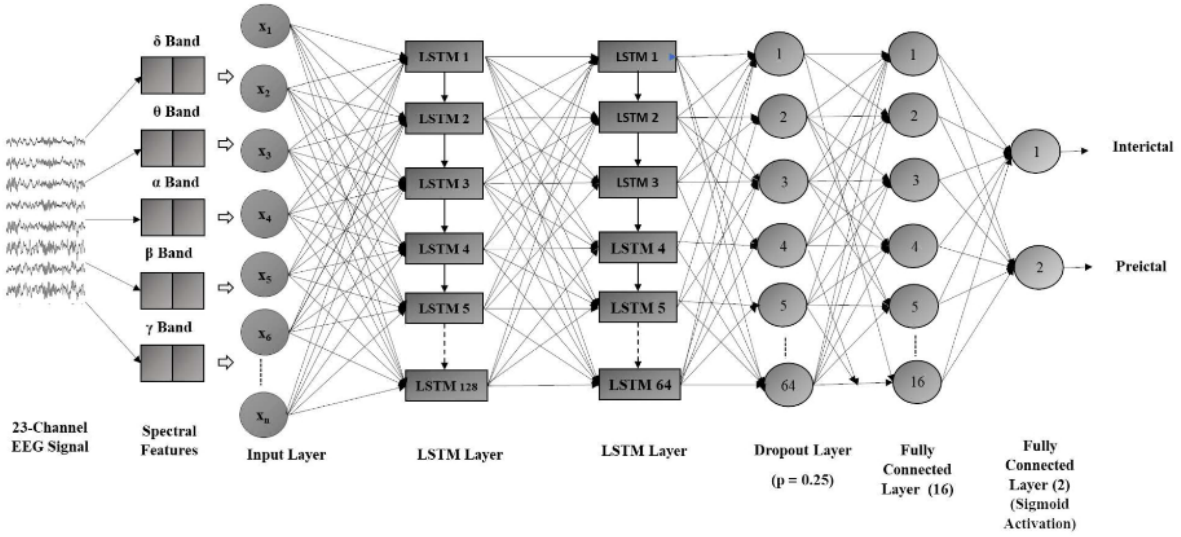

Figure 17: Recurrent neural network with memory cells for processing sequential data

sigmoids (values ranging from 0 to 1), make decisions about what to retain, read, write, or erase. Unlike digital storage in computers, the analog gates are differentiable, making them suitable for backpropagation.

These gates process incoming signals and, like nodes in a neural network, they filter and control the flow of information based on its relevance and strength, regulated by their own sets of weights. These weights, similar to those modulating input and hidden states, are adjusted through the iterative learning process of making predictions, backpropagating errors, and updating weights using gradient descent. [14]

3.2.4 The Vision Transformer (ViT)

The Vision Transformer (ViT) is a type of transformer model designed specifically for handling vision-related tasks, such as image processing and analysis. It utilizes the transformer architecture to effectively process visual data.

While transformers, like LLM's and GPT-4, initially gained prominence in natural language processing (NLP) tasks, convolutional neural networks (CNNs) have traditionally been the go-to choice for image processing systems. CNN models such as Xception, ResNet,

EfficientNet, DenseNet, and Inception are widely recognized in the field of image processing.

Transformers excel at capturing the connections between input tokens, such as words in text strings, through attention mechanisms. However, when it comes to images, the analysis is typically performed at the pixel level. Calculating relationships between every pair of pixels in an image is computationally intensive and memory-consuming. To address this, Vision Transformers (ViTs) divide the image into smaller sections, usually around 16x16 pixels. Within each section, relationships are computed, and positional embeddings are added. These sections, along with their embeddings, are then arranged in a sequential manner and fed into the transformer model for further processing. This approach significantly reduces the computational cost while still allowing the model to analyze image content effectively.

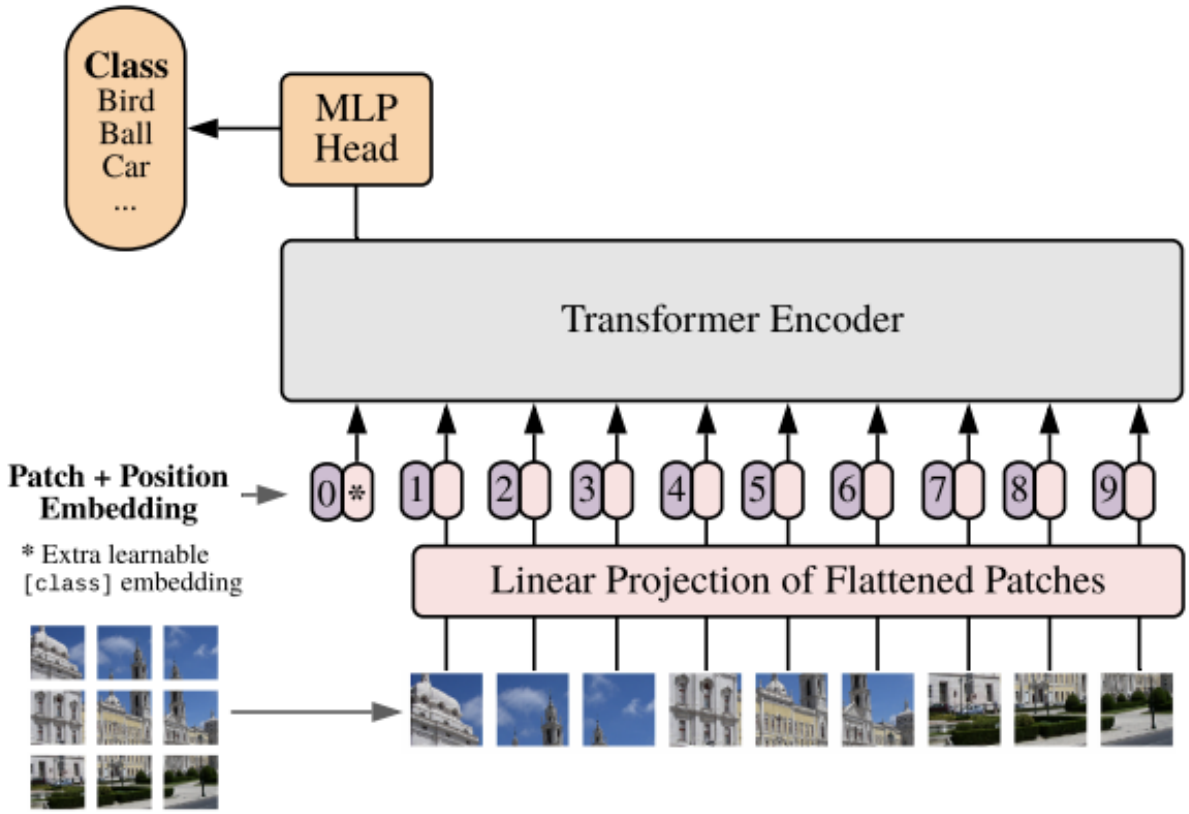

Figure 18: ViT Architecture

Similar to LLM , the class token holds a crucial role in classification tasks. It serves as a special token that is exclusively used as the input for the final MLP Head. The class token

captures the influence and information from all the other tokens in the sequence, allowing it to summarize the context and contribute significantly to the classification process.

3.3 Technical Approach for Model Training

3.3.1 Convolutional Neural Network (CNN)

Methodology

To conduct our experiments, we transformed our vibration signals from time-series data into spectrograms as images. Our aim is to use a CNN architecture for chewing detection as an image classification task, therefore we carried out a survey of the most well-known CNN architectures. Next, we determined best architectures for our task depending on their model size and accuracy to examine their performance on our spectrograms dataset.

Data augmentation

Due to the low amount of data available for our model training, it was necessary for us to use data augmentation. We used frequency masking and time masking methods from SpecAugment, which is a data augmentation method developed by Park et al. for automatic speech recognition [15]. For each spectrogram in our train dataset we applied randomly one or both of the frequency masking and time masking with a random value for the maximum possible length of the mask from a range between 5 and 80.

Evaluated architectures

In this section we discuss and investigate the architectures we employed for our chewing detection task.

ResNet-152

ResNet-152 architecture has residual nets with a depth of up to 152 layers, being 8 times deeper than VGG-19 nets but still having lower complexity. By achieving 3.57% error on the ImageNet test set, it won the 1st place on the ILSVRC 2015 classification task [10]. However ResNet-152 has more number of parameters in comparison to ResNet-34, ResNet-50 and ResNet-101, it had the highest ImageNet top-1 accuracy. Thus, we decided to investigate its performance in our task.

DenseNet-201

DenseNet has some key benefits such as mitigating vanishing gradient and also its efficiency in number of parameters. As DenseNet-201 has about 3 times fewer parameter in comparison to ResNet-152, we made the decision to examine this architecture for our task.

MobileNetV3

Howard et al. constructed MobileNetsV3, which is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm followed by improvement through novel architecture advances. They created two new MobileNet models including MobileNetV3-Large and MobileNetV3-Small, which are targeted for high and low resource use cases [16]. According to the state-of-the-art results they achieved for mobile classification [16] and considering the nature of our task which is chewing detection in smart glasses, we included both MobileNetV3-Large and MobileNetV3-Small architectures in our experiments.

EfficientNetV2-S

EfficientNetV2 is a new family of convolutional networks that have faster training speed and better parameter efficiency than previous models.[r14] Tan and Le concluded that in comparison to EfficientNet and more recent works, EfficientNetV2 trains up to 11x faster while being up to 6.8x smaller [17]. EfficientNetV2 models including EfficientNetV2-S, EfficientNetV2-M and EfficientNetV2-L have 24M, 55M and 121M parameters respectively [17]. As we prefer an architecture with smaller number of parameters, due to our goal which is a deployment on a mobile device, we decided to investigate the performance of EfficientNetV2-S in our experiment.

3.3.2 Support Vector Machines (SVMs)

In SVMs, our main objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum margin hyperplane in the following 2 step –

1) Find the optimal hyperplanes that effectively separate the classes by maximizing the margin between them. Numerous hyperplanes can be considered for classifying the data, but we aim to identify the hyperplane that offers the greatest separation.

2)The selection of the hyperplane is based on maximizing the distance between the hyperplane and the support vectors on either side. When such a hyperplane exists, it is referred

to as the maximum margin hyperplane, and the corresponding linear classifier is known as a maximum margin classifier.

In practice, the SVM algorithm employs a kernel and utilizes a technique called the kernel trick. Essentially, a kernel is a function that maps the data to a higher-dimensional space where it becomes separable. By transforming the input data space into a higher dimension, the kernel enables the handling of non-linearly separable problems as if they were linearly separable. This augmentation of dimensions allows for the construction of a more precise classifier. As a result, the kernel trick is particularly valuable in scenarios involving non-linear separation problems.

In the context of SVMs, there are 4 popular kernels – *Linear kernel*, *Polynomial kernel*, *Radial Basis Function (RBF) kernel* (also called Gaussian kernel) and *Sigmoid kernel*

We will implement svm using all the kernels one by one:

Linear kernel

The linear kernel is represented by a linear function in the following form:

$$Linear\ kernel : K(x_i, x_j) = x_i T.x_j \tag{1}$$

The linear kernel is employed when the data can be separated using a single line, indicating linear separability. It is one of the most commonly used kernels and is particularly useful when dealing with data-sets that have a large number of features. Training with a linear kernel is generally faster compared to other kernels because it only requires optimizing the C regularization parameter. In contrast, when training with alternative kernels, the γ parameter also needs to be optimized. Consequently, performing a grid search to find the optimal parameters may take more time when using non-linear kernels.

Polynomial Kernel

The polynomial kernel captures the similarity between vectors (training samples) in a feature space by considering polynomials of the original variables. It not only examines the given features of input samples to determine their similarity but also takes into account combinations of the input samples. For degree-d polynomials, the polynomial kernel is defined as follows:

$$Polynomial\ kernel : K(x_i, x_j) = (\gamma x_i T x_j + r)d,\ \ where\ \gamma > 0 \tag{2}$$

Polynomial kernel: K(xi, xj) = (xiT xj + r) d, where $\gamma > 0$ In the above polynomial kernel equation, the parameter γ (gamma) controls the influence of the inner product between x_i and x_j. It scales the importance of different polynomial terms in the kernel function. When

γ is small, the influence of the inner product is reduced, resulting in a smoother decision boundary. On the other hand, when γ is large, the inner product has a stronger impact, leading to a more complex and potentially over-fitting decision boundary. The appropriate choice of γ depends on the specific data-set and problem at hand. It is typically determined through cross-validation or grid search techniques to find the optimal value that yields the best performance and generalization on the training and test data.

Radial basis kernel

Radial basis function kernel is a general purpose kernel. It is used when we have no prior knowledge about the data. The RBF kernel on two samples x and y is defined by the following equation –

$$RBF\ kernel : K(x_i, x_j) = exp(-\gamma(|x_i - x_j|)^2) \tag{3}$$

In this equation, γ (gamma) is a parameter that controls the smoothness and flexibility of the decision boundary. It determines the influence of each training example on the classification of new data points. A higher γ value results in a more localized decision boundary, fitting the training data more closely. Conversely, a lower γ value produces a smoother decision boundary with a wider influence range.

The optimal value of γ depends on the specific data-set and should be determined through techniques like cross-validation or grid search, considering factors such as over-fitting, under-fitting, and the complexity of the data.

Sigmoid kernel

The sigmoid kernel function is defined as follows:

$$Sigmoid\ kernel : K(x_i, x_j) = tanh(\gamma x_i T x_j + r) \tag{4}$$

In this equation, γ (gamma) and r are parameters that control the shape and behavior of the sigmoid function. The parameter γ determines the curvature of the function, while r shifts the function horizontally.

The sigmoid kernel can be useful for handling non-linearly separable data, but it is generally less popular compared to other kernels like the linear, polynomial, or radial basis kernels. It has some limitations, such as being sensitive to the choice of parameters and potentially leading to over-fitting.

The choice of optimal parameters γ and r for the sigmoid kernel is crucial, and it is typically determined through techniques like cross-validation or grid search to find the values that result in the best performance on the training and test data.

3.3.3 The Vision Transformer (ViT)

The most common architecture for image classification primarily relies on the Transformer Encoder to process the input tokens effectively. However, it's worth noting that there are other applications where the decoder part of the traditional Transformer architecture is also utilized. In our scenario, we only use the Transformer Encoder part to handle the input data and generate meaningful output. This expanded architecture allows for more complex tasks and enables the model to capture dependencies and generate contextual representations in a comprehensive manner.

The summary is a sequence of layers, including Dense layers, LayerNormalization layers, MultiHeadAttention layers, and others, with various activation functions and dropout layers for regularization. The final layer is a Dense layer with 5 output units, indicating a multi-class classification task with 5 classes.

Total params: 399,151,877

Trainable params: 399,151,877

Non-trainable params: 0

4 Results

4.1 Convolutional Neural Network (CNN)

To address computational resource limitations, the project was divided into three rounds, each involving different numbers of training epochs for the selected architectures. In the first round, five architectures were evaluated over 40 epochs. For ResNet-152, a learning rate of 0.00001 was chosen to achieve a smoother validation loss plot, resulting in a test accuracy of 0.7493. DenseNet-201 exhibited similar behavior, and a learning rate of 0.00001 was employed to achieve a smoother plot, yielding a test accuracy of 0.7889. MobileNetV3-Large initially had fluctuations with a learning rate of 0.0001, but reducing it to 0.00001 for 40 epochs improved generalization, albeit with a test accuracy of 0.7863. MobileNetV3-Small started with a learning rate of 0.0001, but to stabilize performance, the learning rate was lowered to 0.00001 for 40 epochs, resulting in a test accuracy of 0.7573. EfficientNetV2-S displayed fluctuating validation loss and accuracy with a learning rate of 0.0001, but using 0.00001 during round 1 led to a test accuracy of 0.8391. In subsequent rounds, EfficientNetV2-S and DenseNet-201 were prioritized, with training extended to 80 epochs.

However, DenseNet-201's test accuracy dropped to 0.7784 despite using a learning rate of 0.00001 and batch size of 64. In contrast, EfficientNetV2-S's test accuracy increased to 0.8443 after 80 epochs. In the final round, EfficientNetV2-S was further explored with a budget of 200 epochs, using a learning rate of 0.00001 and batch size of 64, achieving a test accuracy of 0.8654.

4.2 Support Vector Machines (SVMs)

Kernel	C (Hyperparameter)	Accuracy
RBF	100	0.62184
RBF	1000	0.6436
Linear	1	0.5901
Linear	100	0.6178
Linear	1000	0.6238
Polynomial	1	0.5683
Polynomial	100	0.5921
Sigmoid	1	0.4911
Sigmoid	100	0.4436

Table 2: Evaluation Table

Here, "C" stands for the regularization parameter or the cost parameter. In a Support Vector Machine (SVM) algorithm, the C parameter is a hyperparameter that determines the trade-off between maximizing the margin (distance between the decision boundary and the support vectors) and minimizing the classification error on the training data. It controls the soft margin, allowing some misclassification of training examples to find a better decision boundary that generalizes well to unseen data. A smaller value of C will create a wider margin but may allow more misclassifications on the training data. On the other hand, a larger value of C will lead to a narrower margin and penalize misclassifications more heavily. Choosing an appropriate value for C is critical in SVM as it influences the model's generalization performance.

4.3 Long Short Term Memory Networks (LSTM)

The model achieved an accuracy of 0.2457 during evaluation, and early stopping was employed to prevent overfitting and ensure model generalization.

4.4 The Vision Transformer (ViT)

The model achieved an accuracy of 0.4418 during evaluation, and early stopping was employed to prevent overfitting and ensure model generalization.

5 Discussion

The table below shows the performance parameters (Accuracy and F1 Score) of different models which were implemented.

Model	Accuracy	F1 Score
CNN	0.87	0.86
SVM	0.58	NA
LSTM	0.24	NA
Vision Transformer	0.45	NA

Table 3: Obtained Accuracy and F1 Score of different Models

It is clear that Convolutional Neural Networks (CNN) has proven to be the most accurate, when it comes to classify food categories with chewing vibration data.

And among various types of models,CNN has the best accuracy below is the detailed overview of the CNN models.

However the MobileNetV3-Small attains the best generalization in comparison to all other models according to its smaller generalization gap between training and validation loss after 40 epochs, its test accuracy was much lower than EfficientNetV2-S. Similar to the first round, EfficientNetV2-S achieved the highest test accuracy during the second round which was 80 epochs and finally accomplished the training task after 200 epochs during the last round with 86.54% test accuracy. It is true that there is still some generalization gap between its training and validation loss, but we should consider very low value of learning rate which causes the optimization process to be very slow. Furthermore, the low amount of data available for model training may also be another reason for that generalization gap.

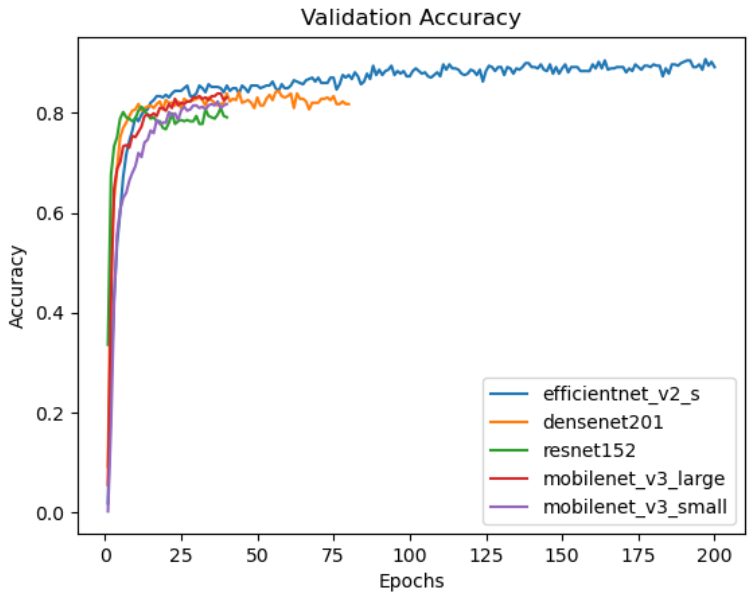

Figure 19: Validation accuracy of all trained models

6 Market Research

A food journal is a daily log through which people keep track of what they eat throughout the day. Keeping a food journal is incredibly beneficial when it comes to losing weight, improving diet or simply be aware of one's food habits to improve health and lifestyle. Food journals are sometimes recommended by doctors and dietitians, who can use them to better understand your eating habits. In some cases, a healthcare professional will also use them to determine which foods or ingredients you may be sensitive to. Keeping a diet-log manually is often a burden on the person and ultimately relies on their memory which introduce inaccuracies in the data.

The Automated Dietary Monitoring System with wearable sensor technology and Artificial Intelligence can overcome these constraints of manual self-reporting by automating this process.

6.1 Market Size

As the awareness of fitness and healthcare is increasing among people, the demand for app based wearable sensor technologies is rising.

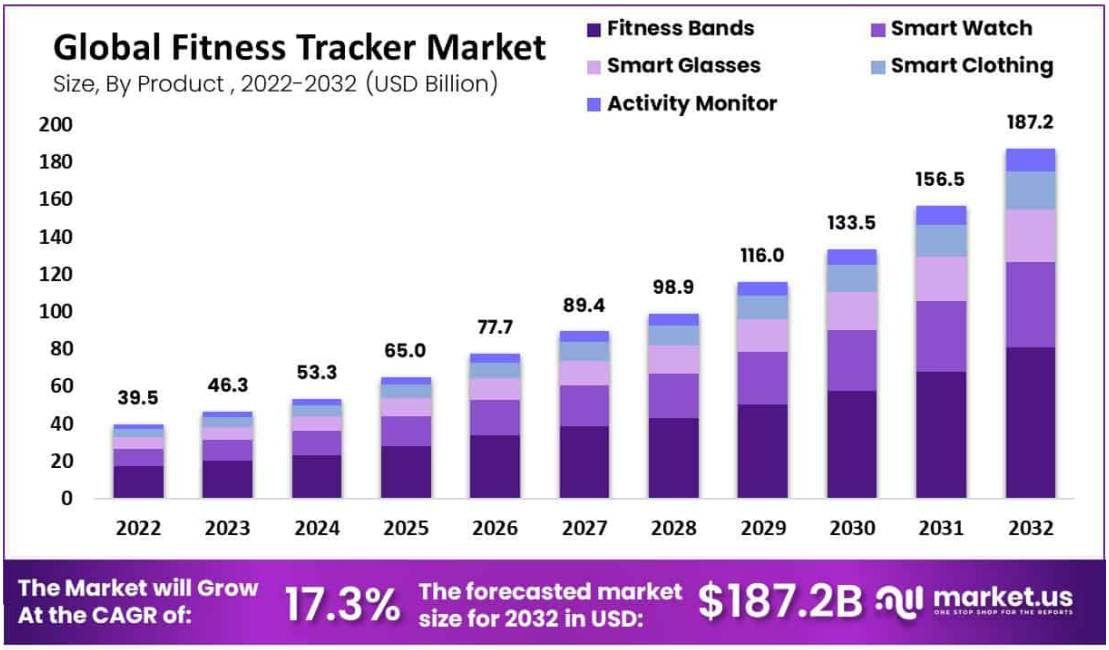

Figure 20: Global Fitness Tracker Market [18]

Wearable fitness technology has carved out such a significant space for itself in the healthcare industry, that devices such as FitBits and smartwatches are now viewed as mainstream. The use of wearable technology has more than tripled in the last four years, in accordance with consumers' increased interest in monitoring their own health and vital signs. Demand for wearables is expected to continue to rise in the next few years, as consumers exhibit interest in sharing their health data with providers and insurers. The US Smart wearable user market is poised to grow 25.5% YoY in 2023, up from 23.3% YoY growth in 2021, per an October 2021 forecast by Insider Intelligence. [19]

The Wearable Technology Market was valued at around USD 39.5 Billion in 2022 and is estimated to be worth approximately Market Size USD 187.2 Billion in 2032, growing at a CAGR of slightly above 17.3% between 2023 and 2032 (see Figure 21). Awareness of these devices has grown mainly during the Covid-19 pandemic. With rising acceptance, novel market players compete to attain the growing need and obtain a more significant market share. This has encouraged a rise in research activities and device innovation in wearable technologies.

By region, the global wearable technology market has been separated into North America, APAC, Europe, Latin America, and MEA. However, North America is the most lucrative market share and will dominate the market share with 42% in 2022. The market in Europe is projected to grow from USD 14.61 billion in 2022 to USD 35.13 billion by 2027, growing

Attribute	Details
Market Value (2022)	USD 39.5 Billion
Market Size (2032)	USD 187.2 Billion
CAGR (from 2023 to 2032)	17.3%
North America Revenue Share	42.0%
Europe Revenue Share	20.7%
Historic Period	2016 to 2022
Base Year	2022
Forecast Year	2023 to 2032

Table 4: Global Market Share Attributes. [19]

at a CAGR of 19.18% from 2022 to 2027. Regionally, the UK fitness trackers market accounted for the leading share of the European market in 2021, followed by Germany. The growing number of fitness tracking apps go hand in hand with these devices. People opt for android or iOS modes of compatible devices that are accurate and convenient. [20]

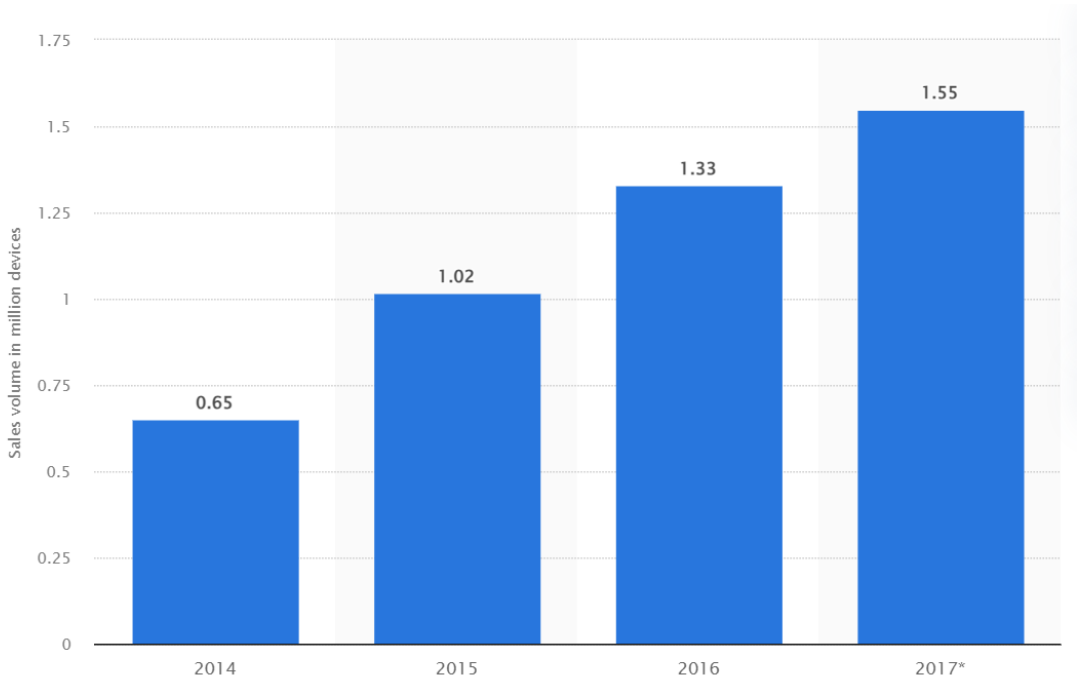

Figure 21: Wearable Fitness Devices' Sales in Germany [21]

6.2 Driving Factors

The market is experiencing significant growth due to the increasing awareness of staying healthy and fit and the need to monitor diet activities. These tracking products have evolved from basic pedometers to sophisticated smart devices in last few years. The fitness industry is still in its early stages, but it foresees substantial adoption, particularly among the younger generation. Many individuals now prefer health clubs and gyms to counter the side-effects of their busy lifestyles. Regular workouts are valued for their ability to reduce stress, anxiety, and depression.

The surge in health issues has prompted people to focus on maintaining a healthy diet. Fitness trackers have emerged as valuable tools to help individuals monitor their exercises, further fueling the demand for health monitoring products. Additionally, certain economic factors such as increased per capita healthcare expenditure, improved healthcare infrastructure, and greater investment in innovative technologies are expected to boost the global market's growth rate by 2028. Thus, the growing health awareness is anticipated to be a key driver for market growth.

The rise in sedentary lifestyles has led to an alarming increase in obesity, a condition that is often accompanied by other health issues such as diabetes, sleep disorders, and various diseases. This escalating obesity crisis has prompted individuals to understand the crucial role of mindful eating and maintaining a food journal in combating weight gain and promoting overall health. As awareness grows about the importance of a healthy lifestyle in preventing obesity, there is a surge in demand for food journaling. This trend is expected to significantly boost the market for automated diet monitoring products.

Furthermore, these devices offer customers enhanced flexibility, enabling them to take preventive measures against major health conditions. Consequently, rising health consciousness is expected to be a major catalyst for the market's expansion.

6.3 Restraining Factors

Safety and privacy concerns are major restraining factors for the growth of Automated Dietary Monitoring devices. These devices rely on collecting user's personal data, which raises concerns about data theft and potential privacy breaches that could lead to significant harm. Many users remain unaware of the privacy implications associated with the potential misuse of their data, especially when accumulated over time or combined with other information. Recent incidents of data theft through fitness tracking devices have

compromised the privacy of millions of users, which are likely to impede market growth for diet monitoring devices.

An additional restraining factor is the low prevalence of suitable wearable devices. Despite the technological advancements, the adoption rate of wearable devices suitable for automated dietary monitoring is still not widespread. This could be due to factors such as cost, lack of awareness, or accessibility issues in certain regions. This limited prevalence poses a significant challenge to the growth of the automated diet monitoring market.

7 Conclusion

This report presents a pioneering study that explores the utilisation of Artificial Intelligence for food types detection.

The research involved an in-lab investigation, where vibration signals were captured while individuals consumed foods with varying textures and hardness. Deep learning models were developed to identify patterns in chewing sequences and associate them with specific food types.

The development of these holds great promise as an automated dietary monitoring tool. It addresses the gap in the wearable devices market as well as in the nutrition and health market, by providing a device capable of quantifying masticatory patterns during food consumption, aiding in the recognition of food type and contributing to improved digestive health.

Convolutional Neural Networks (CNN) proved to be the most performant model for food classification based on the collected signals, achieving high test accuracy. Despite some generalization gaps between training and validation loss, the potential of this technology is significant, especially when coupled with a larger dataset.

Overall, this research showcases the exciting potential of integrating Artificial Intelligence for efficient dietary tracking and analysis. With further advancements and a larger dataset, this technology could revolutionize dietary monitoring and contribute to improved health and well-being.

Appendix

References

[1] "Amft, o., et al. "analysis of chewing sounds for dietary monitoring." section. 5.1, pp."

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[4] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[6] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[14] "A beginner's guide to lstms and recurrent neural networks." [Online]. Available: https://wiki.pathmind.com/lstm

[15] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[17] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[18] "Fitness tracker market." [Online]. Available: https://www.globenewswire.com/en/news-release/2023/04/06/2642293/0/en/Fitness-Tracker-Market-Predicted-to-Garner-US-187-2-Bn-by-2032-At-CAGR-17-3.html

[19] "Latest trends in medical monitoring devices and wearable health technology." [Online]. Available: https://www.insiderintelligence.com/insights/wearable-technology-healthcare-medical-devices

[20] "Wearable fitness devices' market in europe." [Online]. Available: https://www.marketdataforecast.com/market-reports/europe-fitness-trackers-market

[21] "Wearable fitness devices' sales in germany." [Online]. Available: https://www.statista.com/statistics/483282/fitness-tracker-sales-volume-germany/