SAPIENZA
UNIVERSITÀ DI ROMA

Leveraging Self-attention mechanism in Deep Learning for Alcohol Intoxication Detection and Drunkenness Identification

Faculty of Information Engineering, Informatics and Statistics

Corso di Laurea Magistrale in Statistical Methods and Applications

Candidate

Yelizaveta Falkouskaya

ID number 2009414

Thesis Advisor

Prof. Ioannis Chatzigiannakis

Academic Year 2023/2024

Thesis defended on 30 May 2024
in front of a Board of Examiners composed by:

Prof. Luca Tardella (chairman)
Prof. Marco Alfo'
Prof. Pierpaolo Brutti
Prof. Ioannis Chatzigiannakis
Prof. Fulvio De Santis
Prof. Umberto Ferraro Petrillo
Prof. Giovanna Jona Lasinio

Leveraging Self-attention mechanism in Deep Learning for Alcohol Intoxication Detection and Drunkenness Identification
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: y.falkouskaya@gmail.com

Abstract

Alcohol poses a significant threat, with approximately 25% of all road deaths in Europe being alcohol-related [18]. Traditional methods to detect intoxication, such as field sobriety tests, breathalyzer tests, and blood alcohol concentration (BAC) tests, are commonly used but have inherent limitations and specific requirements. This thesis aims to develop an innovative, non-invasive method to identify individuals under the influence of alcohol using a Deep Learning approach. Our proposed Convolutional Attention network surpasses current state-of-the-art methods in both thermal and visible data analysis, and it also provides the capability to visualise attention masks.

Contents

Chapter 1

Introduction

1.1 Overall problem

Alcohol intoxication poses a serious threat to individuals' health and well-being, impairing cognitive and motor functions and increasing the risk of accidents, injuries, and fatalities. Additionally, it may jeopardise public safety by contributing to disruptive behaviours, violence, and disturbances in various settings, ranging from public events and entertainment venues to transportation hubs.

Traditional approaches to identify drunkenness typically include field sobriety tests, breathalyser tests, and blood alcohol concentration (BAC) tests. Each of them are widely employed however come with their inherent limitations and requirements. For example, a breathalyser test may produce false positives or negatives under certain conditions such as an individual's metabolism, medical conditions, or the presence of substances other than alcohol in the breath [5]. Additionally, these tests require specialised equipment and trained personnel, making them less practical for immediate and widespread application in various settings. With the advancements in technology and research, new, non-intrusive methods to discriminate drunk people became possible. These methods rely solely on observable features such as facial expressions, speech patterns, or body movements.

Accurate and non-invasive techniques to detect alcohol-intoxicated individuals can enhance safety and security across various domains such as entertainment venues, customs and border controls, DUI prevention in cars and beyond.

1.2 The approach and goals of the thesis

The primary objective of this thesis is to develop an innovative and non-invasive method to identify individuals under the influence of alcohol, using the Deep Learning approach. Specifically, the focus will be on analysing and interpreting facial images as a key indicator of alcohol intoxication. The thesis aims to advance the existing knowledge base by exploring cutting-edge Deep Learning techniques tailored to the unique challenges posed by this context.

To achieve this goal, the research will employ diverse publicly available datasets encompassing both RGB (Red, Green, Blue) and LWIR (Long-Wave Infrared) imaging. The inclusion of multiple types of imaging data allows to enhance the versatility and robustness of the final model across various environmental conditions. While RGB images provide detailed colour information, LWIR imaging, sensitive to thermal emissions, can offer valuable insights not captured by visible light.

Building upon the foundation of current state-of-the-art models in the context of alcohol intoxication detection, the research will systematically investigate and analyse their efficacy, delving into their methodologies and assessing their strengths and limitations. Furthermore, there exists a deliberate intention to refine and improve these models, pushing the boundaries of their capabilities to achieve a novel solution that surpasses the current standards. Given the relative novelty and limited scope of research in this field, the thesis will delve into the application of the most modern deep learning models, exploring whether their incorporation contributes to more accurate and reliable outcomes in the context of alcohol intoxication detection.

1.3 Structure of the thesis

This thesis is structured to provide a comprehensive exploration and analysis of the methodologies employed in alcohol intoxication detection. The chapters are organised as follows. In Chapter 2, we conduct an in-depth review of the current state-of-the-art research in the field of alcohol intoxication detection using facial images. We critically examine existing literature, methodologies, and findings to establish a foundation for our further exploration. Chapter 3 provides an overview of available data, including its characteristics, collection methods, and preprocessing techniques. Chapter 4 involves the practical implementation of a vanilla Convolutional Neural Network (CNN), which is identified as a promising approach in alcohol intoxication detection based on the literature review. This chapter serves as a vital link between theoretical knowledge and practical application, allowing for the assessment of real-world effectiveness. In Chapter 5 a novel method for classification of alcohol individuals, in particular Convolutional Attention Networks, is introduced and applied to the used datasets to assess its potential in improving the accuracy and efficiency of alcohol intoxication detection using facial images. Chapter 6 is dedicated to the comparison of the implemented models in terms of accuracy, time consumption and applicability. We aim to identify the strengths and limitations of each model, providing a foundation for possible improvement. Finally, Chapter 7 contains the findings and conclusions made during the project, as well as possible future objectives.

Chapter 2

Background

Many scholarly works investigate the identification of drunkenness for various purposes, such as preventing driving under influence (DUI) and conducting fitness-for-duty (FFD) tests. Seeking alternatives to conventional direct measurement techniques like blood tests and breath analysis, researchers in this field leverage a variety of observable data sources, including photo and video recordings capturing facial expressions and eye movements, audio recordings capturing speech nuances, and behavioural data reflecting driving and walking patterns. Within this context, this chapter provides an overview of various statistical and Deep Learning state-of-the-art approaches to identify individuals under the influence of alcohol.

2.1 Face image analysis

Alcohol can manifest on a person's face through various physiological and behavioural changes [11][59]. These changes may be subtle or more pronounced, depending on factors such as the amount of alcohol consumed, the individual's tolerance, and other physiological variables. Some of the physical reactions include flushed or reddened skin due to dilated blood vessels, noticeable changes in pupil size reflecting autonomic nervous system effects, bloodshot eyes from vasodilation, and facial swelling caused by alcohol's diuretic properties leading to dehydration. Additionally, alcohol impacts facial muscle control, resulting in subtle changes in expressions, and its central nervous system depressant effects induce drowsiness and fatigue, evident through drooping eyelids and slowed movements.

A face image of an individual can be obtained using various types of sensors, including visible light cameras (RGB), near-infrared (NIR) and long-infrared (LWIR) sensors. Each of them operates in its own range (Figure 2.1) and comes with its own benefits and limitations. The choice between Thermal, RGB and NIR imaging for drunkenness detection depends on the specific goals, available resources, and the characteristics of the target application.

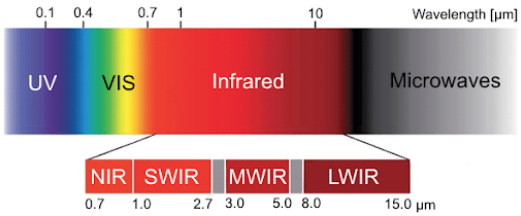

Figure 2.1. Wavelength spectrums of ultra-violet, visible and infrared light [21]

2.1.1 RGB imaging

One of the widely utilised and commercially accessible camera sensors is the visible light camera, which operates within the 400-700 nm range corresponding to the spectrum observable by the human eye. Visible cameras employ red, green, and blue wavelengths (RGB) to capture light, facilitating accurate colour representation and providing a detailed representation of facial features. This can be valuable for identifying subtle changes in skin tone, eye colour, or facial expressions that may be indicative of drunkenness. The colour information in RGB images is human-interpretable, making it easier for researchers and developers to understand and interpret the features learned by the deep learning model. Due to its popularity, RGB images allow for training on diverse datasets that include variations in lighting conditions, backgrounds, and facial expressions, and the integration with existing computer vision techniques and the availability of pre-trained models allows for more efficient and less resource intensive model development. However, it is essential to acknowledge the limitations of RGB imaging, particularly its restriction to daytime and clear sky conditions. Visible cameras rely on ambient light for operation, rendering them ineffective in low-light situations, such as in-vehicle sensors when driving at night. Additionally, atmospheric conditions such as fog, haze, smoke, heat waves, and smog can significantly impair their performance.

The topic of drunkenness detection from RGB images and video has been widely studied by the scientific community. Researchers have explored various methods to classify intoxicated individuals, often leveraging publicly available sources like YouTube videos [65] or publicly accessible photos, comes with inherent challenges. The reliability of the labels is questionable, as there is no guarantee that a video labeled as depicting a drunk person actually contains such content.

In 2018, Mehta et. al [65] introduced a new dataset called *DIF* (Dataset of perceived Intoxicated Faces) which contains audio-visual data of intoxicated and sober people obtained from online sources. The authors tested various models including CNN (Convolutional Neural Network) - RNN (Recurrent Neural Network), 3D CNN, audio-based LSTM (Long Short-Term Memory), and DNN (Deep Neural Network), both on separate audio and video channels and combined. The work reports accuracy of 76.37% using the VGG-LSTM model and 88.39% when coupled with audio materials.

This dataset has been extensively used in research. For example, Kamath et al. [31] propose Graph Neural Networks using facial landmarks for alcohol intoxication detection. In this method, the images are converted to graphs, where nodes represent facial features. Then, a network of Graph Convolutional Layers is trained, achieving promising training accuracy of 86.69 ± 2.29 with validation and testing accuracy of 86.73 ± 2.6 and 86.4 ± 2.4 respectively.

In *"DrunkSelfie: Intoxication Detection from Smartphone Facial Images"* [64] researchers introduce an Android application, *DrunkSelfie* (Figure 2.2), that estimates the subject's drunkenness from a selfie. The model utilises a photo series capturing people before and after several glasses of wine. It utilises facial landmark vectors as features and reports 81% accuracy using Gradient Boosted Machines for classifying subjects as either "sober" (0 or 1 glasses of wine) or "drunk" (2 or 3 glasses of wine).

Another end-to-end system detecting the alcohol content was proposed by Rachakonda et al. [48] Their IoT enabled edge device *Donot-DUEye* uses pupil dilation, raise in blood pressure and eye redness level to analyse the subject's driving capability. The authors utilise SSD MobileNET for classification and achieve the accuracy of approximately 95%. However, this type of classifier requires the ground truth information, making it inapplicable in the situations where such information

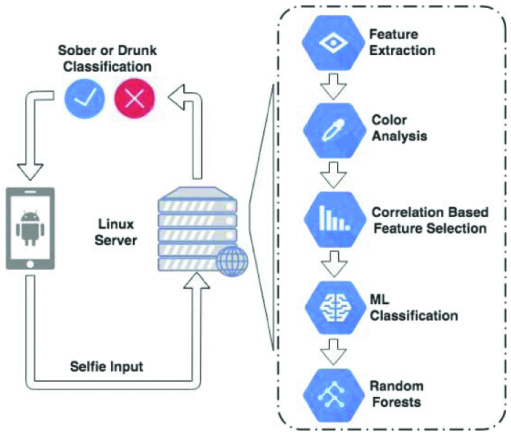

Figure 2.2. Architecture of DrunkSelfie Android App [64]

is unavailable.

Lastly, Chang et al. [13] proposed dividing the subjects into three distinct age brackets (18–30, 31–50, and >50 years), presuming varying facial expression changes across age groups.Their neural network employed a simplified VGG network in the first stage to determine the subject's age range, while the second stage utilised the simplified Dense-Net to identify facial features indicative of drunk driving. In drunk driving recognition tests among the three age groups, accuracies of 94%, 83%, and 81% were obtained, respectively, supporting their theory that age influences how facial changes associated with alcohol intoxication manifest.

2.1.2 Thermal imaging

Long-wave infrared or LWIR is a subset of the infrared band of the electromagnetic spectrum, covering the wavelengths ranging from 8 µm to 14µm (8,000 to 14,000nm). Using thermal imaging for deep learning-based drunkenness identification models can offer several advantages over RGB and NIR imaging due to its ability to capture and visualise temperature variations associated with physiological changes due to alcohol consumption. Unlike visible light, thermal imaging is unobtrusive and invariant to lighting conditions, operating effectively in low-light or darkness, making it suitable for discreet monitoring. Thermal cameras are less dependent on facial expressions and less susceptible to disguises, makeup, or facial changes, primarily relying on temperature differences. While thermal imaging has these advantages, it's essential to consider its limitations, such as potential challenges in capturing fine details, lower spatial resolution compared to RGB, inability to see through some materials (Figure 2.3) and the need for specialised hardware. The effectiveness of thermal imaging for drunkenness identification will depend on the specific characteristics of the dataset and the features associated with intoxication.

Recent research has been actively exploring ways to study and categorise people under the influence of alcohol using thermal imaging. This interest is largely due to the availability of a unique dataset gathered at the University of Patras [36] specifically focusing on sober and drunk individuals. The dataset provides a valuable resource for scientists to investigate the thermal changes associated with different levels of alcohol consumption.

Georgia Koukiou, one of the creators of the dataset and a prominent researcher in the field of drunk person classification using thermal imaging, has made significant

Figure 2.3. Long Infrared and visible imagery of a subject [17]

contributions to this area since 2009. Throughout her career, she has explored various methods and strategies to improve the accuracy and reliability of detecting intoxicated individuals. Koukiou's research has examined a range of aspects, including optimal locations for drunkenness identification [32], patterns of eye temperature distribution [38], thermal signatures of the eyes [39], blood vessels activity [33], local difference patterns [40], and dissimilarity features [41]. She has experimented with a variety of approaches, beginning with simple neural networks and progressing to more complex models.

In one of her most recent works, *"Intoxicated Person Identification Using Markov Chains and Neural Networks"* [34], Koukiou proposed an innovative method that utilizes Markov chains to model the statistical behavior in forehead images. This approach combined with neural networks resulted in high success rates for identifying intoxicated individuals. Her extensive research and diverse methodologies have significantly advanced the field of thermal-based drunk person classification.

In 2018, Kamath et al. [30] explored intoxicated individual identification using Convolutional Neural Networks on body parts like hands, ears, and face, with facial profiles yielding the highest accuracy (82.16%) and hands the lowest (75.55%). Combining multiple body parts via a Backpropagation Neural Network (BPNN) improved accuracy. The research *"Detection of Intoxicated Person using Thermal Infrared Images"* [7] employed Non Subsampled Contourlet Transform (NSCT) to extract texture features from eye, face, hand, and ear thermal images. Support Vector Machine classification achieved impressive accuracies: 93% for eyes, 94.1% for face, 95.1% for hands, and 100% for ears.

"Drunkenness Diagnosis Using a Neural Network-Based Approach" (2017) [45] introduced a model with four stages: thermal infrared image acquisition, Pulse-Coupled Neural Network (PCNN) segmentation, feature selection via Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), and Support Vector Machine (SVM) classification, achieving a 97.5% detection score. In 2020, using the same database the researchers tested an ensemble of CNNs, attaining a 95.75% detection score.

"Deep Learning Technology for Drunks Detection with Infrared Camera" [28] (2020) proposed a handheld IR-based detection system operating on an iPhone operating system (iOS) mobile phone with 85.10% accuracy for multi-level classification (sober, 1 glass, 2 glasses, or 3 glasses) and 74.07% for binary identification. In *"Facial Thermal Image Analysis with Deep Convolutional Neural Network Architectures for Subject Dependent Drunkenness Diagnosis"* (2021) [55], three different Deep Convolutional Neural Network architectures were compared, with GoogLeNet achieving the highest accuracy. *"Drunkenness Detection Using a CNN with adding*

Gaussian Noise and Blur in the Thermal Infrared Images" (2022) [27] applied a CNN model to thermal infrared images with added noise and blur, achieving 93% accuracy. "Identification of Drunk People Among Crowds Using Thermography and Machine Learning" [49] compared Support Vector Machine and transfer learning approaches. "Face Recognition and Drunk Classification Using Infrared Face Images" [61] employed Weber local descriptor (WLD) and local binary pattern (LBP) for feature extraction, achieving 100% performance in face recognition and 86.96% in drunk identification. *"Thermal Biometric Features for Drunk Person Identification Using Multi-Frame Imagery"* [35] utilised multi-frame thermal imagery and morphological operations for feature extraction, achieving over 86% success rate. Lastly, *"Prediction Model of Alcohol Intoxication from Facial Temperature Dynamics Based on K-Means Clustering Driven by Evolutionary Computing"* [42] proposed a segmentation model based on clustering algorithms and evolutionary optimization, allowing for classification of facial temperature areas into segmentation classes for tracking alcohol-temperature features during intoxication.

2.1.3 NIR imaging

Near-Infrared Light (NIR) is a subset of electromagnetic radiation (EMR) wavelengths nearest to the range of the naked eye but just past what we can see. It covers the wavelengths ranging from 0.7 to 1.4 microns and sometimes offers clearer details than what is achievable with visible light imaging (Figure 2.4). NIR is very close to human vision but removes the colour wavelengths, which results in most objects looking very similar to an image that has been converted to black and white.

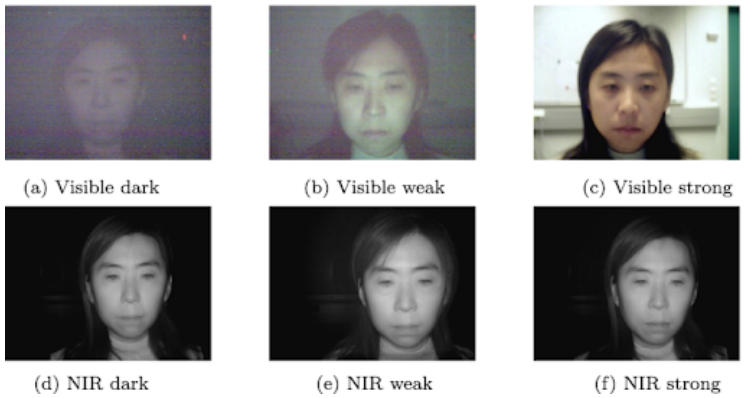

Figure 2.4. Near Infrared imagery compared to visible light imagery [51]

Using Near-Infrared (NIR) imaging can be an alternative or complementary approach for drunkenness detection since it offers its unique advantages over RGB imaging. Due to its greater penetration depth NIR imaging can capture information from deeper tissue layers, potentially revealing additional physiological changes associated with intoxication that are not visible in the RGB spectrum. Additionally, NIR imaging is often less sensitive to variations in lighting conditions compared to RGB imaging making it more robust in different environments, which can be beneficial for real-world applications where lighting conditions may vary. Finally, longer wavelengths of the NIR spectrum are able to penetrate haze, light fog, smoke and other atmospheric conditions better than visible light making it perfect for long-distance imaging. Despite the advantages, NIR imaging comes with its own drawbacks and challenges. Since NIR detects light between 700nm to 1400nm, we're

seeing information beyond the wavelengths that make up colour, so the image is represented in grayscale values. The interpretability of NIR images may be more challenging for humans, as the information captured is not visible to the naked eye. Additional considerations include the need for specialised hardware, potential costs, and the necessity for a dataset that includes NIR images for training and evaluation.

The existing research on the detection of drunkenness using Near-Infrared imaging faces limitations due to the scarcity of publicly available data and the challenges associated with independent data collection. Many studies within this domain predominantly concentrate on NIR images of the eye.

For instance, Juan Tapia Farias and colleagues conducted experiments aimed at classifying individuals deemed unfit for duty. Their work, *"Behavioural Curves Analysis Using Near-Infrared-Iris Image Sequences,"* [12] involves analysing a stream of Near-Infrared (NIR) iris video frames to examine the effects of external factors such as alcohol, drugs, and sleepiness on pupil and iris movements. To facilitate their research, they developed the "FFD NIR iris images Sequences database" (FFD-NIR-Seq), which contains sequences of periocular NIR images captured in four conditions: control, under the influence of alcohol, under the influence of drugs, and drowsiness (Figure 2.5). They utilised pupil-iris-ratio as a feature vector for each sequence and employed standard machine learning algorithms, including Random Forest (RF), Gradient Boosting Machine (GBM), and Multi-Layer Perceptron (MLP) Neural Network, to classify both the four states and fit/unfit states. Their results showed that the Multi-Layer Perceptron and Gradient Boosted Machine achieved the best results in all groups, with an overall accuracy of 74.0% and 75.5% for the Fit and Unfit classes, respectively.

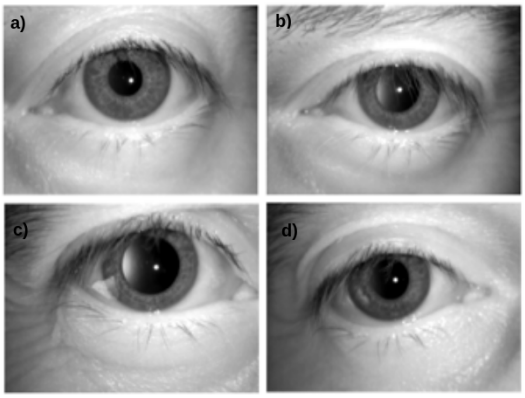

Figure 2.5. Examples of the NIR images captured. a) Control, b) Alcohol, c) Drug, and d) Sleep images [12]

In their subsequent work titled *"Learning to Predict Fitness for Duty using Near Infrared Periocular Iris Images,"* [57] they proposed a modified MobileNetV2 to classify iris NIR images taken from subjects under the influence of alcohol/drugs/sleepiness. Their results demonstrated that the MobileNetV2-based classifier could robustly detect the Unfit alertness condition from iris samples captured after alcohol and drug consumption, achieving detection accuracies of 91.3% and 99.1%, respectively. However, the sleepiness condition proved to be the most challenging, with an accuracy of 72.4%. For two-class grouped images belonging to the Fit/Unfit classes, the model achieved accuracies of 94.0% and 84.0%, respectively, while utilising a smaller number of parameters than the standard Deep Learning Network algorithm.

In *"Alcohol Consumption Detection from Periocular NIR Images Using Capsule Network,"* [58] a novel Fused Capsule Network (F-CapsNet) was proposed to classify iris NIR images taken from subjects under the influence of alcohol consumption. They introduced a new database called IAL-I, which includes NIR periocular images captured from 30 volunteers under the influence of alcohol. Their results showed that the F-CapsNet algorithm could detect alcohol consumption in iris NIR images with an accuracy of 92.3%, while using half the parameters as the standard Capsule Network algorithm.

Finally, in their latest work titled *"Fitness-for-Duty Classification using Temporal Sequences of Iris Periocular images,"* [67] the researchers attempted to leverage not only spatial information but also temporal information. They classified Fitness-for-Duty using sequences of 8 iris images using Convolutional Neural Networks based on VGG-16 and Long Short Term Memory Networks (LSTM). The model delivered better results compared to classic machine learning algorithms, achieving an overall accuracy of 83.6%.

2.2 Alternative data modalities

In addition to facial image data, various other data types have been explored in efforts to identify intoxicated individuals, including speech [52][9][10], texts [4][29][44], gait [43][47], driving patterns [66][8] and so on.

Identification of drunkenness through speech falls within the domain of paralinguistics, which started gaining popularity in the research community after the *"INTER-SPEECH 2011 Speaker State Challenge"* by Schuller et al. [52] This challenge included a subtask focused on predicting speakers' intoxication levels using the "Alcohol Language Corpus," a dataset comprising 162 speakers aged 21–75 from five different locations in Germany. For example, Biadsy et al. [9] conducted a series of experiments discovering that phonotactic features extracted from phone durations play a significant role in classification. Their approach, based on the hypothesis that certain phonemes manifest differently in intoxicated and sober speakers, employs Support Vector Machines with a kernel function computing similarities between adapted phone Gaussian Mixture Models, which encapsulate speakers' phonetic characteristics in their utterances. Bonela et al. [10] present their Audio-based Deep Learning Algorithm to Identify Alcohol Inebriation (ADLAIA), which utilises mel spectrograms derived from 12-second audio clips as input. Mel spectrograms visualise the spectrum of a signal, mapping frequency bands to the mel scale, approximating human auditory perception. These spectrograms are then fed into a pretrained ResNet-18 convolutional neural network. ADLAIA demonstrates superior performance, particularly on subjects with higher Blood Alcohol Concentration (BAC) levels, supporting the hypothesis of increasing impairment of cognitive and psychomotor abilities with rising BAC levels, as previously suggested by Brumback et al. in 2007 [11].

Drunk texting classification has been explored by several studies. These works typically employ traditional feature extraction techniques for text classification, encompassing tasks like tokenization, stopword removal, normalisation of mentions/URLs, along with stylistic features such as discourse connectors, word count, and capital letter frequency, alongside sentiment analysis. Various classification algorithms are experimented with, including Random Forest, Support Vector Machine, Generalised Linear Model (GLM), and Naive Bayes. For instance, Joshi et al. [29] proposed a pipeline utilising n-grams in Tweets and achieved a classification accuracy of 78.1% for drunk Tweets. Meanwhile, Maity et al. [44] analysed Twitter

profiles to differentiate drunk and sober users based on their Tweets. Additionally, Grzeca et al. [20] explored integrating semantic enrichment and word embeddings to identify drunk texting in short, noisy text pieces like tweets, although they faced challenges due to limited dataset size.

The methods mentioned above show promise for identifying alcohol intoxication. However, our focus is on passive measurements that don't rely on active participation, such as gait, cognitive tests, or speaking. This broader applicability leads us to choose facial image data for our work.

Chapter 3

Datasets

The success of any deep learning model heavily depends on the quality and quantity of the data used to train them. Quality data helps better model training and generalisation, can reduce overfitting, enhances model robustness and helps more accurate evaluation. This chapter discusses the datasets used in this study, detailing the collection and description processes, data preprocessing techniques, and exploratory data analysis. In section 3.1, we explore the methods used to collect the datasets and provide a comprehensive description of the data, including key attributes and characteristics. This section outlines the source of the data, the collection process, and any special considerations taken into account during data acquisition. Section 3.2 delves into the data preprocessing techniques applied to prepare the datasets for analysis and modeling. This includes cleaning, normalization, and other transformations to ensure data consistency and quality. We also discuss specific challenges encountered during preprocessing and how they were addressed to maintain data integrity. Section 3.3 focuses on exploratory data analysis and outlier detection. This section describes the techniques used to understand the underlying structure of the data, identify patterns, and detect anomalies. Methods such as clustering, visualization, and statistical analysis are employed to gain insights into the datasets, allowing for more informed decision-making in subsequent analysis and modeling stages.

3.1 Collection and description

Following an extensive literature review, we compiled a list of potentially usable datasets for this study. We first checked which of these datasets were available online. For those that were not publicly accessible but seemed promising, we reached out to the authors to request access. While some did not respond, others informed us that the data could not be shared due to privacy policies. Luckily, a few authors agreed to share their datasets with us.

Unfortunately, despite our efforts, we were unable to obtain any datasets containing near-infrared images of sober and intoxicated individuals. As a result, we will not be able to include this domain in our tests. Nonetheless, we have a selection of datasets that will allow us to conduct a comprehensive study. The list of the available datasets alongside with its features can be found in the Table 3.1.

Dataset Name	Domain	Subjects	Ethnicity	Controlled Environment
Sober-Drunk Database [36]	LWIR	42	Greece	Yes
PUCV-DTF [23]	LWIR	46	Chile	Yes
Kubicek et. al [42]	LWIR	20	Czechia	Yes
DIF [65]	RGB	166	Mixed	No
3 Glasses After [3]	RGB	53	Brazilian	Yes

Table 3.1. Available datasets

Sober-Drunk Database (Figure 3.1)

The most popular and publicly available dataset is the Sober-Drunk database collected at the University of Patras. The database includes the facial image data collected from 41 individuals, comprising 31 males and 10 females. The subjects were recorded using the ThermoVision Micron/A10 Model infrared FLIR camera in two distinct states: sober and 30 minutes after consuming four glasses of wine (62.4 ml of alcohol). Each recording consists of 50 sequential frames of the same object, captured at intervals of 100 milliseconds, resulting in a total acquisition time of 5 seconds for all 50 frames. Consequently, each person has a set of 100 frames recorded in two sessions, leading to a comprehensive database of 4100 images. The infrared images have a resolution of 128 x 160 pixels.

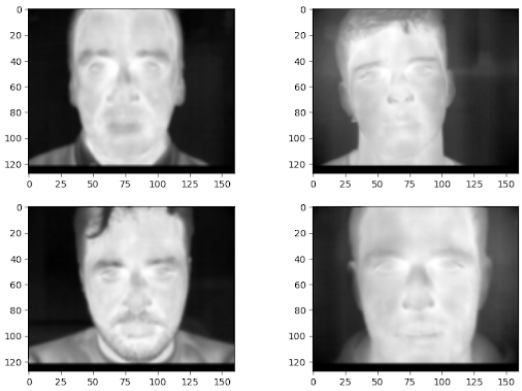

Figure 3.1. Examples of subjects from the Sober-Drunk Database

Pontificia Universidad Católica de Valparaíso-Drunk Thermal Face database (PUCV-DTF) (Figure 3.2)

A total of 46 individuals, consisting of 40 men and 6 women, participated in this study. Their average age was 24 years, with a standard deviation of approximately 3 years; the age range was 18 to 29 years. All participants were in good health with no history of alcohol-related issues, as confirmed by a screening test that excluded regular alcohol consumers.

Before the experiment, subjects rested for 30 minutes in the robotics lab to acclimate to the temperature conditions. The procedure involved drinking a 355 mL can of 5.5° beer, followed by a 30-minute rest period. This cycle was repeated four times, after which subjects with a blood alcohol level of around 0.8 g/L were considered "drunk." They remained in the lab until their blood alcohol level dropped

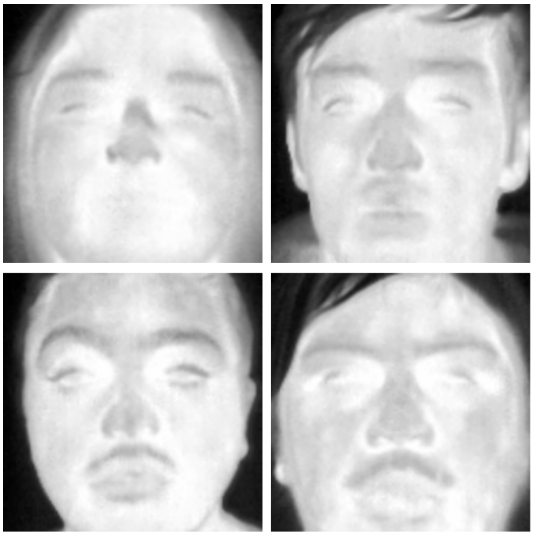

Figure 3.2. Examples of subjects from the PUCV-DTF database

below 0.2 g/L, verified by a breath test. A paramedic was present throughout the experiment to ensure participant safety.

The thermal camera used to capture the data was a FLIR TAU 2, with a resolution of 640 × 480 pixels, a framerate of 30 frames per second, thermal sensitivity of 50 mK, and a spectrum range of 7.5 to 13.5 μm.

The dataset comprised 250 images per subject, across five subsets, each containing 50 images. These subsets were labeled as "Sober," "1 Beer," "2 Beers," "3 Beers," and "4 Beers," reflecting the number of beers consumed during the experiment.

Kubicek et. al (Figure 3.3)

This study involved 20 volunteers to measure alcohol intoxication using infrared (IR) imaging. During the experiment, six images were captured for each participant to track the dynamic progression of alcohol intoxication. The first IR image represented the sober state, and the subsequent images reflected increasing levels of alcohol consumption. The measurements were conducted at the Faculty of Safety Engineering at the Technical University of Ostrava. The measuring room was air-conditioned, with controlled conditions to ensure consistency. The IR camera was positioned two meters from each participant, and the data were exported in a .mat format for further analysis using MATLAB. Participants were instructed not to consume any food before the experiment. After 30 minutes of rest to stabilize from outdoor temperature, participants consumed 38% alcohol in 0.04L doses, with 30-minute intervals between each dose. Blood pressure was measured before the start, and breath alcohol levels were monitored using a Dräger Alcotest® 7510. The experiment had ethical approval, and all participants signed informed consent forms in Czech to allow the publication of their data and acknowledge the effects of alcohol consumption. A physician was present throughout the experiment to ensure safety.

The IR images were captured using a FLIR T640 camera with a resolution of 640 × 480 pixels and thermal sensitivity (NETD) of less than 0.035 °C. The IR camera was calibrated with a Flat Field Correction (FFC), performed during startup and as the camera changed temperature. The background for the IR images was white stucco plaster, and the emissivity was set at 0.96.

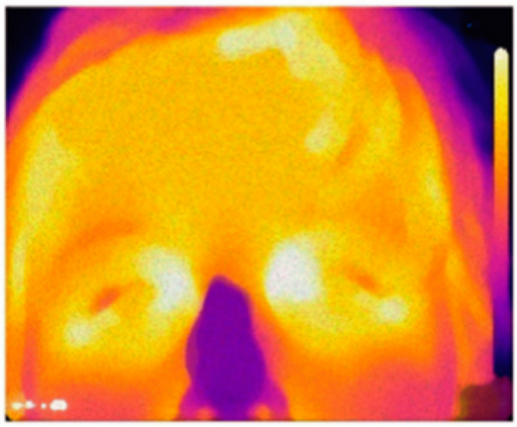

Figure 3.3. An example of data from the Kubicek et. al Database

Due to the need to protect personal data, all records were at least partially anonymised. To achieve this, a median filter with a 10x10 kernel mask was applied. This filtering process has an additional benefit: it creates a smoother temperature color map, where colors blend more seamlessly with each other. However, this approach can distort some of the original temperature information.

Dataset of Perceived Intoxicated Faces (DIF) (Figure 3.4)

The dataset was collected by sourcing videos from social networks, primarily YouTube and Periscope. The selection process involved search queries like "drunk reactions," "drunk review," and "drunk challenge" to gather content featuring intoxicated people. Videos of sober individuals were also collected using similar methods. Due to the nature of these sources, the videos were recorded in real-world, unconstrained settings, providing a broad representation of intoxicated and sober behavior. The DIF dataset contains:

- 91 videos in the drunk category with an average length of 6 minutes. This category consists of 88 unique subjects, including 55 males and 33 females.
- 78 videos in the sober category with an average length of 10 minutes. This category consists of 78 unique subjects, including 35 males and 43 females.

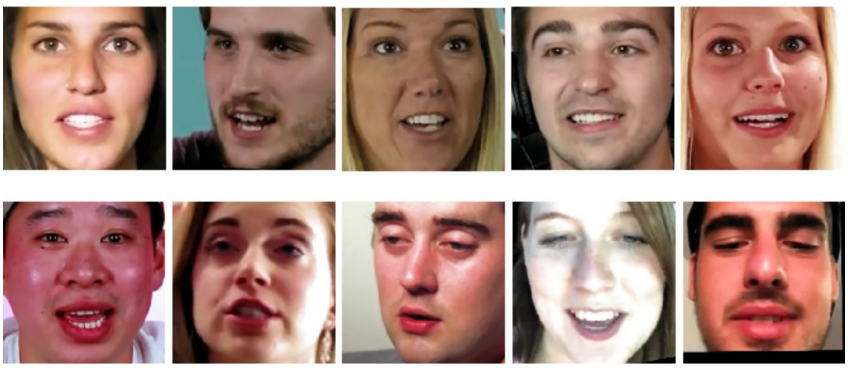

Figure 3.4. Examples of subjects from the DIF database

Given the nature of the video sources, class labels are based on the video title and description. These labels are considered "weak labels" since there's a chance that a subject labeled as "drunk" may not exhibit clear signs of intoxication, or a subject labeled as "sober" may not be entirely devoid of intoxication effects. Moreover, some videos might have been uploaded for their humorous content, adding an additional layer of ambiguity. To process these videos, several steps were followed:

1. Shot Detection and Face Tracking: Using the pyannote-video library, shot detection was performed to isolate individual camera shots. Face detection and tracking were then conducted to identify and follow the faces in each shot.

2. Clustering and Unique Subject IDs: Since some videos contain multiple subjects, clustering was used to extract unique subject identities from the face tracks. This allowed for cropping faces in each frame to create individual data samples.

3. Face Alignment and Standardization: The OpenFace toolkit was employed for face alignment, ensuring a consistent orientation of facial features. The final data samples were fixed at 10 seconds in length.

This comprehensive processing pipeline resulted in a total of 4,658 face videos for the intoxicated category and 1,769 face videos for the sober category, each containing tracked faces of the respective subject.

3 Glasses After (Figure 3.5)

The dataset features photographs by Brazilian photographer Marcos Alberti. The project provides a snapshot of how individuals' expressions and moods evolve with each successive glass of wine. Marcos Alberti set out to test this concept by inviting a diverse group of friends to his studio, with the aim of capturing their changing expressions as they consumed wine. The dataset comprises photographs taken in a controlled studio environment. The process involved:

- Initial Capture: The first photograph was taken as soon as the guests arrived at the studio, capturing their natural state, often displaying signs of stress or fatigue from their daily routines and rush-hour traffic.

- Subsequent Captures: After each glass of wine, another snapshot was taken. Each participant was photographed three times during the course of the evening, with each photo representing a different stage of wine consumption.

- Simple Setting: The photos were taken against a plain wall, focusing solely on the subject's face. The simplicity of the setup allows for a clear view of the subject's expressions without any distractions.

The dataset includes 53 unique subjects from various backgrounds, representing a wide range of professions, such as music, art, fashion, dance, architecture, and advertising. The subjects were photographed across four different states of inebriation, providing a broad spectrum of expressions and reactions to wine consumption.

3.2 Data preprocessing

Preparing the data is an essential step when working with machine and deep learning; this includes tasks like data cleaning and data augmentation. During this phase, it's important to ensure consistency and address any anomalies in the dataset.

Figure 3.5. Examples of data from the 3 Glasses After Database

Using visual inspection and basic Python commands, we checked each image in the dataset for consistent size and format.

For the Sober-Drunk Database, our inspection revealed several issues. Some images had incomplete pixel data, resulting in black lines at the bottom, while others used a 16-bit format instead of the expected 8-bit format. To address these discrepancies, we applied the following corrections:

- Cropping: We removed any black lines or incomplete pixel data by cropping the affected images, ensuring that they had consistent and clean edges.
- Resizing: We adjusted the cropped images to match the dimensions of the rest of the dataset, ensuring that all images were of uniform size.
- Normalizing to 8-bit: Since some images were in 16-bit format, we normalized all images to 8-bit, aligning them with the majority of the dataset.

Normalisation

Normalising image data is a common preprocessing step in machine learning and computer vision, especially when dealing with image classification tasks. It involves adjusting the pixel values in images to a specific range or distribution. This step is crucial in deep learning for several reasons. First, it standardises the scale across images, ensuring consistent input for neural networks, which improves the stability and convergence during training. This is because normalisation addresses large variations in scale and helps the optimization algorithms like gradient descent function more effectively. Additionally, normalisation reduces bias in training by ensuring that no single image or feature disproportionately influences the learning process. It also mitigates sensitivity to variations in factors like illumination and contrast, allowing for better generalisation to new data. Furthermore, when using pretrained models like VGG, ResNet, or Inception, normalisation is essential, as these models were trained on standardised data; failing to normalise can lead to significant performance issues.

There are several methods to perform normalisation, namely:

- **Scaling to a range**: The most common approach is to scale pixel values to a specific range, typically [0,1] or [-1,1]. This can be done by dividing the pixel values by a constant (e.g., 255 for 8-bit images) or by adjusting them to fall within a desired range.
- **Mean and Standard Deviation Normalisation (Z-Score Normalisation)**: This involves subtracting the mean of the dataset and dividing by the standard deviation, resulting in data with a mean of 0 and a standard deviation of 1. This approach is common when there's a known dataset mean and standard deviation (e.g., when using datasets like ImageNet).
- **Histogram Equalization**: A more complex technique that adjusts the pixel intensity distribution to equalise the histogram. This is less common for simple normalisation and is typically used for specific tasks where contrast needs enhancement.

In this work, we are going to utilise z-score normalisation with the well-known formula $z = \frac{x-\mu}{\sigma}$, where x is the original value, μ is the mean of the dataset and σ is the standard deviation of the dataset. To perform z-score normalisation on an image dataset, we need to find the mean and standard deviation for each channel (e.g., Red, Green, Blue in RGB images or for one channel in NIR and LWIR images) across the entire dataset. To calculate the mean for each colour channel across a dataset of images, we need to find the sum of pixel values for each channel, then divide by the total number of pixels.

For each colour channel c, the mean μ_c is calculated as:

$$\mu_c = \frac{1}{N * H * W} \sum_{n=1}^{N} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{n,i,j,c},$$

where:

- N is the total number of images in the dataset.
- H is the height of each image.
- C is the number of colour channels (C=3 for RGB images and C=1 for NIR and LWIR images).
- $x_{n,i,j,c}$ is the pixel value for image n, at row i, column j, in channel c.

Standard deviation is a measure of the dispersion or spread of values around the mean. It requires calculating the variance and then taking the square root.

For each colour channel c, the variance σ_c^2 is calculated as:

$$\sigma_c^2 = \frac{1}{N * H * W} \sum_{n=1}^{N} \sum_{i=1}^{H} \sum_{j=1}^{W} (x_{n,i,j,c} - \mu_c)^2$$

The standard deviation for each channel c, σ_c, is then:

$$\sigma_c = \sqrt{\sigma_c^2}$$

After obtaining the mean and standard deviation for each channel, we can apply z-score normalisation to the dataset. The normalised value for a pixel $x_{n,i,j,c}$ is given by:

$$z_{n,i,j,c} = \frac{x_{n,i,j,c} - \mu_c}{\sigma_c}$$

This ensures that the pixel values for each channel are centred around zero with a standard deviation of one.

Data augmentation

To validate and test the final model, we divided the dataset into two subsets: a training set containing 80% of the data and a validation set containing the remaining 20%. The division was designed to ensure that all data from a given subject was included in only one of these subsets, preventing data leakage between training and validation. Due to the relatively small size of the dataset, we did not create a separate test set.

To increase the effective size of the dataset and introduce more variability, we performed additional data augmentations:

- **Flipping:** Flipping involves creating a mirror image of the original data, effectively doubling the number of samples. This can be done horizontally, vertically, or both. Horizontal flipping is more common for image data augmentation, as it provides a simple yet effective way to simulate variations in image orientation. In the context of our dataset, we assign the probability of $p = 0.5$ of an image to be horizontally flipped.

- **Adding Noise:** Adding noise to images is a common data augmentation technique used in machine learning and deep learning to improve a model's robustness and generalization capabilities. Two common types of noise used in image processing are Gaussian noise and salt-and-pepper noise. In this work, we are going to use Gaussian noise because it introduces smooth random variations, which can mimic real-world imperfections and camera sensor noise. Gaussian noise is generated from a Gaussian (normal) distribution, characterized by its mean and standard deviation. Given an image $\mathbf{I}$, Gaussian noise can be added by superimposing a noise matrix $\mathbf{N}$ on the image, where the elements of $\mathbf{N}$ are drawn from a Gaussian distribution with mean $\mu = 0$ and standard deviation σ:

  $$\mathbf{N} \sim \mathcal{N}(0, \sigma^2).$$

  The noisy image $\mathbf{I}'$ is obtained by adding the noise matrix $\mathbf{N}$ to the original image $\mathbf{I}$:

  $$\mathbf{I}' = \mathbf{I} + \mathbf{N}.$$

  The standard deviation σ controls the intensity of the noise, with larger σ values resulting in more pronounced noise. By training with noisy images, the model learns to extract meaningful patterns despite the noise, which can improve its generalization capabilities.

  An example of an image with Gaussian noise can be seen in Figure 3.6.

- **Blurring:** Gaussian blurring, also known as Gaussian smoothing, is a common image processing technique used to reduce image noise and detail by applying a Gaussian function to smooth out sharp transitions in pixel values. This method is widely used in data augmentation and preprocessing for its ability to retain essential features while eliminating minor irregularities or noise.

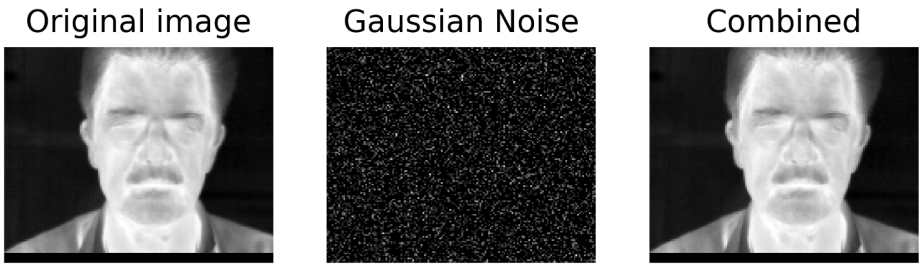

Figure 3.6. Adding Gaussian noise to an image

The Gaussian function used for blurring is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} exp(-\frac{x^2 + y^2}{2\sigma^2}),$$

where:

- x and y represent the horizontal and vertical coordinates, respectively.
- σ is the standard deviation, which determines the spread of the Gaussian distribution.

To apply Gaussian blurring, a Gaussian kernel is generated using the Gaussian function, typically with a predefined kernel size (3x3 in our case) and standard deviation (σ). This kernel is then convolved with the image to produce the blurred version. The convolution operation involves sliding the Gaussian kernel over the image and computing the weighted sum of the neighboring pixel values, where the weights are given by the Gaussian function. This process results in a smooth transition of pixel values, reducing high-frequency noise and small details.

In our case, Gaussian blurring is used to focus on general temperature distributions on the face, allowing us to downplay minor variations while emphasizing larger patterns. This is particularly useful when analyzing thermal images, where the goal is to identify broader temperature trends without getting distracted by fine-grained details or artifacts.

By applying Gaussian blurring, we can:

- Reduce noise and smooth out minor inconsistencies in the image.
- Create a more uniform representation of temperature distributions on the face.
- Enhance the model's ability to recognize broader features and patterns, which are more relevant to our analysis.

An example of an image before and after adding Gaussian blur can be seen in Figure 3.7.

- **Rotation:** Rotation is a technique where images are rotated by a certain degree, typically within a specified range, such as 90 degrees, 180 degrees, or other angles. This technique simulates different orientations of the image, allowing the model to learn from varied perspectives. It is especially useful in contexts where the object's orientation might vary, providing more diverse

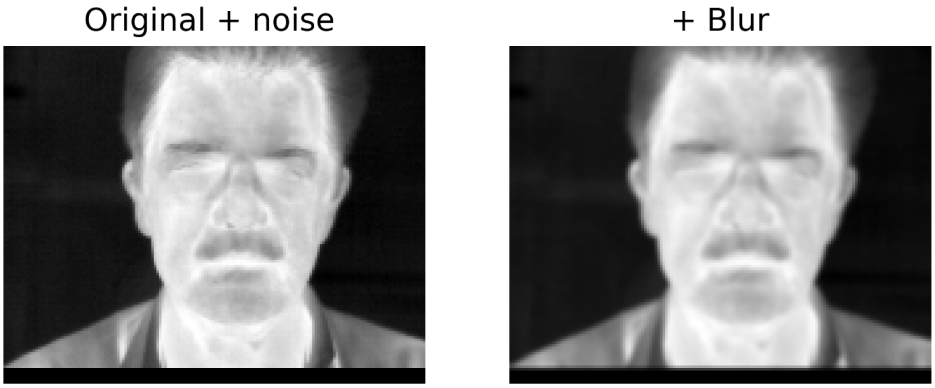

Figure 3.7. Applying Gaussian blur to an image

training examples to the model. In our case, we want to simulate the situation where the subject rotated their head a little bit. Hence, a rotation angle of 10 degrees is applied randomly to some images.

Each augmentation has a 50% chance of being applied to an image. As a result, some images might undergo all augmentations, while others may not be augmented at all (Figure 3.8) . In addition to these augmentations designed to increase dataset diversity, we also applied resizing and normalization to ensure consistency across the entire dataset.

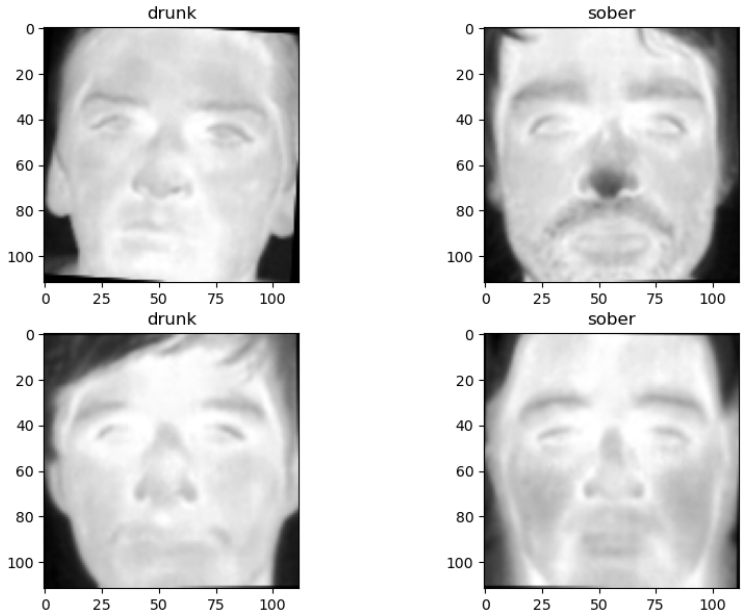

Figure 3.8. A sample of training data after the augmentation

3.3 Exploratory Data Analysis and outlier detection

Due to the limited availability of publicly accessible data, it is important to ensure the quality of the data utilised for training our model. This involves verifying

that subjects labelled as drunk indeed exhibit the characteristics of intoxication, and that external factors such as fluctuations in room temperature do not influence their facial temperature distribution. Even when datasets are collected under controlled conditions, it remains challenging to guarantee the suitability of every subject for the study. Furthermore, in case of thermal data, given that temperature distribution cannot be visually inspected, a reliable tool is needed to autonomously identify potentially 'bad' training data in an unsupervised manner.

Various techniques for anomaly detection, such as outlier detection and novelty detection methods, are available. Our focus primarily lies on outlier detection, as novelty detection methods require prior knowledge of which samples are considered normal or abnormal.

In the following subsection, we use *Pontificia Universidad Católica de Valparaíso Drunk Thermal Face database (PUCV-DTF)* as an example, however, the remaining datasets, except DIF, were studied in the same way.

To begin our analysis, we examined the mean pixel values for sober and drunk classes (Figure 3.9) . While the average pixel value for sober faces was slightly higher than for drunk faces, suggesting that alcohol-intoxicated faces are not consistently 'hotter' overall, specific areas such as the forehead, cheeks, and nose exhibited increased warmth after alcohol consumption. Additionally, when plotting the pixel values of average drunk and sober faces (Figure 3.10) we can notice that the mean of the distribution of the average drunk face is higher than that of sober, implying higher temperatures of some parts of the face.

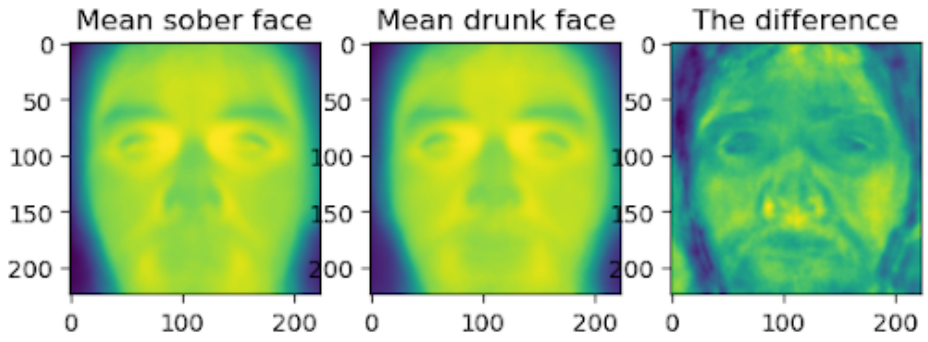

Figure 3.9. Average frame across all subjects for sober and drunk classes

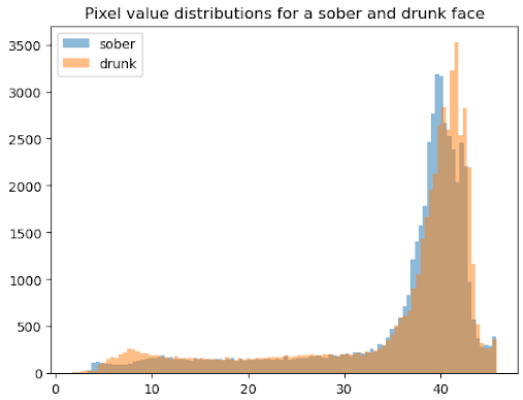

Figure 3.10. Pixel value distribution for sober and drunk face

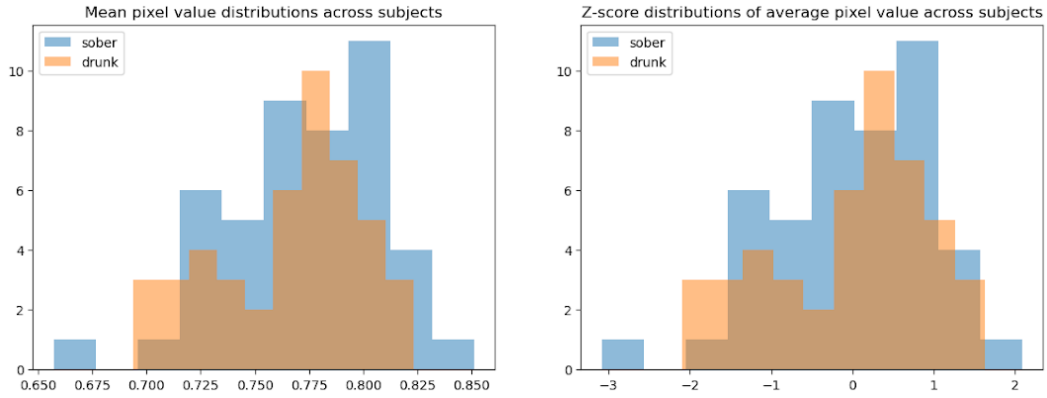

Figure 3.11. Mean pixel value distributions across subjects

Further investigation involved plotting the average pixel value for each subject via histograms (Figure 3.11). In the case of drunk faces, we noticed two peaks in the histogram without any noticeable outliers, whereas sober faces displayed a normal distribution with one subject exhibiting notably lower temperatures. This revealed that one subject's image temperature deviated significantly from the average. Visual inspection indicated that this anomaly might be attributed to the subject standing farther from the camera compared to others.

To test whether the difference is statistically significant, we use a Z-score. The Z-score, also known as standard score, is a statistical measure that indicates how many standard deviations a data point is from the mean of the dataset. In the context of outlier detection, the Z-score can be used to identify observations that deviate significantly from the average. If the population mean and population standard deviation are known, a raw score x is converted into a standard score by

$$z = \frac{x - \mu}{\sigma}$$

where μ is the mean of the population and σ is the standard deviation of the population.

Calculating z using this formula requires use of the population mean and the population standard deviation, not the sample mean or sample deviation. However, knowing the true mean and standard deviation of a population is often an unrealistic expectation, except in cases such as standardized testing, where the entire population is measured. When the population mean and the population standard deviation are unknown, the standard score may be estimated by using the sample mean and sample standard deviation as estimates of the population values. In these cases, the z-score is given by

$$z = \frac{x - \bar{x}}{S}$$

where $\hat{x}$ is is the mean of the sample and S is the standard deviation of the sample.

A common threshold for identifying outliers using Z-scores is ± 3 or ± 2.5, meaning that any data point with a Z-score greater than 3 or less than -3 (or 2.5 depending on the choice of the statistician) is considered an outlier. In our case, z-score of 3 was chosen.

It's important to note that the Z-score method assumes that the data is normally distributed. If the data is not normally distributed, other outlier detection

techniques might be more appropriate. Additionally, while the Z-score is useful for identifying univariate outliers, for multivariate outlier detection, other methods such as Mahalanobis distance or clustering-based approaches may be more suitable.

Clustering for outlier detection

Another approach we explored for outlier detection involved a two-step process: first transforming images into feature representations and then applying clustering techniques to identify outliers. This method allowed us to find images that did not conform to the main distribution patterns, aiding in the isolation of potential outliers.

1. **Feature Extraction:** To extract feature representations, we used a VGG16 style network pre-trained on the ImageNet dataset. This step basically serves as a dimensionality reduction technique, where intermediate features are derived from one of the later layers. This process transformed the images from raw pixel data into a more abstract form, capturing high-level patterns while reducing the size.

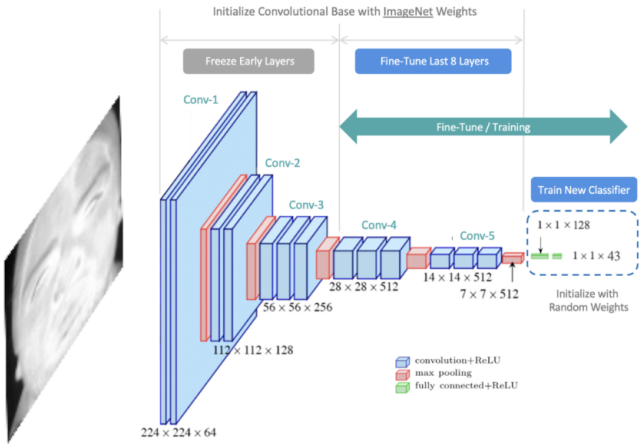

Figure 3.12. Fine-tuning pretrained model

2. **Clustering with K-Means** After obtaining these features, we used the K-means clustering algorithm to group the data into clusters. K-means works by dividing data into a predefined number of clusters and then iteratively adjusting the cluster centers to minimize the distance between points and their corresponding cluster centers. To identify the optimal number of clusters for K-means, we used two common methods: the *silhouette method* and the *elbow method*.

    - Elbow Method: This method involves plotting the sum of squared distances between data points and their nearest cluster centers for a range of cluster counts. The "elbow" in this plot indicates the point where adding more clusters is not "worth" as it doesn't reduce the sum of squared distances.The reasonable number of clusters is usually the point where the rate of decrease in distortion starts to level off. However, as can be see in the (Figure 3.13), there is no such distinct point in our case, so we cannot utilise this method for picking the amount of clusters.

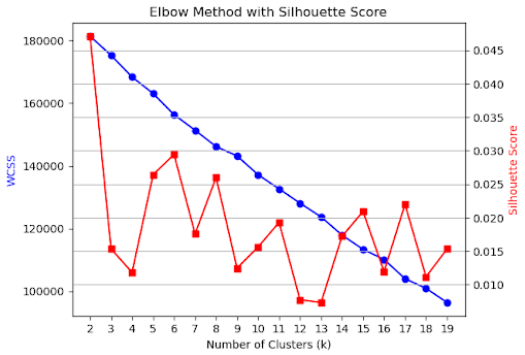

Figure 3.13. Clustering for outlier identification: Elbow Method with Silhouette Score

- Silhouette Method: This method measures how similar a data point is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, with a higher score indicating that the data point is well-clustered. A silhouette score close to 1 indicates that the point is far from other clusters, while a score close to -1 indicates that it might be in the wrong cluster. We used this method to determine the optimal number of clusters by selecting the value that produced the highest average silhouette score across all clusters (in this case, 6).

Once the clusters were established, we identified outliers by measuring the Euclidean distance between each data point's feature representation and its nearest cluster center. The Euclidean distance is a common measure used to calculate the distance between two points in a multidimensional space.

Given a data point $x = (x_1, x_2, \ldots, x_n)$ in an n-dimensional space, and a cluster center $c = (c_1, c_2, \ldots, c_n)$, the Euclidean distance between the data point and the cluster center is calculated as follows:

$$d(x, c) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2}$$

The distances are then transformed into z-score representation. As can be seen from the Figure 3.14, this approach did not detect any unusual subjects.

Autoencoders for outlier detection

Finally, another powerful tool, namely autoencoders, was utilised. Autoencoders work great for outlier detection due to their ability to learn a compressed representation of the input data. Autoencoders consist of an encoder and a decoder (Figure 3.15). The encoder compresses the input data into a lower-dimensional latent space representation, while the decoder reconstructs the input data from this representation. During training, the autoencoder learns to reconstruct the input data with minimal error.

For this task, we employ a simple autoencoder with two convolutional layers and one MaxPooling layer in the encoder part:

1. Encoder Layers:

Figure 3.14. Clustering for outlier identification: Elbow Method with Silhouette Score

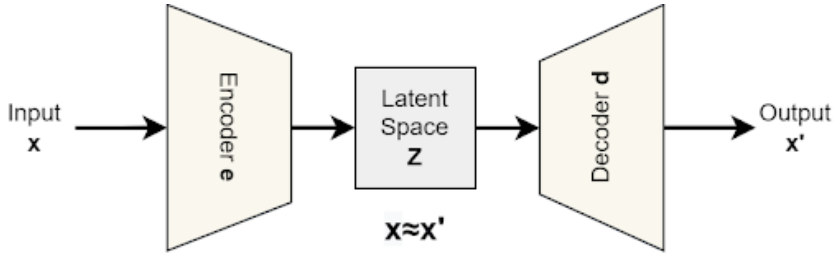

Figure 3.15. Autoencoders architecture [24]

- Convolutional Layer 1: This layer convolves the input grayscale image using a 3x3 kernel, producing 16 feature maps. The stride of 2 reduces the spatial dimensions by half, and ReLU activation introduces non-linearity.
- MaxPooling Layer: Following the first convolution, max-pooling with a 2x2 kernel and a stride of 2 further reduces the spatial dimensions while retaining essential features.
- Convolutional Layer 2: This layer convolves the feature maps from the previous layer using a 3x3 kernel, producing 8 feature maps. Like the previous layer, it applies ReLU activation for non-linearity.

2. Decoder Layers:

- Transpose Convolutional Layer 1: The decoder starts by applying a transpose convolution to upsample the encoded features. This layer takes the 8 feature maps from the encoder and produces 16 feature maps. The 3x3 kernel with a stride of 2 increases the spatial dimensions, and ReLU activation introduces non-linearity.
- Transpose Convolutional Layer 2: Similarly, this layer takes the 16 feature maps and produces 16 feature maps. The 3x3 kernel with a stride of 2 increases the spatial dimensions, and ReLU activation is applied.
- Transpose Convolutional Layer 3: This final layer aims to reconstruct the original grayscale image. It takes the 16 feature maps and outputs a single-channel image. The 3x3 kernel with a stride of 2 increases the spatial dimensions, and ReLU activation is applied.

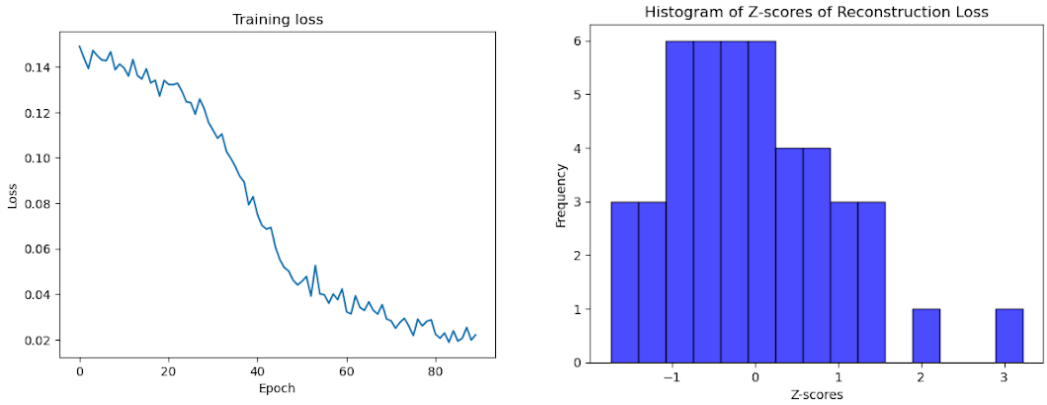

Figure 3.16. Autoencoder training results

- Sigmoid Activation: The output of the decoder passes through a sigmoid activation function. This function squashes the pixel values to the range [0, 1], ensuring they represent valid grayscale intensities.

For outlier identification, we utilise the reconstruction error (MSE) as a measure of how well the autoencoder can reconstruct each input sample. The formula for MSE can be defined as follows:

$$\text{MSE} = \frac{1}{h \times w \times c} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{c} (X_{i,j,k} - X'_{i,j,k})^2,$$

where:

- h is the height of the image.
- w is the width of the image.
- c is the number of color channels (e.g., 3 for RGB, 1 for grayscale).
- $X_{i,j,k}$ represents the pixel value at position (i, j) in the k-th color channel of the original image.
- $X'_{i,j,k}$ represents the corresponding pixel value in the reconstructed image.

A higher reconstruction error suggests that the input sample is dissimilar to the majority of the data the autoencoder was trained on, indicating that it may be an outlier. We utilise the Adam optimizer with a learning rate of 0.001 to update model parameters over 20 epochs. We dont really need more epochs because the dataset is small and can be reconstructed quiet easily after a couple of steps. Since the goal of this approach is not to build a reliable autoencoder but rather see what image the model struggles reconstructing the most, we do not separate the data into training and validation sets. The training loss over the epochs can be see in the Figure 3.16.

After the training process is finished, we evaluate the model by running each individual image and find its reconstruction loss. The values are then transformed to z-scores to identify outliers. In this scenario, the Z-score quantifies how many standard deviations an individual sample's reconstruction error deviates from the mean reconstruction error of the dataset.

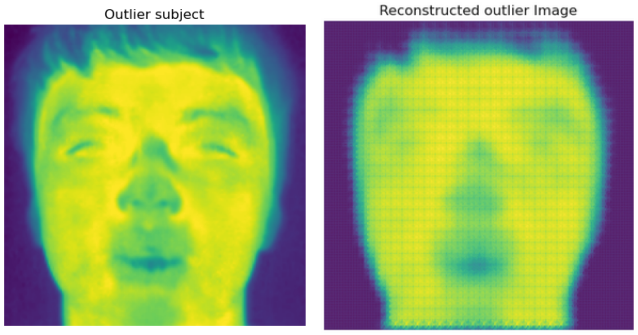

Figure 3.17. Outlier subject and its reconstructed image

Once again, we have the same sample with the Z-score higher than 3. Since two out of three methods indicated subject 23 as a potential outlier, we are going to test the model's performance with and without it.

Chapter 4

Drunkenness detection using Convolutional Neural Networks

Following a comprehensive review of the existing literature, we found that Convolutional Neural Networks have gained significant popularity. A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that learns feature engineering by itself via filters optimization. The layers are arranged in such a way so that they detect simpler patterns like lines and curves first and more complex patterns like faces and objects further along. It is one of the most popular approaches for image classification and these models have been employed in a variety of studies and yielded promising results.

In this chapter, we implement a basic Convolutional Neural Network for Drunkenness Identification to serve as a baseline model. This provides a starting point for assessing the effectiveness of CNNs and lays the groundwork for future improvements. The decision to focus on this approach was influenced by several factors, including the widespread use of classical CNNs, their broad applicability, the ready availability of data, and the potential for further refinement. To evaluate its performance, we test this model across multiple datasets, including both thermal and RGB data. This method allows us to examine the versatility of classical CNNs across different data domains, offering insights into the potential strengths and weaknesses of this approach in the context of Alcohol Intoxication Detection.

4.1 Model Architecture

Typical CNN model architecture includes convolution, pooling and fully-connected layers. An example of a Convolutional Neural Network can be seen in Figure 4.1.

Input Layer: The first layer of the CNN, where the input image is fed into the network. Each pixel in the image is considered as a separate input neuron.

Convolutional Layer: Convolutional layers are the foundational elements of Convolutional Neural Networks. These layers utilise a collection of filters (also known as kernels) to perform convolution operations on the input image. The filters help in detecting patterns, features, or edges in the image. Within a convolutional layer, a kernel moves across the 2D input data in a sliding manner, performing an element-wise multiplication. Consequently, the results are accumulated, yielding a single output pixel (Figure 4.2). The kernel repeats the same operation at every position it traverses, leading to the transformation of a 2D feature matrix into an altered 2D feature matrix. Mathematically, it can be expressed as follows:

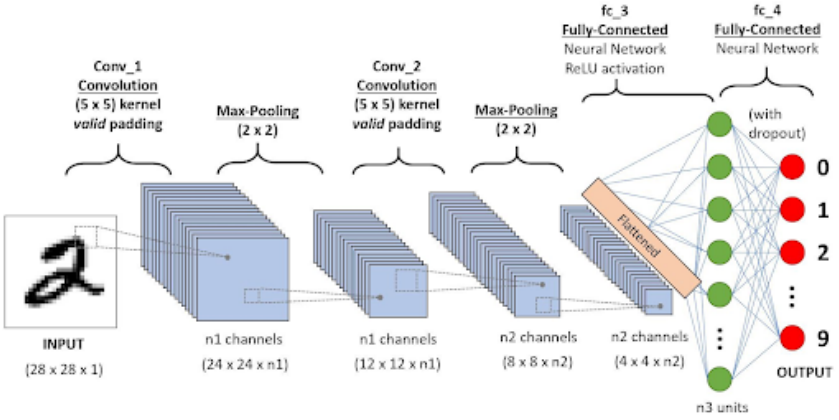

Figure 4.1. An example of a CNN [19]

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k]$$

where f is an input image, h is the kernel, and m and n are the indexes of the rows and columns of the result matrix, respectively.

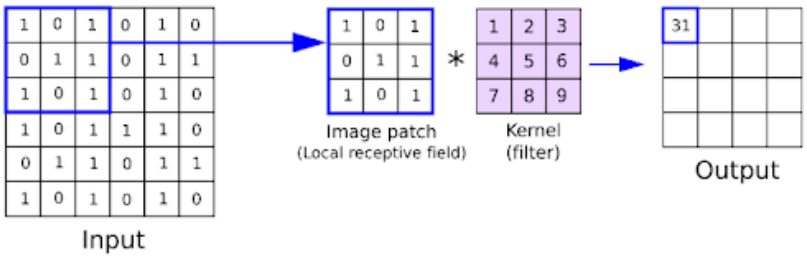

Figure 4.2. Convolution operation [50]

Activation Function: After the convolution operation, an activation function is applied to the output of each neuron. This step introduces non-linearity to the system, allowing the network to learn complex patterns. Activation functions decide whether a neuron should be activated (output a signal) or not, based on the weighted sum of its inputs. One of the most popular activation functions, which is featured in this study, as well, is called ReLU (Rectified Linear Unit):

$$f(x) = max\{0, x\}$$

It sets all negative values to zero and leaves positive values unchanged, ranging from 0 to infinity. Another activation function incorporated in the work is the sigmoid activation function. This function maps the output to a (0,1) scale, making it well-suited for models where predicting the probability of an item belonging to a class is essential. Typically applied in the final stage of forward propagation, the sigmoid function is expressed as:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Graphs illustrating these two activation functions are presented in Figure 4.3.

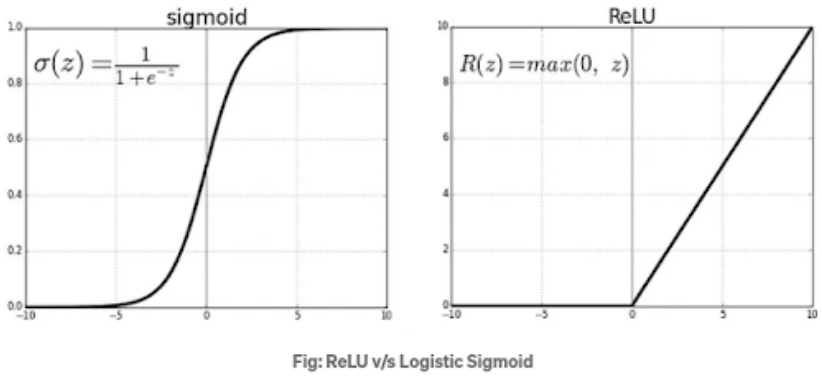

Figure 4.3. Sigmoid and Rectified Linear Unit activation functions [53]

Pooling (Subsampling or Down-sampling) Layer: Pooling layers are used to reduce the spatial dimensions of the feature maps, whilst still retaining the most relevant information. Max Pooling operation, used in this study, is applied independently to each region of the input, and retains the maximum value while discarding the rest. The result is a downsampled representation of the input, where each region is represented by its maximum value (Figure 4.4). Mathematically, the max pooling operation for a 2x2 window with a stride of 2 can be represented as follows:

$$MaxPooling(X)_{i,j,k} = max(x_{2i,2j}, x_{2i,2j+1}, x_{2i+1,2j}, x_{2i+1,2j+1})$$

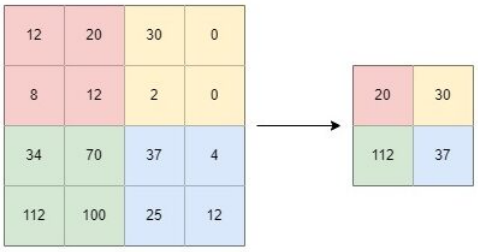

Figure 4.4. An example of MaxPooling operation [2]

Flattening: After several convolutional and pooling layers, the data is represented in the form of a 3D or a 4D tensor (height, width, depth/channel), where each dimension reflects a different aspect of the features detected in the input data. The flattening operation is used to transform the multi-dimensional output into a one dimensional vector by simply stacking the tensor values. This prepares the data to be used in the fully connected layers, requiring one-dimensional input.

Fully Connected (Dense) Layers (Figure 4.5): In these layers, every neuron is connected to every neuron in the previous and subsequent layers. These layers make the final predictions or classifications based on the learned features from earlier layers.

Output Layer: The final layer of the network, responsible for producing the output. The number of neurons in this layer depends on the specific task. For

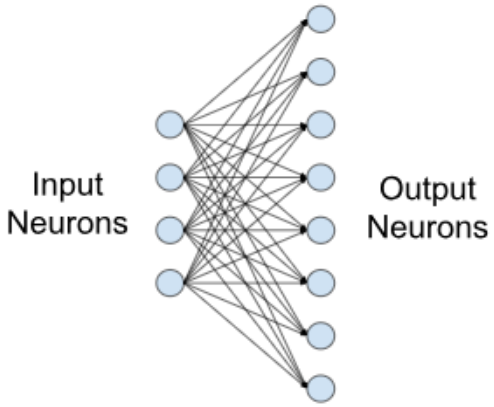

Figure 4.5. An example of a fully connected layer [1]

instance, in our binary classification problem the output layer has one neuron, representing the probability of the subject belonging to the 'drunk' class.

Full model architecture used in this work can be seen in Figure 4.6.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [64, 2, 113, 113]              20
         MaxPool2d-2            [64, 2, 56, 56]               0
            Conv2d-3            [64, 2, 54, 54]              38
         MaxPool2d-4            [64, 2, 27, 27]               0
            Conv2d-5           [64, 32, 25, 25]             608
         MaxPool2d-6           [64, 32, 12, 12]               0
            Linear-7                   [64, 8]          36,872
       BatchNorm1d-8                   [64, 8]              16
            Linear-9                   [64, 1]               9
              Net-10                   [64, 1]               0
================================================================
Total params: 37,563
Trainable params: 37,563
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 3.23
Forward/backward pass size (MB): 31.12
Params size (MB): 0.14
Estimated Total Size (MB): 34.49
----------------------------------------------------------------
```

Figure 4.6. Our CNN model architecture

4.2 Training algorithm

Training a Convolutional Neural Network (CNN) involves adjusting the network's parameters (weights and biases) to minimise the difference between the predicted outputs and the actual target values.

1. **Initialisation:** Initialise the weights and biases of the network randomly. This step sets the initial conditions for the network, from which forward propagation takes place. To ensure the reproducibility of the experiments, we fixed the initialised weights by fixing seed value of the random number generator.

2. **Forward Propagation:** Input image is fed through the network layer by layer in the forward direction according to the model architecture. The result

produced at this step is the probability value of the image to belong to a drunk person.

3. **Loss Computation:** The predicted output generated by the forward pass is compared with the actual target values using a loss function. The loss function quantifies the dissimilarity between the empirical distribution of training data and the distribution induced by the model. Common loss functions include categorical cross-entropy for classification tasks and mean squared error for regression tasks. Since this work focuses on the problem of discriminating drunk from non-drunk individuals the binary cross entropy loss, which works well for binary classification tasks, is used. Binary Cross Entropy/Log Loss measures the dissimilarity between the actual labels and the predicted probabilities of the data points being in the positive class. It penalises the predictions that are confident but wrong. Mathematically, it can be expressed as:

$$BCE = -\frac{1}{n}\sum_{i}^{n}(y_i \times log(\hat{y_i}) + (1 - y_i) \times log(1 - \hat{y_i}))$$

where n is the size of the dataset, y_i is the actual label of the i-th subject (note that $y_i \in \{0, 1\} \forall i = 1, ..., n$) and y_i is the predicted probability of the point i being in the positive class (not necessarily either 0 and 1).

4. **Backpropagation:** Calculate the gradient of the loss with respect to the network's parameters (weights and biases) using the chain rule of calculus. This gradient information is propagated backward through the network.

5. **Optimisation:** Once the gradients are computed, gradient-based optimization algorithm is used to update the weights and biases. The goal is to adjust the parameters in the direction that minimizes the loss. Some of the popular optimisation algorithms are Stochastic Gradient Descent (SGD) or Adam (Adaptive Moment Estimation), which was used in our study. It combines the advantages of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). The algorithm consists of the following steps:

   (a) **Initialisation of parameters:**
   - Learning rate (η), often initialised at 0.001.
   - Exponential decay rates for the moment estimates (β_1 and β_2). Commonly, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
   - Small constant (ϵ to avoid division by zero, usually set to 1×10^{-8}).

   (b) **Initialise moments**: Before the optimization loop, initialize the first and second moment estimates, $m_0 = 0$ and $v_0 = 0$, for each parameter.

   (c) **First and Second Moment Estimates**: In each iteration (or "time step"), Adam computes the biased first and second moment estimates: First moment (Mean of Gradients):

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t$$

   Second moment (Mean of Squared Gradients):

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2$$

   where g_t is the gradient of the loss function with respect to the parameters at time step t.

(d) **Bias Correction** The estimates for m_t and v_t are biased towards zero, especially in early iterations. Adam corrects these biases:
Bias-Corrected First Moment:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

Bias-Corrected Second Moment:

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

These corrections ensure that the estimates are unbiased as the iteration count increases.

(e) **Parameter Update**: Using the bias-corrected estimates, the parameter update is calculated as:

$$\theta_t = \theta_{t-1} - \eta \times \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

where θ_{t-1} represents the parameter value at time step $t - 1$, θ_t is the updated parameter and the learning rate, η, can be constant or decay over time.

This process is repeated for all layers and continues for multiple iterations (or epochs) until the model converges, or the training stops for other reasons (like early stopping or a set number of epochs).

Steps 2)-5) are repeated for multiple batches of training data. The complete pass through the training data is known as an epoch. The current model has been trained for 100 epochs, using training part of the data to adjust weights/biases and validation part to ensure the generalisation ability of the network.

4.3 Results

Experimental setting

The hardware used in this study included a laptop equipped with an NVIDIA GeForce RTX 3050 GPU, featuring CUDA support. Development work was carried out in Visual Studio Code (VSCode), serving as the Integrated Development Environment (IDE). The deep learning frameworks used for model development and training were PyTorch and PyTorch Lightning. Additional software libraries included NumPy, Pandas, Matplotlib, Scikit-learn and other, providing tools for data processing, visualization, and model evaluation.

To ensure reproducibility, initial weights were fixed and a consistent random seed was used. During training, model checkpointing was implemented to save the model whenever it achieved a new high in validation accuracy or a new low in validation loss. Additionally, an early stopping mechanism was employed to prevent overfitting, stopping the training process when the validation loss would not decrease for at least 20 epochs, however such condition was met rarely during the experiments.

The chosen loss function, nn.BCEWithLogitsLoss(), applies a sigmoid activation function followed by a binary cross-entropy loss, providing robust gradient calculation while addressing numerical stability. For optimization, we used optim.Adam, which

combines adaptive learning rates with first- and second-moment estimates, set with a learning rate of $\eta = 0.001$. Additionally, an L2 regularization (weight decay) of 0.001 is applied to prevent overfitting.

These parameters were chosen empirically, following rigorous testing of various combinations to determine the optimal configuration. We settled on 150 epochs to provide enough time for model convergence, with a batch size of 16 to balance computational efficiency and stable gradient estimation. This experimental setup represents the best outcome from multiple iterations, reflecting the performance characteristics and practical constraints of our training environment.

Model evaluation

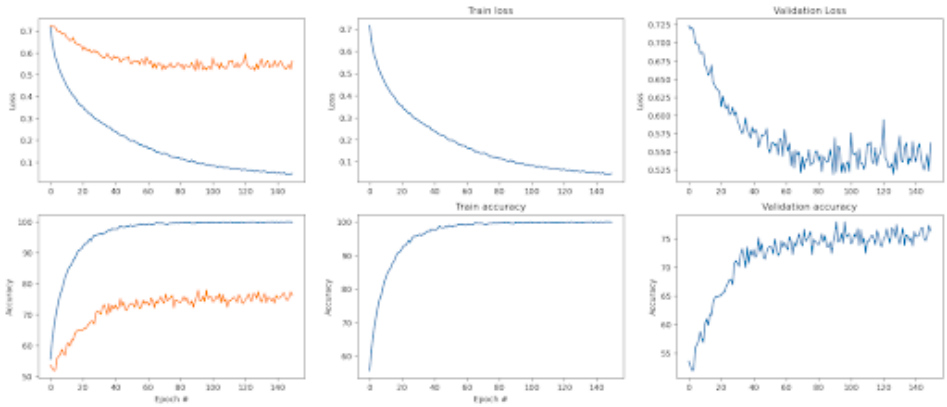

Figure 4.7. Training and validation loss

Figure 4.7 shows that the model did not overfit during the training process and was acting fairly stable. As mentioned earlier, we saved two checkpoints for highest accuracy and lowest loss on the validation data. The classification results of the two can be found in the Figure 4.1. Further in the evaluation, we will refer to the accuracy checkpoint only.

Split	Lowest loss	Highest accuracy
train	0.9997	1
validation	0.737	0.754

Table 4.1. Classification accuracies for train and validation set in two checkpoints

The classification summary and confusion matrix can be seen in Table 4.2 and Figure 4.8.

The classification report provides several metrics to evaluate the performance of a model, including precision, recall, f1-score, and support for each class, as well as overall accuracy and average metrics. Let's break down the key metrics:

- **Precision** measures the proportion of true positive predictions among all positive predictions made by the model. The formula for precision is:

$$\text{Precision} = \frac{TP}{TP + FP},$$

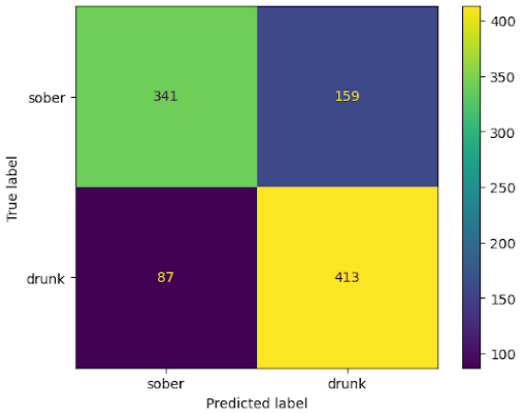

Figure 4.8. Confusion matrix

Class	Precision	Recall	F1-Score	Support
Sober	0.80	0.68	0.73	500
Drunk	0.72	0.83	0.77	500
Accuracy			0.75	1000
Macro Avg	0.76	0.75	0.75	1000
Weighted Avg	0.76	0.75	0.75	1000

Table 4.2. Classification Summary

where TP represents true positives, and FP represents false positives.

For the "sober" class, precision is 0.80, indicating that 80% of predictions labeled as "sober" are correct. For the "drunk" class, precision is 0.72, suggesting that 72% of predictions labeled as "drunk" are correct.

- **Recall**, also known as sensitivity, measures the proportion of true positive predictions among all actual positive cases. The formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN},$$

where FN represents false negatives.

For the "sober" class, recall is 0.68, indicating that the model correctly identified 68% of actual "sober" cases. For the "drunk" class, recall is 0.83, indicating that 83% of actual "drunk" cases were correctly identified.

- The **f1-score** is the harmonic mean of precision and recall, providing a balance between the two. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The f1-score for the "sober" class is 0.73, the f1-score for the "drunk" class is 0.77.

- **Support** represents the number of samples in each class. Both the "sober" and "drunk" classes have 500 samples, indicating a balanced dataset.
- **Accuracy** represents the proportion of correct predictions across all classes. It is defined as:

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

  where TN represents true negatives. The overall accuracy of the model is 0.75.
- We also compute macro and weighted averages for precision, recall, and f1-score. Macro averages treat all classes equally, while weighted averages account for the support of each class:
  - **Macro Average**: Unweighted average across classes. Precision, recall, and f1-score are 0.75, 0.75, and 0.75, respectively.
  - **Weighted Average**: Accounts for the support of each class. Precision, recall, and f1-score are 0.75, 0.75, and 0.75, respectively.

In summary, the classification report shows that the model has a higher recall for the "drunk" class but a lower recall for the "sober" class. The precision for the "sober" class is relatively high, suggesting the model is more cautious in predicting this class. This could be useful when the goal of the model is to prevent misclassification of drunk individuals, for example in the situations where intoxicated individuals pose danger (driving under influence and other). The accuracy and macro averages indicate balanced overall performance, but there's room for improvement in achieving greater consistency across classes. An example of classification results on the validation data can be seen in Figure 4.9.

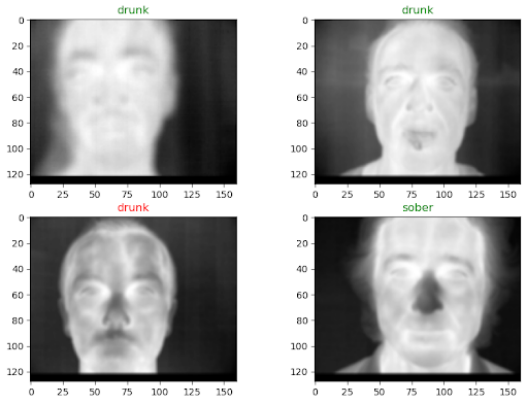

Figure 4.9. An example of classification results on validation data. Red indicates incorrect predictions, while green represents correct ones

Variability of predictions across frames

Given that each subject has 50 frames, after evaluating the classification results, we posed a question: *"Do the predictions vary across frames within a single subject?"* To investigate this, we conducted a test to classify each frame and then visualised the results on a per-subject basis. In the Figure 4.10, we display how predictions

fluctuate across different frames (red indicates incorrect predictions, while green represents correct ones). Given the high frame rate of these images, the subjects typically remained relatively still. Thus, any misclassifications are likely due to changes in the temperature beneath the nose caused by breathing or changes in the distribution due to the blood vessel activity. This observation suggests a new avenue for future research into the use of sequences of frames for classification. By examining how the class predictions change across a sequence of frames for each subject, we could explore whether temporal information can help improve classification accuracy. This approach could be useful for identifying patterns or trends in a sequence of frames that a single frame might not reveal, offering a more robust classification method that accounts for the dynamics of the subject's behavior or physiological changes.

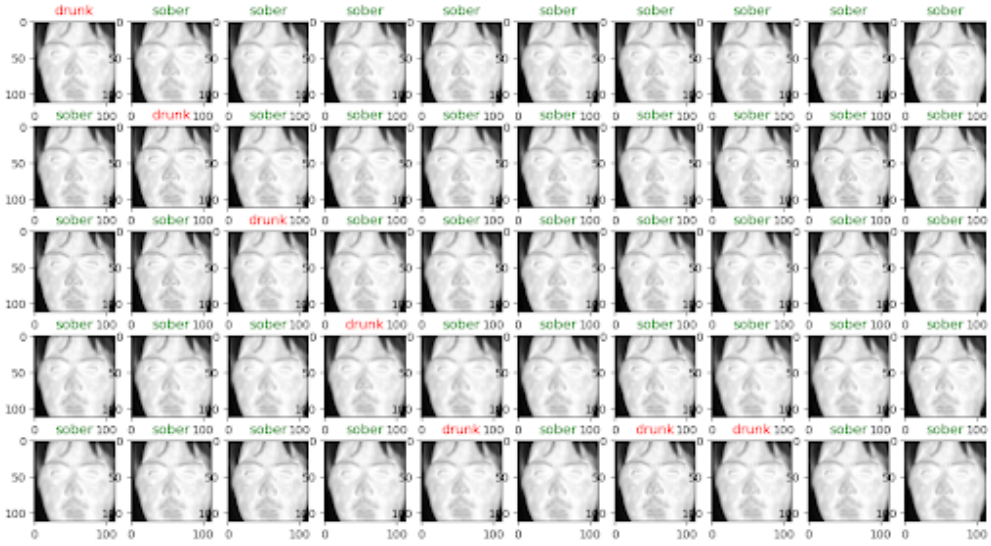

Figure 4.10. An example of classification across frames of a single subject. Red indicates incorrect predictions, while green represents correct ones

Based on this observation, our next step was to examine how confident the predictions were for each subject. Specifically, we wanted to know how many frames were classified correctly for each subject. To do this, we re-ran the classification on the validation set, but instead of calculating the overall accuracy, we calculated the score for each subject. This approach allows us to understand the consistency of predictions within a sequence of frames for each subject.

Given a subject with n frames, let's assign a score of 1 if a frame is classified as "drunk" and 0 if classified as "sober." The classification score for the subject is then calculated as follows:

$$\text{Score} = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

where n is the number of frames for each subject, and x_i is the classification result for the i-th frame ($x_i = 1$ for "drunk," $x_i = 0$ for "sober").

This formula gives the proportion of frames classified as "drunk" within the subject's sequence of frames. A score of 1 indicates that all frames were classified as "drunk," and a score of 0 indicates that all frames were classified as "sober."

Next, we compared this calculated score with the actual label for each subject. By plotting these two values, we can visualize the "confidence" of the predictions for

each subject, providing insights into how consistently the model classified a sequence of frames. From the Table 4.3 we can see that generally, the model was confident for all the predictions, and misclassifications of several frames within a subject did not affect the general accuracy of the model.

Subject	Classification Score	True Label
1	0.02	0
2	0.0	0
3	0.0	0
4	1.0	0
5	1.0	0
6	0.32	0
7	0.0	0
8	0.0	0
9	0.12	0
10	0.0	0
11	1.0	1
12	1.0	1
13	0.06	1
14	1.0	1
15	1.0	1
16	1.0	1
17	1.0	1
18	1.0	1
19	0.08	1
20	1.0	1

Table 4.3. Classification scores and true labels of validation subjects. The score is calculated as the proportion of frames classified as "drunk" within the subject's sequence of frames

Now, let's recalculate the overall accuracy as the proportion of correctly classified subjects. We will consider a subject classified correctly if more than 50 percent of his frames were classified correctly. Mathematically,

$$OverallAccuracy = \frac{\sum_{i=1}^{n} \mathbb{1}(\frac{c_i}{f_i} > 0.5)}{n}$$

where n is the total number of subjects, c_i is the number of correctly classified frames for subject i, and f_i is the total number of frames for subject i. $\mathbb{1}(x)$ is the indicator function which returns 1 if x is true and 0 otherwise.

Using this method, 16 out of 20 subjects were correctly identified, leading to an improved total accuracy of 0.8. However, it's important to note that this approach relies on recording the entire subject for several seconds instead of capturing a single frame.

Chapter 5

Alcohol intoxication identification using Convolutional Attention Networks

In this chapter, we introduce a novel model for identifying alcohol intoxication using a convolutional network with an attention mechanism. We delve into the approach and background of the model, outline its architecture, and analyze the results along with the attention masks.

5.1 Approach

Early concept of attention started getting wide recognition in the scientific community after the work *"Neural Machine Translation by Jointly Learning to Align and Translate"*, 2014 by Badanau et al [6]. This work continued the ideas of Cho et al. (2014) [15] and Sutskever et al. (2014) [56], who introduced an RNN-based encoder-decoder framework for neural machine translation. In this framework, a variable-length source sentence is encoded into a fixed-length vector, which is then decoded to produce a variable-length target sentence. However, Bahdanau et al. identified a critical limitation in this approach. They noted that encoding a variable-length input into a fixed-length vector can lead to loss of information, especially when dealing with longer sentences. This compression of information can cause the performance of the basic encoder-decoder model to decline as the input sentence length increases.

To address this issue, Bahdanau et al. proposed a new approach that uses attention to replace the fixed-length vector with a variable-length one. This attention mechanism allows the model to focus on different parts of the source sentence during translation, thereby improving performance and enabling more accurate alignments between the source and target languages. The researchers tested their ideas on English-to-French texts and reported that their model outperformed all the current conventional encoder-decoder models. This result was a significant advancement in neural machine translation, demonstrating that attention can play a crucial role in enhancing model flexibility and performance.

Later attention models got implemented in the area of computer vision as well. With the introduction of residual connections by He et al.[22] and their ResNet,

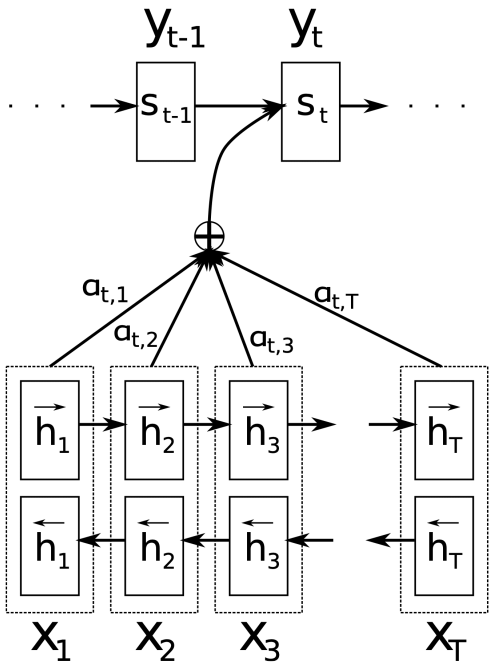

Figure 5.1. The Bahdanau architecture depicting generation of the t-th target word y_t given a source sentence $(x_1, x_2, ..., x_T)$ [6]

more complex and deeper network architectures became possible and so did the additional features, including attention mechanisms. Some of the most promiment works in the area include:

- **Self-Attention in Transformers (2017)**: The seminal work *"Attention Is All You Need"* by Vaswani et al. [60] introduced self-attention in the context of sequence-to-sequence learning, primarily in NLP. This concept inspired many subsequent developments in computer vision.
- **Squeeze-and-Excitation Networks (2017):** Hu, Shen, and Sun [26] introduced the Squeeze-and-Excitation (SE) mechanism, a form of channel-wise attention. In this approach, each channel in a convolutional layer is weighted according to its importance, allowing the network to dynamically scale different features.
- **Non-Local Neural Networks (2018):** Inspired by self-attention, Wang et al. [63] proposed Non-Local Neural Networks, where the model learns relationships between distant points in the feature space, akin to self-attention but within a CNN context. This technique helped improve the ability to capture global context in CNNs.
- **Transformers in Vision (ViT, 2020):** Dosovitskiy et al. [16] introduced the Vision Transformer (ViT), which applied the Transformer architecture from NLP directly to image data, effectively replacing convolutional layers with self-attention mechanisms. This marked a significant shift in how attention was used in computer vision, leading to numerous Transformer-based architectures.

The field is developing rapidly, giving space to even more complex models that could combine the strengths of CNNs and Transformers, using attention mechanisms

to improve performance in various tasks, including image classification, object detection, and segmentation. Additionally, more and more different types of attention mechanisms have emerged, such as

- multi-scale attention,
- axial attention,
- and deformable attention,

allowing for more flexible and powerful attention within CNNs.

Attention Mechanism in Convolution neural networks intuitively mimics human visual attention, which can focus on a certain region of an image with high resolution while perceiving the surrounding image in low resolution. It allows a model to weigh the importance of different positions in the input data differently when producing an output. It calculates "soft" weights for each pixel in the frame.

5.2 Model architecture

The idea of the proposed Convolutional Attention Network (CAN) to have an attention channel training in parallel with a classical CNN channel. The architecture was inspired by the work of Chen et al. [14], where the researchers proposed a system for video-based measurement of bio signals such as heart and breathing rate using CAN. In their work, the CNN channel takes normalised frame difference that acts as a motion representation based on a skin reflection model, while the attention mechanism uses appearance information to guide motion estimation. In our case, since the idea is to use only the data available at time $C(t)$, both channels take the same image as an input (Figure 5.2).

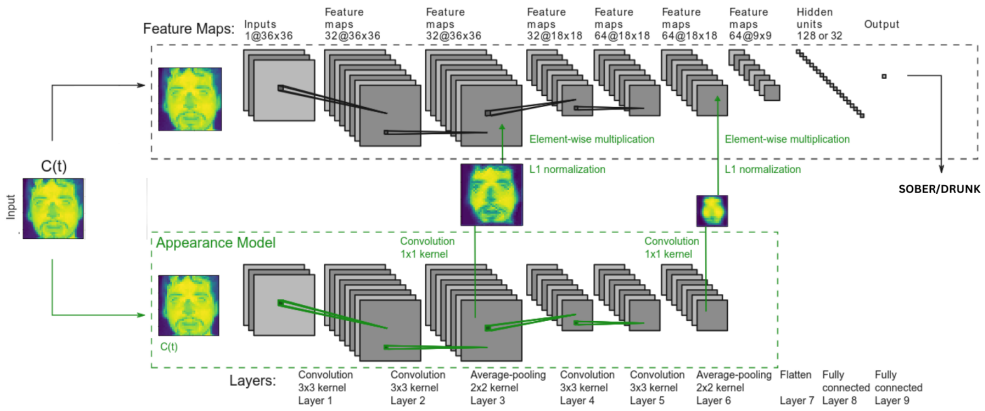

Figure 5.2. The architecture of our convolutional attention network. A frame is given as an input to the appearance and classical channel at the same time. The network learns spatial masks, that are shared between the models, and features important for classifcation.

CNN channel

Our convolutional channel comprises four convolutional layers and two max pooling layers. The max pooling layers are strategically placed after the second and

fourth convolutional layers to capture the most salient features within each region of the image. Max pooling retains the highest activation within each pool, making it effective in identifying localized patterns or areas of interest. This approach is particularly suitable for our task, where specific localized changes, such as alterations in temperature around the nose or forehead regions, may indicate drunkenness. To prevent overfitting, two dropout layers were introduced after second and third convolutional layers with the dropout rate $p = 0.5$.

Following the max pooling layers, we employ a fully connected layer. For activation functions, we empirically chose Leaky ReLU to address the issue of "dying ReLU," where neurons become stuck in a deactivated state due to a gradient of 0 for negative inputs. Leaky ReLU introduces a small slope for negative inputs, preventing this problem. Mathematically, it is expressed as:

$$LeakyReLU(x) = max(0, x) + negative\_slope \times min(0, x)$$

Here, x represents the input, and $negative\_slope$ is a small positive constant.

In our binary classification task, we aim to use Binary Cross Entropy (BCE) as the loss function. However, directly applying the Sigmoid activation followed by BCE loss can lead to numerical instability, especially for large or small logits. To address this, we utilize the Binary Cross Entropy with Logits loss function (BCEWithLogitsLoss). This approach combines the Sigmoid activation function with the Binary Cross Entropy loss function in a numerically stable manner. It allows the model to output raw scores (logits) without the need for an explicit Sigmoid activation in the network's architecture. By doing so, potential overflow or underflow issues during training are mitigated, leading to improved computational efficiency and avoiding redundant calculations that would occur if these operations were performed separately.

Attention channel

Attention channel is introduced to discard the areas of the image that do not contribute to the classification. We would like to focus on the face and not take into account background noise or movement. As was noticed in the previous research as well as in our exploratory analysis, cheeks and forehead area were warmer for the alcohol intoxicated person [37], so if this assumption is correct, the masks should highlight those areas.

During the training process, the model learns soft-attention masks, assigning greater importance to skin areas with stronger signals. The architecture of this channel closely resembles that of our CNN model, with the exclusion of the last three layers. There are many types of attention mechanisms as well as ways of implementing them. In our work, we want to implement a simple attention mechanism that takes the output of a hidden layer as input, transforming it using weights and biases of the same dimension. This transformed output is then subjected to element-wise multiplication with a vector and subsequently scaled to values between 0 and 1 through the Sigmoid operation. This process is commonly referred to as Soft-Attention. The resulting output is then multiplied element-wise with the original output of the hidden layer, effectively masking out less relevant details in the input. To allow the generation of masks from various levels of visual features, we position two of them, one before each pooling layer.

Let $x_a^j \in \mathbb{R}^{C_j \times H_j \times W_j}$ and $x_m^j \in \mathbb{R}^{C_j \times H_j \times W_j}$ represent the feature maps from the appearance and CNN channels at layer j, respectively, and C_j, H_j and W_j be

the number of channels, height and width. The attention mask $q^j \in \mathbb{R}^{1\times H_j \times W_j}$ is computed as follows:

$$q^j = \frac{H_j W_j \cdot \sigma(w^{j^T} x_a^j + b^j)}{2\|\sigma(w^{j^T} x_a^j + b^j)\|_1},$$

where $w^j \in \mathbb{R}^{C_j}$ is the 1×1 convolution kernel, b^j is the bias, and $\sigma(\cdot)$ is the sigmoid function. Unlike the conventional softmax function, our approach utilises sigmoid activation followed by $L1$ normalization, resulting in a softer and less extreme mask.

The attention mask is then applied to the CNN channel feature map via element-wise multiplication:

$$z_m^j = (\mathbf{1} \cdot q^j) \odot x_m^j,$$

where $z_m^j \in \mathbb{R}^{C_j \times H_j \times W_j}$ is the masked feature map, $\mathbf{1} \in \mathbb{R}^{C_j}$ is a vector with all ones, and $\odot$ denotes element-wise multiplication.

5.3 Results

To ensure the comparability of the results from several classification models we process the data in the same way as describe in Section 3.2. This includes removal of the outliers, normalisation, and augmentation techniques such as flipping and rotation.

Adadelta optimiser with a learning rate of 0.001 and 100 epochs were chosen empirically, showing the best classification results. As can be seen in Figure 5.3, the model started learning much quicker compared to the classical CNN model. In this case, we stopped training after 70 epoch, having achieved a good result on both training and validation data (Figure 5.1).

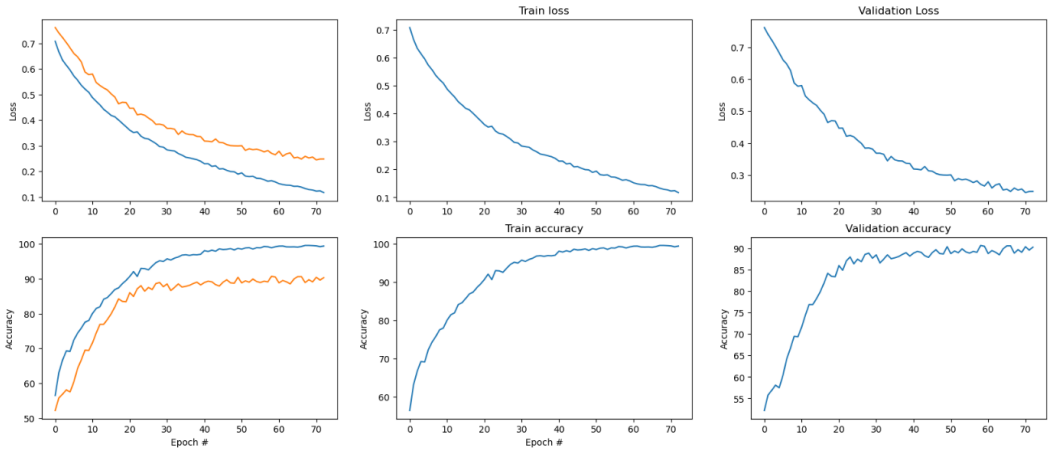

Figure 5.3. Training and validation loss and accuracy

Further exploration includes analysing the classification metrics (Figure 5.2) and the confusion matrix (Figure 5.4).

These classification results indicate a well-performing model with balanced precision and recall values for both classes:

Split	Lowest loss	Highest accuracy
train	0.9995	0.9998
validation	0.914	0.896

Table 5.1. Classification accuracies for train and validation set in two checkpoints

Class	Precision	Recall	F1-Score	Support
Sober	0.91	0.88	0.89	500
Drunk	0.89	0.91	0.90	500
Accuracy			0.90	1000
Macro Avg	0.90	0.90	0.90	1000
Weighted Avg	0.90	0.90	0.90	1000

Table 5.2. Classification Summary of the CAN model

- The precision for the "sober" class is 0.91, indicating that out of all instances predicted as "sober" by the model, 91% were actually "sober." Similarly, the precision for the "drunk" class is 0.89, meaning that 89% of instances predicted as "drunk" were actually drunk. These high precision values suggest that the model has a low false positive rate, meaning it is good at avoiding misclassification of instances.
- The recall for the "sober" class is 0.88, indicating that the model correctly identified 88% of all actual "sober" instances. Likewise, the recall for the "drunk" class is 0.91, meaning that the model captured 91% of all actual "drunk" instances. These high recall values suggest that the model has a low false negative rate, meaning it is good at capturing instances of both classes.
- The F1-score for the "sober" class is 0.89, and for the "drunk" class is 0.90, indicating a good balance between precision and recall for both classes.

Our final evaluation metric is Receiver Operating Characteristic (ROC) curve. ROC curves are graphical representations of the performance of a binary classification model across different threshold settings. It provides a visual representation of the trade-off between true positive rate and false positive rate for different threshold settings and helps to assess the discriminatory power of the model and compare the performance of different classifiers. In our case (Figure 5.5), with an AUC of 0.97, it indicates that the model has high discriminatory power and can effectively differentiate between the two classes (e.g., "sober" and "drunk"). This suggests that the model's predictions are accurate across a wide range of thresholds, making it highly reliable for the given task.

Attention weights visualisation

An advantage of the proposed attention-based model is its ability to visualize attention masks. This feature enables us to confirm earlier hypotheses regarding the temperature distribution on the faces of drunk and sober individuals, offering insights into the model's focus areas. Additionally, it contributes to improving the model's explainability — a critical step toward making neural networks more

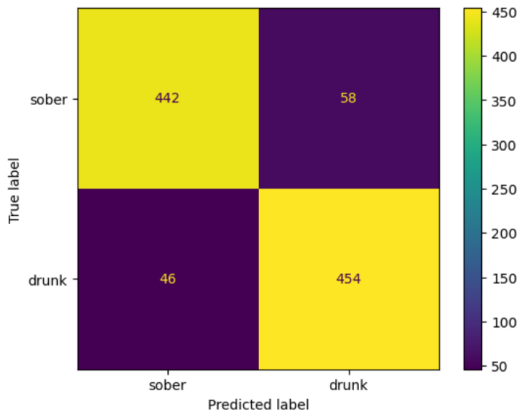

Figure 5.4. Confusion matrix after classification with CAN

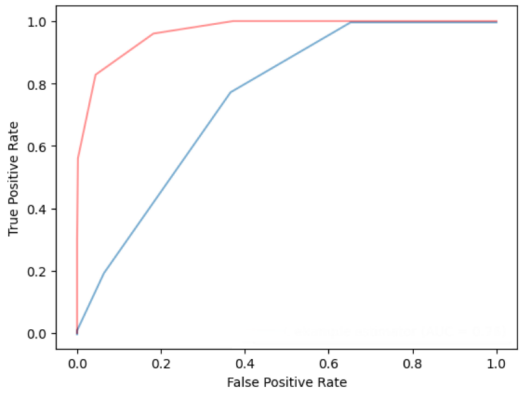

Figure 5.5. ROC curves for CNN (blue) and CAN (red) models. $\text{AUC}_{CNN} = 0.76, AUC_{CAN} = 0.97$

transparent. The explainability of neural networks is a relatively recent research focus, as traditional CNNs have often been viewed as "black boxes," with limited visibility into their inner workings.

As shown in Figure 5.6, the model gradually learns the attention weights. During the first epochs, the weights are randomised and scattered, while closer to the end of the training process they become more structured and resemble a face shape. We can also notice how the model avoids the background, facial hair and noses, prioritizing the rest of the face.

During our analysis, we observed an intriguing thing (Figure 5.7): upon concluding the training process and examining the final attention masks, we discovered that the model exhibited a focus on the ear area for certain subjects. While there is limited research directly linking alcohol consumption to fluctuations in ear temperature, it's plausible that the increased blood flow associated with vasodilation could lead to a slight increase in ear temperature. Although challenging to validate with our current dataset, this observation presents a potential avenue for future research.

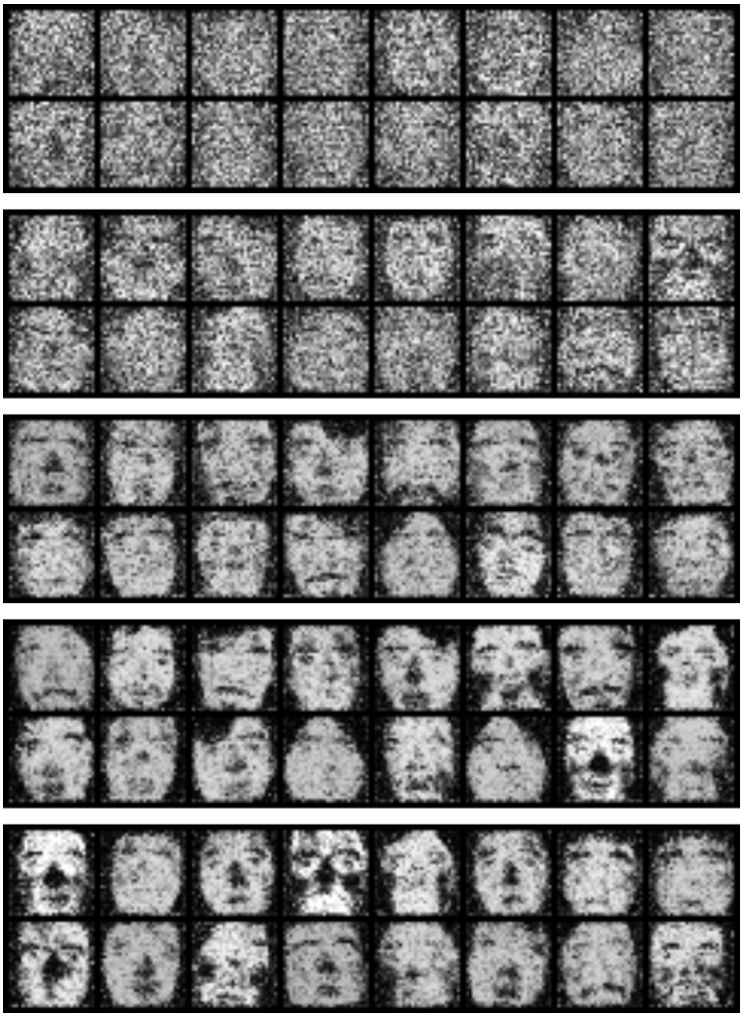

Figure 5.6. Progressive improvement of attention masks across epochs (1, 10, 20, 30, 40)

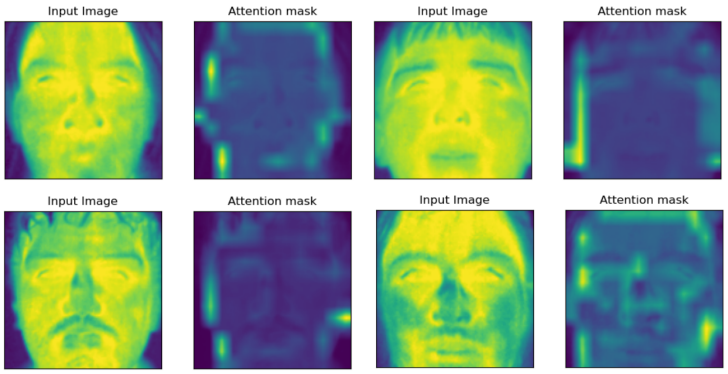

Figure 5.7. Attention focus on ear area

Chapter 6

Results and comparison

In the previous chapters, we implemented two different Deep Learning models for alcohol intoxication detection: a classic Convolutional Neural Network (CNN) and a CNN with a Self-Attention Mechanism. In this chapter, we evaluate and compare their performance across different available datasets. Furthermore, we present classification metrics from well-known classic CNN architectures like ResNet, VGG, and others, allowing for a comprehensive comparison of the results.

6.1 Model comparison

To evaluate the effectiveness of our attention mechanism model, we conducted experiments across five distinct datasets, comprising three thermal and two RGB datasets (more information about the data can be found in Chapter 3). These datasets were chosen to represent diverse scenarios and challenges commonly encountered in alcohol intoxication detection using facial images.

In our comparative analysis, we included three state-of-the-art convolutional neural network architectures: ResNet18 [22], VGG16 [54], and MobileNet v2 [25]. Each of these models offers unique characteristics and performance profiles, making them suitable candidates for our task. ResNet18 is renowned for its depth and skip connections, VGG16 for its simplicity and effectiveness, and MobileNet v2 for its efficiency and compactness.

Given that these models were primarily designed for RGB images, we adapted them to grayscale datasets by stacking the images to create three-channel inputs. Additionally, we introduced an extra layer to convert the classification task from multiclass to binary, aligning with our specific objective of distinguishing between sober and drunk individuals.

To accommodate variations in model complexities and generalization challenges across different datasets, we tuned several parameters, including the number of hidden units, training epochs, optimizer selection, and batch sizes. The results can be visualised in the Table 6.1. These results underscore the absence of a one-size-fits-all solution when it comes to designing convolutional neural network architectures. Certain architectures excel on specific datasets while underperforming on others, and vice versa. For instance, ResNet18 and VGG16 demonstrated strong performance on the DIF dataset (88% and 85%, respectively), which is characterized by its size and diversity. However, they exhibited poor generalisation when applied to datasets with smaller training samples. This discrepancy may come from the deep structure of these architectures, which typically show better results with larger datasets.

Conversely, our proposed simple CNN showcased superior performance on relatively small datasets such as Kubicek et. al and 3 Glasses After. This suggests that simpler architectures may offer better generalization capabilities when training data is limited.

Of particular note is the effectiveness of the proposed attention mechanism, which consistently outperformed all state-of-the-art models. Across every dataset, the attention mechanism yielded higher validation accuracies, underscoring its potential to enhance classification performance in alcohol intoxication detection tasks.

Dataset	ResNet18	VGG16	MobileNet v2	Our CNN	CAN
Sober-Drunk	0.5	0.57	0.653	0.78	0.92
PUCV-DTF	0.54	0.643	0.53	0.75	0.9
Kubicek et. al	0.664	0.571	0.766	0.761	0.8
DIF	0.883	0.851	0.779	0.671	0.917
3 Glasses After	0.58	0.574	0.6	0.645	0.7

Table 6.1. Accuracy of different classification models across various datasets

6.2 Datasets merging

To expand the training data, we attempted to merge two datasets, PUCV-DTF and the Sober-Drunk dataset, given their relative similarities. However, the latter dataset contained additional information, such as neck and shoulders, whereas the former focused mainly on facial information. Thus, a processing strategy was employed to ensure consistency across the merged dataset:

1. **Face Detection**: We used the Haar Feature-based Cascade Classifier to detect faces within the images. Introduced by Viola and Jones [62], the algorithm uses edge and line detection features. The OpenCV module cv.CascadeClassifier() was utilized to identify facial regions (Figure 6.1).

Figure 6.1. Face Detection with Haar Cascade Classifier

2. **Manual Adjustment of Bounding Boxes:** After face detection, we manually reviewed and adjusted the bounding boxes to ensure they were square-shaped and contained all relevant information. This step was crucial to ensure consistency across the merged dataset.

3. **Cropping and Resizing:** Once the bounding boxes were adjusted, we cropped the desired regions from the images, focusing on the face while removing the neck and shoulders. The cropped images were then resized to match the dimensions of the other dataset.

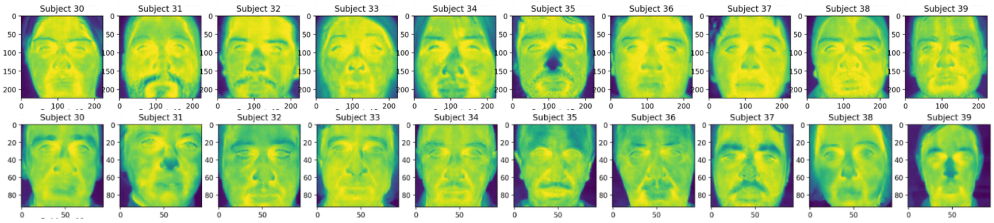

Figure 6.2. Samples from from PUCV-DTF (first row) and Sober-Drunk dataset (second row) after processing

The process resulted in two datasets, as shown in Figure 6.2. Upon visual inspection, differences in face temperatures between the datasets became apparent, with subjects from the PUCV-DTF dataset appearing 'hotter.' To validate this observation, we compared the average pixel values for the sober and drunk classes across both datasets. For the sober class, the average pixel values were quite similar (182.45 for PUCV-DTF vs 182.92 for Sober-Drunk). However, a significant discrepancy emerged for the drunk class (196.71 for PUCV-DTF vs 185.06 for Sober-Drunk). This inconsistency could introduce bias or variability during training, affecting the model's performance. To address this issue, we normalised the data to bring both datasets onto a common scale.

Although achieving a lower overall accuracy (see Table 6.2), this approach diversifies the data and allows for more robust performance of the model on subjects filmed in different settings and varying sensors.

Class	Precision	Recall	F1-Score	Support
Sober	0.84	0.75	0.79	850
Drunk	0.77	0.86	0.81	850
Accuracy			0.90	1700
Macro Avg	0.81	0.80	0.80	1700
Weighted Avg	0.81	0.80	0.80	1700

Table 6.2. Classification report on validation set after merging two datasets

6.3 BAC level labeling

For the PUCV-DTF dataset, information on subjects' Blood Alcohol Concentration (BAC) levels was available. Measurements were conducted across five states: sober, and after consuming 1, 2, 3, or 4 glasses of beer. The Table 6.3 provides breathalyzer test data for the first 13 subjects, illustrating significant variations in BAC levels. For example, subject 12 had a BAC level of 0.392 after just one beer, while subject 3 had a lower BAC (0.335) even after consuming three beers.

This variation occurs because alcohol affects individuals differently, depending on factors such as body weight, alcohol tolerance, metabolism, and more. As a result,

Subject	Sober	One beer	Two beers	Three beers	Four beers
1	0	0.121	0.211	0.401	0.560
2	0	0.000	0.251	0.558	0.886
3	0	0.000	0.200	0.335	0.491
4	0	0.111	0.368	0.637	1.004
5	0	0.187	0.388	0.529	0.767
6	0	0.294	0.603	0.784	0.884
7	0	0.257	0.278	0.392	0.542
8	0	0.295	0.626	0.867	0.996
9	0	0.187	0.419	0.626	0.794
10	0	0.191	0.420	0.559	0.766
11	0	0.225	0.388	0.556	0.823
12	0	0.392	0.683	0.784	1.174
13	0	0.200	0.213	0.432	0.633

Table 6.3. BAC level measurement results (PUCV-DTF database)

classifying individuals as "drunk" or "sober" based solely on the number of drinks consumed may not be entirely accurate.

To address this, we proposed a new labeling approach based on BAC levels. According to this method:

1. A person is classified as "drunk" if their BAC exceeds 0.5.

2. A person is classified as "sober" if their BAC is 0.

This approach aligns with the legal limit for blood alcohol concentration in many European countries, including Italy, where driving with a BAC exceeding 0.5‰ is prohibited. By adopting this threshold, we aimed to adhere to established standards and enhance clarity in our classification process.

Following this relabeling, our dataset comprised 6300 frames: 2300 sober frames and 4000 drunk frames. To ensure the validation data exclusively features unseen subjects from training, we meticulously partitioned it. Consequently, our training dataset consists of 5000 images (850 sober, 3150 drunk), while the validation dataset comprises 1300 images (450 sober, 850 drunk).

Given the dataset's imbalance, we opted for weighted Binary Cross-Entropy loss over standard BCE to address this skew.

After performing this new labeling method, we retrained the data and got the results that can be found in the Table 6.4:

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Sober	0.96	0.83	0.89	450
Drunk	0.92	0.98	0.95	850
Accuracy			0.93	1300
Macro Avg	0.94	0.91	0.92	1300
Weighted Avg	0.93	0.93	0.93	1300

Table 6.4. Classification report on validation set after relabelling

Based on the evaluation metrics, the classification model demonstrates high performance:

- Precision for identifying sober individuals is 0.96, indicating that when the model predicts a subject to be sober, it is correct 96% of the time. For drunk individuals, the precision is 0.92, indicating that when the model predicts a subject to be drunk, it is correct 92% of the time.
- Recall for sober individuals is 0.83, suggesting that the model correctly identifies 83% of all sober subjects in the dataset. Recall for drunk individuals is high at 0.98, meaning the model correctly identifies 98% of all drunk subjects.
- The F1-score for sober and drunk classification is 0.89 and 0.95, respectively, indicating a strong balance between precision and recall.
- The macro-average F1-score and weighted-average F1-score are 0.92 and 0.93, respectively, which proves strong overall performance across both classes.

However, it's crucial to emphasize that these results are not directly comparable to previous approaches due to the inclusion of different training and validation subjects. Moreover, the prevalence of the 'drunk' class in the training and validation results in the model learning this class weights better. This approach serves merely as an illustration of how a more precise and objective labeling strategy can impact model performance.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this research work, we delved into methodologies for identifying intoxicated individuals. Through a comprehensive review of existing literature, data collection, and the implementation of a CNN as the current benchmark model, we laid the groundwork for our exploration. Additionally, we proposed an end-to-end network tailored for non-contact alcohol measurement, leveraging a Convolutional Attention Network. This novel approach allows for the visualisation of attention masks, offering insights into potential regions of interest.

Our evaluation encompassed multiple datasets comprising both RGB and LWIR data, comparing our method against established CNN architectures. Impressively, our proposed method surpassed all prior state-of-the-art approaches, demonstrating its effectiveness across diverse demographics, skin types, and lighting conditions. These results confirm the generalisation capabilities of our supervised method to varying environmental factors and individual characteristics.

7.2 Limitations and future directions

7.2.1 Lack of training data

One of the primary limitations faced by this project is the lack of training data collected in a controlled environment, particularly in the thermal and near-infrared domains. A table presenting publicly available data and its characteristics can be in Section 3 and Table 3.1.

This limitation arises from the ethical concerns associated with inducing alcohol consumption in subjects and recording them to create a dataset. To address this limitation, expanding the research and collecting a new, more diverse dataset including subjects of various age and ethnic groups would contribute to training more effective and generalisable models.

Regarding the thermal data, which captures thermal information about the face, there is a need for further in-depth research and comparisons. This entails studying subjects with variations in face temperature due to factors like fever, exercise, and other causes of facial temperature changes. Additionally, conducting research with subjects wearing face masks, glasses, and other potential obstacles could enhance the current model's performance.

Furthermore, as highlighted in earlier sections, there is limited existing research on classifying drunk individuals using NIR sensor data. These sensors, often more

cost-effective, present opportunities for application in diverse situations, necessitating further exploration and investigation.

7.2.2 BAC prediction level

Another avenue for potential future work involves the prediction of precise Blood Alcohol Concentration levels based on facial information. Currently, the classification has been conducted using 'raw' definitions of drunk and sober. For instance, in the Sober-Drunk dataset, individuals who have consumed three glasses of wine were categorised as drunk, while in the RGB data, we relied on the labels provided in the YouTube video names. However, the impact of the same amount of alcohol can vary among individuals.

Several factors influence how intoxicated an individual becomes after consuming alcohol [46]. These include:

- body weight,
- gender,
- metabolism,
- tolerance,
- rate of consumption,
- alcohol content and type,
- medications and health conditions,
- emotional state and environment.

For example, in the Table 6.3, we already noticed that one of the subjects was 'drunker' than the other even if he only drank one beer instead of three. Therefore, the inclusion of BAC measurements, obtained through breathalyzer or blood tests, is crucial for making more objective predictions. This approach not only allows us to determine if a person exceeds the drunk limit but also paves the way for studying the exact level of drunkenness through advanced techniques such as multilabel classification or regression tasks.

7.2.3 Temporal sequences for alcohol intoxication prediction

Up to this point, our focus has been on predicting subjects' levels of intoxication using individual frames. However, particularly in the case of thermal images, we've observed instances where the model misclassifies different frames of the same subject (Figure 4.10). This discrepancy can be attributed to factors like blood vessel activity and fluctuations in temperature under the nose caused by breathing patterns.

Given this observation, a promising avenue for future research involves leveraging the entire sequence of frames as input data. By employing recurrent neural network (RNN) architectures such as Long Short-Term Memory (LSTM) networks or other suitable models, we can potentially capture temporal dependencies and patterns across frames. This approach may lead to more accurate and robust predictions by considering the dynamic evolution of facial temperature over time.

Bibliography

[1] Linear/Fully-Connected Layers User's Guide - NVIDIA Docs — docs.nvidia.com. https://docs.nvidia.com/deeplearning/performance/dl-performance-fully-connected/index.html. [Accessed 04-05-2024].

[2] What Are Channels in Convolutional Networks? | Baeldung on Computer Science — baeldung.com. https://www.baeldung.com/cs/cnn-channels. [Accessed 04-05-2024].

[3] Marcos Alberti. WINE PROJECT — masmorrastudio.com. https://www.masmorrastudio.com/wine-project. [Accessed 04-05-2024].

[4] Yin Aphinyanaphongs, Bisakha Ray, Alexander Statnikov, and Paul Krebs. Text Classification for Automatic Detection of Alcohol Use-Related Tweets: A feasibility study. 2014.

[5] Helen F Ashdown, Susannah Fleming, Elizabeth A Spencer, Matthew J Thompson, and Richard J Stevens. Diagnostic accuracy study of three alcohol breathalysers marketed for sale to the public. *BMJ Open*, 4(12), 2014.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[7] M. K. Bhuyan, Kangkana Bora, and Georgia Koukiou. Detection of intoxicated person using thermal infrared images. In *2019 IEEE 6th Asian Conference on Defence Technology (ACDT)*, pages 59–64, 2019.

[8] M. K. Bhuyan, Suchit Dhawle, Pradipta Sasmal, and Georgia Koukiou. Intoxicated person identification using thermal infrared images and gait. In *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 1–3, 2018.

[9] Fadi Biadsy, William Wang, Andrew Rosenberg, and Julia Hirschberg. Interspeech 2011 intoxication detection using phonetic, phonotactic and prosodic cues. pages 3209–3212, 08 2011.

[10] Abraham Albert Bonela, Zhen He, Aiden Nibali, Thomas Norman, Peter G. Miller, and Emmanuel Kuntsche. Audio-based deep learning algorithm to identify alcohol inebriation (adlaia). *Alcohol*, 109:49–54, 2023.

[11] Ty Brumback, Dingcai Cao, and Andrea King. Effects of alcohol on psychomotor performance and perceived impairment in heavy binge social drinkers. *Drug and alcohol dependence*, 91:10–7, 11 2007.

[12] L. Causa, J. E. Tapia, E. Lopez-Droguett, A. Valenzuela, D. Benalcazar, and C. Busch. Behavioural curves analysis using near-infrared-iris image sequences, 2022.

[13] Robert Chen-Hao Chang, Chia-Yu Wang, Hsin-Han Li, and Cheng-Di Chiu. Drunk driving detection using two-stage deep neural network. *IEEE Access*, 9:116564–116571, 2021.

[14] Weixuan Chen and Daniel McDuff. Deepphys: Video-based physiological measurement using convolutional attention networks, 2018.

[15] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[17] Ahmad Eid. *Combined Classifiers for Invariant Face Recognition*. PhD thesis, 06 2004.

[18] Directorate General for Transport European Road Safety Observatory. Brussels, European Commission. Road safety thematic report – alcohol and drugs, 2023.

[19] PhD Everton Gomede. Understanding the Learning Mechanism of Convolutional Neural Networks. https://pub.aimind.so. [Accessed 04-05-2024].

[20] Marcos Grzeca, Karin Becker, and Renata Galante. Improving the classification of drunk texting in tweets using semantic enrichment. pages 190–197, 12 2018.

[21] Jens Hashagen. Swir applications and challenges: A primer, Jun 2015.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[23] Gabriel Hermosilla, José Verdugo, Gonzalo Farias, Esteban Vera, Francisco Pizarro Torres, and Margarita Machuca. Face recognition and drunk classification using infrared face images. *Journal of Sensors*, 2018:1–8, 01 2018.

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

[25] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[26] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[27] Nguyen Huynh Phuong Thanh and Kha Huynh. Drunkenness detection using a cnn with adding gaussian noise and blur in the thermal infrared images. *International Journal of Intelligent Information and Database Systems*, 1:1, 01 2022.

[28] Pisit Iamudomchai, Pattanawadee Seelaso, Satjana Pattanasak, and Wibool Piyawattanametha. Deep learning technology for drunks detection with infrared camera. In *2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, pages 1–4, 2020.

[29] Aditya Joshi, Abhijit Mishra, Balamurali R, Pushpak Bhattacharyya, and Mark Carman. A computational approach to automatic prediction of drunk-texting. 07 2015.

[30] Aditya K. Kamath, A. Tarun Karthik, Leslie Monis, Manjunath Mulimani, and Shashidhar G. Koolagudi. Sobriety testing based on thermal infrared images using convolutional neural networks. In *TENCON 2018 - 2018 IEEE Region 10 Conference*, pages 2170–2174, 2018.

[31] Vighnesh Bhaskar Kamath, Sagar S Pai, Shwetha S Poojary, Ananth Rastogi, and K S Srinivas. Drunkenness face detection using graph neural networks. In *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pages 1–6, 2021.

[32] G. Koukiou and V. Anasassopoulos. Face locations suitable drunk persons identification. In *2013 International Workshop on Biometrics and Forensics (IWBF)*, pages 1–4, 2013.

[33] G Koukiou and V Anastassopoulos. Infrared signature of facial blood vessels for intoxicated person discrimination. *Australian J Forens Sci*, 48:326–338, 2016.

[34] Georgia Koukiou. Intoxicated person identification using markov chains and neural networks. *Neural Computing and Applications*, 33, 04 2021.

[35] Georgia Koukiou. Thermal biometric features for drunk person identification using multi-frame imagery. *Electronics*, 11(23), 2022.

[36] Georgia Koukiou and Vassilis Anastassopoulos. Sober-drunk database. http://physics.upatras.gr/sober/index.html, 2008. [Online; accessed 19-July-2022].

[37] Georgia Koukiou and Vassilis Anastassopoulos. Drunk person identification using thermal infrared images. *International Journal of Electronic Security and Digital Forensics*, 4:229–243, 03 2012.

[38] Georgia Koukiou and Vassilis Anastassopoulos. Eye temperature distribution in drunk persons using thermal imagery. In *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, pages 1–8, 2013.

[39] Georgia Koukiou and Vassilis Anastassopoulos. Drunk person screening using eye thermal signatures. *Journal of Forensic Sciences*, 61, 11 2015.

[40] Georgia Koukiou and Vassilis Anastassopoulos. Drunk person identification using local difference patterns. In *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 401–405, 2016.

[41] Georgia Koukiou and Vassilis Anastassopoulos. Fusion of dissimilar features from thermal imaging for improving drunk person identification. *International Journal of Signal Processing Systems*, 5:106–111, 09 2017.

[42] Jan Kubicek, Dominik Vilimek, Alice Krestanova, Marek Penhaker, Eva Kotalova, Bastien Faure-Brac, Clément Noel, Radomir Scurek, Martin Augustynek, Martin Cerny, and Tomas Kantor. Prediction model of alcohol intoxication from facial temperature dynamics based on k-means clustering driven by evolutionary computing. *Symmetry*, 11(8), 2019.

[43] Ruojun Li, Emmanuel Agu, Atifa Sarwar, Kristin Grimone, Debra Herman, Ana Abrantes, and Michael Stein. Fine-grained intoxicated gait classification using a bilinear cnn. *IEEE Sensors Journal*, PP:1–1, 12 2023.

[44] Suman Kalyan Maity, Ankan Mullick, Surjya Ghosh, Anil Kumar, Sunny Dhamnani, Sudhansu Bahety, and Animesh Mukherjee. Understanding psycholinguistic behavior of predominant drunk texters in social media. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 01096–01101, 2018.

[45] Victor-Emil Neagoe and Serban-Vasile Carata. Drunkenness diagnosis using a neural network-based approach for analysis of facial images in the thermal infrared spectrum. In *2017 E-Health and Bioengineering Conference (EHB)*, pages 165–168, 2017.

[46] Marketing Communications: Web // University of Notre Dame. Absorption Rate Factors // Rev. James E. McDonald, C.S.C., Center for Student Well-Being // University of Notre Dame — mcwell.nd.edu. https://mcwell.nd.edu/your-well-being/physical-well-being/alcohol/absorption-rate-factors/. [Accessed 10-05-2024].

[47] Suah Park, Byunghoon Bae, Kyungmin Kang, Hyunjee Kim, Mi Song Nam, Jumyung Um, and Yun Jung Heo. A deep-learning approach for identifying a drunk person using gait recognition. *Applied Sciences*, 13(3), 2023.

[48] Laavanya Rachakonda, Saraju Mohanty, and Elias Kougianos. Donot-dueye: An iot enabled edge device to monitor blood alcohol concentration from eyes. In *2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, pages 87–92, 2019.

[49] Sivakumar Rajagopal, Deepikaa Balaji, Vidhyalakshmi Venkatesh, Kanishka S, and Rahul Soangra. Identification of drunk people among crowds using thermography and machine learning. *ECS Transactions*, 107(1):1867, apr 2022.

[50] Snehal Rathi, Omkar Mirajkar, Shubhangi Shukla, Laukik Deshmukh, and Lokesh Dangare. Advancing crack detection using deep learning solutions for automated inspection of metallic surfaces. *Indian Journal of Information Sources and Services*, 14:93–100, 03 2024.

[51] Nilu Salim, Srinath Venkobarao, Umarani Jayaraman, and Phalguni Gupta. Recognition in the near infrared spectrum for face, gender andfacial expressions. *Multimedia Tools and Applications*, 81, 01 2022.

[52] Björn Schuller, Anton Batliner, Stefan Steidl, Florian Schiel, and Jarek Krajewski. The interspeech 2011 speaker state challenge. pages 3201–3204, 08 2011.

[53] SAGAR SHARMA. Activation Functions in Neural Networks — towardsdatascience.com. https://towardsdatascience.com/

activation-functions-neural-networks-1cbd9f8d91d6. [Accessed 04-05-2024].

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

[55] Alexandru-Ionel Soltuz and Victor-Emil Neagoe. Facial thermal image analysis with deep convolutional neural network architectures for subject dependent drunkenness diagnosis. In *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–4, 2021.

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[57] Juan Tapia, Daniel Benalcazar, Andres Valenzuela, Leonardo Causa, Enrique Lopez Droguett, and Christoph Busch. Learning to predict fitness for duty using near infrared periocular iris images, 2022.

[58] Juan Tapia, Enrique Lopez Droguett, and Christoph Busch. Alcohol consumption detection from periocular nir images using capsule network. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 959–966, 2022.

[59] Jerin Varghese and Sarika Dakhode. Effects of alcohol consumption on various systems of the human body: A systematic review. *Cureus*, 14, 10 2022.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[61] Gabriel Hermosilla Vigneau, Jose Luis Verdugo, Gonzalo Farías Castro, Esteban Vera, Francisco Pizarro, and Margarita Machuca. Face recognition and drunk classification using infrared face images. *J. Sensors*, 2018:5813514:1–5813514:8, 2018.

[62] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[63] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks, 2018.

[64] Colin Willoughby, Ian Banatoski, Paul Roberts, and Emmanuel Agu. Drunkselfie: Intoxication detection from smartphone facial images. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 496–501, 2019.

[65] Devendra Pratap Yadav and Abhinav Dhall. DIF : Dataset of intoxicated faces for drunk person identification. *CoRR*, abs/1805.10030, 2018.

[66] Ying Yao, Xiaohua Zhao, Hongji Du, Yunlong Zhang, Guohui Zhang, and Jian Rong. Classification of fatigued and drunk driving based on decision tree methods: A simulator study. *International Journal of Environmental Research and Public Health*, 16:1935, 05 2019.

[67] Pamela C. Zurita, Daniel P. Benalcazar, and Juan E. Tapia. Fitness-for-duty classification using temporal sequences of iris periocular images, 2023.