Raspoznavanje dišnog signala iz toplinskih snimki

Cvjetko, Marko

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:978004

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-29**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 595

RASPOZNAVANJE DIŠNOG SIGNALA IZ TOPLINSKIH SNIMKI

Marko Cvjetko

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 595

RASPOZNAVANJE DIŠNOG SIGNALA IZ TOPLINSKIH SNIMKI

Marko Cvjetko

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 595

Pristupnik:	**Marko Cvjetko (0036500931)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Raspoznavanje dišnog signala iz toplinskih snimki**

Opis zadatka:

Raspoznavanje videa važan je zadatak računalnog vida s mnogim zanimljivim primjenama koje mogu pospješiti istraživanja u biologiji i medicini. Ovaj rad razmatra raspoznavanje fizoloških funkcija laboratorijskih životinja u toplinskim snimkama. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće diskriminativne arhitekture utemeljene na konvolucijama i pažnji. Pribaviti skupove snimki te oblikovati podskupove za učenje, validaciju i testiranje. Odabrati i prilagoditi prikladan model za promatranu primjenu te uhodati postupke učenja i validiranja hiperparametara. Primijeniti naučene modele te prikazati i ocijeniti postignutu točnost. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2024.

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 595

BREATHING SIGNAL RECOGNITION FROM THERMAL FOOTAGE

Marko Cvjetko

Zagreb, August, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 595

RASPOZNAVANJE DIŠNOG SIGNALA IZ TOPLINSKIH SNIMKI

Marko Cvjetko

Zagreb, kolovoz, 2024.

The master thesis assignment text in English goes here.

Ovdje dolazi tekst zadatka diplomskog rada na hrvatskom jeziku.

I want to thank Professor Siniša Šegvić and Petra Bevandić for their guidance during my studies.

Special thanks to Professor Sebastian Haesler for giving me the opportunity and continued support to work and study at the Neuro-Electronics Research Flanders. I also want to thank Michiel, Cagatay, Kacper and many other lab members for making my stay at NERF pleasurable.

Last but not least, I am grateful for the love and support of my family and friends.

Contents

1 Introduction 3

2 Background 5

2.1 Monitoring Respiration in Rodents 5

2.2 Computer Vision 7

2.3 Deep Learning 9

2.3.1 Transformer model 13

2.4 Transfer learning 14

2.4.1 Masked Image Modeling 15

2.4.2 Evaluation of trained models 15

3 Methodology 19

4 Dataset 21

4.1 Pressure sensor signal 22

4.2 Manual Annotation 23

4.3 Preprocessing of camera footage and the pressure sensor signal 24

5 Results and Discussion 26

5.1 Evaluation on manual annotations 27

5.2 Evaluation on intranasal cannula pressure sensor signal 28

5.3 Error analysis 29

6 Conclusion and Future work 31

6.1 Future work 31

References 33

Abstract **37**

Sažetak **38**

1 Introduction

Measuring behaviour is crucial aspect of many fields of biomedical research. Depending on the experiment's nature, different behavioural traits, such as eye movement, pupil dilation, respiration and others can be particularly informative. Throughout history, accurate behavioural measurements have been challenging to obtain due to technological constraints. Today however, microphones, cameras, electroencephalograms (ECG) and other measurement tools allow us to simultaneously gather experimental data in multiple modalities (Figure 1.1).

This thesis focuses on monitoring breathing behaviour of mice, which have been the animal models of choice in many biomedical studies. The advantages of using rodents as animal models include their small size, ease of maintenance, short life cycle, and abundant genetic resources [2]. Breathing is often studied in the context of odor guided behaviours and olfactory perceptual tasks, and is relevant in many other applications, such as drug discovery or disease modeling.

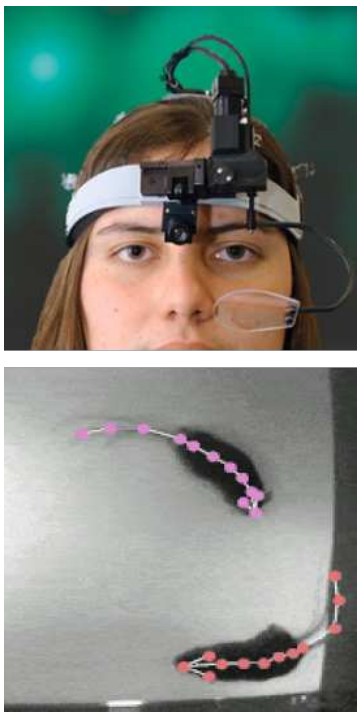

Figure 1.1: Examples of recording behavioural data. Above, an eye tracking device records eye movements. Below, movements of mice are tracked with a 2D bodypart tracking algorithm [1].

Monitoring breathing behaviour is still an open problem and many approaches have been proposed, each with its own advantages and drawbacks, usually related to invasiveness, cost, ease of use, latency and flexibility in terms of experimental setups. Therefore, choosing the appropriate method for a specific task can be challenging [3].

This work aims to create a machine learning-based framework for monitoring breathing in infrared (IR) camera. The method's main benefits are its non-invasiveness and ease of use (once the initial camera setup has been prepared). The proposed machine learning model is intended to work with manually annotated labels, thus reducing, or completely removing the need for gathering reference breathing measurements. The work intends to be easily accessible to researchers in biomedical sciences and compatible with the Three Rs Principle [4]. The Three Rs are ethical guidelines adopted by European Union for using animals in science. The Three Rs stand for:

- **Replacement:** Seeking alternatives to using animals in experiments whenever possible, e.g. with cell cultures, or computational models.
- **Reduction:** using the minimum number of animals necessary to obtain reliable results, through careful experimental design and statistical analysis.
- **Refinement:** includes minimizing pain and distress, providing appropriate housing and care, and enriching their environment to promote natural behaviours.

The proposed framework is evaluated on dataset of manually labeled inhalation onsets, and a dataset of intranasal cannula pressure sensor breathing measurments, and is compared with the previous method used within Haesler lab[1].

[1] https://haeslerlab.sites.vib.be/en#/

2 Background

This chapter provides a succint overview of methods for measuring respiration in rodents, and introduces the fields of computer vision and deep learning. For a more comprehensive overview of measuring respiration in rodents, the reader can refer to the review by Grimaud and Murthy [3].

2.1 Monitoring Respiration in Rodents

Breathing is the involuntary act of inhaling, while sniffing refers to active sampling of odors for the purpose of information acquisition. Sniffing is typically identified by changes in respiratory frequency or amplitude, e.g. while interacting with others or exploring the environment. Breathing and sniffing are related behaviours and are measured with the same tools. The frequency of breathing of dormant mice ranges from 3 to 5 Hz, and increases up to 12 Hz during bouts of exploratory behaviour [5]. Several methods for breathing monitoring are described below and illustrated in Figure 2.1:

Breathing behaviour can be monitored through neuronal or muscular activity. Attempts have been made at placing an electrode into brain regions which govern respiratory muscle activity in order to measure breathing, but have been shown to lack reliability at higher breathing frequencies [6]. On the other hand, recording breathing from muscle activity has been more successful and usually involves implanting electromyogram (EMG) electrodes on the diaphragm. Other muscles, such as the tongue and those involved in whisking, have also been found to contract in accordance with the rhythm of breathing, however they are less correlated at lower breathing rates. EMG recordings are regarded as one of the most precise measurements of respiration. However, both methods have the drawback of being extremely invasive to the animal.

Plethysmography refers to measuring volume changes within an organ or the entire body. For example, a plethysmograph can capture the expansions and contractions of the chest cavity, providing the breathing behaviour measurements. Alongside measuring breathing rate, plethysmography has an added advantage of being able to measure the volume of air passing at each inhalation and exhalation. Several plethysmograph designs exist: lung plethysmographs connect the trachea to the instrument, others press a mask against the face, or place the animal in a chamber, with the last method being most suited for small rodents. Plethysmographs also differ depending on what they measure: airflow, pressure, or capacitance.

Another method involves sensors such as pressure transducers or thermistors to record olfactory behaviour. However, measuring respiration of smaller animals is generally done through intubation, or implantation of sensors in the intranasal cavity, making the processes invasive to the animal. Furthermore, intubation can only be performed on anesthetized rodents, while the intranasal probes require puncturing the roof of the nasal cavity, which is suspected to affect the way the animal breaths, and disturb odor perception.

Lastly, breathing behaviour can be extracted using infrared thermography (IRT). By capturing the temperature changes of the air around the nostrils during breathing, an infrared (IR) camera can provide the necessary data to extract the breathing signal. The main limitation of IR camera is its price, as the camera needs to be of high enough frame rate, and have sufficient thermic sensitivity to accurately capture the heating and cooling of the air around the nostrils. At the time of writing this thesis, the camera Rabell and Haesler used in their publication (FLIR A325sc infrared camera) costs several thousand US dollars, not including the dedicated lens [7]. Additionally, the camera is relatively large and needs to be placed very close to the animal. This method is currently restricted to head-restrained animals, since movements might omit the nostrils. After recording, the breathing signal is extracted through numerous video-provessing steps.

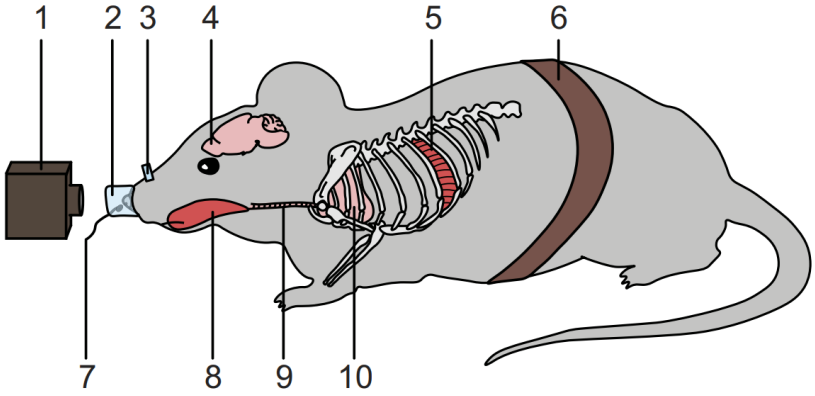

Figure 2.1: Visual summary of the methods for breathing measurement in rodents: 1) infrared camera and video monitoring, 2) face mask, 3) intranasal cannula and implanted temperature probe, 4) brain recording, 5) electromyogram (EMG) of respiratory muscles, 6) movement sensor, 7) external temperature probe, 8) EMG of nonrespiratory muscles, 9) intubation, and 10) lung plethysmograph [3].

2.2 Computer Vision

Computer vision is a field of computer science concerned with extracting meaningful information from images, which can take many forms, such as videos, medical imaging scans and multiple camera images. The field arose from the desire to mimic the capabilities of the human visual system, such as recognizing faces, or navigating traffic. Thus, it is intimately intertwined with neuroscience, a relationship from which both have benefited [8]. Several fundamental tasks in computer vision are listed below and illustrated in figure 2.2:

1. **Image Classification:** The task of assigning a label to an image from a predefined set of categories. The aim is to determine the primary subject or content of the image. For example, classifying an image as a "cat" or a "dog".

2. **Object Detection and Localization:** This task involves not only identifying objects within an image but also determining their spatial location by drawing bounding boxes around them. This is relevant in applications like autonomous driving, where the precise location of pedestrians, vehicles, and traffic signs is a necessity.

3. **Semantic Segmentation:** This task takes object recognition a step further by assigning a class label to each pixel in an image, resulting in a pixel-level understanding of the scene, where each pixel is labeled according to the object it belongs to.

4. **Instance Segmentation:** Building upon semantic segmentation, instance segmentation goes further by distinguishing individual objects within the same class. For example, an instance segmentation model would identify each car as a separate instance, even if they belong to the same class ("car").

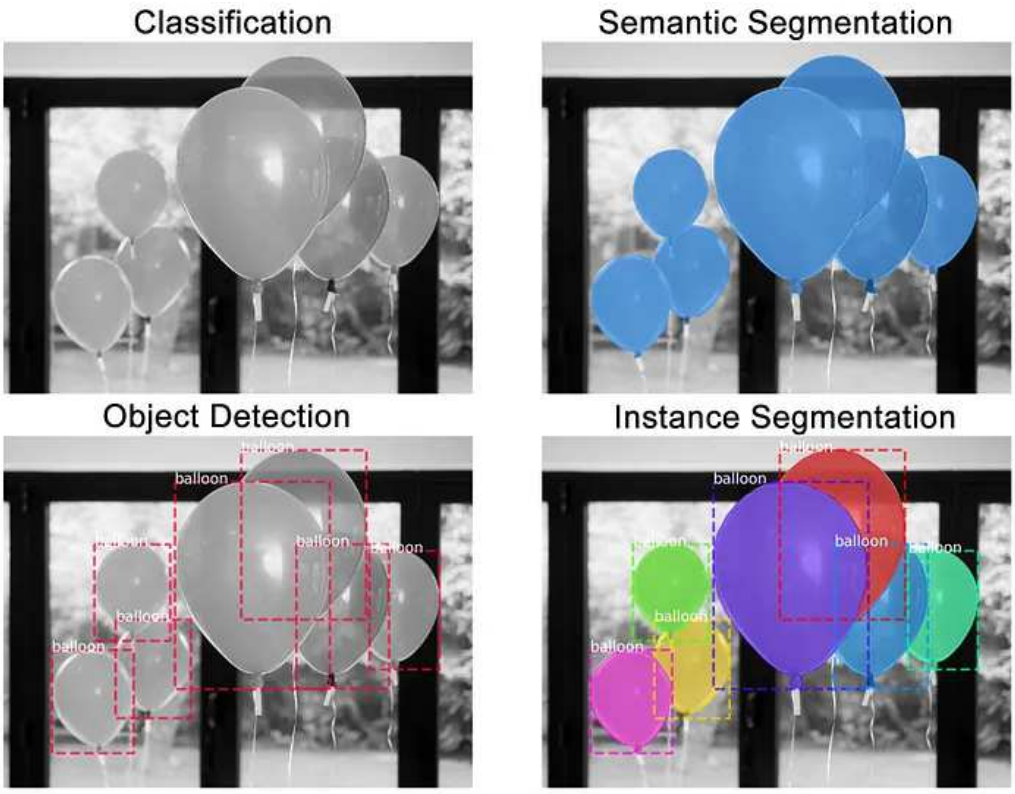

Figure 2.2: Illustration of different fundamental computer vision task (image source: [9]).

As the field becomes more capable, more complex tasks gain traction, such as generating images from text, editing images based on text descriptions (inpainting), and creating videos from text or image [10][11].

2.3 Deep Learning

Deep Learning (DL) is a subset of Machine Learning (ML), which falls under the broader umbrella of Artificial Intelligence (AI). While AI encompasses various subfields and definitions, Russell and Norvig define it as the development and study of methods and software that enable machines to perceive their environment and take intelligent actions to achieve defined goals [12]. ML focuses on systems that automatically learn and improve from experience without explicit programming [13]. It can be further divided into several branches, of which three are discussed below and illustrated in Figure 2.3:

- **Supervised learning**: refers to algorithms that learn from a set of data that contains inputs (features) and the desired outputs (labels). Consider an email service provider which would like to automatically report spam emails. The provider could train a supervised ML algorithm using email header information, sender's address, title, etc., as features, and the users' spam reports as the desired output. The goal would be learn to identify feature patterns in spam emails and automatically flag them for future users. Another example is estimating house prices by looking at their age, location, square footage, etc. An important difference between these examples is that the former outputs categorical values, while the latter outputs real values. These two cases are called classification and regression, respectively.

- **Unsupervised learning**: this class of algorithms learn and discover patterns from data which is not labeled. Common tasks include density estimation, content generation, data compression and anomaly detection. For example, a retailer may want to create targeted advertisements by grouping customers based on their purchasing behaviour, such as the amount of money spent, purchasing frequency and types of products purchased.

- **Self-supervised learning**: this approach leverages unlabeled data by creating pretext tasks that generate surrogate labels from the data itself. For example, in natural language processing, a model can be trained to predict masked words in a sentence with respect to visible words, or in computer vision, to reconstruct a masked portion of an image [14] [15]. By learning to solve these pretext tasks, the model learns useful representations of the data that can be transferred to other

downstream tasks, such as image classification or object detection. Self-supervised learning tasks are often used in combination with transfer learning, which is further described in Section 2.4.

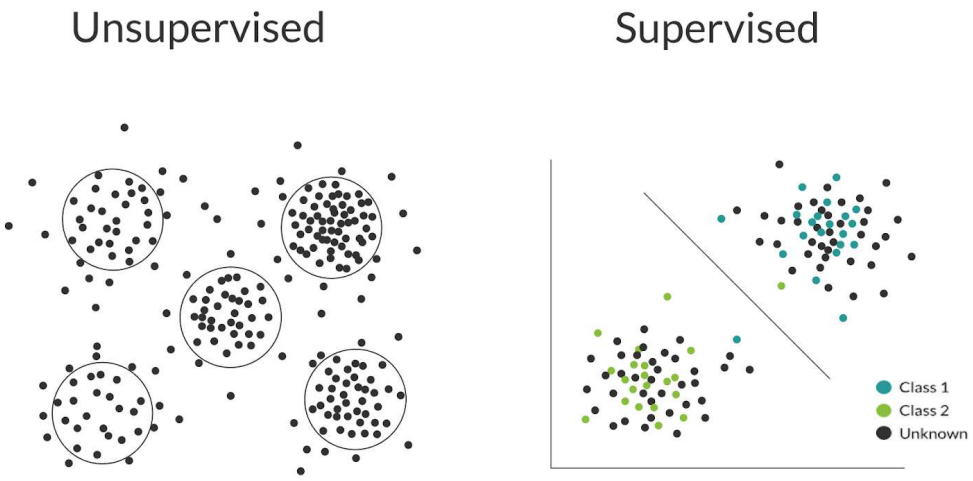

Figure 2.3: Illustration of unsupervised and supervised learning. On the left, a ML model learns to group samples based on their features, without using explicit labels. On the right, a model is trained to assign labels to samples, based on known feature-label pairs, in this case by learning a boundary in the feature space which best separates known samples of classes 1 and 2.

DL is specialized subset of ML using methods based on artificial neural networks. The adjective "deep" refers to the use of multiple hierarchical layers in the network. Although current DL architectures do not intend to model the human brain, the field has produced many successful architectures that were heavily inspired by neuroscience [16].

A quintessential DL architecture is the perceptron model (Figure 2.4), and its multilayer generalization (Figure 2.5). They are examined below:

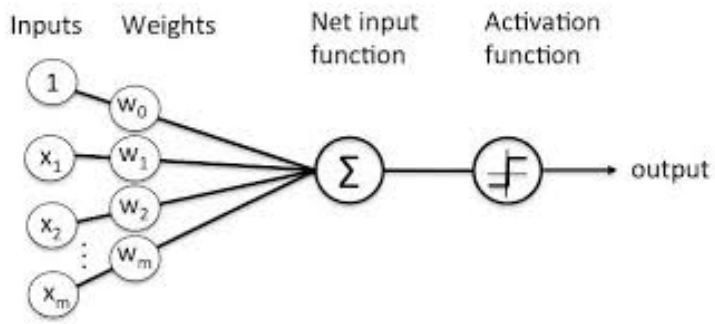

Figure 2.4: The perceptron model.

A neural network's task is to learn an arbitrary function:

$$y = f^*(\mathbf{x}) \tag{2.1}$$

where x is the input vector of features and y is the target label. This can be represented as a training set consisting of many input-output pairs of data:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_n, y_n) \tag{2.2}$$

The perceptron model can be expressed with the following equation:

$$y = \sigma(\sum_{i=1}^{m} w_i x_i + b) \tag{2.3}$$

where x_i are input features, w_{ij} and b are model parameters (weights), and σ is the activation function, which is some non-linear function, such as the sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

or a ReLU (Rectified Linear Unit) function:

$$\text{ReLU}(x) = \max(0, x) \tag{2.5}$$

Essentially, for a given input sample, the perceptron computes a weighted sum of the feature vector and passes the resulting value through a nonlinearity. The training of a perceptron (and other DL architectures) amounts to finding the set of model parameters $\mathbf{w}$ which assign a correct label y, given the input features $\mathbf{x}$. This is achieved by defining a loss function and using an optimization algorithm to minimize it.

The loss function quantifies the error between the model's prediction and the true target value. This error is typically a non-negative number where smaller values indicate better performance. A common loss function in regression tasks is the Mean Squared Error (MSE):

$$MSE = (y_{predicted} - y_{true})^2 \tag{2.6}$$

For binary classification tasks, Cross-Entropy Loss is commonly used:

$$L = -y_{true} ln(y_{pred}) - (1 - y_{true}) ln(1 - y_{pred}) \tag{2.7}$$

The goal of the optimization algorithm is to minimize the loss function by adjusting model parameters. Today, most neural networks are optimized with iterative algorithms using backpropagation, such as the Stochastic Gradient Descent (SGD). For a given sample, the SGD calculates the loss based on the forward pass through the network, it then uses the calculus' chain rule to propagate this loss backward through the network, computing the gradient of the loss with respect to each parameter. With the gradients, the algorithm updates the model's parameters in a direction that reduces the overall loss, as shown in the following expression:

$$\hat{w_i} = w_i - \eta \frac{\partial L}{\partial w_i}, \tag{2.8}$$

where η is the learning rate, a hyperparameter which modulates the strength of the update. Care must be taken when choosing the learning rate, because overly large learning rates will result in overshooting the optimal solution or divergence, while too small learning rates result in a very slow and inefficient convergence. The whole process is repeated for many iterations, and with each iteration the model's ability to approximate the target function improves.

The multilayer perceptron (MLP) is a natural extension of the base perceptron, and is formed by stacking multiple perceptron layers on top of each other. The outputs of one layer become the inputs to the next. MLPs are also commonly referred to as feedforward neural networks due to the unidirectional flow of information. Each layer in an MLP applies a linear transformation to its input, followed by a non-linear activation function. The hidden layers, situated between the input and output layers, learn to extract progressively more complex and abstract representations of the input data. An MLP with two hidden layers is represented with the following equations:

$$\mathbf{h_1} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_2) \tag{2.9}$$

$$\mathbf{h_2} = \sigma(\mathbf{W}_2\mathbf{h_1} + \mathbf{b}_2) \tag{2.10}$$

$$\mathbf{y} = \sigma(\mathbf{W}_3\mathbf{h_2} + \mathbf{b}_3) \tag{2.11}$$

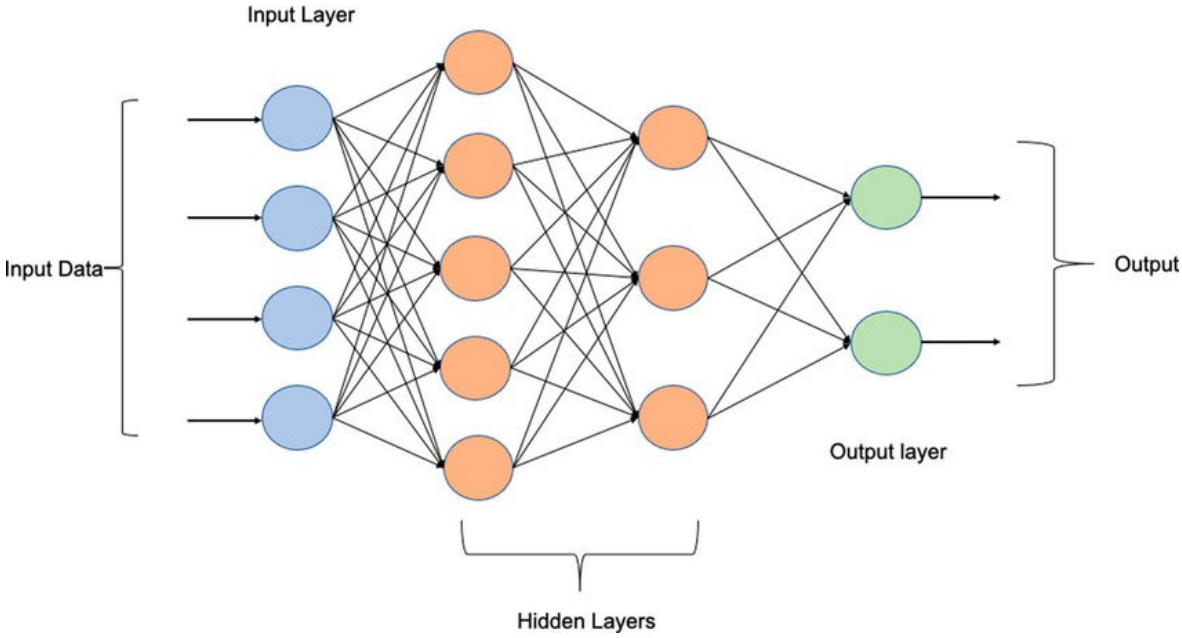

Figure 2.5: A depiction of a multilayer perceptron architecture

Although the base perceptron and its multilayer generalization are considered simple compared to contemporary architectures, the core ideas surrounding them are still present in most DL algorithms.

2.3.1 Transformer model

Transformer models are a type of DL architecture that has revolutionized the field [17]. They were initially used for tasks involving sequential data (such as text), but have over time been adapted to handle data of other modalities. Since their introduction in 2017., transformers models have often been described as *groundbreaking* and have been successfully applied in a wide variety domains [18].

The core innovation of transformers is the self-attention mechanism, implemented using a set of query, key, and value matrices. For each element (token) in the input sequence, a query vector is computed, which is then used to compute attention scores with all other elements in the sequence based on their corresponding key vectors. These attention scores are then used to weight the value vectors, producing an output representation for that element. The scaled dot-product attention, used in the original transformer

paper, can be expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimensionality of the key vectors. In addition to self-attention, transformers typically include feedforward layers and residual connections.

While the original transformer model used an encoder-decoder architecture, not all transformer models follow this structure. Some are encoder-only (like those used for tasks like sentiment analysis), and others are decoder-only (such as those used for text generation).

Transformer architectures have been successfully adapted for computer vision tasks, giving rise to Vision Transformers (ViTs) [19]. The ViT architecture is similar to BERT, with main differences being in early layers of the network, which transform inputs into a form suitable for the attention mechanism [14]. Images are transformed by splitting into patches, which are then flattened and passed through a linear embedding layer. ViTs led to significant advancements in many vision related tasks.

Transformers have been applied to a wide range of tasks, including machine translation, text-to-image generation and are the key component for the success of large language models.

2.4 Transfer learning

An important question when training DL models is what set of initial model parameters to use, as it has been shown to greatly impact the speed and convergence of the training procedure [20]. In the past, initialization techniques generally involved assigning random values to model parameters by sampling from the probability distributions, such as the uniform or Gaussian distribution. More recently, an idea emerged to reuse weights of previously trained (pretrained) models on new tasks. The intuition is that the knowledge obtained by solving a particular task will transfer well into solving similar tasks, hence the name transfer learning. For example, a model trained to classify images of dogs has

likely learned to extract features which are relevant for detecting other animals. A pretrained model can be used either as a fixed feature extractor, in which case its weights are *frozen*, or its weights can be updated, but generally with a lower learning rate than during pretraining.

With the growing sizes of DL architectures, training from scratch has become increasingly laborious, and transfer learning has stood out an especially useful concept, offering numerous benefits. When used appropriately, transfer learning has been shown to improve generalization capabilities of the model, significantly shorten training time, and reduce the amount of training data necessary to solve a task. It has also played a key role in bringing sophisticated DL models to a wider audience, thus democratizing accessibility.

2.4.1 Masked Image Modeling

Masked Image Modeling (MIM) has recently emerged as a powerful technique for self-supervised pre-training of vision transformers [21][15]. MIM aims to learn robust image representations by randomly masking a percentage of an image and training a model to predict the masked content, conditioned by the visible portions of the image. This approach learns to translate the inherent structure and patterns within images into meaningful representations without relying on explicit labels. MIM draws inspiration from the success of Masked Language Modeling (MLM) in natural language processing, where models are pretrained by predicting masked words in a sentence.

Several implementations of MiM exist, usually consisting of the following core steps: First, the image is divided into patches, of which a random percentage is masked. An encoder extracts features of the visible patches, which are then passed to the decoder, that attempts to predict either the masked patches, or the whole image. An example architecture is illustrated in Figure 2.6.

2.4.2 Evaluation of trained models

Evaluation of trained models is an essential step in DL, and ML in general. It allows us to quantify how well the model performs on unseen data and understand its strengths and weaknesses.

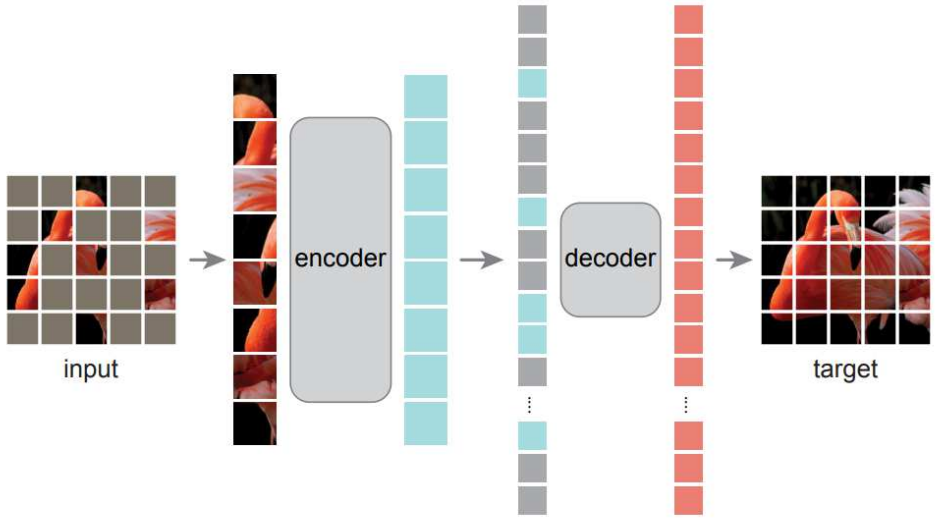

Figure 2.6: The Masked Autoencoder architecture [15]. During pretraining, the encoder receives the visible patches of the image. The features obtained from these are passed to the decoder that reconstructs the original image. After pre-training the decoder is discarded and the encoder applied to uncorrupted images.

Training, Test and Validation Datasets

In order to properly evaluate the performance of a model, the available dataset is split into a training, validation and test sets. The training set is used to adjust model parameters during the training process. The validation set is used to monitor model performance on unseen data during training process and to fine-tune hyperparameters, which are set before the training process begins. Examples of hyperparameters are the learning rate in the SGD algorithm, or the number of hidden layers in a multilayer perceptron. The test set provides an unbiased evaluation of the final model fit on the training dataset. It is important that the test set remains unseen during training to provide an accurate estimate of how the model will generalize to new data.

Classification Metrics

When evaluating the performance on a binary classification task, each prediction can be assigned to one of four categories:

- **True Positive (TP):** The model correctly predicts the positive class.
- **False Positive (FP):** The model incorrectly predicts the positive class.

- **True Negative (TN):** The model correctly predicts the negative class.
- **False Negative (FN):** The model incorrectly predicts the negative class.

Using these four categories, we can calculate various classification metrics:

- **Accuracy:** The fraction of correct predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.13}$$

  This metric is suitable when the dataset is balanced, meaning that the number of positive and negative samples is roughly equal. In this case, accuracy provides a good overall measure of model performance.

- **Precision:** Measures the quality of positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.14}$$

  Precision is suitable when the cost of false positives is high. For example, in a spam filter (where a spam is denoted as a positive sample), it is more important to avoid classifying a legitimate email as spam than to miss a few spam emails.

- **Recall (Sensitivity):** The fraction of actual positives that the model identifies correctly.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.15}$$

  Recall is used when the cost of false positives is high. A classic example is medical diagnosis system, in which it is more important to identify all patients with a disease even if it means some healthy patients are misdiagnosed.

- **F1 Score:** The harmonic mean of Precision and Recall, providing a balanced measure of the model's performance.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.16}$$

It is important to note that previously mentioned metrics are not well suited for time series event detection, since they do not consider temporal information. For example, a prediction that is a few miliseconds off the actual event could be considered a false positive, even though it may be practically useful. However, they can be easily adapted using the concept of the *tolerance window*: a neighbourhood is defined around the groundtruth events. Detections falling within this neighbourhood are considered true positives. The size of the window can be adjusted to reflect the desired level of tolerance (Figure 2.7).

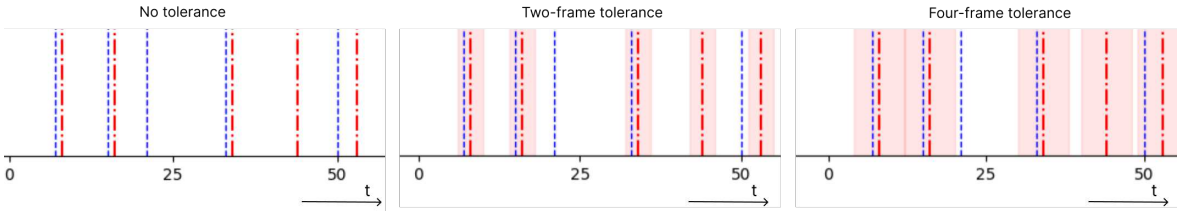

Figure 2.7: Impact of the tolerance window in a time series event detection task. Red and blue lines represent actual and predicted events, respectively. In the first image, where no tolerance is used, zero actual events would be considered correctly predicted, even though several predictions differ by only one frame from the nearest actual event. In the centre and right image, a tolerance is introduced represented by the red areas around actual events, is introduced. Predictions falling within these areas are considered true positives.

Many other metrics exist. In general, the choice of metric depends on the specific problem and the relative importance of different types of errors. Appropriately evaluating the performance of trained models is crucial for understanding how well are they about to generalize to new data and for identifying areas where they can be improved. By using appropriate evaluation metrics and understanding their implications, informed decisions can made about model selection and deployment.

3 Methodology

The goal of extracting the breathing behaviour of mice through IR footage is formulated as a supervised learning video classification task, where each frame is labeled as an inhalation onset or not. The dataset used for the experiments consists of many laboratory trails collected as part of the research in Haesler lab. During all trails, an IR camera was used to record the behaviour of lab mice. The dataset can be split into three distinct subsets: (i) trails for which breathing behaviour was measured with an intranasal cannula, (ii) trails with manually labeled inhalation onsets, (iii) trails without labels. The dataset description and the preprocessing methodology is further described in chapter 4.

A small, custom sized ViT is used as the backbone of the DL architecture throughout all experiments. A smaller size was deemed sufficient considering the narrow domain of the task at hand. The backbone splits the image into patches of 16x16 pixels, with an embedding dimension of 64, 6 encoder layers, and 2 encoder heads. The training procedure consists of the following steps: First, the backbone is pretrained on a large dataset of unlabeled trails using the masked image modeling task. Then, the model is finetuned using the dataset with manual annotations. Finally, an inhalation onset extraction algorithm is tuned on the generated probabilistic output. The architecture is illustrated in Figure 3.1. The resulting model is evaluated on both the test set of manually annonated dataset and the intranasal cannula dataset.

The backbone is pretrained with a masking ratio of 0.5, using the mean squared error loss, Adam optimizer with the cosine annealing learning rate with warmup, and a batch size of 1280 [22]. The pretraining is stopped when the validation loss starts increasing. The finetuning uses binary cross entropy as the loss function, and the Adam optimization algorithm alongside the cosine annealing learning rate scheduler. To adjust for the label imbalance, a weighted random sampler is used.

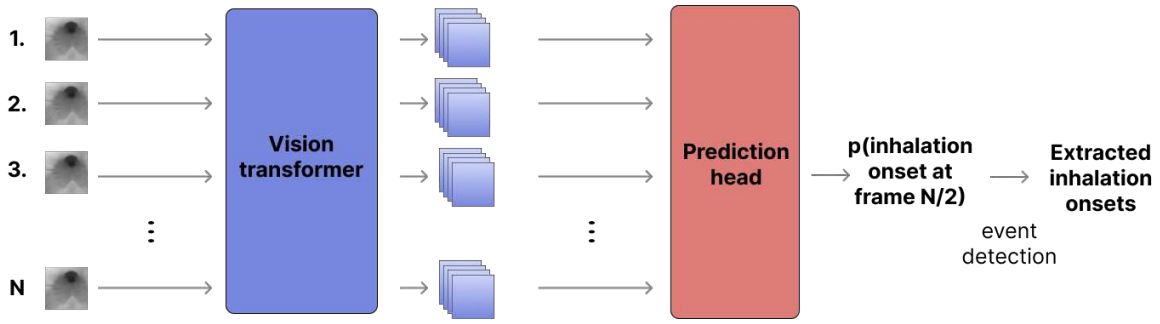

Figure 3.1: Proposed archtecture for extracting the inhalation onsets. A shared ViT outputs features of N consecutive frames (each frame independently), which are fed to the predicton head. The prediction head outputs the probability of the inhalation onset in the center frame. After the probabilties of all frames in a video have been generated, inhalation onsets can be extracted.

Several hyperparameters are considered during training: (i) aside from using a pre-training with Masked Image Modeling, the backbone is also trained from scratch, (ii) different sets of stochastic data augmentations are considered, the first consists of applying horizontal flipping, cropping and resizing, and rotation of the original image, while the second additionally adjusts brightness, contrast and applies Gaussian blurring. (iii) a fully connected layer and an LSTM neural network are considered for the classification head [23].

The experiments are conducted using the *Python* programming language, its DL libraries *Pytorch*[1] and *DeepLabCut*[2], and *scikit-learn*[3], a library offering simple and efficient tools for predictive data analysis [24][1][25]. The labeling was done using the *CVAT.ai* framework[4] [26]. All experiments were run on a single machine with four *Nvidia GeForce GTX 980 Ti* GPUs.

[1]https://github.com/pytorch/pytorch
[2]https://www.mackenziemathislab.org/deeplabcut
[3]https://github.com/scikit-learn
[4]https://github.com/cvat-ai

4 Dataset

The initial dataset comprises of experimental trials on mice. During each trail, mice are presented with an odor, which triggers a behavioural response. The response strength varies depending on the mouse's familiarity with the odor: novel odors generally evoke stronger responses than familiar ones. To ensure consistent experimental conditions, the mice are head-restrained inside a sound and light isolated box. During the experiments, a 60 Hz IR camera is placed under the mouse's nose (illustrated in Figure 4.1). Additionally, an intranasal cannula was connected to a pressure sensor, recording the airflow inside the nostrils (signal sampled at 3 kHz). The recordings last about 10-15 seconds.

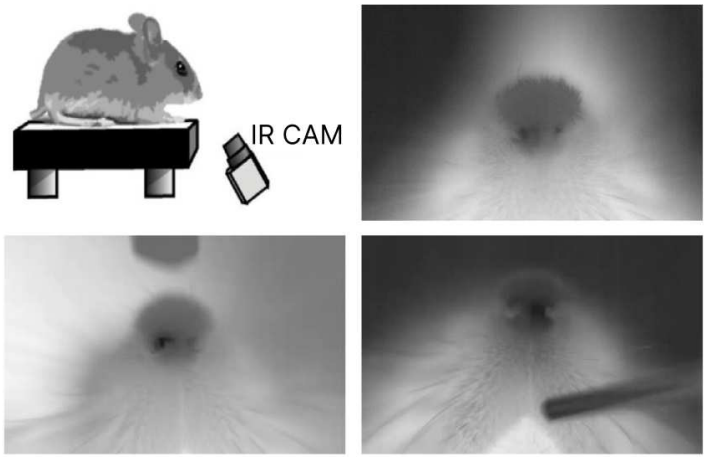

Figure 4.1: Illustration of the camera placement (based on [27]) and several frames captured from different trails

The dataset consists of five mice, with a total 541 trails, however the number of trails per mouse is variable, as shown in Table 4.1.

Mouse Name	Edirne	Jose	Lausanne	Montreux	Neuchatel
# Trails	3	9	137	203	189

Table 4.1: Number of experimental trails per mouse.

In addition to the labeled dataset, the Haesler lab collected IR camera recordings between 2019 and 2024 of approximately 268 000 trails from 220 different mice.

4.1 Pressure sensor signal

Intranasal cannula pressure sensor signals are considered the reference (groundtruth) measurement for breathing behaviour. As the mouse inhales, the air flowing through the nostrils results in lowered voltage output of the pressure transducer. For exhalations, the voltage increases. The recorded signals exhibit a high degree of inter-individual variability. The signals have a varying amounts of noise, slope steepness, local minimum and maximum amplitudes and other features, resulting in distinct breathing cycles for each mouse (as shown in Figure 4.2). This could be due to behavioural differences between mice, but could also be caused by artefacts of the recording method.

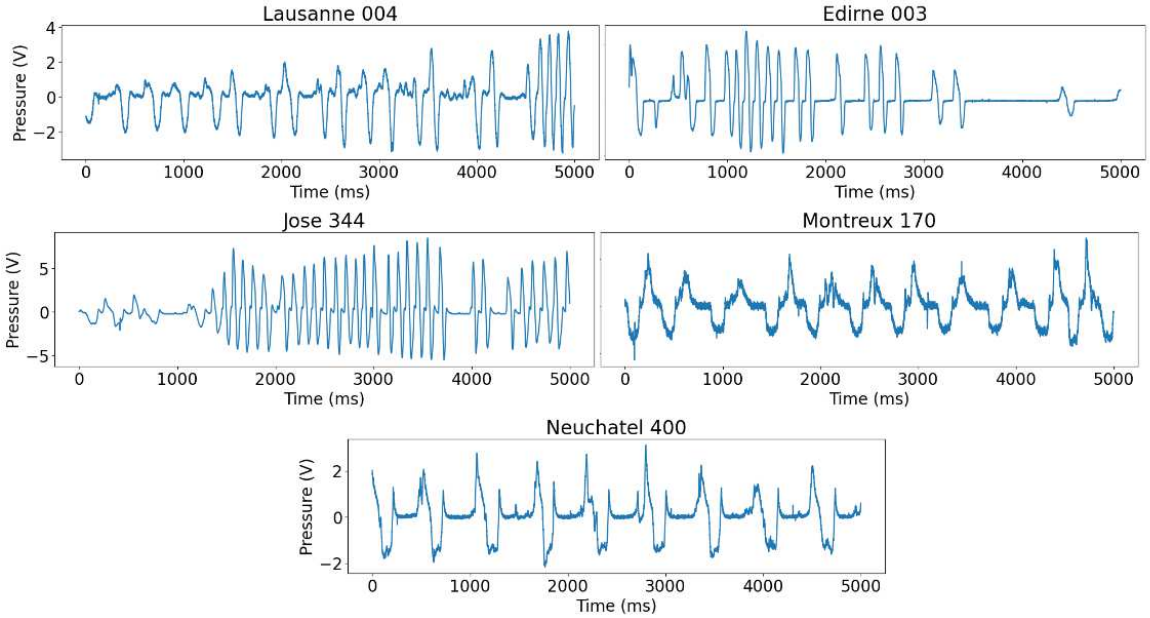

Figure 4.2: Cannula pressure sensor signal samples (groundtruth) from each mouse of the dataset, with mouse and trail names written above the plots.

The initial pressure sensor dataset faces several challenges when examined through the lens of supervised ML. A major drawback is that it only contains five mice, two of which only have several trails, which means that the dataset is likely a poor representation of the whole distribution of experimental recordings that can be found "in the wild". The limited amount of mice for which groundtruth signals exist also severely limits the possibility of robustly testing a trained ML model. Ideally, to conduct proper testing of a ML model, the dataset samples are independently and identically distributed, and split into subsets for training, validation and testing. For the purpose of this task, this implies that different data subsets must not share trails of the same mouse. As there are only

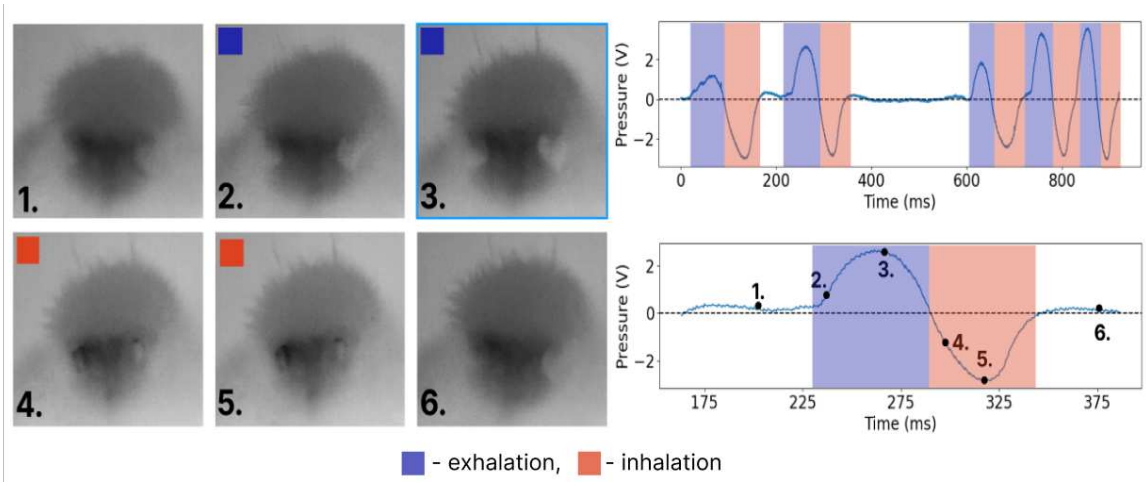

Figure 4.3: Relationship of the IR camera footage and the pressure sensor breathing signal. On the left, a typical breathing cycle is depicted with 6 representative frames. On bottom right, a pressure sensor signal is plotted with enumerated points mapping the signal to the camera frames. The top right plot showcases the signal over multiple breathing cycles. Inhalations and exhalation periods are marked on frames and signals with blue and red color, respectively.

five mice, it follows that each stage of the ML workflow contains one, two or three mice. Even if the trained model performs well on the test set, it would have been tested on a very few mice, making it hard to make conclusions on the generalizing capabilities of the model. Cross-validation can be used to alleviate the latter issue, but it still holds that the available labeled dataset is inherently limited and might not guarantee a truly generalizable model. Additionally, the small sample size of mice increases the risk of the dataset being biased, reflecting specific characteristics of these particular mice rather than the broader population.

For these reasons, alongside the high intervariability of the pressure signal, the initial dataset was not deemed sufficient for using supervised ML to extract the breathing signal from IR camera footage.

4.2 Manual Annotation

Since the labeled dataset was deemed not satisfactory, an alternative approach is considered. Instead of extracting the whole breathing signal, it would be sufficient to only extract inhalation onsets. The proposed changes reduce the complexity of the problem without significantly reducing the usefulness of the framework, as inhalation onsets themselves are often a behavioural trait of interest in biomedical experiments, and they

allow for computing of the breathing frequency, which is another important trait [28]. Focusing on predicting of specific moments of the breathing behaviour offers advantages as it reduces the problem of pressure signal intervariability. Additionally, since the inhalation onsets are characterized by sudden cooling of air around the nostrils (visible as darkening of pixels on the IR camera), it becomes possible to manually label new trails of other mice, thus expanding the dataset for a more robust training procedure.

130 trails were randomly sampled for annotation from the large unlabeled dataset. The sampled subset contains data from multiple researchers and experimental setups, thus being a representative sample of data generated in Haesler lab in the recent years. First and last few seconds of the sampled videos were cut, as they mostly contain control phases of the trail, where the mouse doesn't percieve and odor, and generally behaves in a passive manner with a stable, low-frequency breathing rate.

During the labeling process, frames which capture the sudden darkening of the pixels around the nostrils (as shown on frames in Figure 4.3) are labeled as the inhalation onsets. 11 trails have been discarded due to poor video quality.

4.3 Preprocessing of camera footage and the pressure sensor signal

The most informative part of the footage for breathing extraction is the area around the nostrils; however, the camera has a much broader field of vision, introducing irrelevant infromation for the DL model. This raises several potential issues: first, the model is at risk of learning bad correlations, i.e. mapping uninformative features to the breathing behaviour, second, if a model would be trained on the whole frame (as opposed to only the image of the nostrils), it would require significantly more computational power and likely a DL model with a higher capacity (i.e. more parameters).

To address this, nose position was extracted for each frame and the frames were cropped to only contain the nose. The positions were extracted with DeepLabCut (DLC), a DL framework for pose estimation in animals [1]. DeepLabCut streamlines the whole data collection, training and inference workflow by offering a graphical user interface for labeling key points, several pretrained models, and out-of-the-box training and infer-

ence pipelines. The nose is cropped by extracting four keypoints, from which a cropping bounding box is created (Figure 4.4). To maintain uniform size, the cropped frames are resized to 128x128 pixels.

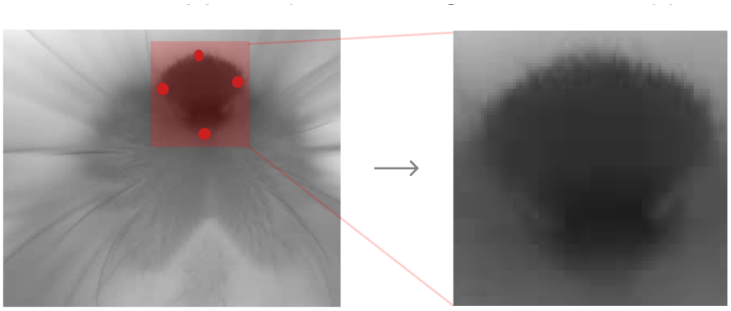

Figure 4.4: An example of the keypoints and the resulting cropped frame using the DeepLabCut framework.

Inhalation onsets were extracted from the groundtruth pressure sensor signals in the following manner: for each trail, the signals were centered by substracting their mean, then a lowpass filter was applied to reduce noise. Peaks were detected from the resulting signals using scikit-learn's *find_peaks* algorithm, and the first zero-crossing after each peak was taken as the inhalation onset. Extracted onsets were visually inspected before being used for testing purposes. An example of inhalation onsets extracted from the pressure sensor signal is plotted in Figure 4.5.

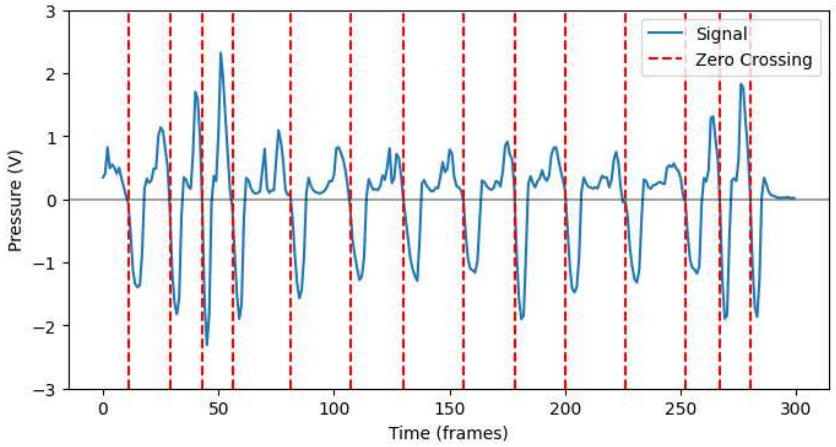

Figure 4.5: Automatically extracted inhalation onsets from the pressure sensor signal.

5 Results and Discussion

The model with a backbone trained from scratch, milder data augmentations and an LSTM classification head achieved the lowest validation loss during training. Even though pretraining generally yields better results, in this case both the pretrained and from scratch trained backbone converged to a very similar validation loss. However, the pretrained backbone converges faster compared to training from scratch. The similar end result can mean that: (i) the finetuning dataset is large enough to properly train the model without extra data, (ii) the features learned with the pretraining task are not informative for extracting inhalation onsets. The best performing model is further evaluated on the test set of the manually annotated dataset and the intranasal cannula dataset. Additionally, comparisons are made with previous method used within Haesler lab.

Previous method

The previous method is based on supervised machine learning using two mice of the intranasal cannula dataset and an AlexNet model [29]. The model is trained to predict the pressure sensor signal value from a context of three consecutive IR frames. The frames are concatenated and passed to the model as an RGB image would be, that is, the channel dimensions represents a temporal relation, rather than RGB values. This is not ideal as AlexNet employs only 2D convolutions, which collapse temporal information into single-channel feature maps, preventing any temporal reasoning to happen in subsequent layers [30]. Another issue arises from the organization of data subsets for training, validation and testing. Namely, the subsets share trails from the same mice, leading to data leakage.

5.1 Evaluation on manual annotations

Precision, recall and F1 score are computed for various tolerance windows (Figure 5.1). For the strictest evaluation (no tolerance), the model achieves 0.63 F1 score. The relatively poor performance for the strictest evaluation is expected since the dataset contains human annotated samples, which are prone to some margin of error. Additionally, extracting the exact moment of the inhalation onset through an IR camera, with a limited frame rate is subject to ambiguity. The model performs significantly better for a tolerance of one, achieving 0.92 F1 score. Considering that one frame represents 16.67 miliseconds, the high performance for such a low tolerance window demonstrates the consistency of annotations and is deemed a satisfactory result. Further increase of the tolerance yields diminishing returns.

The previous method performs significantly worse with no tolerance, however some of its poor performance can be explained with the fact that the models are evaluated on manually annotated labels. Considering that our method was trained on the manual annotations, and the previous one was trained on pressure sensor signals, the former has an inherent advantage.

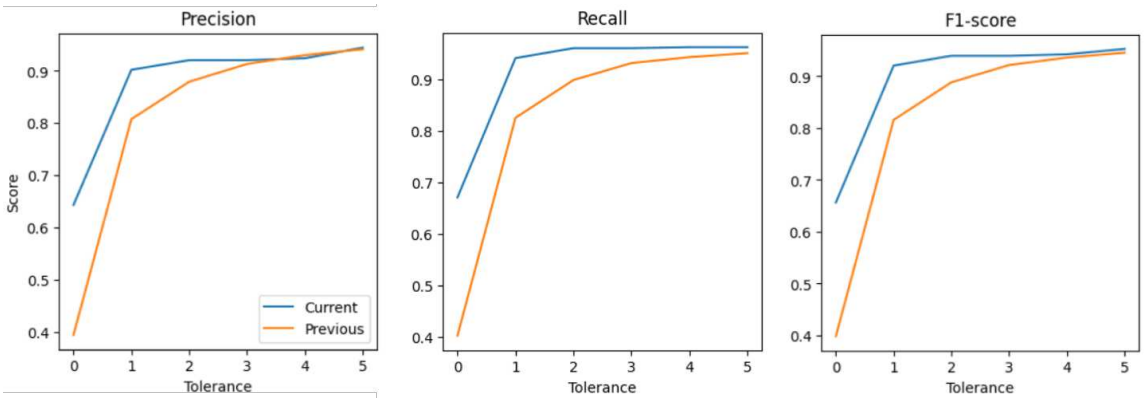

Figure 5.1: Proposed and previous architectures evaluated on the manually annotated dataset.

Since breathing frequency is a common behavioural trait of interest for researchers conducting olfaction experiments, the number of annotated and predicted inhalation onsets is compared for each trail in the test set (5.2).

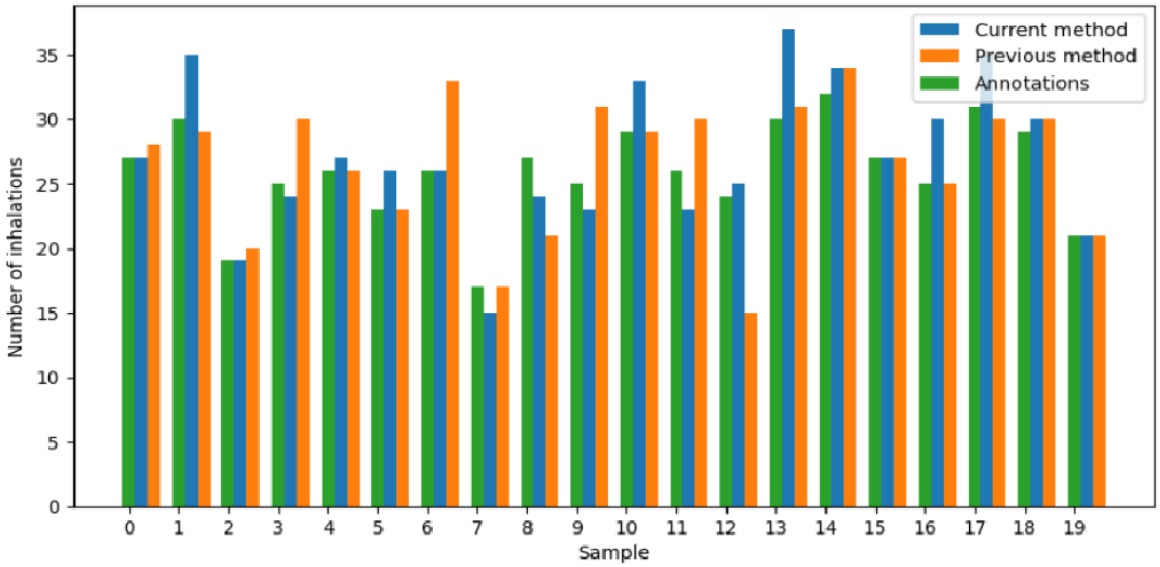

Figure 5.2: A histogram comparing the number of inhalation onsets extracted by the proposed method and the previous method, and the number of manually annotated onsets .

5.2 Evaluation on intranasal cannula pressure sensor signal

Similar to manual annotations, precision, recall and F1 scores are calculated for different tolerances for each mouse of the dataset (Figure 5.3). Although the model show promising results, the quality of the inference varies between mice. The best performance is achieved for Edirne, with the results being slightly lower than the manual annotation test set. The results of other mice are far below the manual annotation test set and the model is not be considered reliable. Importantly, comparisons with the previous method are considered uninformative, since the previous method used this dataset as its training subset.

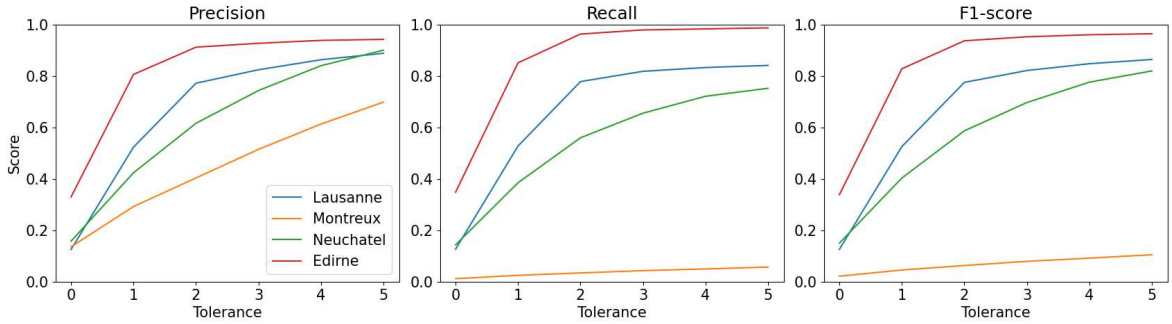

Figure 5.3: Model performance on different mice in the intranasal cannula dataset.

Further work should be done to conclude the exact reason for degraded performance,

but the most likely cause is the difference in IR footage collection and preprocessing methods between the intranasal cannula and manually annotated dataset. Specifically, it remains unclear whether the camera calibration procedures employed during the collection of the two datasets were consistent, and which preprocessing method was used to derive pixel values from the raw IR footage of the intranasal cannula dataset. Either of these can results in videos of significantly different pixel distributions, making them unrecognizable to the DL architecture.

An attempt was made to standardize manually labeled and intranasal cannula dataset pixel distributions by performing Z-score normalization on individual videos (previously, same normalization parameters were used for all videos). The model was retrained with the rest of hyperparameters unchanged, with the evaluation results shown in Figure 5.4. Model performs is slightly better on the previously worst performing mouse, while other mice are almost unchanged. In conclusion, individual Z-score normalization did not solve the problem of poor perfomance on the intranasal cannula dataset.

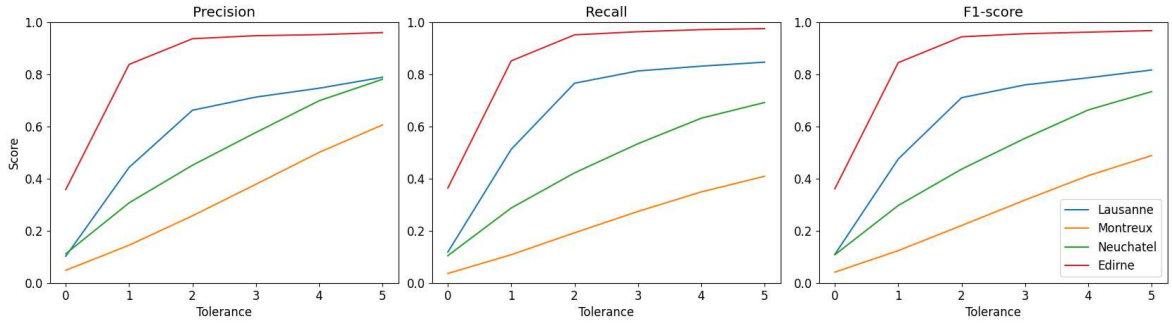

Figure 5.4: Model performance on different mice in the intranasal cannula dataset.

5.3 Error analysis

There are several failure modes of the proposed architecture (Figure 5.5). Firstly, cropping the frames with DLC during preprocessing might produce crops which don't contain the nose. Naturally, it is not possible to extract the breathing signal from such frames. However, most of the times when such a failure occurs, it is caused by obstruction of the nose (e.g. by the paws of the mouse), which is a limitation of the recording method, rather than the DL architecture. Bad croppings are easily detected since DLC provides robust, very high confidence scores for correctly extracted nose keypoints, hence the inference algorithm can track confidence scores and inform the end user of uncertain pre-

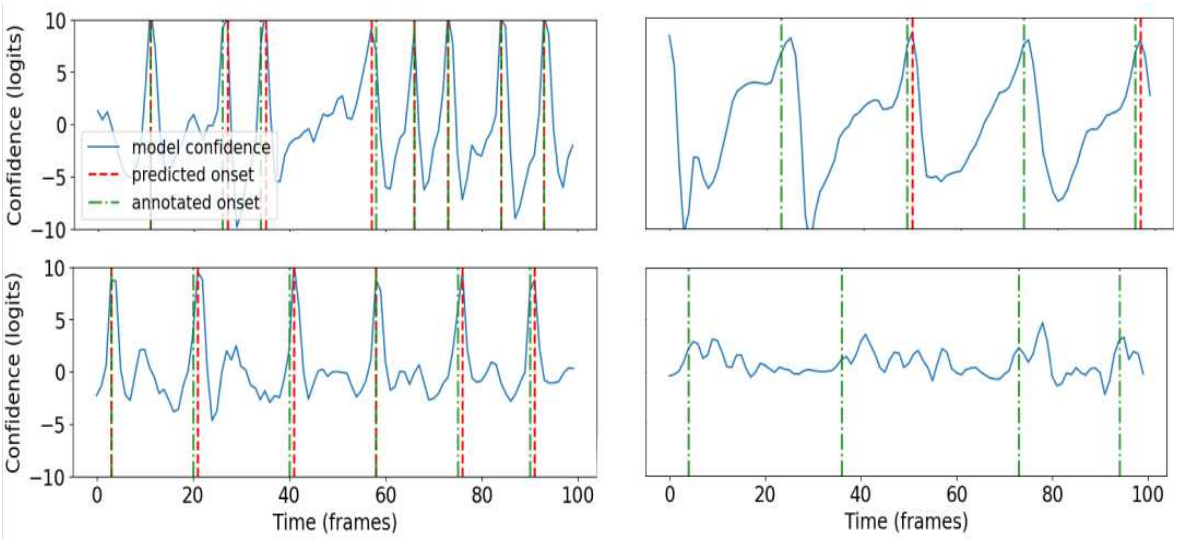

Figure 5.5: Several examples of inference on the manually annotated test set. On the left, two trails with satisfactory predictions are plotted. The top right plot demonstrates a fault in the event detection algorithm. The bottom right plot showcases model outputs which do not correlate with the annotated inhalation onsets.

dictions.

If the croppings are correct, two further failure modes are possible. The model might fail to provide reliable probability scores for inhalation onsets, for which several causes exists. Sometimes the IR camera is not properly calibrated, resulting in blurry footage, which leads to a loss of relevant information for breathing extraction. In this case, the main remedy is ensuring proper camera calibration before starting the experiments. Additionally, different IR cameras might have different color mappings for producing images from temperatures of objects in the camera frame. Again, a potential solution is to properly calibrate the camera. Alternatively, it is possible to manually label several trails for which the model underperforms and retrain the model. Similarly to the errors during cropping, these faults can be automatically reported to the user by tracking output the model output; a model that is uncertain will produce very low logits throught the trail.

If the model produces correct croppings and informative confidence scores, a failure can still happen when the event detection algorithm extracts inhalation onsets.

6 Conclusion and Future work

This thesis approaches extracting mice breathing behaviour from IR camera footage by formulating it as a supervised learning video classification task. We train a ML model to extract inhalation onsets from IR videos, leveraging the recent DL advancements, namely the ViT architecture. The model is trained with a combination of self-supervised pretraining and supervised finetuning on manually annotated inhalation onsets. The framework is compatible with the Three Rs guidelines for animal welfare and is intended to be easily transferable and user-friendly for biomedical science researchers.

The DL architecture is evaluated on the test set of trails with manual annotations, and a smaller dataset of trails which use an intranasal cannula pressure sensor to monitor breathing, considered a reference measurement for breathing behaviour. Although the proposed solution achieves excellent performance on the manually annotated test set, it lacks reliability on the intranasal cannula dataset.

6.1 Future work

Additional work is required to determine why the trained model achieves poor performance on the intranasal cannula dataset. The most likely cause is the difference between collecting and preprocessing IR videos between the two datasets.

To further improve model robustness, performance and speed, several architectural changes can be considered: (i) the ViT transformer backbone can be modified to generate a single embedding for multiple frames, as opposed to generating frame-wise embeddings. This would promote learning of temporal features, relevant for extracting breathing behaviour, (ii) instead of extracting inhalation onset probabilities frame by frame, the model can be modified to produce multiple probabilities simultaneously, thus enabling

training on a wider context, and making each training sample more informative, (iii) the experiments presented in this thesis do not vary the DL architecture size. Reducing the number of parameters will result in a faster model, potentially allowing for real-time inference and also reducing the risk of overfitting.

Even though the pretraining tasks increased the speed of finetuning convergence, it didn't reduce the final validation loss. Other pretraining tasks might produce a backbone better suited for extracting inhalation onsets. Tasks such as next frame prediction or video masked image modelling are considered as promising approaches [31].

References

[1] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis, "Using deeplabcut for 3d markerless pose estimation across species and behaviors," *Nature Protocols*, vol. 14, no. 7, pp. 2152–2176, Jul. 2019. https://doi.org/10.1038/s41596-019-0176-0

[2] E. C. Bryda, "The mighty mouse: The impact of rodents on advances in biomedical research," *Missouri Medicine*, vol. 110, no. 3, pp. 207–11, May-Jun 2013.

[3] J. Grimaud and V. N. Murthy, "How to monitor breathing in laboratory rodents: a review of the current methods," *Journal of Neurophysiology*, vol. 120, no. 2, pp. 624–632, 2018. https://doi.org/10.1152/jn.00708.2017

[4] R. C. Hubrecht and E. Carter, "The 3rs and humane experimental technique: Implementing change," *Animals (Basel)*, vol. 9, no. 10, p. 754, Sep 30 2019. https://doi.org/10.3390/ani9100754

[5] D. W. Wesson, T. N. Donahou, M. O. Johnson, and M. Wachowiak, "Sniffing behavior of mice during performance in odor-guided tasks," *Chemical Senses*, vol. 33, no. 7, pp. 581–96, Sep. 2008. https://doi.org/10.1093/chemse/bjn029

[6] A. G. Khan, M. Sarangi, and U. S. Bhalla, "Rats track odour trails accurately using a multi-layered strategy with near-optimal sampling," *Nature Communications*, vol. 3, no. 1, p. 703, 2012. https://doi.org/10.1038/ncomms1712

[7] J. Esquivelzeta Rabell and S. Haesler, "Probing olfaction in space and time," *Neuron*, vol. 108, no. 2, pp. 228–230, 2020. https://doi.org/https://doi.org/10.1016/j.neuron.2020.10.007

[8] D. D. Cox and T. Dean, “Neural networks and neuroscience-inspired computer vision,” *Current Biology*, vol. 24, no. 18, pp. R921–R929, 2014. https://doi.org/https://doi.org/10.1016/j.cub.2014.08.026

[9] W. Abdulla, “Splash of color: Instance segmentation with mask r-cnn and tensorflow explained by building a color splash filter,” https://heartbeat.fritz.ai/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-explained-by-building-a-657558fe9471, 2024, accessed July 31, 2024.

[10] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, “Text-to-image diffusion models in generative ai: A survey,” 2023. [Online]. Available: https://arxiv.org/abs/2303.07909

[11] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh, “Video generation models as world simulators,” 2024. [Online]. Available: https://openai.com/research/video-generation-models-as-world-simulators

[12] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.

[13] A. Alam, “What is machine learning?” 08 2023. https://doi.org/10.5281/zenodo.8231580

[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[15] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, “Masked autoencoders are scalable vision learners,” *CoRR*, vol. abs/2111.06377, 2021. [Online]. Available: https://arxiv.org/abs/2111.06377

[16] “Convergence of artificial intelligence and neuroscience towards the diagnosis of neurological disorders-a scoping review.”

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017.

[Online]. Available: http://arxiv.org/abs/1706.03762

[18] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, and W. Pedrycz, "A comprehensive survey on applications of transformers for deep learning tasks," 2023. [Online]. Available: https://arxiv.org/abs/2306.07303

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[20] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, "A review on weight initialization strategies for neural networks," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 291–322, 2022. https://doi.org/10.1007/s10462-021-10033-z

[21] H. Bao, L. Dong, and F. Wei, "Beit: BERT pre-training of image transformers," *CoRR*, vol. abs/2106.08254, 2021. [Online]. Available: https://arxiv.org/abs/2106.08254

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] CVAT.ai Corporation, “Computer Vision Annotation Tool (CVAT),” Nov. 2023. [Online]. Available: https://github.com/cvat-ai/cvat

[27] K. Mutlu, J. E. Rabell, P. Martin del Olmo, and S. Haesler, “Ir thermography-based monitoring of respiration phase without image segmentation,” *Journal of Neuroscience Methods*, vol. 301, pp. 1–8, 2018. https://doi.org/https://doi.org/10.1016/j.jneumeth.2018.02.017

[28] J. Morrens, Çağatay Aydin, A. Janse van Rensburg, J. Esquivelzeta Rabell, and S. Haesler, “Cue-evoked dopamine promotes conditioned responding during learning,” *Neuron*, vol. 106, no. 1, pp. 142–153.e7, 2020. https://doi.org/https://doi.org/10.1016/j.neuron.2020.01.012

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[30] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” *CoRR*, vol. abs/1711.11248, 2017. [Online]. Available: http://arxiv.org/abs/1711.11248

[31] Z. Tong, Y. Song, J. Wang, and L. Wang, “Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training,” 2022. [Online]. Available: https://arxiv.org/abs/2203.12602

Abstract

Breathing Signal Recognition from Thermal Footage

Marko Cvjetko

Video recognition is an important computer vision task with many interesting applications that can improve it research in biology and medicine. This thesis considers the recognition of the physiological functions of laboratory animals in thermal imaging. In the scope of this work, it is necessary to select a framework for automatic differentiation and to familiarize oneself with the libraries for handling tensors and images. To study and briefly describe the existing architectures based on convolutions and attention. Obtain sets of recordings and form subsets for training, validation and testing. Select and adapt a suitable model for the observed application and find hyperparameters using learning and validation procedures. Apply the trained models and evaluate the achieved performance. Suggest directions for future work. Attach to the work the original and executable code of the developed procedures, test sequences and results, along with the necessary ones explanations and documentation. Cite the literature used and indicate the help received.

Keywords: Respiration recording; computer vision; deep learning

Sažetak

Raspoznavanje dišnog signala iz toplinskih snimki

Marko Cvjetko

Raspoznavanje videa važan je zadatak računalnog vida s mnogim zanimljivim primjenama koje mogu pospješiti istraživanja u biologiji i medicini. Ovaj rad razmatra raspoznavanje fizoloških funkcija laboratorijskih životinja u toplinskim snimkama. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće diskriminativne arhitekture utemeljene na konvolucijama i pažnji. Pribaviti skupove snimki te oblikovati podskupove za učenje, validaciju i testiranje. Odabrati i prilagoditi prikladan model za promatranu primjenu te uhodati postupke učenja i validiranja hiperparametara. Primijeniti naučene modele te prikazati i ocijeniti postignutu točnost. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Ključne riječi: Mjerenje respiracije; računalni vid; duboko učenje