1290

UNIVERSIDADE Đ
COIMBRA

Gonçalo Rebelo de Almeida Moreira

Neuromorphic Event-based Facial Identity Recognition

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, no ramo de Automação, orientada pelo Professor Doutor Jorge Manuel Moreira de Campos Pereira Batista e apresentada ao Departamento de Engeharia Eletrotécnica e de Computadores.

Outubro de 2021

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

Gonçalo Rebelo de Almeida Moreira

Neuromorphic Event-based Facial Identity Recognition

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, no ramo de Automação, orientada pelo Professor Doutor Jorge Manuel Moreira de Campos Pereira Batista e apresentada ao Departamento de Engeharia Eletrotécnica e de Computadores.

Outubro de 2021

Agradecimentos

Quero começar por agradecer ao meu orientador Professor Doutor Jorge Batista pelo voto de confiança que me deu para embarcar num tema altamente inovador e pouco explorado, e por todo o apoio incansável que me foi dando ao longo do ano, com vista a fazer o melhor trabalho possível. Quero também agradecer aos meus colegas de laboratório Bruno Silva, Pedro Martins, Eurico Almeida, André Graça e Alessio Silva pela ajuda e *brainstorming* que proporcionaram, alavancando o meu trabalho.

Quero agradecer aos meus pais, Isabel Rebelo e João Nuno Moreira, aos meus irmãos, Sofia Moreira e Pedro Moreira, aos meus avós, Luísa, Álvaro, Milú e Fernando, aos meus tios, Anica e João, Daniela e Nuno, e Marissol e Pedro, e aos meus primos, Teresinha, Vasco, Leonor, Miguel e Rodrigo, por todos e quaisquer momentos pelos quais passamos e que me ajudaram a crescer e completar-me como pessoa para além da Universidade. Sem a sua influência em mim, hoje não era quem sou e nada disto seria possível. Estou eternamente grato pela vida que me deram e pelas oportunidades que me proporcionaram.

Quero agradecer a todos os meus amigos e contemporâneos da faculdade, por todas as vivências e experiências que proporcionaram e que melhoraram a minha vivência académica na Universidade de Coimbra. É impossível mencionar todos, mas faço especial referência aos colegas de curso Alessio Silva, Afonso Castiço, Tomás Pedrosa, Tomé Ventura, André Galvão, David Pereira, Martin Estorninho, Hugo Sousa e Hugo Ferreira. Quero também agradecer a todos os amigos que o basquete me deu ao longo dos últimos anos, uma autêntica segunda família, não irei referir nenhum, porque não é possível mencionar um, sem mencionar todos. Mas sem dúvida tornaram a minha experiência académica, social, desportiva e pessoal absolutamente inesquecível. Muito obrigado a todos!

Quero e tenho de agradecer à minha namorada Madalena Cardoso que tanto contribuiu para a minha vida académica, como para o meu crescimento pessoal, por todo o apoio que me deu nestes anos, e pela ajuda que me deu a ultrapassar os vários obstáculos que foram aparecendo pelo caminho. Acima de tudo, ensinou-me a amar e a viver a vida com mais paixão, e é, para mim, uma inspiração para ser melhor todos os dias.

Não posso deixar de agradecer à família da Madalena que tanto enriqueceram a minha vivência académica, que tanto influenciaram e tornaram especial o caminho que percorri nos últimos anos dentro e fora do curso, e por me terem acolhido como família.

Resta-me agradecer a toda a gente que teve influência na pessoa em que me tornei e que aqui não foi mencionada, amigos de longa data, familiares mais distantes, professores e treinadores.

Dedico este trabalho ao falecido Professor Catedrático jubilado, ex-Reitor da Universidade de Coimbra e meu avô, Fernando Manuel da Silva Rebelo, por toda a sua contribuição de excelência para a Universidade na qual, hoje, orgulhosamente, me formo, e por me ter inspirado em tornar-me na pessoa que hoje sou e que no futuro aspiro ser.

Abstract

Facial recognition research has been around for longer than a half-century, as of today. This great interest in the field stems from its tremendous potential to enhance various industries, such as video surveillance, personal authentication, criminal investigation, and leisure. Most state-of-the-art algorithms rely on facial appearance, particularly, these methods utilize the static characteristics of the human face (*e.g.*, the distance between both eyes, nose location, nose shape) to determine the subject's identity extremely accurately. However, it is further argued that humans also make use of another type of facial information to identify other people, namely, one's idiosyncratic facial motion. This kind of facial data is relevant due to being hardly replicable or forged, whereas appearance can be easily distorted by cheap software available to anyone.

On another note, event-cameras are quite recent neuromorphic devices that are remarkable at encoding dynamic information in a scene. These sensors are inspired by the biological operation mode of the human eye. Rather than detecting the light intensity, they capture light intensity variations in the setting. Thus, in comparison to standard cameras, this sensing mechanism has a high temporal resolution, therefore it does not suffer from motion blur, and has low power consumption, among other benefits. A few of its early applications have been real-time Simultaneous Localization And Mapping (SLAM), anomaly detection, and action/gesture recognition.

Taking it all into account, the main purpose of this work is to evaluate the aptitude of the technology offered by event-cameras for completing a more complex task, that being facial identity recognition, and how easily it could be integrated into real world systems. Additionally, it is also provided the Dataset created in the scope of this dissertation (NVSFD Dataset) in order to facilitate future third-party investigation on the topic.

Keywords: Neuromorphic Vision; Facial Identity Recognition; Event-Cameras; Facial Dynamics; NVSFD Dataset

Resumo

A investigação na área do reconhecimento facial existe já há mais de meio século. O grande interesse neste tópico advém do seu tremendo potencial para impactar várias indústrias, como a de vídeovigilância, autenticação pessoal, investigação criminal, lazer, entre outras. A maioria dos algoritmos estado-da-arte baseiam-se apenas na aparência facial, especificamente, estes métodos utilizam as caraterísticas estáticas da cara humana (*e.g.*, a distância entre os olhos, a localização do nariz, a forma do nariz) para determinar com bastante eficácia a identidade de um sujeito. Contudo, é também discutido o facto de que os humanos fazem uso de outro tipo de informação facial para identificar outras pessoas, nomeadamente, o movimento facial idiossincrático de uma pessoa. Este conjunto de dados faciais é relevante devido a ser difícil de replicar ou de falsificar, enquanto que a aparência é facilmente alterada com ajuda de ferramentas computacionais baratas e disponíveis a qualquer um.

Por outro lado, câmaras de eventos são dispositivos neuromórficos, bastante recentes, que são ótimos a codificar informação da dinâmica de uma cena. Estes sensores são inspirados pelo modo de funcionamento biológico do olho humano. Em vez de detetarem as várias intensidades de luz de uma cena, estes captam as variações dessas intensidades no cenário. De modo que, e comparando com câmaras *standard*, estes mecanismos sensoriais têm elevada resolução temporal, não sofrendo de imagem tremida, e são de baixo consumo, entre outros benefícios. Algumas das suas aplicações são Localização e Mapeamento Simultâneo (SLAM) em tempo real, deteção de anomalias e reconhecimento de ações/gestos.

Tomando tudo isto em conta, o foco principal deste trabalho é de avaliar a aptidão da tecnologia fornecida pelas câmaras de eventos para completar tarefas mais complexas, neste caso, reconhecimento de identidade facial, e o quão fácil será a sua integração num sistema no mundo real. Adicionalmente, é também disponibilizado o Dataset criado no âmbito desta dissertação (NVSFD Dataset) de modo a possibilitar investigação futura sobre o tópico.

Palavras-Chave: Visão Neuromórfica; Reconhecimento de Identidade Facial; Câmaras de Eventos; Dinâmica Facial; NVSFD Dataset

Contents

List of Figures

List of Tables

List of Acronyms

AETS Accumulated Exponential Time Surface. 9–11, 36

API Application Programming Interface. 28

CNN Convolutional Neural Network. 4

ConvNet Convolutional Neural Network. ix, xi, xv, 4, 14–16

HATS Histogram of Averaged Time Surfaces. 25

HOTS Histogram Of Time Surfaces. 25

I3D Two-Stream Inflated 3D ConvNet. ix, xi, 15–18, 25, 31, 35, 59

LIF Leaky Integrate-and-Fire. 12, 13, 36

MSE Mean Squarred Error. 18

NVSFD Neuromorphic-Vision for Speech-induced Facial Dynamics. v, vii, 28, 31–33, 71

RGB Red-Green-Blue. xi, 16, 17, 23, 59

SAE Surface of Active Events. 8, 9, 36

SGD Stochastic Gradient Descent. ix, 19, 31, 35

SKIM Synaptic Kernel Adaptation Method. 25

SLAM Simultaneous Localization And Mapping. v, vii, 1

TBR Temporal Binary Representation. 6–8, 31, 36

1 Introduction

This chapter has the objective to state the thesis of this dissertation, discuss the objectives and contributions, and preview the structure of the document.

1.1 Context and Motivation

Face Recognition algorithms have been around for the past half-century and have maintained their relevance and interest around the Computer Vision research community, thus far. This comes by as no surprise for there is still room for enhancements in robustness and computational cost despite the already great recognition rates. Consequently, a wide range of techniques have been explored and proposed with the intention of achieving these goals.

Furthermore, a trending philosophy of problem solving and algorithm implementation is rising, especially in the field of Computer Vision - bio-inspired techniques. Like the name suggests, the concession of these types of methodologies are inspired by biological systems, in a way that the techniques emulate their functioning. Since everything in nature operates very efficiently and optimally due to the countless years of evolution, it is wise to design systems that mimic its behavior. Thereupon, and regarding facial recognition, the variables humans take into account in the face perception process have been examined. It has been concluded that apart from the static appearance of a face, its characteristic movement also conveys a lot of information on which humans naturally rely on to recognize someone's facial identity [1].

Along with it, a new bio-inspired sensor has taken the Computer Vision and Robotics world by storm - Neuromorphic Vision Sensors - also known as Event-Cameras. This kind of cameras are inspired by the biological working of the human retina. Unlike conventional cameras that output intensity images at a constant frame rate, Event-Cameras asynchronously measure light variation on each pixel and output a stream of events that encode time of activation, pixel position and sign of the variation (polarity). This type of sensor has a number of advantages over the standard camera, such as: high temporal resolution (in the order of μs); very high dynamic range (140 dB), which allows it to perform under extreme lighting conditions; low power consumption, since there is no need to process light information throughout the whole frame, just a simple asynchronous calculation of light variation at each pixel; high pixel bandwidth (in the order of the kHz) which contributes to motion blur reduction [2].

This type of sensors are already used in many Computer Vision and Robotics applications, such as Object Detection and Tracking, Gesture Recognition, Face Detection, Anomaly Detection and SLAM, however, very little work has been published with focus on Facial Recognition.

1.2 Objective

The main purpose of this work is to explore the potential of Neuromorphic Vision Sensors to perform a complex task in facial recognition through facial dynamics derived from speech and to validate how contributory facial motion is for face perception. Secondly, a complementary aim is to publish a Neuromorphic Facial Motion Dataset, which not only served this work's investigation, but also to ease and allow further research on this topic.

1.3 Contributions

The two main contributions of the work developed in the scope of this dissertation are firstly, the creation of a novel approach to perform facial recognition tasks using Neuromorphic Vision Sensors, facial dynamics and a network architecture first conceived to perform action classification, and, secondly, the publication of the Dataset utilized to complete this work with the view to grant the continuity of investigation on this field.

Important Disclaimer: All the participants who volunteered to take part in the Dataset formally consented its use for this work and future third-party investigation.

1.4 Dissertation Structure

This document is divided in six different chapters, of which the first one - Section 1 - is the current one and has introductory purposes. Secondly, in Section 2, some background scientific context is given so as to introduce important concepts about this subject. Subsequently, in Section 3, some relevant applications and studies conducted among the researching community, which contributed or inspired this work in some way or another, are covered and delved into. Thereafter, Section 4 deep dives into the actual work that was conducted in order to prove and defend the thesis of this dissertation. Later, on Section 5, the outcome of the work described in the previous section is carefully reported and discussed intending to clarify some of the questions raised in the scope of this thesis. Finally, Section 6 summarizes all the relevant aspects presented in previous sections, and sheds a light on possible future research paths and on further questions that still need to be answered.

2 Background Knowledge

In this section, a batch of important concepts is presented to better understand the intricacies of this work, while introducing some fundamental terminology as well.

2.1 Neuromorphic Vision Sensors

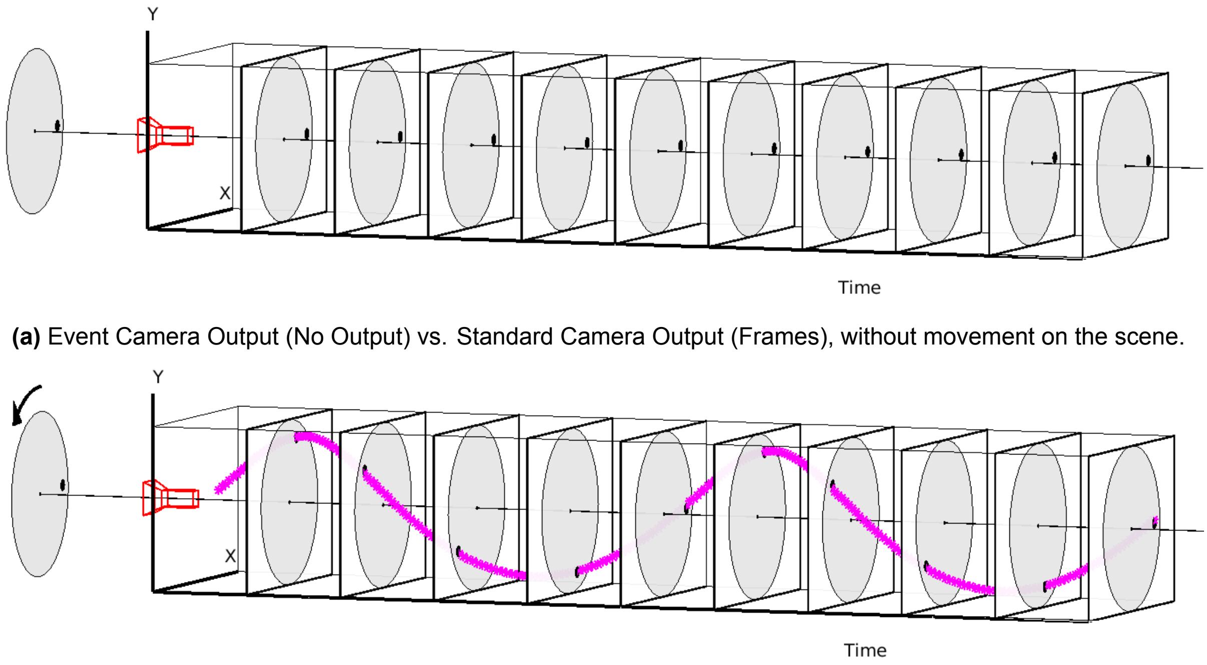
Neuromorphic Vision Sensors, commonly referred to as Event-Cameras, have a distinctive operating mode from standard cameras, as mentioned previously in the section 1.1. For better understanding of the functioning of the camera a visualization is provided with a simple case on figure 1, with two different situations being portrayed. The first scenario (figure 1a) has the camera capture a non-rotating disc with a black dot on it, and since there is no movement on the scene, the event camera does not output any information for there is no light variation, while a standard camera would still capture a sequence of frames at a fixed rate. On the latter case (figure 1b) the disc is rotating, which induces variation of light in the scene caused by the circular motion of the black dot, consequently, triggering the Event-Camera to output a three dimensional point cloud composed of spatial and temporal information. Each element of the point cloud can be represented by a triad as shown in equation (1) and it is usually designated as an Event, e_k.

$$e_k = (x_k, y_k, t_k), \forall k \tag{1}$$

The (x_k, y_k) pair represents the pixel location on the frame where the event was triggered, while t_k represents the time at which the event occurred. Thus, e_k represents the k-th occurrence of an event. Nevertheless, at a higher tier of event encoding, both the sign and value of light variation can be contemplated, which in turn gives way to a richer descriptor of the event, presented in equation (2).

$$e_k = (x_k, y_k, t_k, p_k, v_k), \forall k \tag{2}$$

The sign of the light change that triggers an event is commonly referred to as Polarity, represented by pk, and its value is 1 for Positive variation of light and -1 for Negative variation. On the other hand, the value of the change of light represents how steep that variation was. However, for this work, only the event descriptor given by equation (1) was used.



(a) Event Camera Output (No Output) vs. Standard Camera Output (Frames), without movement on the scene.

(b) Event Camera Output (Magenta) vs. Standard Camera Output (Rotating Disc Frames), with movement on the scene.

Figure 1: Event Camera Output (Magenta) vs. Standard Camera Output (Rotating Disc Frames) both with and without movement on the scene captured from the angle of the Red Camera.

2.2 Frame Representation of Events

The intrinsic nature of an Event-Camera output makes any kind of conventional image processing like a Convolutional Neural Network (CNN), also known as ConvNet, seemingly unfeasible to process it. In fact, there are methodologies that work directly over the event cloud. However, since research using intensity images is decades ahead of the one using event clouds, not only is it viable to use the already available techniques, but it could also present itself as more advantageous. Yet, the event data must, first, be adapted to work with those conventional techniques, which implies converting the point clouds to frames. In order to do that, a Frame Time must be chosen - this is the time taken to accumulate all the events that will take part of each frame. Naturally, it will both determine the frame rate and also how many events contribute to the construction of each image. The image on figure 2 illustrates the concept.

This process, in turn, carries its own disadvantages. One could argue that this type of data formatting neglects the inherent higher temporal resolution of these cameras. Moreover, another argument is that converting a sparse set of data into a dense set such as an image frame, will result in the unavoidable increment of the data size. For instance, if there is a small number of events on a scene, the converted frames will have a lot of pixels set to 0, which is redundant. On the flip side, a Point Cloud solely conveys information about the activated pixels,

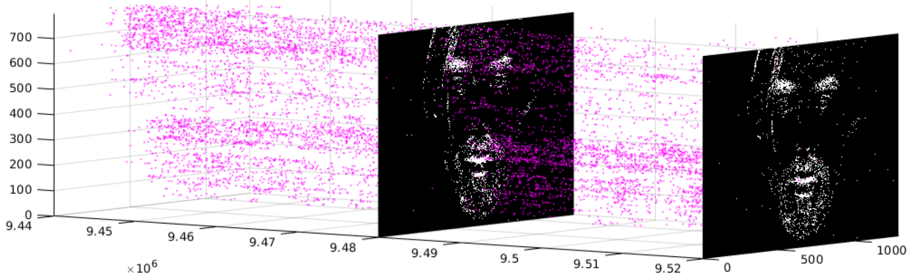


Figure 2: Event-Cloud to Event Frames: This image portrays a small proportion of the event cloud (magenta) and the event frames (black and white) constructed from it. The temporal slice between both frames is called the Frame Accumulation Time or simply Frame Time. All events triggered during this period will be used to build the next event frame.

hence, cutting down the size of the whole video sequence.

Despite the two valid arguments presented, none of them pose too big of a problem for the intended analysis of this work. Perhaps the temporal resolution loss with the conversion may be a factor that might have had a small impact on the results, and should further be studied, however, it was not relevant enough throughout the experimentation process of this work. Furthermore, many other studies have had exceedingly satisfactory results using these types of event representation.

2.2.1 Binary Event Representation

The Binary Event Representation is the simplest representation method to implement. Consequently, it also is the one which conveys the least amount of information. In essence, during the frame accumulation time if an event is triggered at least once, it immediately sets that pixel to its maximum value. In other words, a pixel is either ON or OFF. This, of course, neglects all the temporal resolution of the event cloud that was generated during that frame time. Furthermore, it is especially susceptible to event noise, meaning all noisy events will have the same value as other relevant and, perhaps, more frequent ones. The event frame on figure 3 depicts a real example of an image frame built using this Binary Representation.

Every single pixel that was acitvated during the Frame Accumulated Time is represented with maximum intensity on the image frame. In turn, this makes it hyper sensitive to any existing noise in the scene. For its loss of temporal information, this method was not taken into account on any of the experiments.

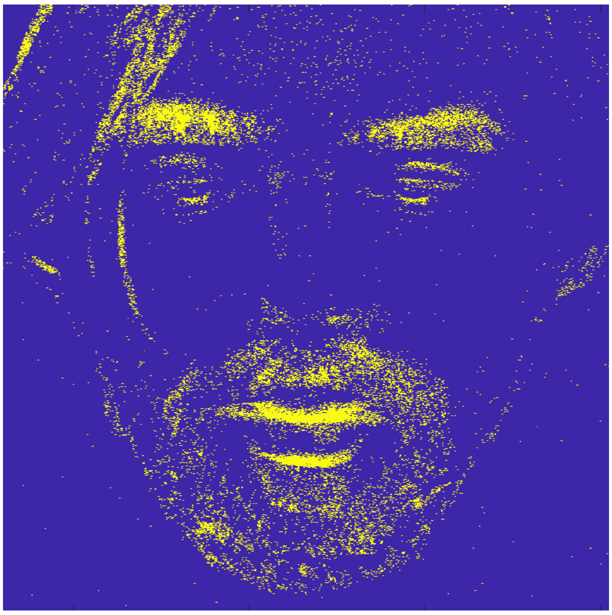


Figure 3: Example of a Binary Event Frame: This is a Binary Event frame of a user face integrated during $40ms$. The yellow frame pixels have the maximum intensity, whereas the dark blue colored pixels have the lowest intensity.

2.2.2 Temporal Binary Representation

The Temporal Binary Representation (TBR) was first proposed by [3]. Its core concept is based on slicing the Frame Accumulation Time evenly, in a predetermined number of slots. For demonstration purposes, figure 4 illustrates the idea with a practical example. Additionally, the algorithm used to compose a TBR frame is depicted in algorithm 1.

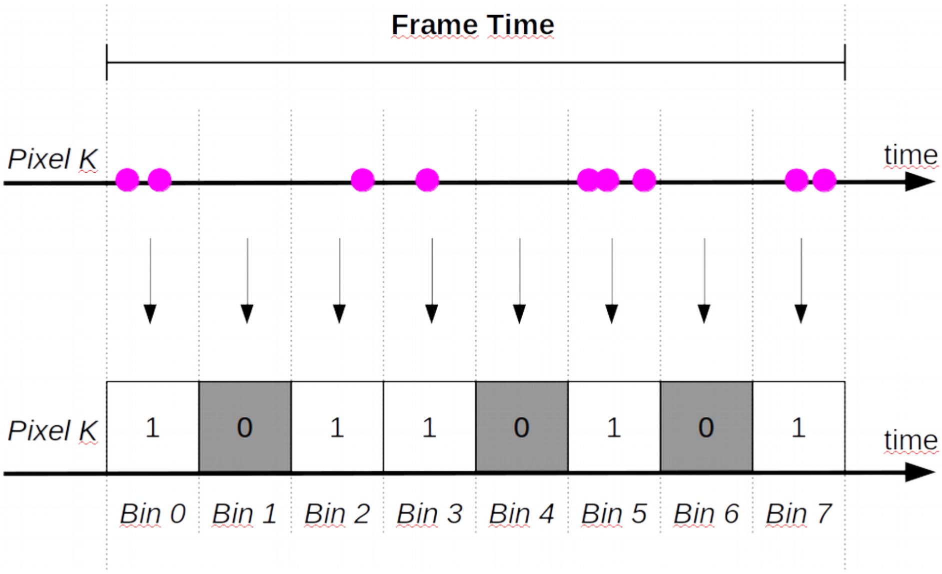


Figure 4: Temporal Binary Representation Frame Pixel Construction: During the Frame Time, Pixel K was activated multiple times (magenta dots). The Frame Time is then divided in 8 time bins and for any activation during N-th Bin, that Bin is immediately activated. Thus, this means that the final value of Pixel K for this current Frame is $(10101101)_2 = (173)_{10}$.

Algorithm 1: Algorithm for constructing a TBR frame.

```
Data: All Event Cloud Points that occurred during the Frame Accumulation Time
$BIN_{NO} \leftarrow$ Number of Total Bins;
$\Delta T \leftarrow$ Frame Time;
$t_0 \leftarrow$ Initial Timestamp of the Frame;
$F \leftarrow$ Frame to be constructed;
for Each Incoming Event given by $(x_i, y_i, t_i)$ do
    $bin_n \leftarrow$ floor($\frac{t_i - t_0}{\Delta T} \times BIN_{NO}$) ;          /* get the slot number */
    $pixel_{bin} \leftarrow$ decimal2binary($F(x_i, y_i)$);
    if $pixel_{bin}[bin_n] = 0$ then
        $F(x_i, y_i) \leftarrow F(x_i, y_i)) + 2^{bin_n}$;          /* activate bin, if not already */
    end
end
Result: $F$
```

The image presented in figure 5 is an example of a final frame constructed through this representation. It is clear that the most active facial features are more noticeable, meaning those were the regions with most movement on the scene.

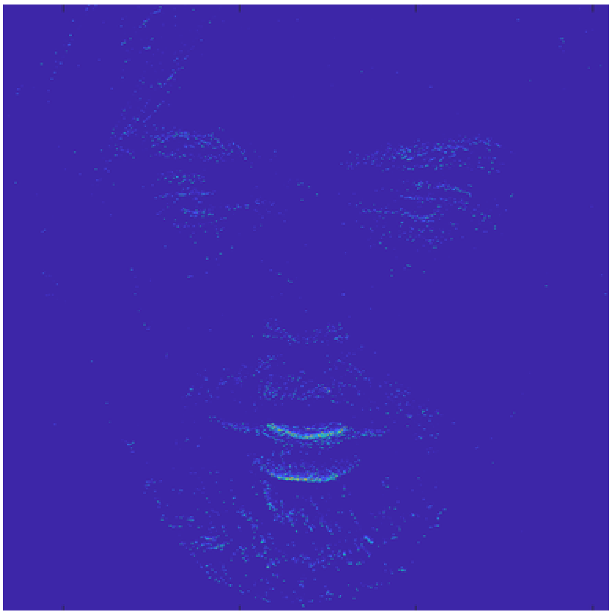


Figure 5: Example of a Temporal Binary Representation Frame: This is a TBR frame of a user face integrated during $40ms$. It can be observed that there is a lot less noise compared to the previous method. Moreover, the intensity variations are noticeable

A great feature of this methodology compared to the others ones is that exact temporal information of the events is encoded in each pixel. Although still losing temporal resolution, this method has $(BIN_{NO})\times$ more resolution than the Binary Event Representation. Namely, if the total number of Bins is 8, $BIN_{NO} = 8$, then it has $8\times$ more resolution. In fact, if $BIN_{NO} = 1$, then they are the same representation. However, this feature is slightly altered if some affine transformations are applied to the image, which makes it rather irrelevant when data augmentation is used in the training phases of a Neural Network. Nevertheless, it is just as valid and useful a representation as the other ones.

2.2.3 Surface of Acitve Events

The Surface of Active Events (SAE) is the oldest frame representation [4] but equally effective and simple to implement. Similarly to TBR, it is a timestamp normalization technique, yet, instead of time discretization through time slots, it works directly with the time value normalization. The algorithm 2 depicts how a SAE `uint8` frame can be constructed.

Algorithm 2: Algorithm for constructing a SAE frame.

```
Data: All Event Cloud Points that occurred during the Frame Accumulation Time
$\Delta T \leftarrow$ Frame Time;
$t_0 \leftarrow$ Initial Timestamp of the Frame;
$F \leftarrow$ Frame to be constructed;
for Each Incoming Event given by $(x_i, y_i, t_i)$ do
    $F(x_i, y_i) = \texttt{round}(255 \cdot \frac{t_i - t_0}{\Delta T})$
end
Result: $F$
```

Figure 6 displays an exmaple frame obtained through the SAE representation. As it can be observed, this is a method that is quite rich in terms of the amplitude of values, so regions of high activity are quite emphasized.

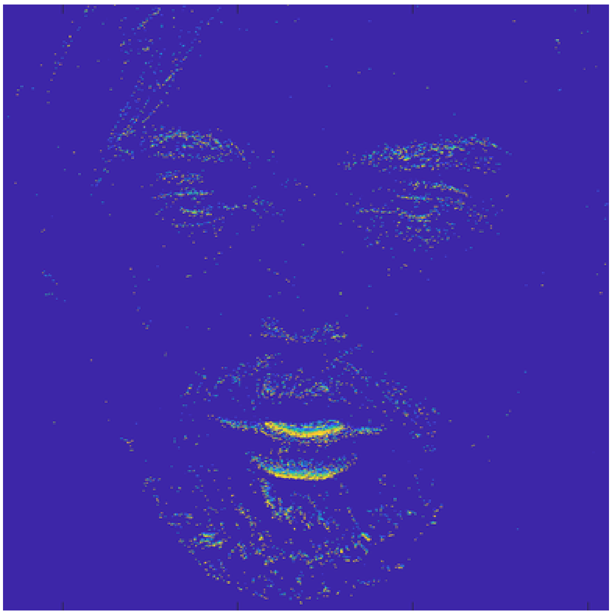


Figure 6: Example of a Surface of Active Events Frame: This is a SAE frame of a user face integrated during $40ms$. It is noticeable that this representation, while low on noise, has the dynamic features of the face highly emphasized.

2.2.4 Time Surface

The representation of Time Surface [5], also called Surface of Time, is a similar approach to SAE. Both methods construct a surface that encodes the timestamp of the last activation of each frame pixel. However, this methodology, not only takes into account the events that occurred during the current Frame Time, but also pixels triggered prior to that, depending on a time decaying factor, τ. To put it in another way, this decaying factor gives memory to each event so its influence is felt throughout time, rather than just being a mere pulse. The work on [5], suggests three different Decaying Time Surfaces: a Linear Decay Time Surface, an Exponential Decay Time Surface and an Accumulated Exponential Time Surface (AETS), depicted by the image on figure 7.

The first scenario represents events that occurred at a Pixel K during a certain period of time and no decay is applied, therefore having no memory. In the second case, a linear decay is applied, therefore in the Time Surface this pixel is active during more time. On the other hand, the following methods implement an exponential decaying behavior to each event. However, the latter of these two cases, accumulates activations, giving it more emphasis on regions that are more frequently triggered.

The representation taken into consideration for this work was the last of all three - AETS. Thus, solely that methodology's formulation is covered in this section. The algorithm 3 demon-

[1] This figure was adapted from [5].

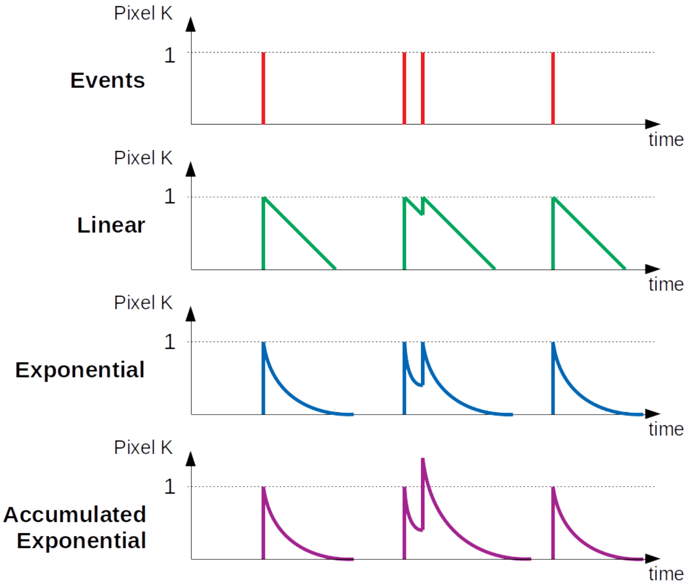


Figure 7: Various Time Surface Decaying methods.[1]

strates the pseudo-code to construct an `uint8` AETS frame. Moreover, the picture on figure 8 portrays a resulting AETS frame. This is the method with the highest sensitivity to the movement in the scene, apart from the Binary Event Representation.

Algorithm 3: Algorithm for constructing an AETS frame.

```
Data: All Event Cloud Points that occurred during the Frame Accumulation Time
$\tau \leftarrow$ Decaying Factor (time unit);
$S \leftarrow$ Map of Timestamps ($t_i$);
$P \leftarrow$ Map of Event Polarity $G \leftarrow$ Surface of Time;
$F \leftarrow$ Frame to be constructed;
for Each Incoming Event given by $(x_i, y_i, t_i)$ do
    $S(x_k, y_k) \leftarrow t_k$;
    $P(x_k, y_k) \leftarrow 1$;
    if Last Incoming Event then
        $G \leftarrow (G + P) \cdot \mathtt{exp}(\frac{S - t_k}{\tau})$;
        $F \leftarrow \mathtt{round}(255 \cdot \frac{G}{\mathtt{max}(G)})$;        /* Only if Polarity is always positive */
    end
end
Result: $F$
```

For this formulation, the temporal decay factor affects neighboring frames, giving the current image some temporal history from past frames.

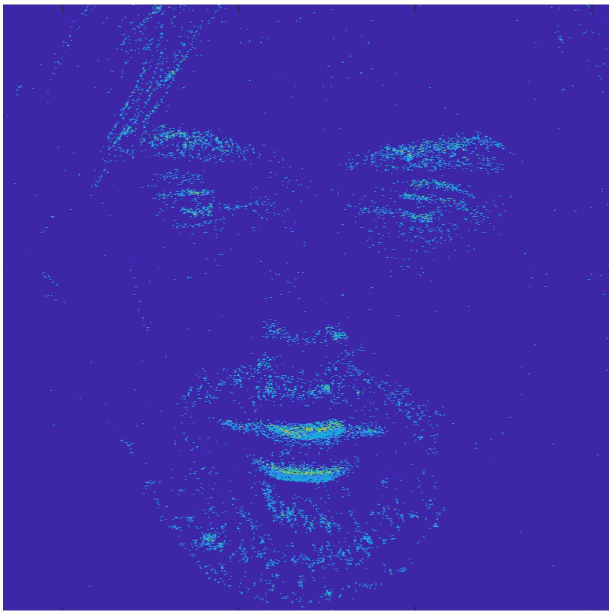


Figure 8: Example of an Accumulation of Exponential Time Surface Frame: This is a AETS frame of a user face integrated during $40ms$.

2.2.5 Event Frequency Representation

The Event Frequency Representation was first proposed by [6]. The concession of this methodology was justified through the fact that the majority of events are triggered in regions that are edges of the scene's subject. In detail, this formulation relies on the frequency of activation of a certain pixel - the more it occurs, the higher that pixel's value will be. Moreover, in consequence, image noise is effectively reduced. Essentially, this is a simple map of counters that is then normalized utilizing a normalization equation inspired by the ”Sigmoid Representation” proposed by [7]. The pseudo-code of this strategy is shown in algorithm 4.

Algorithm 4: Algorithm for constructing an Event Frequency frame.

```
Data: All Event Cloud Points that occurred during the Frame Accumulation Time
$C \leftarrow$ Map of Counter of Pixel Activations, initialized with zeros;
$F \leftarrow$ Frame to be constructed;
for Each Incoming Event given by $(x_i, y_i, t_i)$ do
    $C(x_k, y_k) \leftarrow C(x_k, y_k) + 1$;
    if Last Incoming Event then
        $F(x_k, y_k) \leftarrow 255 \cdot 2 \cdot (\frac{1}{1+\texttt{exp}(C(x_k,y_k))} - 0.5)$
    end
end
Result: $F$
```

The frame illustrated in figure 9 is an example of a resulting frame from this technique. Although looking very similar to other representations, this construction has the drawback of not contemplating time information in any way. Nonetheless, this fact did not pose a problem in the experimental stages of this work, since the Frame Accumulation Time was short.

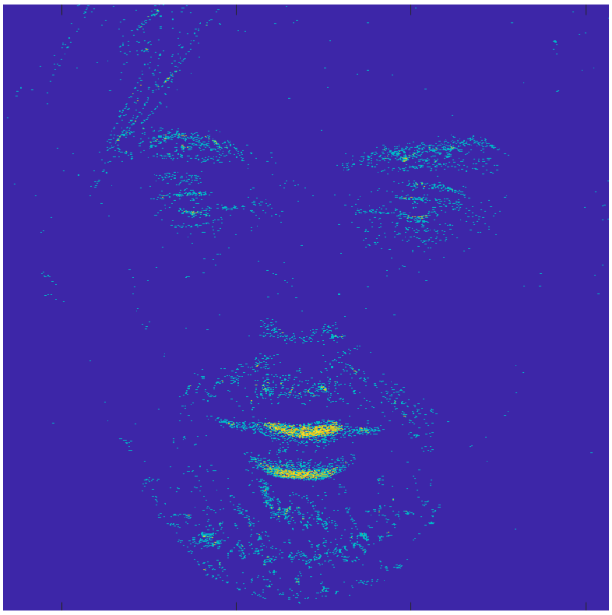


Figure 9: Example of an Event Frequency Frame: This is a Event Frequency frame of a user face integrated during $40ms$.

2.2.6 Leaky Integrate-and-Fire Event Representation

Finally, the last Event Representation covered in this work dives into a method denominated Leaky Integrate-and-Fire (LIF) also proposed by [6], which is inspired in the Leaky Integrate-and-Fire neuron model [8]. For illustration purposes, the image in figure 10 represents an elementary representation of how a neuron operates.

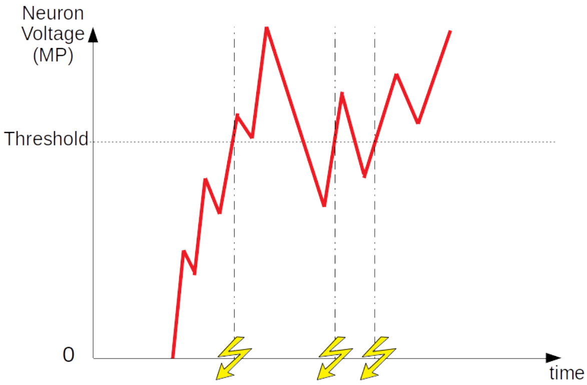


Figure 10: High-level visualization of the functioning of a neuron for the LIF model. Input spikes raise the neuron's Membrane Potential (MP), which then fires once that Voltage exceeds a certain Threshold.

Fundamentally, a neuron's membrane is electrically charged and its voltage increases in steps due to input spikes that discharge throughout time. In addition, the membrane potential value decays at a fixed rate, provided no spikes occur in the meanwhile. Whenever the membrane's voltage exceeds a particular threshold, the neuron fires. Given this, an event image representation analogy can be established. Assuming a Membrane Potential Map the size of the image frame and regarding all events as input spikes, then the Neuron parallelism can be set. Every time an event occurs at a pixel location it increases the value of the Membrane Potential in that location. Unless there is a new discharge, its value will forever decrease at a fixed rate until it is zero. The moment the Membrane Potential Map exceeds the stipulated threshold at those coordinates, the image frame pixel fires at the corresponding location. The resulting frame is a Map of the number of times each pixel was fired. Intrinsically, it is a very similar method to the Event Frequency Representation. Hence, it inherits its downsides. Both methods are frequency-based, which neglects the temporal information given by the event cloud. Alike its counterpart, this did not represent a problem in the experimental stages, quite the opposite in fact.

The algorithm for this strategy is a little more extensive than the rest and since there is an implementation of it online [2] [6], along with implementations for the Event Frequency Representation and Surface of Active Events, it will not be presented in this work. Regardless, an illustration of a resulting frame is provided in figure 11.

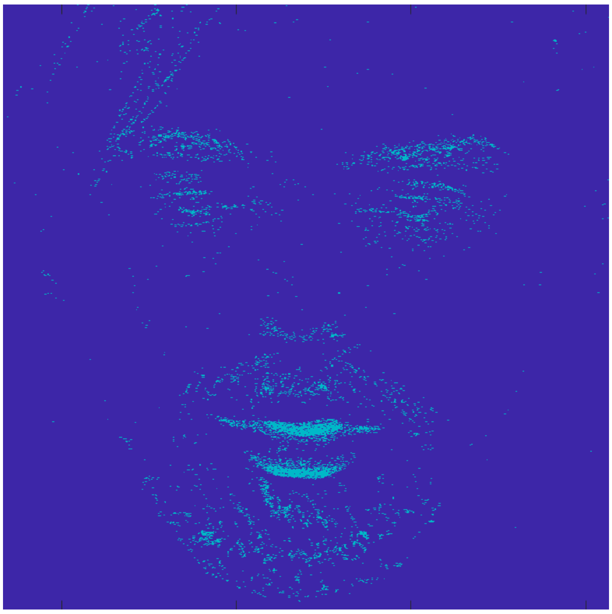


Figure 11: Example of a Leaky Integrate-and-Fire Frame: This is a LIF frame of a user face integrated during $40ms$.

[2] https://github.com/CrystalMiaoshu/PAFBenchmark

2.3 Convolutional Neural Network (ConvNet)

The Convolutional Neural Network was first devised to recognize handwritten digits [9] three decades ago, left disregarded for “it was impractical for real-world applications with complicated images” [10]. Only to be revisited two decades later [11] for the ImageNet LSVRC-2010 contest that attempted to classify over a million images into a thousand different classes, and be one of the top performers.

The working principle of this type of architecture is based on adjusting two-dimensional filters that are convolved with the input image to extract features in the Convolutional Layers, which in turn are compressed in the Pooling Layers and, sequentially, output the correct classification decision. In essence, each element of these filters corresponds to a neuron, hence it is updated and optimized accordingly during training phases. Figure 12 illustrates well the network’s pipeline.

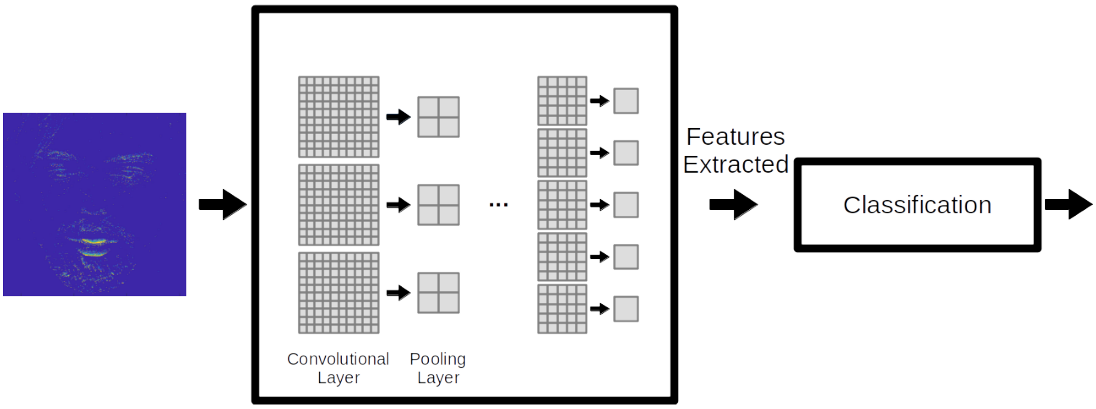
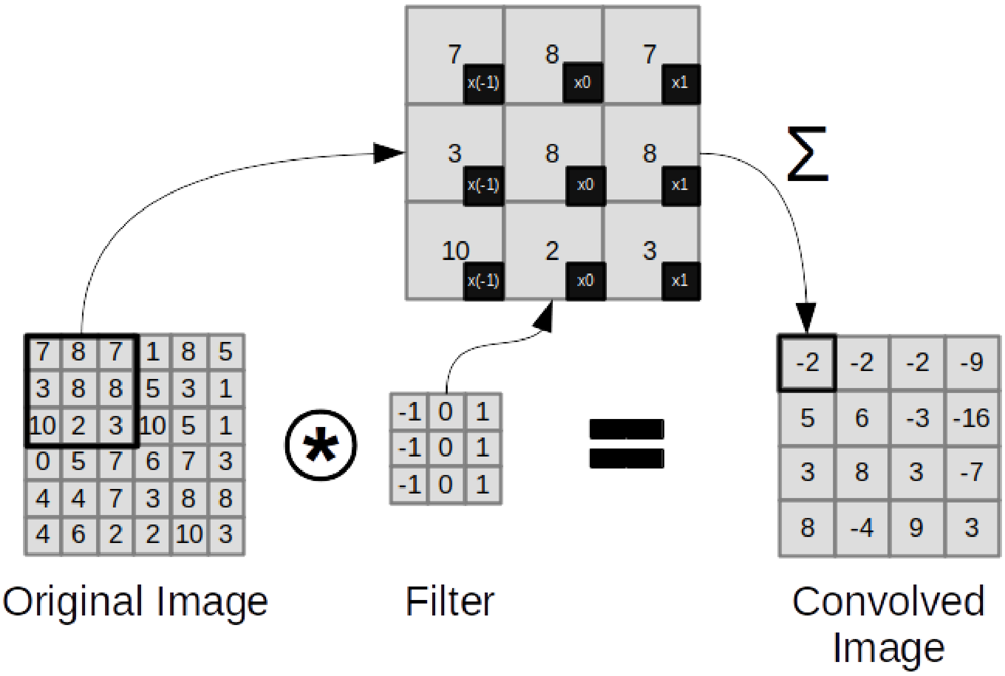


Figure 12: Standard 2D ConvNet pipeline. This figure was adapted from [10]

By way of explanation, on one hand, a Convolution is a mathematical operation between two signals. More specifically, for the case of ConvNets this operation is discrete and two-dimensional. Essentially, the operation is computed between an image and a kernel, commonly referred to as a filter. The operation requires the kernel to slide over the image and execute an element-wise multiplication between the filter and the sub-matrix, which have the same dimensions, followed by a sum of all results. The produced value is then placed in the corresponding coordinates in the subsequent image. This calculation is demonstrated through a simple example on figure 13.

On the other hand, there are various Pooling methods, of which the two most typically used in these sort of architectures are Average Pooling and Maximum Pooling. In short, pooling is a way to compress information. Alike a convolution, there is a kernel (moving window) with a specific stride that can differ from one, albeit it does not convolve with the values. Instead, it



Original Image

7	8	7	1	8	5
3	8	8	5	3	1
10	2	3	10	5	1
0	5	7	6	7	3
4	4	7	3	8	8
4	6	2	2	10	3

Filter

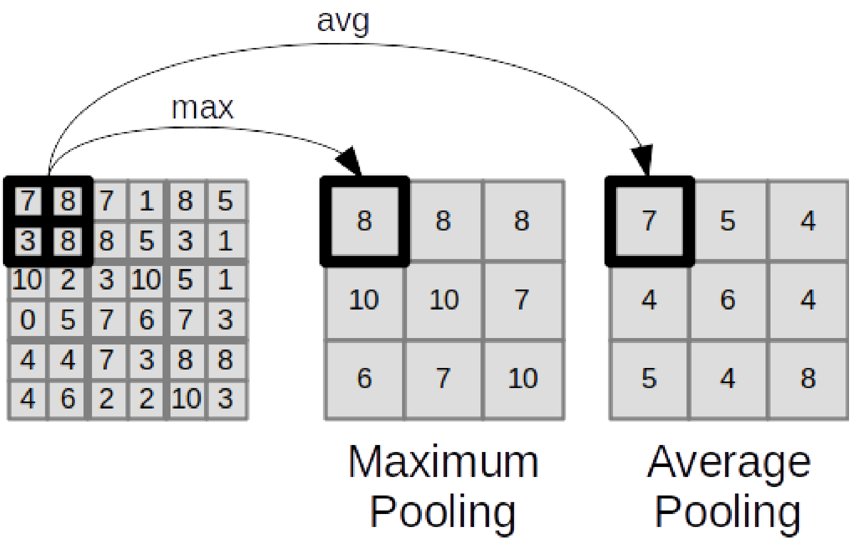
-1	0	1
-1	0	1
-1	0	1

Convolved Image

-2	-2	-2	-9
5	6	-3	-16
3	8	3	-7
8	-4	9	3

Figure 13: Example of a 2D Convolution.

calculates either the average or the maximum between all values covered and outputs it into a new image. Figure 14 provides a visualization of this process.



7	8	7	1	8	5
3	8	8	5	3	1
10	2	3	10	5	1
0	5	7	6	7	3
4	4	7	3	8	8
4	6	2	2	10	3

Maximum Pooling

8	8	8
10	10	7
6	7	10

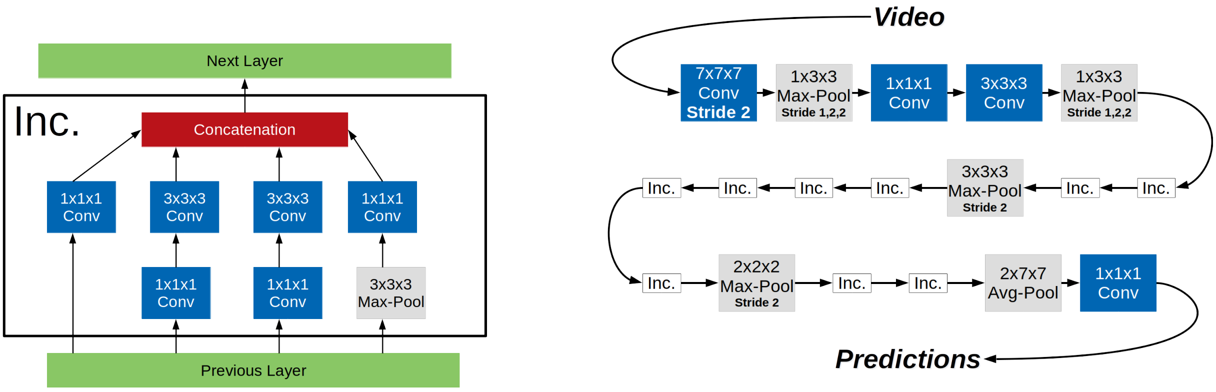
Average Pooling

7	5	4
4	6	4
5	4	8

Figure 14: Maximum and Average Pooling Example: Window-Size = 2×2; Stride = 2.

2.4 Two-Stream Inflated 3D ConvNet (I3D)

The I3D network was first created and proposed by [12]. Its concession purpose was to perform Action Recognition on videos. Unlike other video classification networks, which analyze an image sequence in a frame-by-frame fashion, the main advantage of this architecture lies on the fact that alongside the extraction of a spatial feature, there is a temporal one as well. Specifically, the basis of the network is a 2D ConvNet inflation - the augmentation of the filters and pooling kernels of classification ConvNets into 3D - allowing the network to extract spatio-temporal features, instead of only spatial ones. Thus, this model takes as input a short sequence of frames instead of a single frame, so that three-dimensional convolutional filters also extract temporal characteristics. The network architecture is detailed in figure 15.



(a) Inception Module (Inc.) Architecture.

(b) I3D Network Architecture.

Figure 15: These are adapted images from [12] of the overall I3D Pipeline. The convolutional and pooling layers with no stride default to one, and for simplification purposes the batch normalization layers, ReLu's and the softmax at the end were omitted.

On a separate note, another feature of this architecture is that it admits multiple input channels. For the work on [12], one channel was utilized to input RGB video images, whilst the second channel took the Optical Flow calculated for that RGB sequence - therefore, being termed “Two-Stream”. Basically, the network when fed dynamic information alongside the normal frames, which further enrich each clip, results in a performance enhancement of the system. Nonetheless, a single-stream is as valid an option. Figure 16 provides a holistic view of the data stream for the I3D network used in [12].

On a final note, to understand the upcoming section, it is important to detail in what format the network prediction is outputted. Basically, the returned value is a simple probability distribution across all data categories. Namely, the output is a vector of the model's confidence for each data class. For visual reference an example is orchestrated in figure 17.

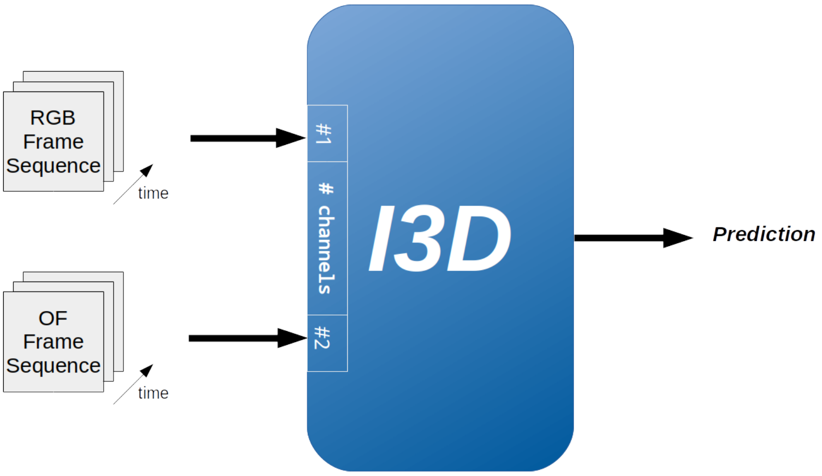
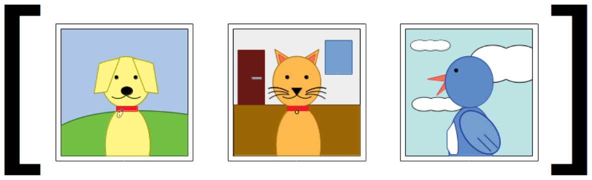


Figure 16: Comprehensive view of the data flow of the I3D network, with two channels one for RGB and another one for Optical Flow, as used in [12].



GROUND-TRUTH	[1 , 0 , 0]	DOG
	[0 , 1 , 0]	CAT
	[0 , 0 , 1]	BIRD
PREDICTION	[0.6 , 0.3 , 0.1]	DOG

Figure 17: Example of a Prediction Output. The first three rows represent the Ground Truth of each category, while the last one is a possible prediction returned by the model. In this case, the network would be 60% confident the input corresponded to a dog, 30% to a cat and 10% to a bird.

2.5 Network Optimization

In this section, a high level explanation of basic network optimization will be covered. In essence, a Neural Network is a colossal mathematical function with millions of parameters (weights and biases) and inputs, which are combined to output a certain value, in most cases, a prediction of what the input represents. The network parameters are initialized with random values and there are a lot of techniques to do so [13], [14]. At this stage if any data is inputted into the network the final accuracy will be close to $\frac{1}{\text{Number of Classes}}$ for a balanced dataset. For instance, if the task is to recognize three different categories like Cats, Dogs and Birds, an untrained network will correctly predict nearly 33% ($\approx \frac{1}{3}$) of the images of the dataset. To enhance its performance, it has to go through a training procedure that updates its parameters. This training operation is completed through inputting a batch of "learnable" objects (i.e. the data structures that the model will predict over) through the model, make a prediction, check how wrong the output is and adjust the parameters. The network is fed in batches instead of single samples at a time for it improves the model's generalization capabilities. Furthermore, the weights and biases tuning is not done at random, it is always executed with the goal to minimize a metric that reflects "how much error the model has" [15]. This measure is calculated via a Loss function. For linear regression, this function would be the Mean Squarred Error (MSE). For the case of classification, and, specifically, for networks that output a probability distribution, which is the case for the I3D, Categorical Cross-Entropy is very commonly utilized.

2.5.1 Categorical Cross-Entropy Loss

This function is expressly utilized to compare "one-hot"[3] ground-truth vectors - commonly referred to as targets - to probability distributed vectors - also regarded as predictions. The function is detailed by equation (3).

$$Loss_j = -\sum_k y_{j,k} \cdot \text{log}(\hat{y}_{j,k}) \tag{3}$$

In which, j is the j-th sample of the whole batch, $Loss_j$ is the loss value for the prediction $\hat{y}_j$, with y_j being the ground-truth, while k corresponds to the k-th index of the vector, that is, the k-th class, and log is the natural logarithm. For the example in section 2.4, figure 17, the computed Loss value would be as follows in equation (4), assuming the target is $y_j = [1, 0, 0]$, which corresponds to a Dog.

$$Loss_j = -\sum_k y_{j,k} \cdot \text{log}(\hat{y}_{j,k}) = -(1 \cdot \text{log}(0.6) + 0 \cdot \text{log}(0.3) + 0 \cdot \text{log}(0.1)) = 0.51082562376 \tag{4}$$

However, for the same prediction, if the input image illustrated a bird instead, meaning the target would be $y_j = [0, 0, 1]$, the calculated Loss would be the one given by equation (5).

[3] "one-hot" vectors are arrays with one value 1 (hot) and the others 0 (cold). For ground-truth that value 1 will correspond to the index of the correct data class.

$$Loss_j = -\sum_k y_{j,k} \cdot \log(\hat{y}_{j,k}) = -(0 \cdot \log(0.6) + 0 \cdot \log(0.3) + 1 \cdot \log(0.1)) = 2.30258509299 \quad (5)$$

As the example above shows, the closer the probability distribution is to the ground-truth, the lower the loss value. Hence, this will be the metric that will determine whether or not the updates of the model's parameters are heading the network in the right direction.

2.5.2 Network Optimization: Stochastic Gradient Descent (SGD)

The SGD is one of the oldest network optimizers, but still highly relevant today. In fact, most optimizers are an adaptation of this technique.

Essentially, like the name suggests, the SGD calculates the descendant direction of the gradient of an Error function relative to the network's parameters (both weights and biases). These parameters are subsequently updated given that calculation.

In mathematical terms, given a mini-batch with M samples of a Dataset X composed by $(input, target)$ pairs,

$$X_m : \quad (x_m, y_m) \quad , \qquad m = \{1, ..., M\} \quad (6)$$

also, $\hat{X}$, the $(input, prediction)$ pairs given by the network for the same set of input data,

$$\hat{X}_m : \quad (x_m, \hat{y}_m) \quad , \qquad m = \{1, ..., M\} \quad (7)$$

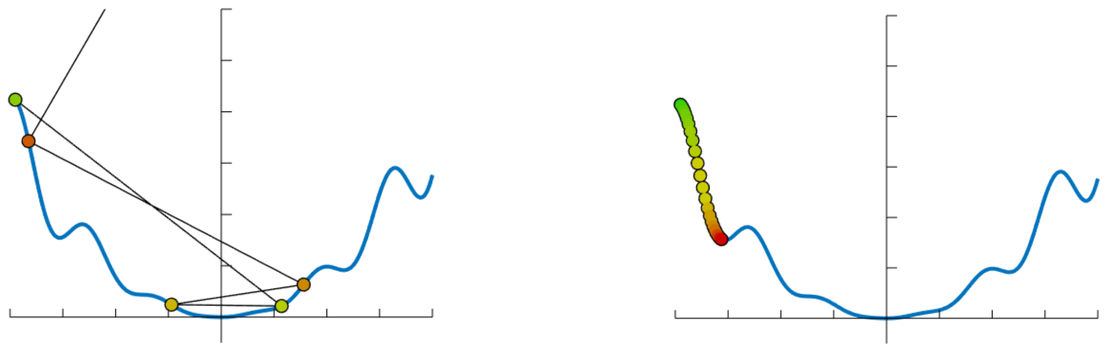
and, E, an error function calculated by the network (e.g. Categorical Cross-Entropy Loss) comparing how close the predictions, $\hat{X}_m$, are to the ground-truth, X_m, for the current set of weights and biases, globally represented by Θ,

$$E(X, \hat{X}, \Theta) \quad (8)$$

The SGD optimizer calculates the gradient of the error function, E, relative to the network parameters, Θ, which are, in turn, updated through equation (9).

$$\Theta_{t+1} = \Theta_t - \alpha \cdot \frac{\partial E(X, \hat{X}, \Theta)}{\partial \Theta} \quad (9)$$

in which, α is the learning rate - a tunable factor that defines how abrupt the updating process is. If too high, the function minimum search might get unstable and diverge. If too low, the minimization process might halt in a local minimum. These two scenarios are depicted in figure 18, for a given Loss function dependent some network parameters.



(a) Example of a high learning rate: The search for the minimum is unstable for its high learning rate, hence, it does not converge to a determined point.

(b) Example of a low learning rate: The search converges to a local minimum.

Figure 18: Learning Rate Example: Too high vs. Too low.

The learning rate only points towards the descent direction of the gradient, this means that when a local minimum is reached the search can go back and forth infinitely, hence halting there. Thus, in order to make the searching process more stable and robust another parameter should be added - momentum. In essence, this parameter gives some inertia to the search by considering the direction of the previous update. Therefore, reducing the possibility of stalling in any minimum found. This parameter, m, is then added to equation (9) as shows in (10).

$$\Theta_{t+1} = m \cdot \Theta_t - \alpha \cdot \frac{\partial E(X, \hat{X}, \Theta)}{\partial \Theta} \tag{10}$$

Addressing the previous example on figure 18, the image on figure 19 illustrates the search using the same learning rate as the one on figure 18b but with the addition of momentum.

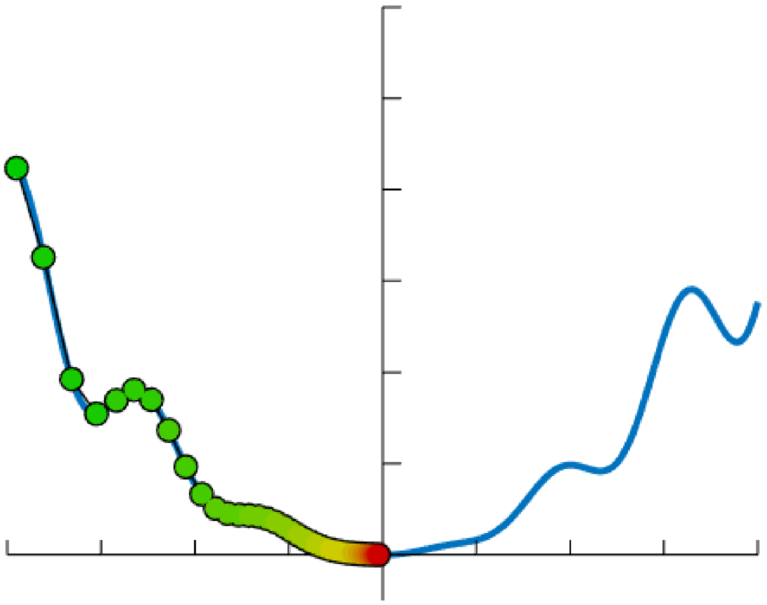


Figure 19: Example of a Minimum Search with Learning Rate and Momentum.

As it can be observed, adding momentum allows the search to overcome the local minimum and converge to the absolute minimum of the function.

3 State of the Art

3.1 Facial Recognition through Facial Dynamics

Facial recognition has been a hot topic for several decades now. Thus, naturally, the advancements in the field are of great magnitude. Recognition rates are absolutely off the charts, surpassing even human capabilities. However, the type of data that is relied upon - static RGB or gray-level images - is very easily attacked or manipulated, hence, it is wise to look for ways to make this type of application more robust. A lot information can be extracted from a person's face that assures a more secure prediction of that individual's identity. Namely, depth information for adding yet another dimension of face features or infrared sensors to guarantee the liveliness of a subject through the radiation emitted by the person's skin. Furthermore, it has been long discussed whether or not one's facial motion is strong enough idiosyncrasy and if it somehow contributes to the human's perception of the face.

In fact, the work on [1] was specially conducted with the aim to answer many of these questions. It was shown that the human eye is trained to be highly sensitive to facial movement, since it is a great source of non-verbal communication cues during social interactions. In addition, it is a good indicator of liveliness, due to the fact that the lack of motion in someone's face is a strong sign of inauthenticity. Nevertheless, a lot of information is still conveyed by a face's static features. Apart from this, that same researching team was involved in another relevant experiment [16] that evaluated how much identity can different facial movements convey. The experiment consisted in recording a group of actors performing three types of facial motion - self-induced facial expressions, facial expressions in a social context and conversational-induced expressions. Subsequently, the movements were embedded into an avatar face to normalize all faces and remove appearance features. Taken everything into account, it was concluded that the conversational expressions revealed more identifiable traits than the rest.

On a separate note, one hypothesis that could be presented as to why facial dynamics analysis improves accuracy and robustness over static analysis could be that the latter uses quite less samples. This is not the case, however, as it was proven by [17]. The proof was based on an experiment in which a group of people had to recognize a certain facial expression through a static frame, a normal video sequence and a corrupted video sequence. The altered video had noise masks inserted between frames, without the removal of any frame, thereby, eliminating the dynamic component of the face in the video. The results showed that the rates of correct identification of expression were better for single frame images compared to the multiple frame sequences that lacked dynamic stimuli. On top of that, the accuracy was actually best for the raw video sequences, that is, those of which had maintained the dynamic features of the face in the video.

Following all the advantages listed previously, plenty of research has been conducted with facial recognition via facial dynamics being the main object of study [18]–[23]. Moreover, facial motion has been used to identify potential attacks on face recognition systems [24]. In addition, many studies have delved into mapping phonemes[4] to visemes[5] and which visemes generate the most recognizable dynamic traits [25]–[27]. Based on this concept, lip reading applications for speech recognition [28] and even language identification (visual-only) [29] are possible. Besides, another similar application is facial expression recognition, both for standard emotions [30], such as happiness or sadness, as well as for more subtle and micro expressions [31]–[33].

3.2 Event-Based Face Detection, Tracking and Alignment

On a more particular subject, there has already been work developed with neuromorphic vision sensors and faces, mostly in the field of face detection, tracking and alignment. For instance, this has been done for faces in constant movement, thus, being always depicted in the frame [34], [35]. As a matter of fact, some have gone as far as creating and publishing datasets to facilitate further research [36], [37]. These resource, however, do not focus on still heads with dynamic frontal faces for their original goal is simple face detection and alignment, which, in turn, does not fully serve the purposes of this work.

Alternatively, other applications use more than just head or face shape for detection-and-tracking tasks, also taking advantage of facial dynamics. Namely, the work on [38] makes use of a very strong dynamic feature of any face - eye blinking. Indeed, throughout time, an eye blink is a very apparent feature captured by an event camera and strongly indicates the presence of a face. Far and beyond goes the research on [39], which evaluates a driver's level of drowsiness based on the frequency of eye blinking and mouth opening. The event camera's low power consumption features, high temporal resolution, low latency, high dynamic range and reduced motion blur make face analysis in these scenarios much simpler and robust in contrast with conventional cameras.

3.3 Event-Based Object Detection and Classification

An issue not directly related to faces, highly relevant nonetheless, regards feature extraction, object detection and classification from events. Similarly to face detection and such, the event

[4] The smallest unit of speech sound that, when grouped with others, composes the pronunciation of a word.

[5] The facial motion induced by voicing a specific phoneme.

cloud needs to be processed and most methods take a frame-based approach, apart from some works that use graphs [40] or operate directly over the point cloud [41]. Apart from this, investigation on this field has led to the creation of large-scale datasets [42], [43].

In respect to the most common frame-based approaches, the Time Surface representation (mentioned in section 2.2.4 of this work) is widely adopted. Various studies have used this representation for feature extraction and tracking [5], [44], [45], meanwhile others [5], [42], [43] have gone as far as to using it for object detection and classification tasks. Some methodologies embraced for classification were a Histogram of Averaged Time Surfaces (HATS) [42], a Synaptic Kernel Adaptation Method (SKIM) [5] classifier (originally proposed by [46]) and a Histogram Of Time Surfaces (HOTS) [47]. This last work also used their pattern recognition technique for a facial recognition task for heads in constant movement that achieved a top identification rate of 79%.

3.4 Action and Gesture Recognition

In the realm of classifying actions and gestures, there has been work developed prior to the surge of event cameras, with intensity-image cameras (both colored and gray-level). Namely, an interesting approach uses a Dynamic Image Network [48], which condenses a video sequence into one frame and then classifies over it. One other method, in fact, one that is this work's greater source of inspiration created the network architecture I3D [12], which obtained state of the art classification on the HMDB-51 dataset with 80.9% and on the UCF-101 dataset with 98.0%.

Eventually, along with the rise of neuromorphic vision sensors, algorithms for identifying actions and gestures using this technology have been created. Similarly to other application fields, there have been proposed multiple event processing techniques, ranging from direct point cloud analysis [49] to graph oriented methods [50] to the most common frame-based approaches [3], [51], including ones with stereo cameras [52]. The work on [51] actually developed a gesture dataset - IBM DVS128 Gesture Dataset - on which most studies benchmark algorithm performance, including another research [3] that had great influence on this dissertation. Particularly, this work [3] achieved state of the art recognition using a self-proposed event frame representation - Temporal Binary Representation, covered in section 2.2.2 - and the I3D network architecture [12], obtaining a top recognition accuracy of 99.58% for 10 classes and 99.62% for 11 classes.

4 Proposed Methodology

This chapter focuses on the body of work done for this dissertation. It covers the materials used, and the techniques and procedures implemented to answer the primary questions of this work.

4.1 CeleX-V Camera

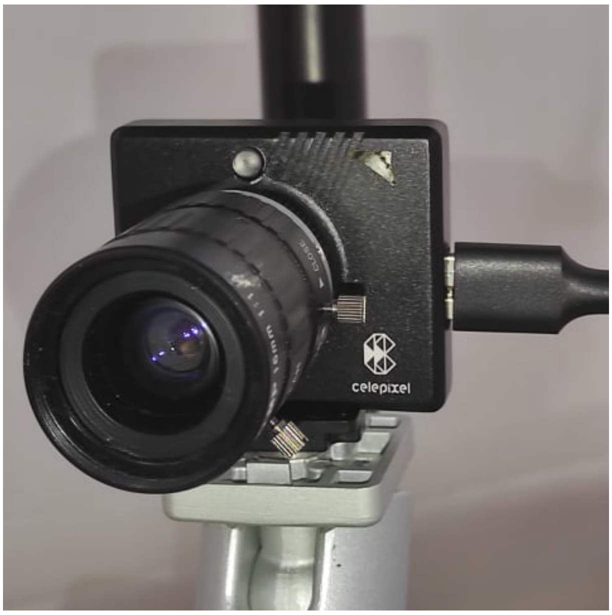


Figure 20: Picture of the CeleX-V Event Camera.

The Event-Camera used for the data acquisition process of the whole experiment was the CeleX-V, from the company CelePixel, recently acquired by Will Semiconductor. The main motivator for this choice was the superior specifications in comparison to its market competitors, in terms of camera resolution and ability to output both events and grayscale images. The CeleX-V specifications are detailed in table 1.

Table 1: CeleX-V Specifications and Market Comparison[6]

Specificatoins	CeleX-V	Market Comparison (out of 13)
Resolution (pixels)	1280 x 800	Top 2 best
Latency (μs)	8	Top 2 best
Dynamic Range (dB)	120	Top 5 best
Power Consumption (mW)	400	Top 1 worst
Pixel Size (μm^2)	9.8 x 9..8	Top 5 best
Stationary noise (ev/pix/s) at 25º C	0.2	Top 1 worst
Grayscale Output	Yes	7 out of 13 do
Maximum Frame Rate (fps)	100	Top 1 best

[6] The tabulated values were reported [2] at the date of publication.

On another note, the published camera documentation, along with the API [7], served as the basis of the entire development of the data acquisition software for this work. In addition, as table 1 details, the sensor can also output Grayscale images like a standard camera. Specifically, the camera has two operating modes - Fixed Mode and Loop Mode. Firstly, the Fixed Mode is the default working state of the camera in which it outputs one fixed modality of data continuously. The returned objects range from Event Clouds (with or without Polarity or Intensity values) to Grayscale and Optical Flow Frames. It also outputs some event frame representations, yet, none of them were used or covered in the scope of this work. Lastly, the Loop Mode operates in a manner that up to three different data classes are returned, virtually, simultaneously. In reality, the camera's kernel quickly switches between the various modes, looping through all three, as illustrated in figure 21. By default, it outputs an Event cloud or frame, a Grayscale frame, and an Optical Flow frame.

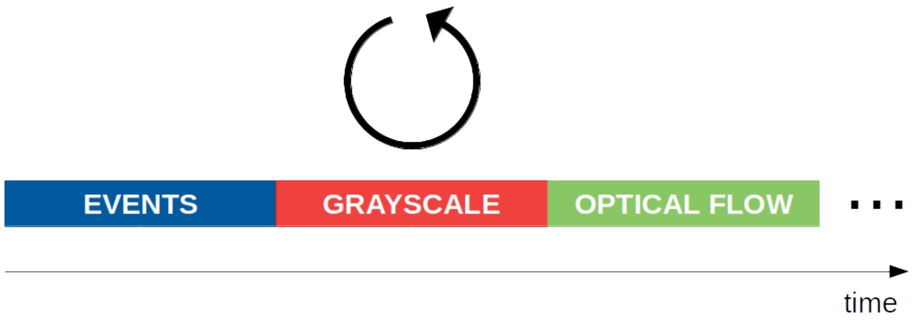


Figure 21: Visualization of the CeleX-V Loop Mode operation.

This comes, however, at a cost. The Event Data quality returned from this operation mode is somewhat reduced in terms of both noise increase and event cloud fragmentation. Instead of a uniform distribution of events along the time axis, the cloud seems to be broken into smaller event clusters.

4.2 Neuromorphic-Vision Dataset for Speech-Induced Facial Dynamics

The fact that there is a massive shortage of neuromorphic resources that target facial analysis, part of this work was focused on recording a considerably large dataset of rather still human faces reading from a text or pronouncing words facing the camera - the Neuromorphic-Vision for Speech-induced Facial Dynamics (NVSFD) Dataset. The aim of the dataset is not only to satisfy the experiments' needs but also to allow further research on the topic.

[7] https://github.com/CelePixel/CeleX5-MIPI

4.2.1 Material and Recording Setup

The camera used for the data acquisition was the previously mentioned CeleX-V, which operated under a highly controlled environment - fixed on a sturdy tripod, inside a photo studio box, with studio lighting to guarantee the most favorable conditions possible, as figure 22 depicts.

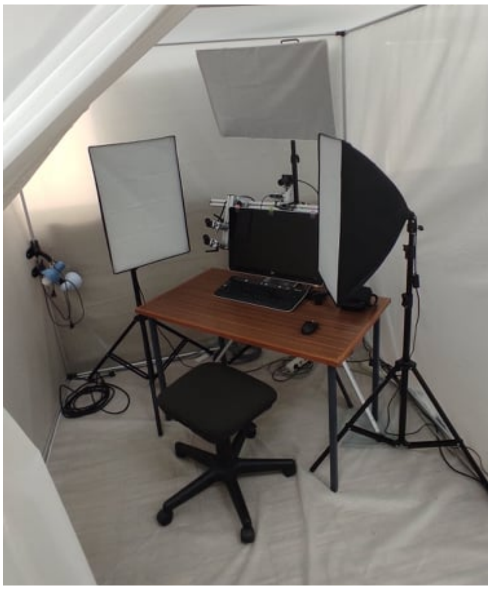


Figure 22: Data Acquisition Setup Environment.

In addition, inside the box, a table was equipped to support the user-interface monitor, keyboard, and mouse, along with a stool for the recorded subject to sit down directly in front of the camera - figure 23 illustrates a closer look into the scenario.

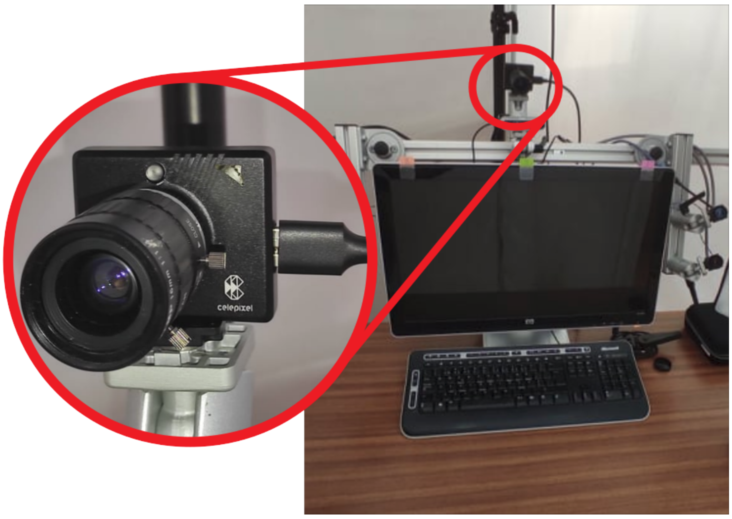


Figure 23: Closer view of the Data Acquisition Setup Environment.

4.2.2 Subject-Task Protocol

The Subject-Task protocol was specifically designed to meet this work's experiment needs. Namely, each individual was closely recorded so most of the frame area would be used to extract the idiosyncratic features of each subject's face. Consequently, problems such as face detection and alignment had no need to be addressed. In addition, all recordings were performed in the Camera's Fixed Mode - for pure Event gathering - and in Loop Mode - for collecting Grayscale frames and Event Clouds in a simultaneous fashion.

Regarding the actual assignments each individual had to perform, there were three for every subject in the dataset and two additional ones for the first nine users. The two first exercises had each person reading two different Nursery Rhymes. The third endeavor consisted of each subject pronouncing their full name multiple times. Finally, the last two tasks were only done by the first nine subjects. One of them was for spoofing purposes and only performed by `user000` and `user001`, whereas the other task had the first nine users say every other user's name of those nine. The full dataset specifications can be found in Appendix I, while the two nursery rhymes used are exhibited in Appendix II. Additionally, the illustration of the various tasks is depicted in figure 24.

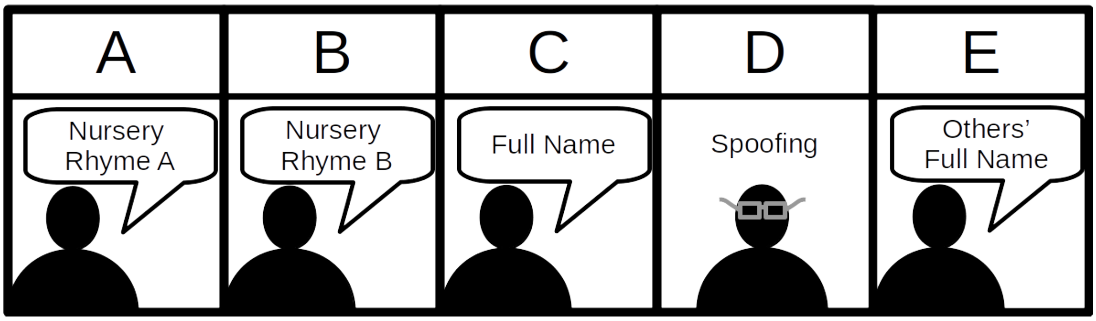


Figure 24: Various Dataset User Tasks: Tasks A-C were completed by all subjects in the dataset; Task D was done by the first two users only; Task E was performed by the first nine subjects only.

This letter notation, A-E, will be used from this section forward to simplify the reference to each one of the five activities performed by the Dataset participants. It is also important to notice that tasks A and B were recorded twice for every subject. For the first recording, every participant read the full nursery rhyme, whereas, for the second instance, only half of the verses were read. The first will be referred to as activity A1, while the latter will be mentioned as activity A2. Surely, it is the same case for task B. In addition, task C was recorded five times but all of them were the same, however, a similar notation such as C1, C2, etc. will also be used.

4.3 Video Processing and Classification

This dissertation workflow was greatly influenced by the work on [3]. Likewise, the event clouds of the individuals speaking were converted to frame sequences and then fed into the I3D network. For this study, however, given the high resolution of the camera and the image shape supported by the network, the image frames had to be square cropped around the face (crop size = 800×800) and resized to 224×224 pixels. In addition, to estimate the location of the face a very rudimentary method was used. Accounting for the fact that throughout the video sequence every subjects' head was essentially still, for each clip it was produced an Event Frequency Frame comprising the whole video, which was then filtered by a Gaussian kernel. The resulting image is then convolved with an Ellipse Mask followed by setting to 1 every pixel of value greater than or equal to 0.2, otherwise, they were set to 0. Finally, since the consequent image corresponds to the area of most event occurrences, the center of mass along the columns gives a general estimation of the location of the center of the speaking face. Regarding the axis along the image rows, it was always chosen half of the height of the image, since all subjects had their heads roughly covering the whole height of the frame. Figure 25 illustrates the automatic crop pipeline.

This methodology, however, is not foolproof as it can be observed by the example presented in figure 26.

As for the classification endeavor, there were three main stages that took place for the validation of this methodology.

I. IBM DVS Gesture Dataset integration with I3D network (replicating the work on [3]).

II. NVSFD Dataset with 9 different subjects.

III. NVSFD Dataset with 40 different subjects.

The first step was dedicated to implementing the I3D network and convert the IBM DVS Gesture Dataset point clouds to TBR frames. The network model used was one published online [8] in PyTorch [53], however, the training and testing were separately carried out. The accuracy results were not the exact same as the one in the paper, but definitely close enough - around 96.692% recognition rate. This was due to the fact that the training procedure was not as exhaustive as the one in the original study. Regardless, the objective of validating the network model was achieved. In order to obtain this result, the model was trained using a SGD

[8] https://github.com/piergiaj/pytorch-i3d

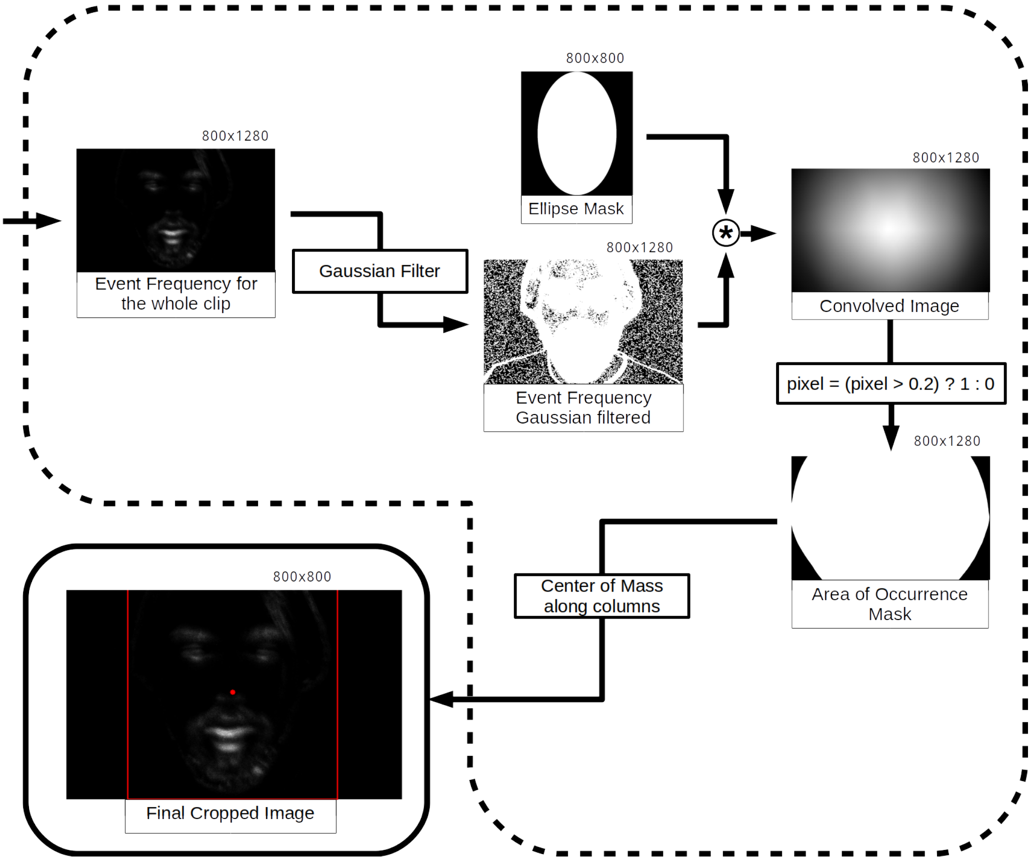


Figure 25: Square Crop Automation: This filtering process was used to avoid cropping every clip by hand.

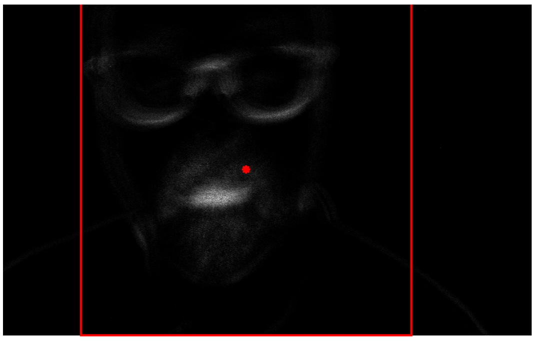


Figure 26: Bad result of Crop Automation: This image crop was calculated via the automatic process described before and it shows it is not perfect for every scenario. Nonetheless, it is good enough for this task, since it successfully crops the whole face. Centering is not a huge problem due to the fact that the video frames are transformed during training for data augmentation purposes.

optimizer with a learning rate and momentum of $\alpha = 0.001$ and $m = 0.9$, respectively, while feeding the network a batch of 13 samples at a time, for a total of 40 training epochs.

Consequently, the second iteration of the execution pipeline required recording people's faces while reading from a text or mouthing their own names. This motivated the experimental creation of the NVSFD Dataset with 9 subjects to determine whether or not the network's performance for gesture classification would translate to facial recognition. This universe of users executed the four different tasks described in section 4.2.2, which were activities A-C and E. However, only the first two subjects performed task D that consisted in reading a few verses of nursery rhyme A while the eye-glasses wearer did it without glasses and the non-wearer used its fellow's glasses.

Eventually, the last phase of the operation workflow was executed to appraise the scalability of this technique, hence, a universe of 40 subjects seemed reasonable to validate this network's feature. Alike the first 9 users, the rest of the subjects performed tasks A-C. Plainly, not only being quite inconceivable that all users would perform activity E, it also did not present considerable benefits for them to perform both tasks D and E.

Furthermore, it is important to state how the samples were extracted from the video sequences, since it also impacted the performance of the models. For the first stage of the process (using the IBM Gesture Dataset), the samples were extracted with a stride of 12 frames, which is the same size of each "learnable object". For the second phase (NVSFD with a universe of 9 speaking users), the stride was 1, so a lot more samples were retrieved from each video. Finally, for the third stage (NVSFD with a universe of 40 users) the stride was 2, for the fact that using a stride of 1 would represent a colossal amount of data, hence, harder to process given the available tools. A visualization of this process for each phase is illustrated in figure 27.

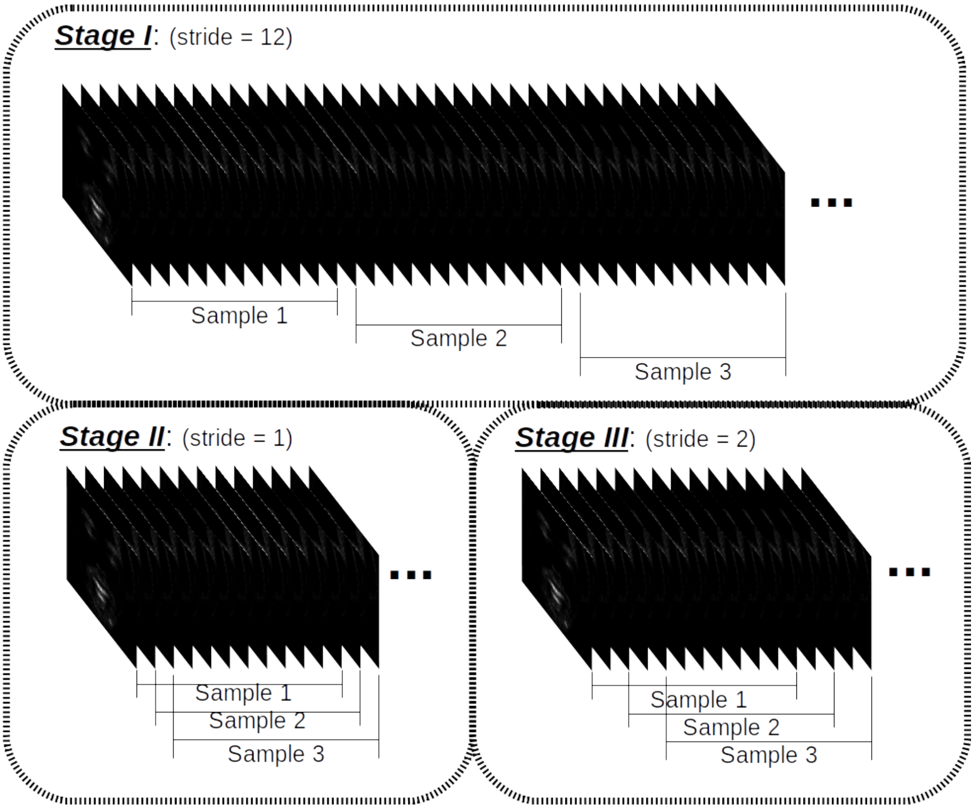


Figure 27: Sample Extraction: Visualization of the stride of the samples extraction of the full clips for each stage.

These samples retrieved from the full video sequences were then used in the training and testing of the models for the major experiments of each stage. Regarding the type of data used for the development and evaluation of the network models, this work targeted mostly the modality of pure events. Nevertheless, a shallow attempt at assessing the impact of using solely Grayscale information or even combining both Events and Grayscale was conducted.

5 Experimental Results

The aim of this chapter is to report and open a discussion on the outcome of the experiments conducted for the last two stages of this work's pipeline described in section 4.3. Since stage 1 was only a validation step for the efficacy of the methodology, its results are irrelevant to the work at hand, although summarized in the previous section. Regardless, the training procedure found best for the last two stages was different from the first stage and it is thoroughly specified in table 2.

Table 2: Stage 2 and 3 - Training Specifications

Training Specifications		Remarks
Optimzer	SGD	-
Learning Rate	0.01	-
Learning Rate Decay	10	Everytime the validation loss stalls.
Minimum Learning Rate	0.0001	No more learning rate decay once this value was reached.
Momentum	0.9	-
Number of Epochs	Ranged between 12-40	Varied with the amount of training data
Batch Size	12 (for stage 2) 32 (for stage 3)	-
Data Augmentation	Affine transformations; Random Erase from top to half.	Rotation: $[-10, +10]°$; Translation: $[-10, +10]pixels$ (each direction); Scaling factor: $[0.8, 1.8]$.

Notice that a training aspect of paramount importance for these stages was Data Augmentation, which allowed the models to shift its features extractors from solely considering the spatial plane to give a little more emphasis to the temporal component as well.

5.1 Stage II - 9 speaking subjects

This stage evaluated the network's ability to perform a facial identity recognition task instead of what it was initially conceived to do - action and gesture classification. Thereby, and to simplify further experiments, only one Event Frame Representation was chosen out of the five representations thoroughly described in section 2.2. All the representations were created with a Frame Accumulation Time of 40 ms and clips of 12 frames were fed into the I3D network during training and testing of the models. In addition, Grayscale and Grayscale alongside Event Representation clips were also tested. The models were trained on samples extracted from activities A1 and B1, and tested on the samples withdrawn from tasks A2 and B2. The stride of sample extraction was 12 frames. Table 3 shows the results obtained from the experiment.

Table 3: Event Representation Comparison: Accuracy Comparison

	Pure Events	with Grayscale
TBR	97.701%	**100%**
AETS	**98.851%**	**100%**
SAE	98.161%	**100%**
Frequency	98.391%	**100%**
LIF	98.391%	**100%**
Only Grayscale	**100%**	

The accuracy values presented were calculated over each sample prediction. As it shows, whenever Grayscale is added the models are able to perfectly predict each and every subject based on the 12-frame samples. Therefore, and since adding the gray-level component was never the main focus of this work, there was no interest in considering it in further experiments. Its discussion is therefore left open for further investigation.

Regarding the Pure Event modality, the actual recognition rates are outstanding given not only its the sparse nature but also the little information each event frame usually conveys. The accuracy rates were very similar between representations, despite the AETS representation clearly being the front runner among all five. Hence, this representation is the one used to carry out the rest of the experiments.

For this stage, four major experiments were conducted to challenge the network and understand whether the dynamics of the face would origin a strong enough signature for each subject. These four studies are listed below.

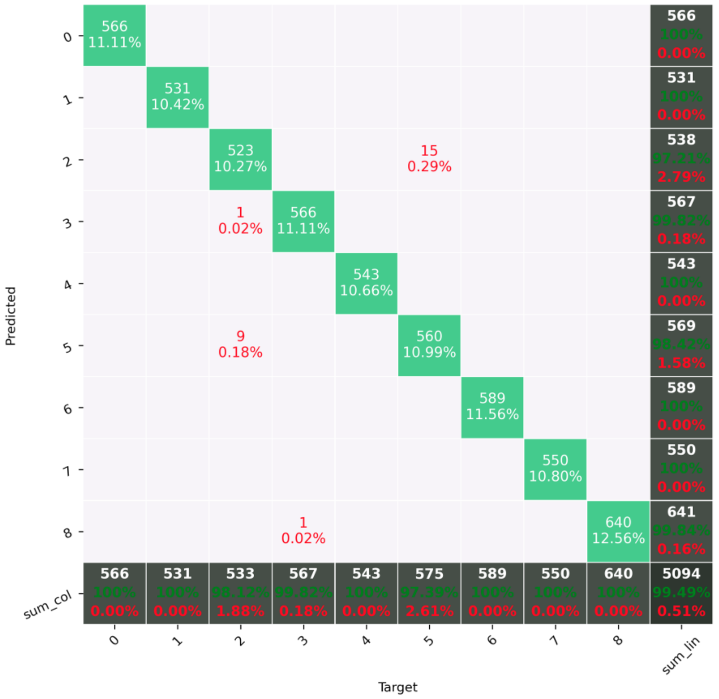
1. Train a model with tasks A1 and B1, and test it with tasks A2 and B2.
2. Train a model with task A and test it with task B.
3. Train one model with tasks C1-C3 and another one with tasks A, B and C1-C3, and test both of them with tasks C4 and C5.
4. Train a model with tasks C1-C3 and test it with task E.

For these experiments, all samples were extracted with a stride of 1.

5.1.1 Experiment 1

The purposes of this first model were to maximize the performance obtained in the preliminary experiment regarding event representations, to understand how much impact the facial dynamic component was having in the model's decision and how prone the model was to spoofing.

Assessing the performance of this first model, the simple fact that the training was done with more samples due to the extraction stride being 1, it improved the model's accuracy by almost an extra 0.25%. Figure 28 presents the confusion matrix obtained for this first scenario.



Predicted \ Target	0	1	2	3	4	5	6	7	8	sum_lin
0	566 11.11%									566 100% 0.00%
1		531 10.42%								531 100% 0.00%
2			523 10.27%			15 0.29%				538 97.21% 2.79%
3			1 0.02%	566 11.11%						567 99.82% 0.18%
4					543 10.66%					543 100% 0.00%
5			9 0.18%			560 10.99%				569 98.42% 1.58%
6							589 11.56%			589 100% 0.00%
7								550 10.80%		550 100% 0.00%
8				1 0.02%					640 12.56%	641 99.84% 0.16%
sum_col	566 100% 0.00%	531 100% 0.00%	533 98.12% 1.88%	567 99.82% 0.18%	543 100% 0.00%	575 97.39% 2.61%	589 100% 0.00%	550 100% 0.00%	640 100% 0.00%	5094 99.49% 0.51%

Figure 28: Confusion Matrix[9]for the model trained on A1 and B1 and tested on A2 and B2, which had a sample extraction stride of 1 frame.

As it can be observed on the bottom right cell, the final accuracy value for all the "learnable objects" extracted from A2 and B2 went up to 99.49%. This, however, is not the same testing set as the one used for the event representation experiment, since the sample extraction stride was different. For that previous testing set this model's accuracy was 99.08%, which is an improvement, nonetheless. Furthermore, the accuracy for each user can be observed along the bottom row in green, while along the rightmost column the green values represent the true positive rate for each subject and the red ones represent the false positive rate. Clearly, from all the information displayed by the matrix, it can be noticed that there was a slight confusion between subject 2 and 5, not all that relevant given its dimension compared to the 5094 samples of the testing set. To put this matrix in perspective with the testing set created with an extraction stride equal to the length of each learnable video sequence (12 frames), figure 29 shows the confusion matrix for this model's performance on that testing set.

Given that each learnable object is roughly a half-second, figure 29 shows that the confusion

[9] This confusion matrix was plotted using the software on `https://github.com/wcipriano/pretty-print-confusion-matrix`

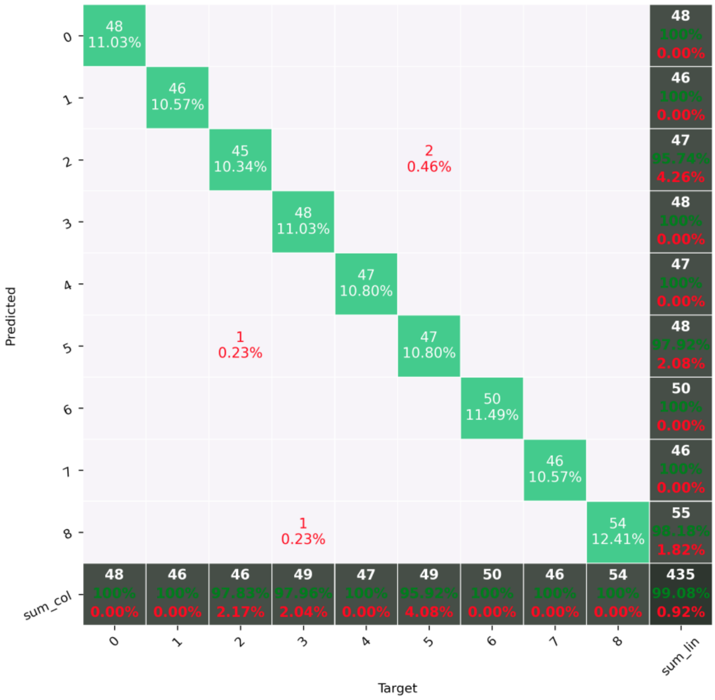


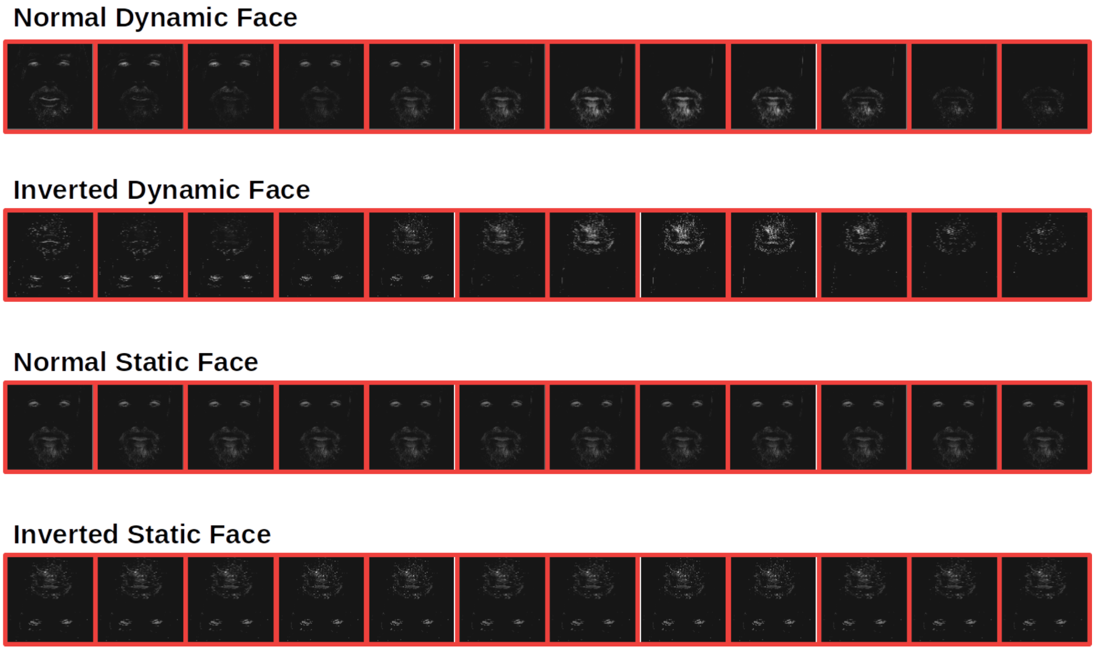
Figure 29: Confusion Matrix for the model trained on A1 and B1 and tested on A2 and B2, which had a sample extraction stride of 12 frames.

presented in figure 28 amounts to only one half-second out of the whole 46 half-seconds (23 seconds) of video for subject 2, one half-second out of 49 (24.5 seconds) for subject 3 and, finally, roughly one full second (or two half-seconds) out of 49 half-seconds (24.5 seconds) for subject 5. This whole assessment has no memory, all the model's decisions are done in half a second, therefore, for a real-time application, it would be wise to consider a short prediction history record to make sure the effect of these sporadic outliers is attenuated.

On another note, one other objective of this experiment was to determine whether the dynamic component of the face was useful in the discrimination process. To prove that point, four different tests were run, all for the testing set sampled at a stride of 1 frame. On each test the following transformations were applied to each learnable video clip.

- No transformation on any plane (normal and dynamic faces).
- A $180°$ rotation on the spatial plane, with no transformation on the temporal plane (inverted and dynamic faces).
- No spatial transformation, with a freezing of the frame, which had the most spatial information, across the rest of the clip (normal and static faces).
- A $180°$ rotation on the spatial plane and a freezing of the frame, which had the most spatial information, across the rest of the clip (inverted and static faces).

The image illustrated on figure 30 helps visualize the four transformations previously described.



Normal Dynamic Face

Inverted Dynamic Face

Normal Static Face

Inverted Static Face

Figure 30: This image represents the four different transformations described above applied for one video sample.

The freezing frame transformation had the goal to remove the temporal evolution of each sample giving it only static appearance information. This experiment was inspired by [54], which advocates in favor of the usefulness of facial dynamics for face identity recognition. This work described many experiments in which people were shown faces in "non-optimal viewing conditions" and it happened that the subjects had more difficulties identifying the face when it was static rather than when it had the dynamic component. Analogously, relative to this experiment, the rotation makes for the "non-optimal viewing conditions" while freezing one frame across the rest simulates a clip of static face that the network can process. The results can be observed in table 4.

Table 4: Accuracies for each one of the four transformations tests

	Normal Faces	Rotated Faces
Dynamic Faces	99.490%	37.515%
Static Faces	94.641%	30.487%
Relative Difference from Dynamic to Static Faces	-4.874%	-18.734%

In fact, it appears that there is a general drop of accuracy from dynamic to static faces, but that does not necessarily prove the hypothesis. Sure, this assures that the network is extracting temporal features, not solely focusing on spatial ones, but the point to be made clear is that dynamic stimuli actually helps the facial perception process, especially, under sub-optimal conditions, such as for inverted faces. The simple fact that there is a drop of 18.734% in recognition rate under abnormal conditions does not prove this by itself. Actually, it is the accuracy drop for rotated faces being higher than the one under normal circumstances that surely confirms that the dynamic aspect is truly relevant for unfavorable situations. Nonetheless, the results support the relevance of dynamic information for facial perception in any circumstance.

On a final note, the last point to be lightly discussed is how prone to spoofing this system can be. In general terms, since neuromorphic vision is such a novel field, there have not been developed many spoofing techniques. However, the use of face masks or the use of videos of other people may be some that could potentially attack this type of systems. The aim of this work was never to develop a system that predicts spoofing methods and prevents them from working. Nevertheless, a simple experiment was conducted to assess how dynamic stimuli was an anti-spoofing mechanism by itself. To do that, Dataset participants 0 and 1 were asked to read the first stanza of nursery rhyme A, with the particularity that user 1 wore subject 0's glasses, while user 0 did not. To evaluate the model's performance under these special conditions another confusion matrix was constructed and it is illustrated on figure 31.

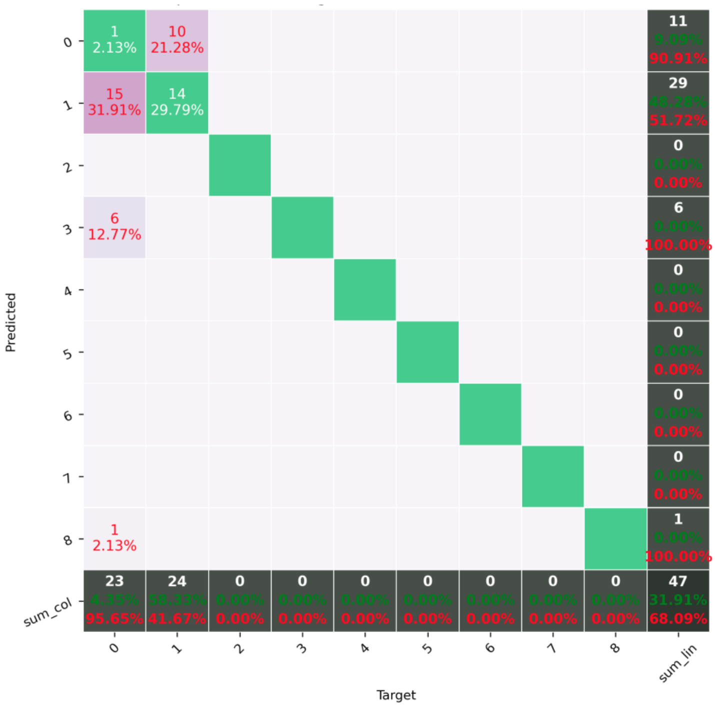
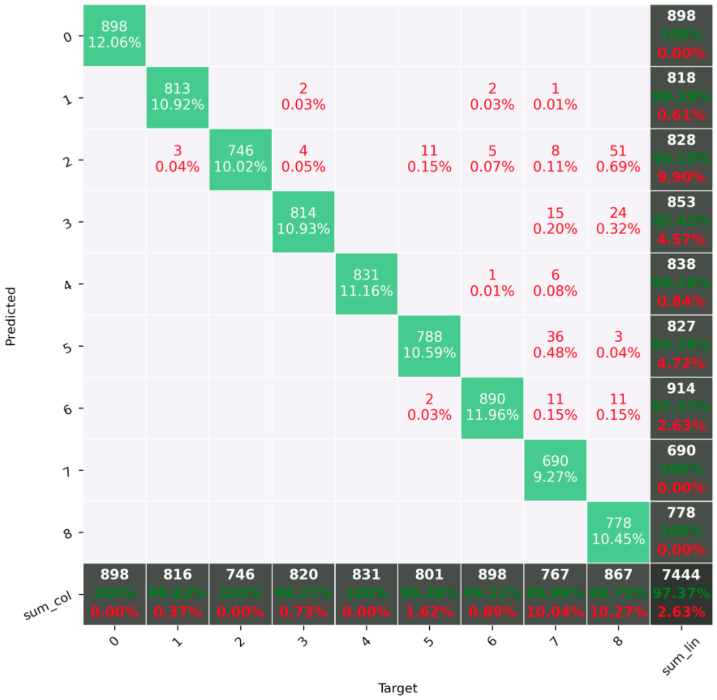


Figure 31: Confusion Matrix for the model trained on A1 and B1 and tested on D, with a sample extraction stride of 1 frame.

As it can be observed, on one hand, for subject 0, the model had great difficulty identifying it as user 0 - classifying it correctly once out of 23 times. This can be easily deduced from the fact that during training, despite the random erase of the top half of the frame, the network never really witnessed this subject without glasses, thus, making it seemingly impossible to make a prediction on that user if the person speaking is not wearing anything. On the other hand, for user 1 the model had an outstanding performance - correctly recognizing it 14 times out of 24 (above 50% accuracy). Even though during training the model never witnessed user 1 with glasses and, in fact, only observed those same glasses on another subject is a very strong sign that the facial dynamics of user 1 stood above the spoofing attempt.

5.1.2 Experiment 2

The aim of the model created under the scope of this second experiment was to understand whether the dynamic traits generated by reading one nursery rhyme would generalize to some other set of speaking words never experienced during the training phase. Hence, this model was trained on task A and tested on B. The evaluation of the model's performance is detailed by the confusion matrix on figure 32.



	0	1	2	3	4	5	6	7	8	sum_lin
0	898 12.06%									898 100% 0.00%
1		813 10.92%		2 0.03%			2 0.03%	1 0.01%		818 99.39% 0.61%
2		3 0.04%	746 10.02%	4 0.05%		11 0.15%	5 0.07%	8 0.11%	51 0.69%	828 90.10% 9.90%
3				814 10.93%				15 0.20%	24 0.32%	853 95.43% 4.57%
4					831 11.16%		1 0.01%	6 0.08%		838 99.16% 0.84%
5						788 10.59%		36 0.48%	3 0.04%	827 95.28% 4.72%
6						2 0.03%	890 11.96%	11 0.15%	11 0.15%	914 97.37% 2.63%
7								690 9.27%		690 100% 0.00%
8									778 10.45%	778 100% 0.00%
sum_col	898 100% 0.00%	816 99.63% 0.37%	746 100% 0.00%	820 99.27% 0.73%	831 100% 0.00%	801 98.38% 1.62%	898 99.11% 0.89%	767 89.96% 10.04%	867 89.73% 10.27%	7444 97.37% 2.63%

Figure 32: Confusion Matrix for the model trained on A and tested on B, with a sample extraction stride of 1 frame.

It is noticeable that the model displays somewhat more confusion, which is natural, given that different tasks force the readers to have different facial behavior. Nonetheless, it is quite remarkable that the accuracy has remained in the high nineties percentage wise, exactly, 97.37%. Furthermore, it can be observed that for each subject the model has maintained the recognition rate around 90% to 100%. Thus, despite being slightly more uncertain, the model's final prediction is still quite confident. In essence, two major conclusions can be deduced from this data. The first one being that the model is able to extract certain dynamic features that can be found generally throughout any speaking action. Secondly, the more diverse data the network is exposed to throughout the training stages, the better it will perform in "never-seen-before" scenarios.

Aside from that, similarly to the previous experiment described in section 5.1.1, a spoofing attempt was also conducted. The outcome of the trial is described in the confusion matrix shown in figure 33.

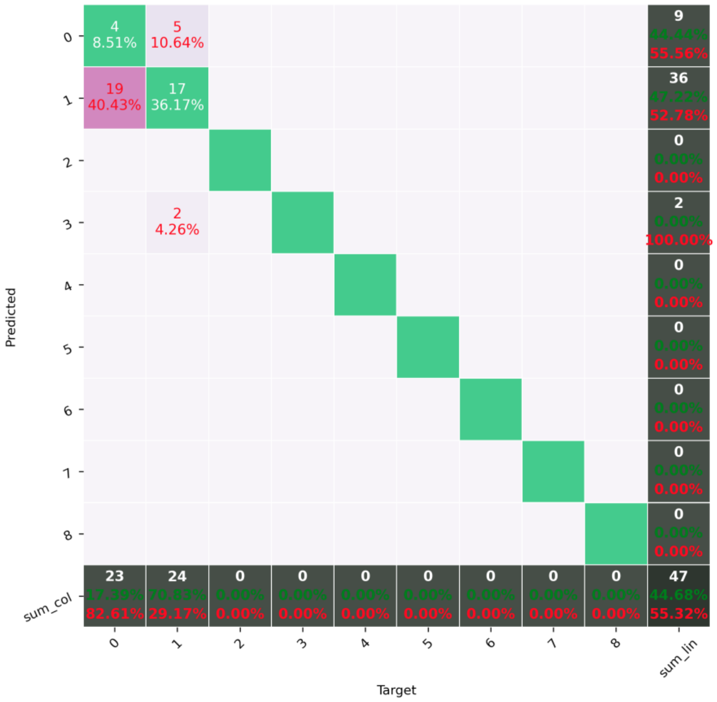


Figure 33: Confusion Matrix for the model trained on A and tested on D, with a sample extraction stride of 1 frame.

It is important to stress that activity D, which is the spoofing task, had the two subjects read the first stanza of nursery rhyme A. Thus, there is great similarity between facial movements caused by each task, even though the facial appearance component is corrupted. Therefore, given the fact that both this and the previous section's models were trained during roughly the same time, the current network has a greater familiarity with the movements executed in task A. Consequently, the dynamic signatures for each user during activity D were better recognized by the model. For subject 0, the accuracy rose from 4.35% to 17.39%, while, for subject 1, it rose from 58.33% to 70.83%. This means that the glasses became a rather less relevant factor of influence in the classification process.

In conclusion, the decision to use people with glasses on the training stages of the models and the hope that the network would be able to set subjects apart for tasks such as activity D was quite an ambitious endeavour. However, it has sparked interesting results and could certainly be considered for further investigation.

5.1.3 Experiment 3

The objective of this third experiment was to perform a sketchy simulation of a biometric access system, which, similarly to the fingerprint system, people would be granted access by pronouncing their full name to the camera. There are two major parameters that need evaluation to make certain that this would be a robust and practical strategy. Firstly, whether one's full name could generate enough dynamic discriminatory stimuli. Secondly, how much data would be necessary in training for the network to build a strong classification model. In attempt to judge these two factors, there were trained two different models. One was trained with information solely from tasks C1-C3, while the other one with information from activities A, B, C1-C3. Both were tested on tasks C4 and C5. The confusion matrices for the models are presented in figures 34 and 35. It should be brought into attention that for performance enhancement reasons, the data augmentation practice was slightly altered. Namely, the rotation transformation range increased to $[-45°, 45°]$, so as to force the model to extract more temporal signatures.

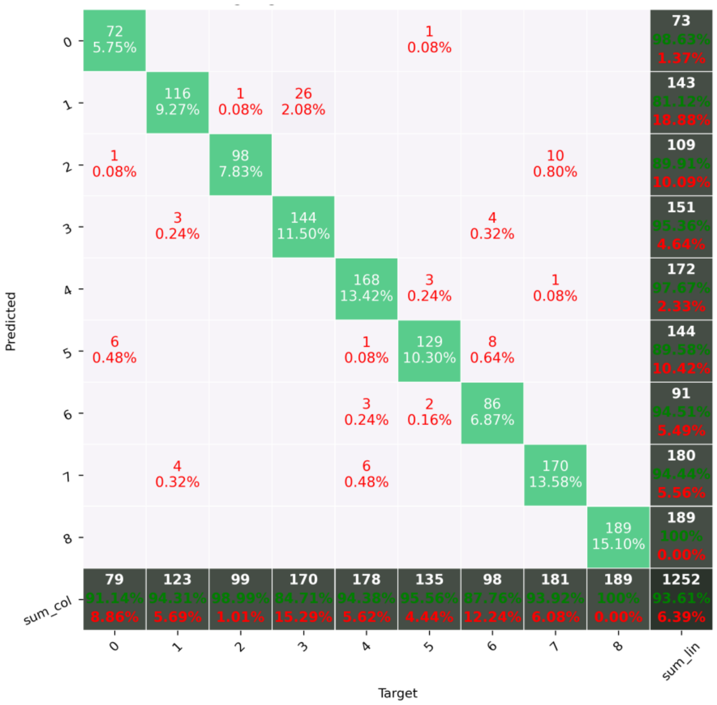
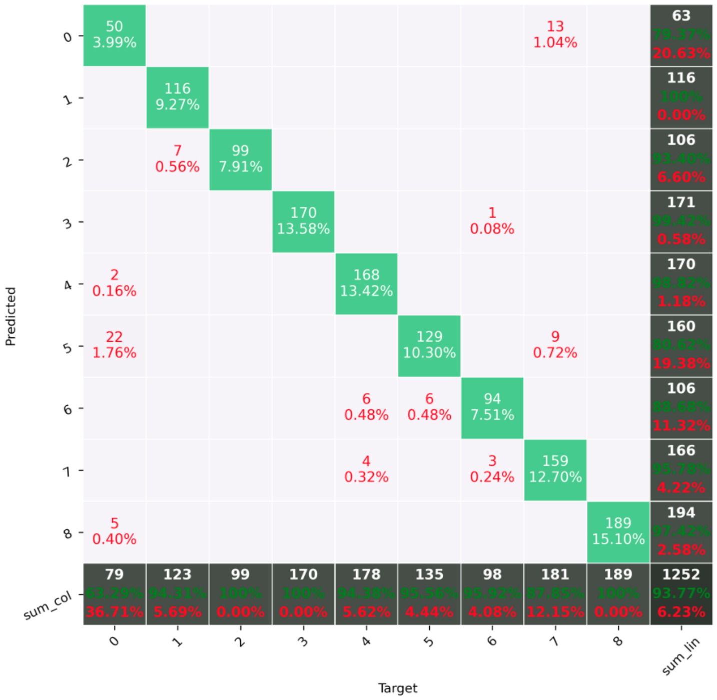


Figure 34: Confusion Matrix for the model trained on C1-C3 and tested on C4 and C5, with a sample extraction stride of 1 frame.

As it can be observed, the model trained with only data from task C performed quite well. For each user, the confidence rates ranged from nearly 85% to 100%. This fact, by itself, confirms the biometric relevance of the type of data stemmed from activity C, since it proved it fits the two criterion stated at the beginning of this section. However, an interesting question to pose would be whether there would be improvements in accuracy, given it was provided more information to the model, aside from solely each one's names, during the training phase.



Predicted \ Target	0	1	2	3	4	5	6	7	8	sum_lin
0	50 3.99%							13 1.04%		63 79.37% 20.63%
1		116 9.27%								116 100% 0.00%
2		7 0.56%	99 7.91%							106 93.40% 6.60%
3				170 13.58%			1 0.08%			171 99.42% 0.58%
4	2 0.16%				168 13.42%					170 98.82% 1.18%
5	22 1.76%					129 10.30%		9 0.72%		160 80.62% 19.38%
6					6 0.48%	6 0.48%	94 7.51%			106 88.68% 11.32%
7					4 0.32%		3 0.24%	159 12.70%		166 95.78% 4.22%
8	5 0.40%								189 15.10%	194 97.42% 2.58%
sum_col	79 63.29% 36.71%	123 94.31% 5.69%	99 100% 0.00%	170 100% 0.00%	178 94.38% 5.62%	135 95.56% 4.44%	98 95.92% 4.08%	181 87.85% 12.15%	189 100% 0.00%	1252 93.77% 6.23%

Figure 35: Confusion Matrix for the model trained on A, B and C1-C3, and tested on C4 and C5, with a sample extraction stride of 1 frame.

The results of the second model's evaluation can be misleading if the reason behind user 0's accuracy is left unexplained. All other subjects' accuracies either rose or maintained, apart from a slight and almost irrelevant drop for user 7. Had it not been for the colossal drop for subject 0, the overall accuracy would definitely increase from the one achieved by the previous model. The two potentially great influencing factors were that, firstly, for tasks A and B user 0 wore glasses, whereas for task C all 9 participants performed without glasses, secondly, subject's 0's name is shorter than the average name length among the 9 subjects. These two aspects combined in having the network train on a quite short amount of "non-wearing-galsses" samples for user 0. For instance, subject 4 also performed A and B with glasses and C without, however, user 4's name is actually longer than the average among all participants. Thus, that should explain the reason behind the drop in accuracy for subject 0 but not for subject 4.

In conclusion, disregarding the effect of user 0 for the reasons presented above, the total accuracy increased from 93.777%, for the first model, to 95.823%, for the second model. This proves, once again, the theory that the more diverse information is given during training, the better the network can generalize for other data.

5.1.4 Experiment 4

This fourth and final experiment had the goal to assess whether the network, trained solely with subjects mouthing their names (activity C), would be induced in error given the case that one user uttered another one's name. The idea was to understand how replicable could the pronunciation of a subject's name be by other users. For instance, if the model predicted anyone who said subject 0's name to be subject 0, then the hypothesis that this methodology could serve as a good biometric would be refuted.

In order to evaluate this scenario, the same model built for the experiment on section 5.1.3 was used, in which the training was done with tasks C1-C3. The testing, however, was conducted with activity E. Indeed, it is important to note that task E is comprised of all nine subjects mouthing every other user's names. Therefore, to assess the performance of the model, two different statistical results were withdrawn. The first one being a standard confusion matrix, whereas the second being a set of nine matrices, one for each subject, each with nine rows and nine columns. Each matrix row represents the number of the subject whose name was uttered, while each column is the number of the user predicted by the model. For instance, for matrix number k, cell (i, j) contains a percentage representing how many times the model predicted subject k to be j, while subject k mouthed user i's name, thus, the sum of all values of each row should be 1. In addition, since task E does not contain subjects pronouncing their own name, row i, in which $i = k$, is all zeros. A visualization on figure 36 is provided for clarification purposes.

In addition, this tool allows to determine whether the final prediction of the whole video is correct, incorrect or inconclusive using the following criteria, given a matrix M_k, k being the number of the speaking subject:

- **Correct:** For a row, i, and a column, j: $M_k(i, j) > 0.5$, $j = k$.
- **Incorrect:** For a row, i, and a column, j: $M_k(i, j) > 0.5$, $j \neq k$.
- **Inconclusive:** For a row, i, and any column, j: $M_k(i, j) < 0.5$, $\forall j$.

An additional illustration of two use-cases for user 3 are provided on figure 37.

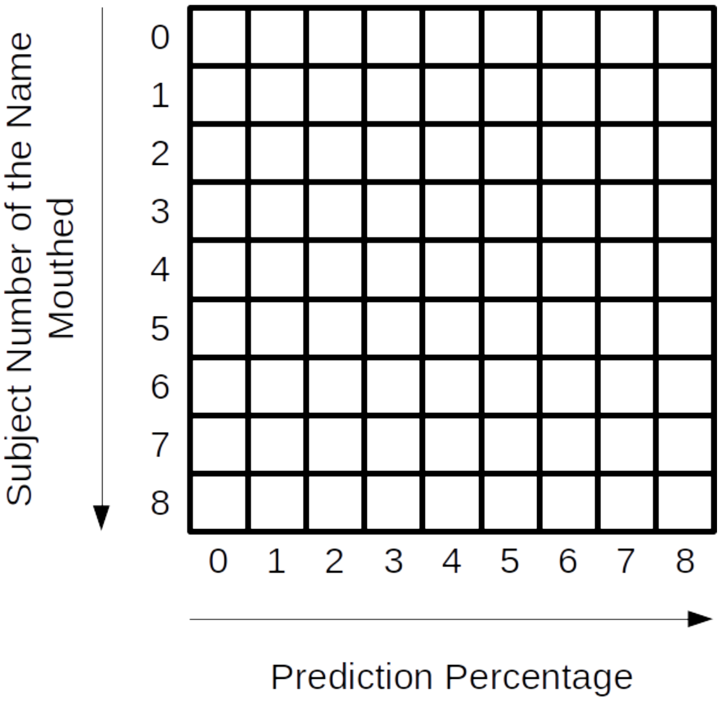
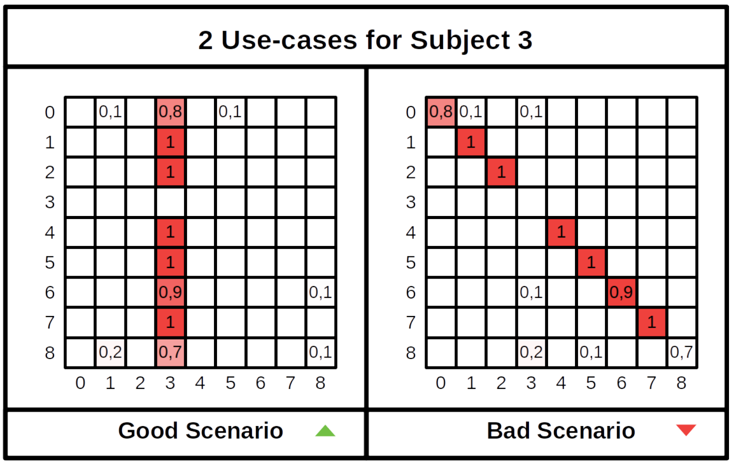


Figure 36: Evaluation Matrix Visualization



2 Use-cases for Subject 3

	0	1	2	3	4	5	6	7	8
0		0,1		0,8		0,1			
1				1					
2				1					
3									
4				1					
5				1					
6				0,9					0,1
7				1					
8		0,2		0,7					0,1

Good Scenario ▲

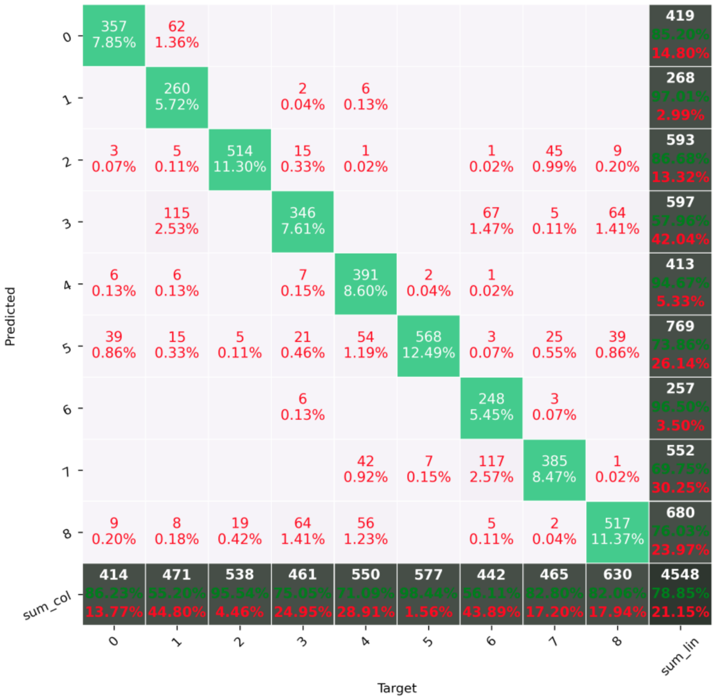
	0	1	2	3	4	5	6	7	8
0	0,8	0,1		0,1					
1		1							
2			1						
3									
4					1				
5						1			
6				0,1			0,9		
7								1	
8				0,2		0,1			0,7

Bad Scenario ▼

Figure 37: Evaluation Matrix Example

On one hand, the first scenario shows that the model predicted subject 3 to be subject 3 more often than not. For example, for the clip in which subject 3 mouthed user 6's name (row 6), 90% of the time it correctly recognized user 3, while 10% of the time it classified it as user 8. On the other hand, for the second scenario, the model predicted user 3 to be the subjects whose name was being uttered. For instance, for the clip in which user 3 pronounced user 6's name, the matrix shows that, for 90% of the samples in the video, the model predicted the speaking subject to be user 6 (incorrectly) and for user 10% to be user 3.

Regarding the actual results from the experiment, figure 38 illustrates the confusion matrix for the general evaluation of the model's performance.



	0	1	2	3	4	5	6	7	8	sum_lin
0	357 7.85%	62 1.36%								419 85.20% 14.80%
1		260 5.72%		2 0.04%	6 0.13%					268 97.01% 2.99%
2	3 0.07%	5 0.11%	514 11.30%	15 0.33%	1 0.02%		1 0.02%	45 0.99%	9 0.20%	593 86.68% 13.32%
3		115 2.53%		346 7.61%			67 1.47%	5 0.11%	64 1.41%	597 57.96% 42.04%
4	6 0.13%	6 0.13%		7 0.15%	391 8.60%	2 0.04%	1 0.02%			413 94.67% 5.33%
5	39 0.86%	15 0.33%	5 0.11%	21 0.46%	54 1.19%	568 12.49%	3 0.07%	25 0.55%	39 0.86%	769 73.86% 26.14%
6				6 0.13%			248 5.45%	3 0.07%		257 96.50% 3.50%
7					42 0.92%	7 0.15%	117 2.57%	385 8.47%	1 0.02%	552 69.75% 30.25%
8	9 0.20%	8 0.18%	19 0.42%	64 1.41%	56 1.23%		5 0.11%	2 0.04%	517 11.37%	680 76.03% 23.97%
sum_col	414 86.23% 13.77%	471 55.20% 44.80%	538 95.54% 4.46%	461 75.05% 24.95%	550 71.09% 28.91%	577 98.44% 1.56%	442 56.11% 43.89%	465 82.80% 17.20%	630 82.06% 17.94%	4548 78.85% 21.15%

Figure 38: Confusion Matrix for the model trained on C1-C3 and tested on E, with a sample extraction of 1 frame.

As it can be confirmed on the bottom right cell, the total accuracy for all samples was 78.85%, which is quite a decrease relative to the testing set used in the experiment of section 5.1.3. Furthermore, it can be concluded that there was a lot of confusion for subject 1 and 6. The model predicted user 1 to be 0 and 3 almost 45% of the time, meanwhile for user 6 the model confused it to be subjects 3 and 7 almost 44% of the time. This comes across as somewhat natural given the low diversity of the training used to build the model. Figure 39 presents the results in the nine-matrices format further detailing how each clip was perceived by the model.

It can be observed that generally all matrices follow the pattern of the good scenario described in figure 37. The confusion for users 1 and 6 that was depicted in the confusion matrix can also be found in this data representation, since for the fact that, in the case of subject 1, columns 0 and 3 are both slightly highlighted. Furthermore, an interesting point this evaluation tool makes is that for some clips the final prediction of the model was, in fact, inconclusive or wrong. For instance, for subject 1 rows 0 and 7 are inconclusive, while the others are correct. Besides, for user 6, rows 1, 3 and 8 are incorrect, whereas the others are correct. Apart from that, there are no clear signs of the bad case depicted in figure 37. Therefore, it can be concluded that the model might be wrong for these type of scenarios, but not necessarily fooled. The spoofing would be working, only if the data representation would take a shape like the second scenario illustrated in figure 37.

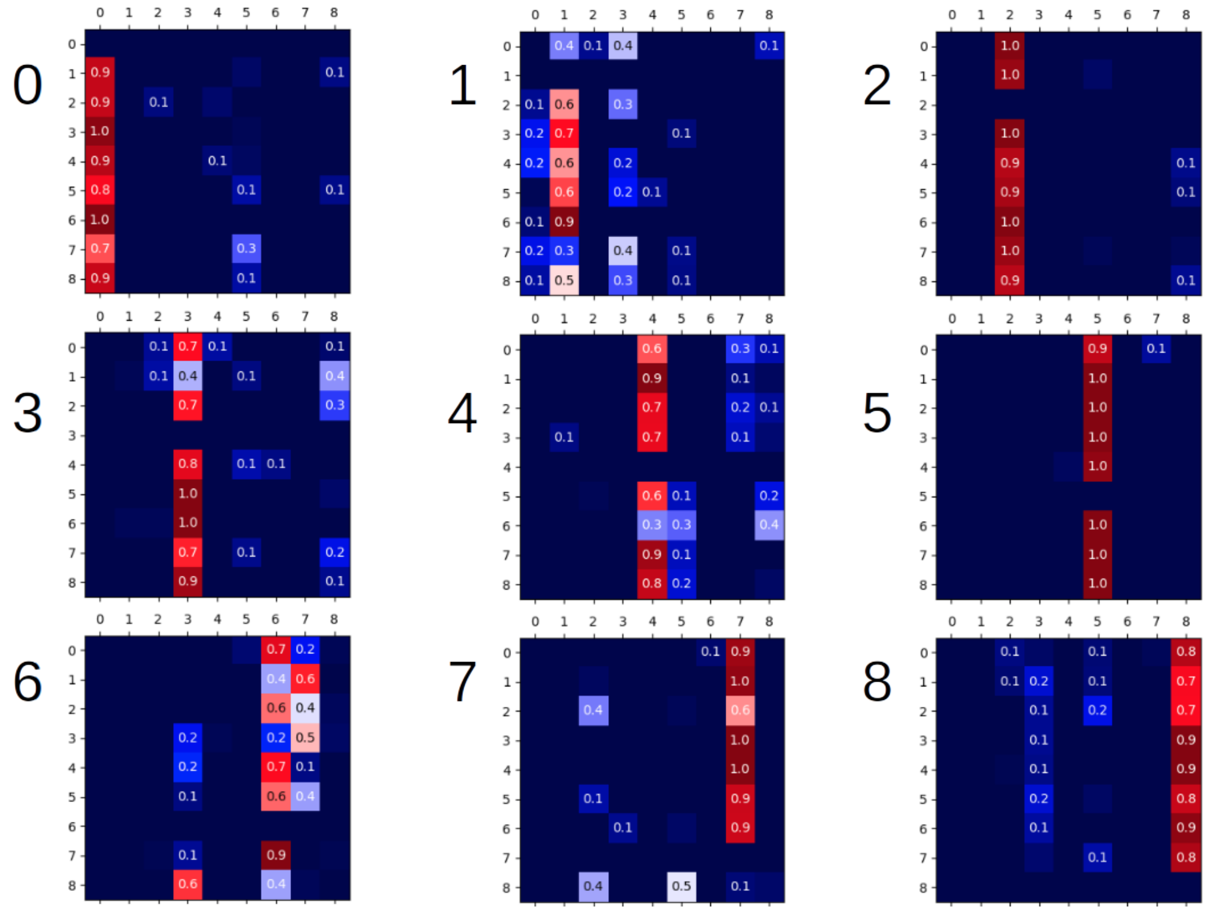
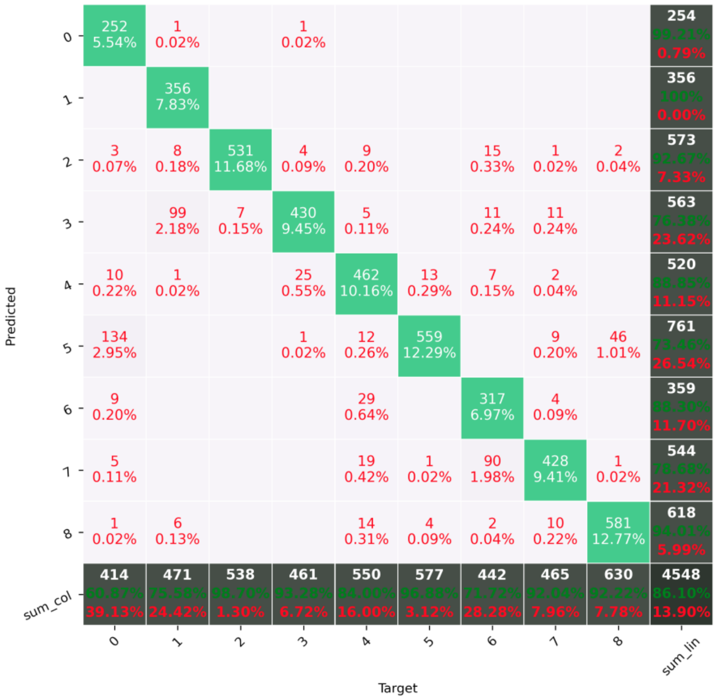


Figure 39: Nine Evaluation Matrices. Red cells are closer to 1, while blue cells are closer to zero. Furthermore, zero cells were omitted for cleaner visualization. On a final note, some row sums might not be exactly 1 because of rounding of the values.

Finally, to check the effect of a more diverse training stage on the network's performance, it was also used a model trained on A1, B1 and C1-C3. The results are shown for the same data representations as before, on figures 40 and 41. However, it is important to note the defect described for user 0 in section 5.1.3 that caused a drop in the model's recognition accuracy for that subject.

As it can be noticed from figure 40, the total accuracy rose from the previous test. Furthermore, all users, but expectantly user 0, saw an increase in accuracy. Disregarding subject 0's effect on the results, the total accuracy rose from 78.298% to 88.631%, which is an extra 10% overall.

Similarly to the experiment on section 5.1.3, and ignoring the accuracy drop of predictions on user 0, there is a general improvement of the model's performance for all subjects, as figure 41 shows. For the case of subject 1, all rows are now correct, meanwhile for user 6 all rows are correct, but for one inconclusive one (row 5). Therefore, once again it was proved that the models benefit from diverse training regimens.



	0	1	2	3	4	5	6	7	8	sum_lin
0	252 5.54%	1 0.02%		1 0.02%						254 99.21% 0.79%
1		356 7.83%								356 100% 0.00%
2	3 0.07%	8 0.18%	531 11.68%	4 0.09%	9 0.20%		15 0.33%	1 0.02%	2 0.04%	573 92.67% 7.33%
3		99 2.18%	7 0.15%	430 9.45%	5 0.11%		11 0.24%	11 0.24%		563 76.38% 23.62%
4	10 0.22%	1 0.02%		25 0.55%	462 10.16%	13 0.29%	7 0.15%	2 0.04%		520 88.85% 11.15%
5	134 2.95%			1 0.02%	12 0.26%	559 12.29%		9 0.20%	46 1.01%	761 73.46% 26.54%
6	9 0.20%				29 0.64%		317 6.97%	4 0.09%		359 88.30% 11.70%
7	5 0.11%				19 0.42%	1 0.02%	90 1.98%	428 9.41%	1 0.02%	544 78.68% 21.32%
8	1 0.02%	6 0.13%			14 0.31%	4 0.09%	2 0.04%	10 0.22%	581 12.77%	618 94.01% 5.99%
sum_col	414 60.87% 39.13%	471 75.58% 24.42%	538 98.70% 1.30%	461 93.28% 6.72%	550 84.00% 16.00%	577 96.88% 3.12%	442 71.72% 28.28%	465 92.04% 7.96%	630 92.22% 7.78%	4548 86.10% 13.90%

Figure 40: Confusion Matrix for the model trained on A1, B1 and C1-C3, and tested on E, with a sample extraction of 1 frame.

5.2 Stage III - 40 speaking subjects

This final stage had the aim to assess the scalability of the network, meaning if it would still perform well for a larger universe of subjects. In order to do so, the first three studies conducted for the previous stage were replicated for the new set of subjects. However, the sample extraction stride was increased to 2, for the increment of the number of users represented a colossal increase in the amount of data to be processed. Given the available tools, a stride of 1 would make the data volume insurmountable to be processed and, thus, a stride of 2 was chosen to cut that volume in half. Furthermore, the details of each experiment replication are listed below.

1. Train a model on tasks A1 and B1, and test it on tasks A2 and B2.

2. Train one model on activity A and another one on B, and test them on task B and task A, respectively.

3. Train one model on tasks C1-C3 and another one on A, B and C1-C3, and test them both on activities C4-C5.

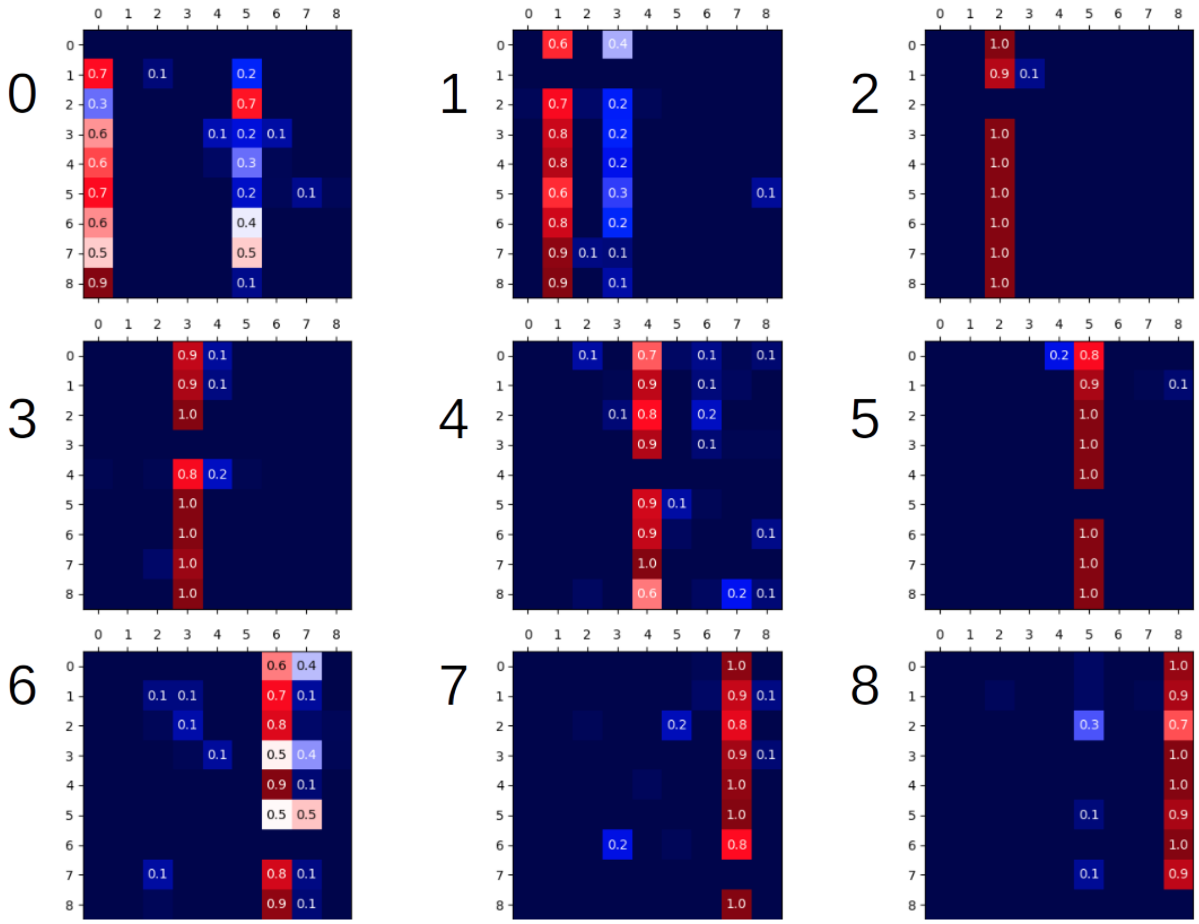


Figure 41: Nine Evaluation Matrices for a more diversely trained model.

5.2.1 Experiment 1

This first experiment had exactly the same procedure as the one on section 5.1.1, the only difference being that during training the size of the batch forwarded into the network was increased from 12 to 32, which was the maximum value the available hardware would allow. This had the purpose to aid the network to iterate on more samples and, thus, improve the generalization between all classes. Ideally, this number would be greater than the total number of classes, so as to allow the model to update its parameters based on an evaluation done over more classes at a time. Apart from that, regarding the evaluation of the trained model, a confusion matrix is provided in figure 42, along with an accuracy histogram for each on of the subjects in figure 43.

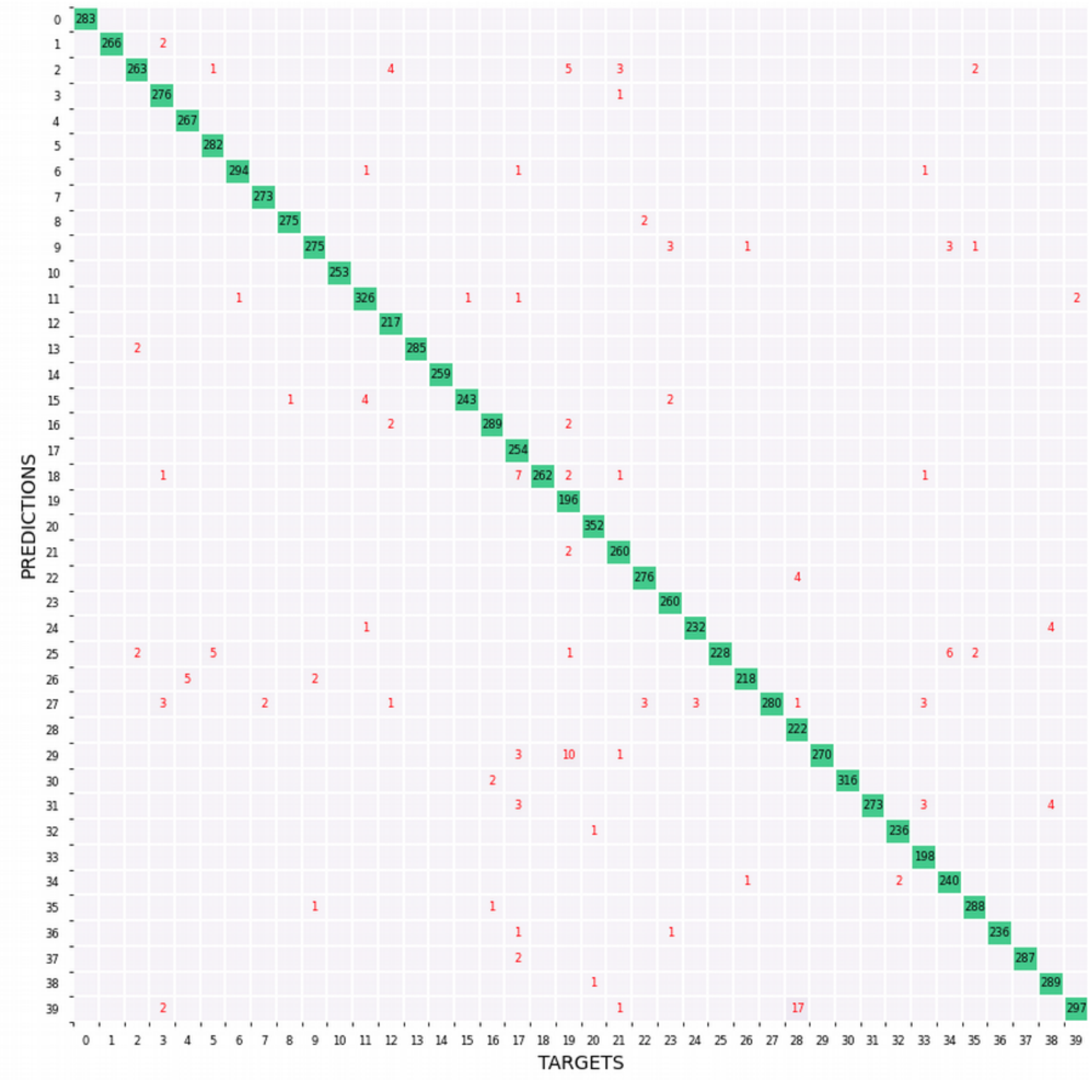


Figure 42: Confusion Matrix for the model trained on A1 and B1, and tested on A2 and B2.

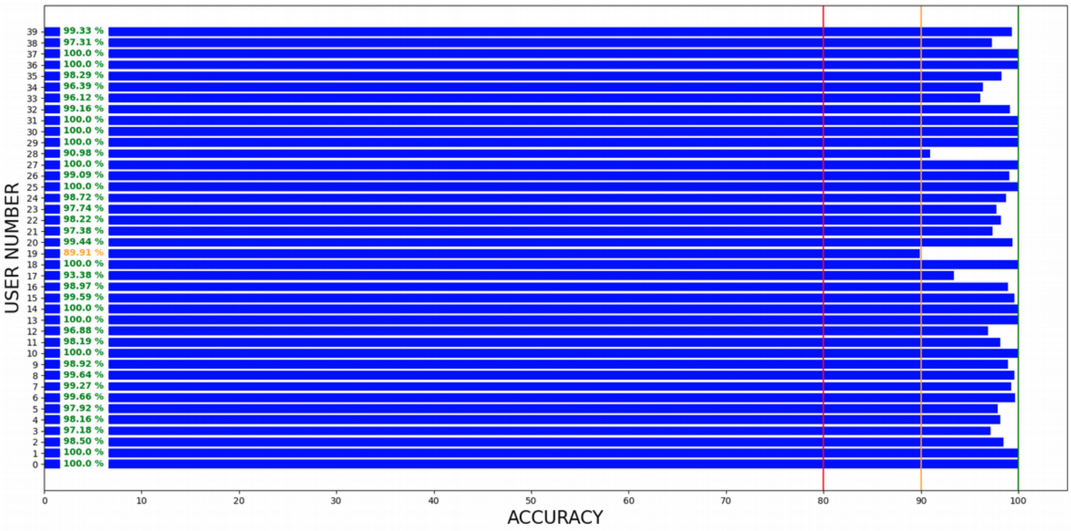
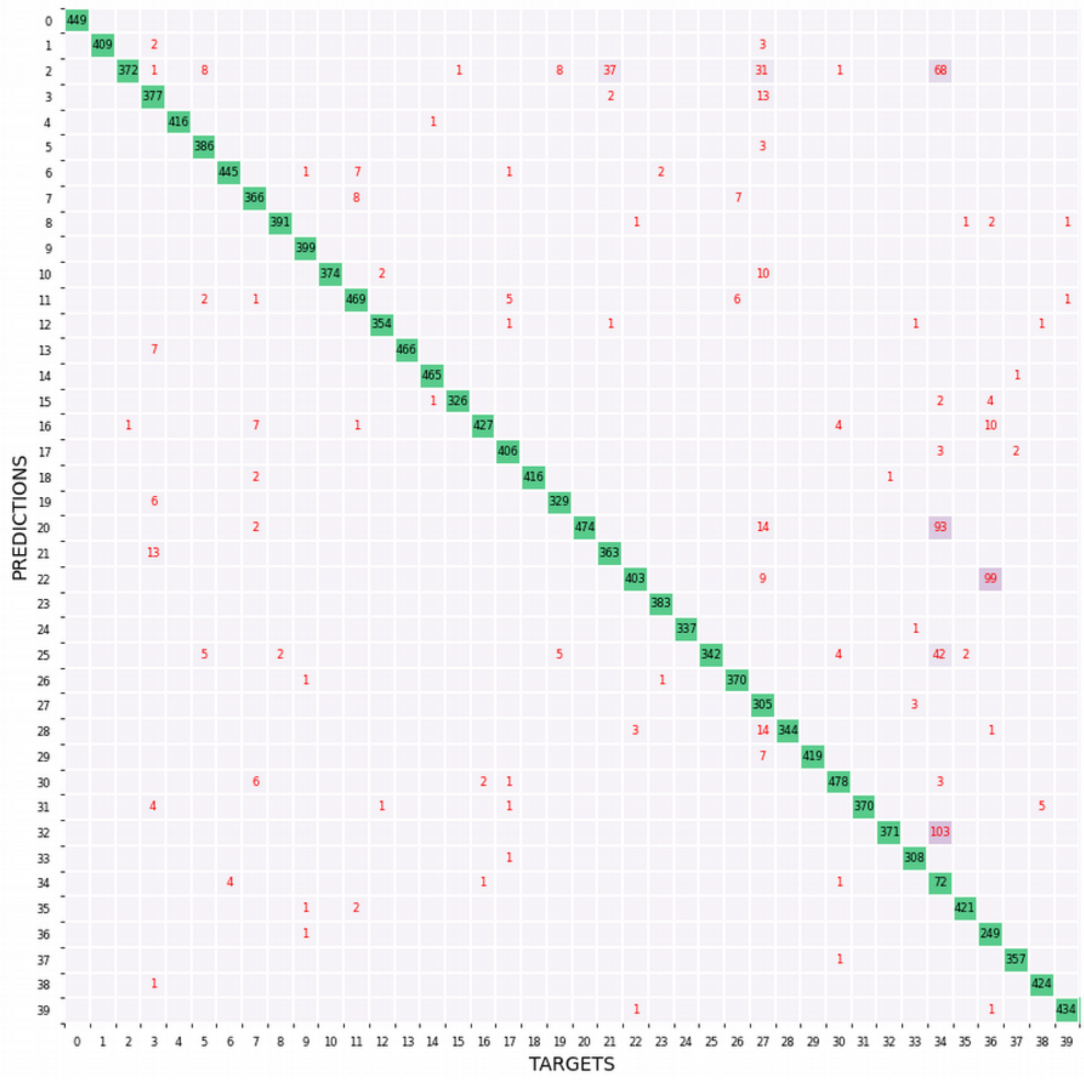


Figure 43: Accuracy Histogram for each subject.

As figure 42 illustrates, no clear confusion between any two users is too relevant. Moreover, the final accuracy of this model was 98.439% and as figure 43 shows, the accuracies per user ranged from nearly 90% up to 100%. Therefore, it can be concluded that, under these training conditions, the network has the potential to scale and be highly effective even for larger sets of speakers.

5.2.2 Experiment 2

This experiment had the goal to ensure that the second experiment of the second stage was scalable. For the purpose of making it more complete, two models were built, one trained on activity A and tested on activity B, and another the other way around. Figures 44 and 45 show the results for the performance test of the first model, while figures 46 and 47 exhibit the outcome of the performance test for the second model.



PREDICTIONS / TARGETS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	449																																							
1		409		2																								3												
2			372	1		8										1				8		37						31			1				68					
3				377																		2						13												
4					416										1																									
5						386																						3												
6							445			1		7						1						2																
7								366				8															7													
8									391														1													1	2			1
9										399																														
10											374		2															10												
11						2		1				469						5									6													1
12													354					1				1												1					1	
13				7										466																										
14															465																							1		
15															1	326																			2		4			
16			1					7				1					427														4						10			
17																		406																	3			2		
18								2											416														1							
19				6																329																				
20								2													474							14							93					
21				13																		363																		
22																							403					9									99			
23																								383																
24																									337									1						
25						5			2											5						342					4				42	2				
26										1														1			370													
27																												305						3						
28																							3					14	344								1			
29																												7		419										
30								6									2	1													478				3					
31				4									1					1														370							5	
32																																	371		103					
33																		1																308						
34							4										1														1				72					
35										1		2																								421				
36										1																											249			
37																															1							357		
38				1																																			424	
39																							1														1			434

Figure 44: Confusion Matrix for the model trained on A and tested on B.

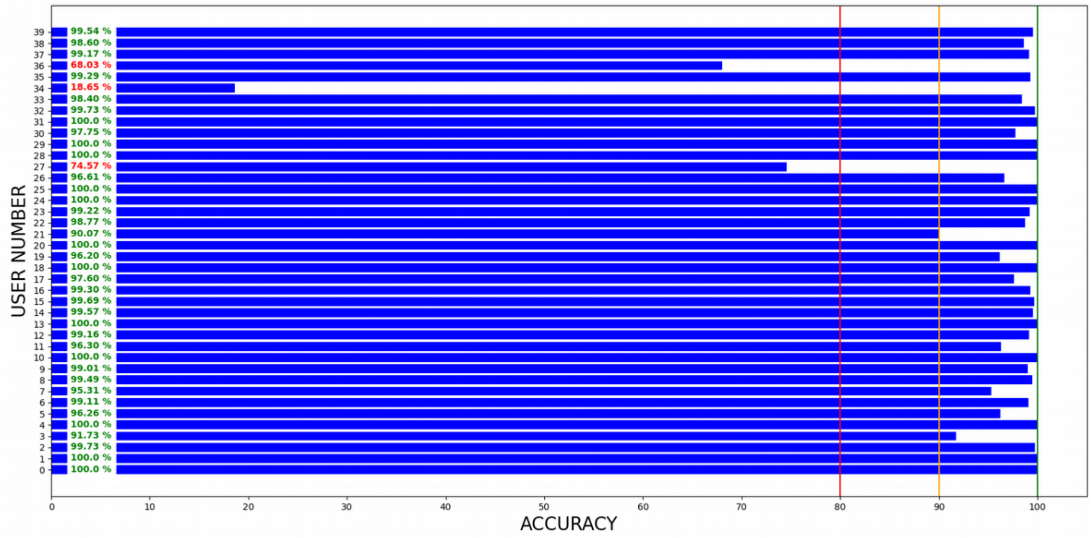
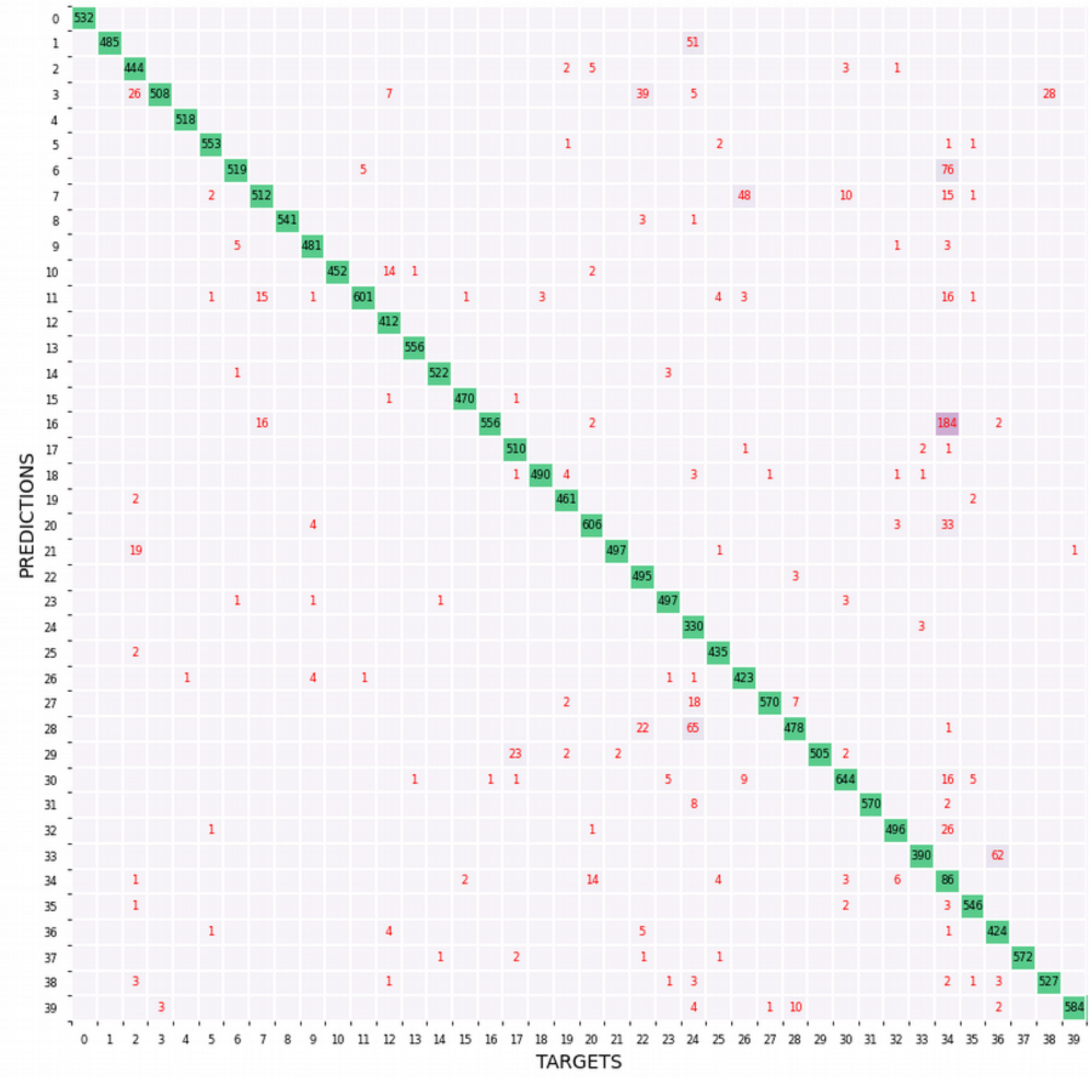


Figure 45: Accuracy Histogram for each subject. Model trained on A and tested on B.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	532																																							
1		485																							51															
2			444																	2	5										3		1							
3			26	508									7										39		5														28	
4					518																																			
5						553														1						2									1	1				
6							519					5																							76					
7						2		512																			48				10				15	1				
8									541														3		1															
9							5			481																							1		3					
10											452		14	1							2																			
11						1		15		1		601				1			3							4	3								16	1				
12													412																											
13														556																										
14							1								522									3																
15													1			470		1																						
16								16									556				2														184		2			
17																		510									1							2	1					
18																		1	490	4					3			1					1	1						
19			2																	461																2				
20										4											606												3		33					
21			19																			497				1														1
22																							495						3											
23							1			1					1									497							3									
24																									330									3						
25			2																							435														
26					1					4		1												1	1		423													
27																				2					18			570	7											
28																							22		65				478						1					
29																		23		2		2								505	2									
30														1			1	1						5			9				644				16	5				
31																									8							570			2					
32						1															1												496		26					
33																																		390			62			
34			1													2					14					4					3		6		86					
35			1																												2				3	546				
36						1							4										5												1		424			
37															1			2					1			1												572		
38			3										1											1	3										2	1	3		527	
39				3																					4			1	10								2			584

Figure 46: Confusion Matrix for the model trained on B and tested on A.

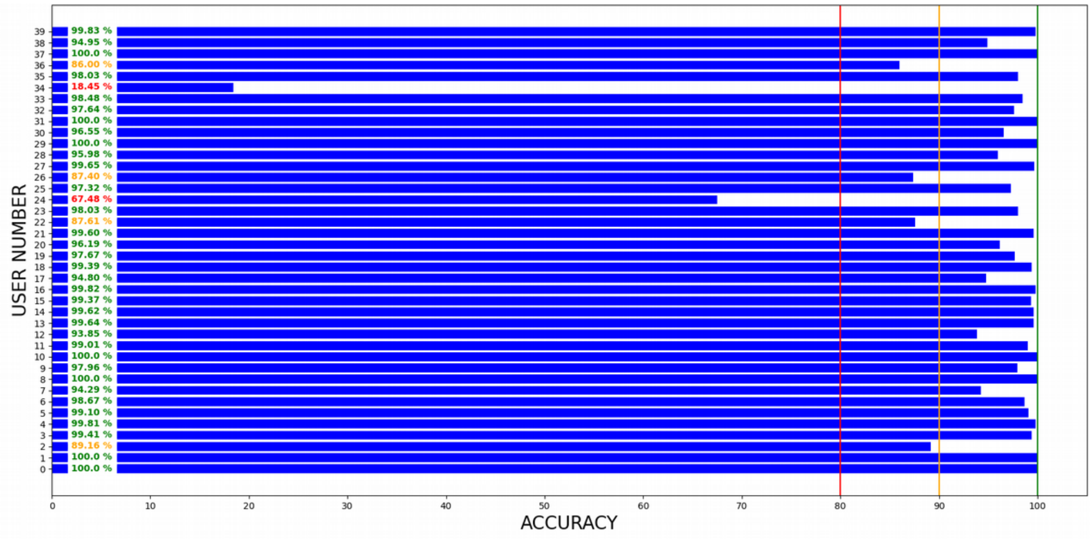


Figure 47: Accuracy Histogram for each subject. Model trained on B and tested on A.

The overall accuracy for the first model was 95.287%, while the second one had a total of 94.809%. However, despite the apparently good generic performance of the models, it can be noticed a big failure from both of them regarding subject 34. Indeed, it was only noticed at the generation of these results, that the user 34 event data from task A was slightly altered, for the camera lens was somewhat unfocused. Hence, the way events were triggered for this subject differed between the two tasks. Consequently, since facial appearance information from event data is quite imperceptible and due to the temporal evolution being disparate between the two tasks, the models had great difficulty recognizing that user. The first model clearly made almost random predictions between four different users, as column 34 shows on figure 44, whereas the second model made its predictions particularly around a set of three different subjects, although being more certain for one of the wrong users.

Nevertheless, there are great results overall for most users. It cannot be left unnoticed, though, that for some subjects the models did not find much replicability of facial stimuli between the two tasks. Moreover, it is important to refer that nursery rhyme A is slightly longer than B, thus, the model trained with A had more data to train with than the one trained with B, which explains the slight drop in recognition rates from the first model to the second one.

On a conclusive note, this experiment was crucial to detect some of the weaknesses of this implementation. Namely, since no facial normalization, alignment or segmentation are done over the images, this methodology relies solely on the networks' capacity to extract features on its own and, thus, is prone to failure on situations in which the training data is largely different from the testing data. This, definitely, questions the reliability of this implementation in a less controlled environment.

5.2.3 Experiment 3

Regarding this final experiment, the main objective was to determine whether models built on data of the sort of task C would still stand as potentially useful, and perform at a reasonable level. One of the biggest factors that could spoil the task C trained models is the scarcity not only of data, but also of diverse data. In any case, the evaluation of the model trained solely on activities C1-C3 is depicted in figures 48 and 49.

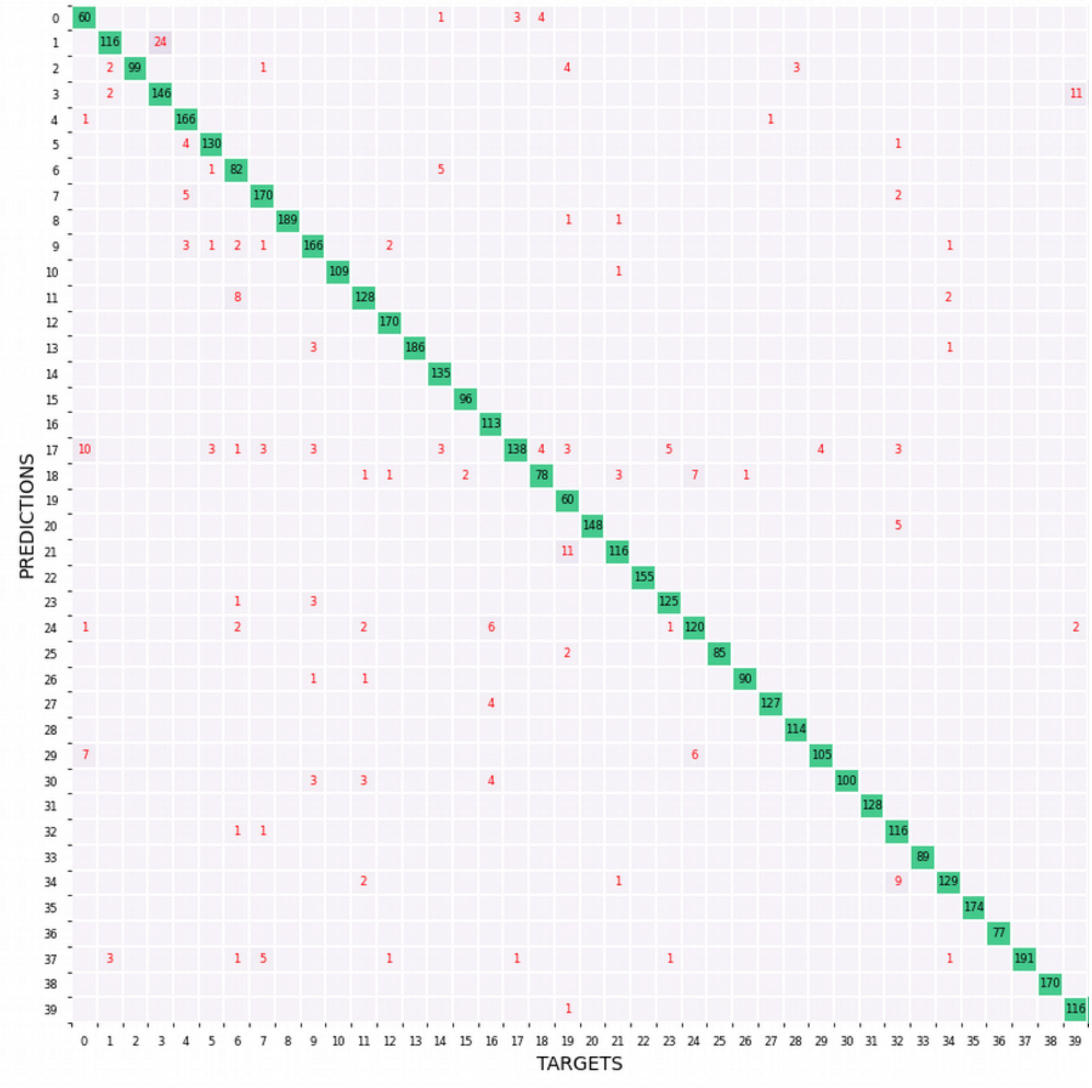


Figure 48: Confusion Matrix for the model trained on C1-C3 and tested on C4 and C5.

As it can be observed in figure 48, there seems to be very little confusion of the model on any particular pair of subjects. However, the model's confidence window for each user suffered a slight drop, ranging from just below 75% and 100%, which is, yet, quite acceptable. Overall, the accuracy of the model for this testing set amounted to 95.213%, still a highly reasonable rate of recognition, proving that, even for this kind of data, the model does not fail to scale. Furthermore, to assess if the model would perform even better provided a richer and more

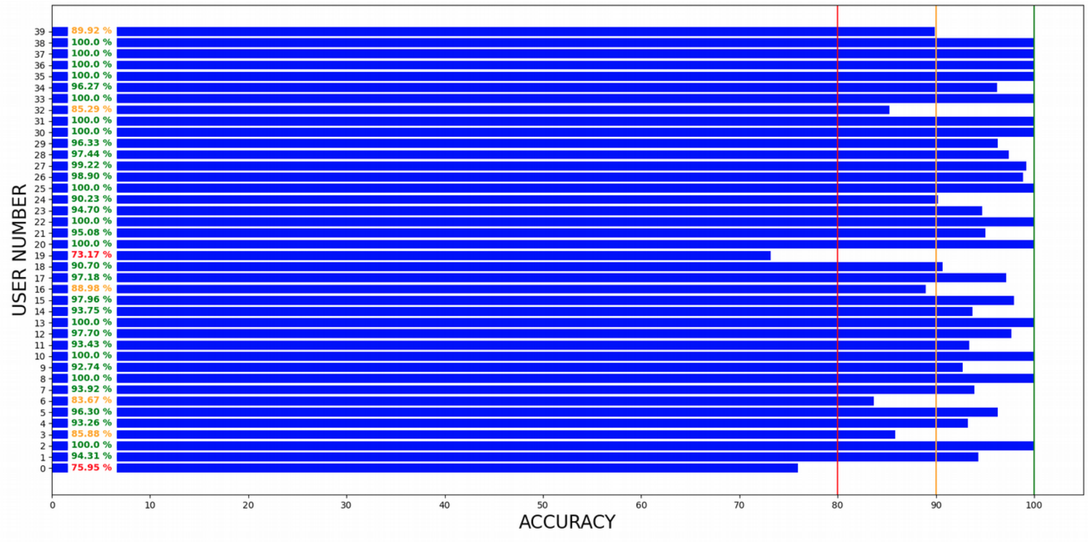


Figure 49: Accuracy Histogram for each subject.

diverse training procedure, the model trained on A, B and C1-C3 was evaluated over the same testing set. The results are illustrated in figures 50 and 51.

For the case of figure 50 it can be appreciated that not a lot of confusion was made by the model, yet again. Besides, figure 51 reveals a great increase in recognition rates for every subject, apart from user 0, which is still affected by the same issue described in the previous stage, on section 5.1.3. Nevertheless, and aside from subject 0, it can be noticed that the majority accuracy values below 90% (red and yellow) rose, in fact, above the 90% mark, thus, proving the importance of diversity in training for better model generalization, once again.

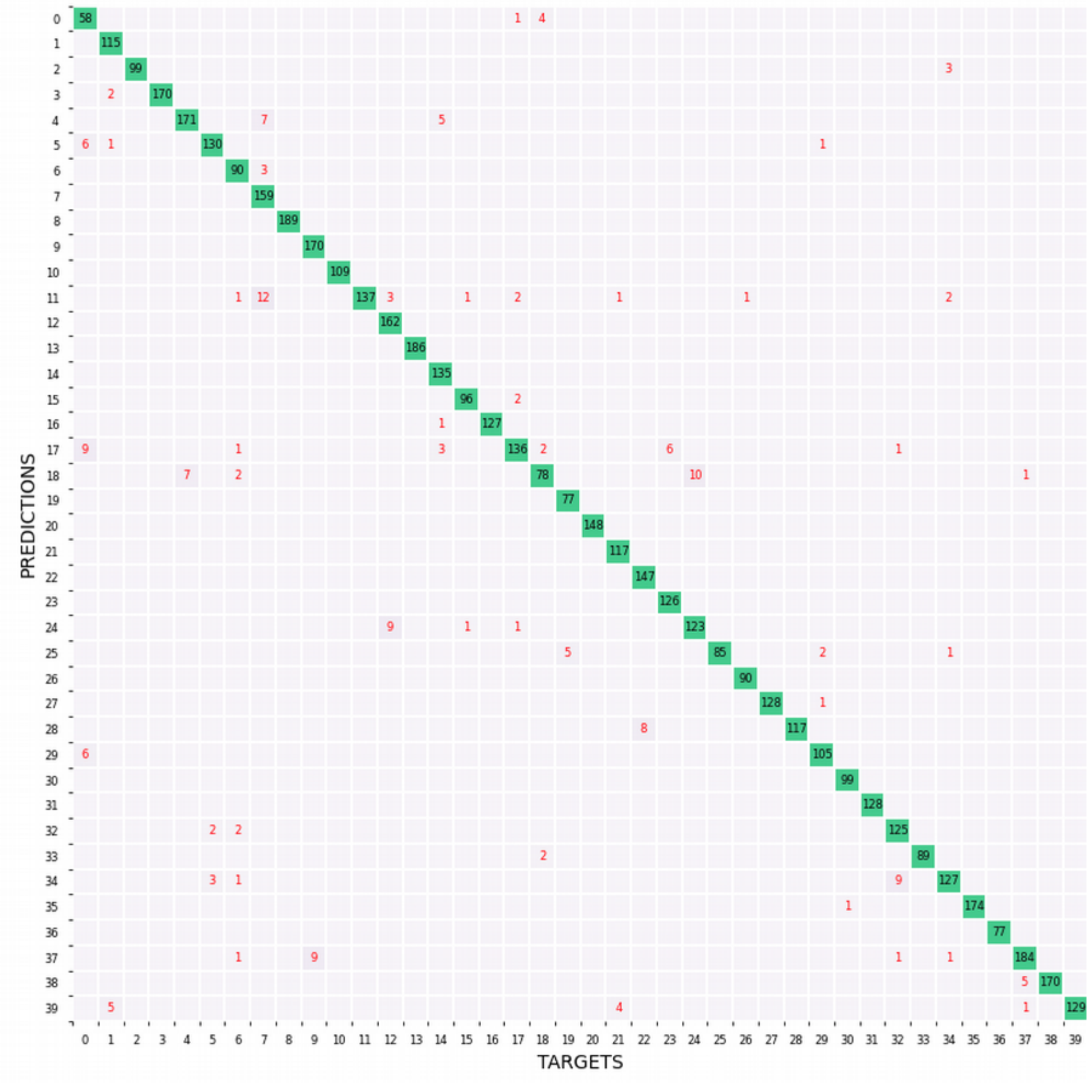


Figure 50: Confusion Matrix for the model trained on A, B and C1-C3, and tested on C4 and C5.

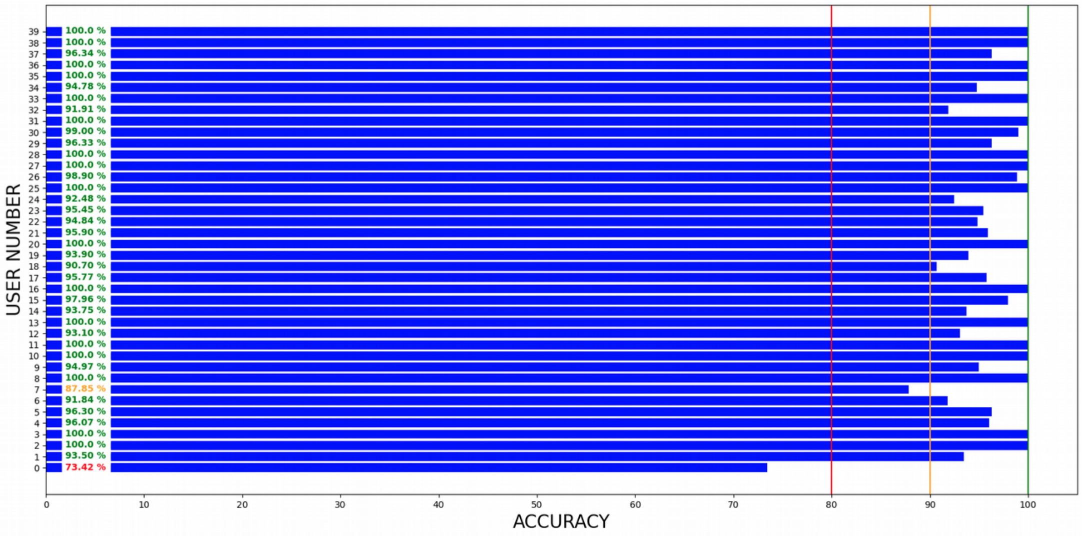


Figure 51: Accuracy Histogram for each subject.

6 Conclusions and Future Work

In conclusion, this work's research contributed to the thesis that facial dynamics is, in fact, a discriminatory trait that aids the process of facial perception and, thus, has great potential to be a strong and robust biometric. Furthermore, it was demonstrated that the I3D network was successful at performing facial recognition tasks and not only what it was first conceived to do - action and gesture recognition. Alike the work on [3], the network proved to operate extremely well, even for sparse data such as events. Not only that, but also the fact that the network was able to maintain high performance levels even for a scaling of the universe of subjects was notable. Naturally, and as expected, it was also concluded that the more visemes and facial expressions shown to the models during training, the better it will do at generally recognizing a subject during the testing stages, independently of what the subject may be saying. Moreover, this work proved that the potential of data provided by Neuromorphic Vision Sensors does not stall in the realm of solely face detection and alignment, but, indeed, is expandable to theoretically more complex tasks such as facial recognition. However, the methodology used in this work is not infallible to the use of glasses, due to the fact that there is not an isolation of facial dynamics from facial appearance, thus, the models consider glasses as part of the facial identity of a person. Nevertheless, interestingly, for subjects, who did not wear glasses for training, the network was still capable of identifying the person when glasses were worn during testing. For a real application system, this might mean that subjects only have to remove their glasses for training stages, while being allowed to wear them when being verified (analogous to the testing phase). Possibly, this is something that can be done with Neuromorphic Vision data, along with spatio-temporal classification, but not with an analysis over single light-intensity-frames, such as RGB or gray-level. Certainly, this is quite an interesting point to be further investigated.

Regarding possible paths and further questions to be studied, apart from the one mentioned previously, the integration of this methodology in a real-time system would be extremely compelling and easy to execute. Additionally, a performance and robustness comparison between Grayscale images and the combination of Grayscale with Events is a highly relevant study and should definitely be further delved into. In relation to the particular methodology applied in this work, there should be more precise solutions for the face normalization and alignment problem, both offline and in real-time. Furthermore, additional studies could focus on possible spoofing mechanisms and formulate techniques that could prevent them from being effective. Finally, the technology used in this work to process event-data is far from being computationally inexpensive. Essentially, for standard image processing techniques to be used, sparse data - the event cloud - is converted to dense information - the event frame - which is redundant for most pixels. Definitely, a more elegant and less taxing methodology would be to process the spatio-temporal data directly over the point clouds, which is already done for other types of work. Moreover, the use of Graphs Neural Networks or even Spike Neural Networks would clearly be a better approach with the view to reduce the computational cost. This reduction in computational expense would benefit the practicability of integration of this technology in a real-world system.

References

[1] K. Dobs, I. Bülthoff, and J. Schultz, “Use and usefulness of dynamic face stimuli for face perception studies—a review of behavioral findings and methodology,” *Frontiers in psychology*, vol. 9, p. 1355, 2018.

[2] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, *et al.*, “Event-based vision: A survey,” *arXiv preprint arXiv:1904.08405*, 2019.

[3] S. U. Innocenti, F. Becattini, F. Pernici, and A. Del Bimbo, “Temporal binary representation for event-based action recognition,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 10 426–10 432.

[4] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, “Event-based visual flow,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 407–417, 2013.

[5] G. K. Cohen, “Event-based feature detection, recognition and classification,” Ph.D. dissertation, Western Sydney University (Australia), 2015.

[6] S. Miao, G. Chen, X. Ning, Y. Zi, K. Ren, Z. Bing, and A. Knoll, “Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection,” *Frontiers in neurorobotics*, vol. 13, p. 38, 2019.

[7] N. F. Chen, “Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 644–653.

[8] A. N. Burkitt, “A review of the integrate-and-fire neuron model: I. homogeneous synaptic input,” *Biological cybernetics*, vol. 95, no. 1, pp. 1–19, 2006.

[9] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” *Advances in neural information processing systems*, vol. 2, 1989.

[10] P. Kim, “Convolutional neural network,” in *MATLAB deep learning*, Springer, 2017, pp. 121–147.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[12] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[13] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[15] H. Kinsley and D. Kukieła, *Neural Networks from Scratch (NNFS)*. [Online]. Available: `https://nnfs.io`.

[16] K. Dobs, I. Bülthoff, and J. Schultz, “Identity information content depends on the type of facial movement,” *Scientific reports*, vol. 6, no. 1, pp. 1–9, 2016.

[17] Z. Ambadar, J. W. Schooler, and J. F. Cohn, “Deciphering the enigmatic face: The importance of facial dynamics in interpreting subtle facial expressions,” *Psychological science*, vol. 16, no. 5, pp. 403–410, 2005.

[18] J. Zhang and R. B. Fisher, “3d visual passcode: Speech-driven 3d facial dynamics for behaviometrics,” *Signal processing*, vol. 160, pp. 164–177, 2019.

[19] S. T. Kim and Y. M. Ro, “Attended relation feature representation of facial dynamics for facial authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1768–1778, 2018.

[20] P. Xie, “Facial movement based human user authentication,” Ph.D. dissertation, Iowa State University, 2014.

[21] H. Dibeklioğlu, A. A. Salah, and F. Gürpınar, “Measurement of facial dynamics for soft biometrics,” in *International Workshop on Face and Facial Expression Recognition from Real World Videos*, Springer, 2014, pp. 69–84.

[22] A. Hadid, M. Pietikäinen, and S. Z. Li, “Learning personal specific facial dynamics for face recognition from videos,” in *International Workshop on Analysis and Modeling of Faces and Gestures*, Springer, 2007, pp. 1–15.

[23] S. T. Kim and Y. M. Ro, “Facial dynamics interpreter network: What are the important relations between local dynamics for facial trait estimation?” In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 464–480.

[24] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. Ho, “Detection of face spoofing using visual dynamics,” *IEEE transactions on information forensics and security*, vol. 10, no. 4, pp. 762–777, 2015.

[25] L. Benedikt, D. Cosker, P. L. Rosin, and D. Marshall, “Assessing the uniqueness and permanence of facial actions for use in biometric applications,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 3, pp. 449–460, 2010.

[26] H. L. Bear and R. Harvey, “Phoneme-to-viseme mappings: The good, the bad, and the ugly,” *Speech Communication*, vol. 95, pp. 40–67, 2017.

[27] H. L. Bear, R. W. Harvey, B.-J. Theobald, and Y. Lan, “Which phoneme-to-viseme maps best improve visual-only computer lip-reading?” In *International Symposium on Visual Computing*, Springer, 2014, pp. 230–239.

[28] H. L. Bear and S. Taylor, “Visual speech recognition: Aligning terminologies for better understanding,” *arXiv preprint arXiv:1710.01292*, 2017.

[29] J. L. Newman and S. J. Cox, “Speaker independent visual-only language identification,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2010, pp. 5026–5029.

[30] M. Behzad, N. Vo, X. Li, and G. Zhao, “Automatic 4d facial expression recognition via collaborative cross-domain dynamic image network,” *arXiv preprint arXiv:1905.02319*, 2019.

[31] N. Yitzhak, S. Gilaie-Dotan, and H. Aviezer, “The contribution of facial dynamics to subtle expression recognition in typical viewers and developmental visual agnosia,” *Neuropsychologia*, vol. 117, pp. 26–35, 2018.

[32] M. Verma, S. K. Vipparthi, G. Singh, and S. Murala, “Learnet: Dynamic imaging network for micro expression recognition,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1618–1627, 2019.

[33] S. Mariooryad and C. Busso, “Facial expression recognition in the presence of speech using blind lexical compensation,” *IEEE Transactions on Affective Computing*, vol. 7, no. 4, pp. 346–359, 2015.

[34] S. Barua, Y. Miyatani, and A. Veeraraghavan, “Direct face detection and video reconstruction from event cameras,” in *2016 IEEE winter conference on applications of computer vision (WACV)*, IEEE, 2016, pp. 1–9.

[35] D. R. Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.-H. Ieng, and R. Benosman, “An asynchronous neuromorphic event-driven visual part-based shape tracking,” *IEEE transactions on neural networks and learning systems*, vol. 26, no. 12, pp. 3045–3059, 2015.

[36] B. Ramesh and H. Yang, “Boosted kernelized correlation filters for event-based face detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2020, pp. 155–159.

[37] A. Savran and C. Bartolozzi, “Face pose alignment with event cameras,” *Sensors*, vol. 20, no. 24, p. 7079, 2020.

[38] G. Lenz, S.-H. Ieng, and R. Benosman, “Event-based face detection and tracking using the dynamics of eye blinks,” *Frontiers in Neuroscience*, vol. 14, p. 587, 2020.

[39] G. Chen, L. Hong, J. Dong, P. Liu, J. Conradt, and A. Knoll, “Eddd: Event-based drowsiness driving detection through facial motion analysis with neuromorphic vision sensor,” *IEEE sensors journal*, vol. 20, no. 11, pp. 6170–6181, 2020.

[40] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, “Graph-based object classification for neuromorphic vision sensing,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 491–501.

[41] Y. Sekikawa, K. Hara, and H. Saito, “Eventnet: Asynchronous recursive event processing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3887–3896.

[42] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, “Hats: Histograms of averaged time surfaces for robust event-based object classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1731–1740.

[43] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, “Learning to detect objects with a 1 megapixel event camera,” *arXiv preprint arXiv:2009.13436*, 2020.

[44] S. Afshar, N. Ralph, Y. Xu, J. Tapson, A. v. Schaik, and G. Cohen, “Event-based feature extraction using adaptive selection thresholds,” *Sensors*, vol. 20, no. 6, p. 1600, 2020.

[45] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, “Asynchronous event-based multikernel algorithm for high-speed visual features tracking,” *IEEE transactions on neural networks and learning systems*, vol. 26, no. 8, pp. 1710–1720, 2014.

[46] J. C. Tapson, G. K. Cohen, S. Afshar, K. M. Stiefel, Y. Buskila, T. J. Hamilton, and A. van Schaik, “Synthesis of neural networks for spatio-temporal spike pattern recognition and processing,” *Frontiers in neuroscience*, vol. 7, p. 153, 2013.

[47] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, “Hots: A hierarchy of event-based time-surfaces for pattern recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1346–1359, 2016.

[48] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, “Action recognition with dynamic image networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2799–2813, 2017.

[49] Q. Wang, Y. Zhang, J. Yuan, and Y. Lu, “Space-time event clouds for gesture recognition: From rgb cameras to event cameras,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 1826–1835.

[50] J. Chen, J. Meng, X. Wang, and J. Yuan, “Dynamic graph cnn for event-camera based gesture recognition,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2020, pp. 1–5.

[51] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, *et al.*, “A low power, fully event-based gesture recognition system,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7243–7252.

[52] J. H. Lee, T. Delbruck, M. Pfeiffer, P. K. Park, C.-W. Shin, H. Ryu, and B. C. Kang, “Real-time gesture interface based on event-driven processing from stereo silicon retinas,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 12, pp. 2250–2263, 2014.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[54] N. G. Xiao, S. Perrotta, P. C. Quinn, Z. Wang, Y.-H. P. Sun, and K. Lee, “On the facilitative effects of face motion on face recognition and its development,” *Frontiers in psychology*, vol. 5, p. 633, 2014.

Appendices

I Neuromorphic-Vision Dataset for Facial Dynamics Specifications

This appendix serves the purpose to detail the layout of the Dataset built for this thesis. The whole folder tree is presented in figure 52.

```
    DATASET
     |
     |__FixedMode
     |   |
     |   |__userAAA
     |       |
     |       |__taskBB
     |           |
     |           |__recordingCC
     |               |
     |               |__pointcloud.csv
     |
     |__LoopMode
         |
         |__Events
         |   |
         |   |__userAAA
         |       |
         |       |__taskBB
         |           |
         |           |__recordingCC
         |               |
         |               |__pointcloud.csv
         |
         |__Grayscale
             |
             |__userAAA
                 |
                 |__taskBB
                     |
                     |__recordingCC
                         |
                         |__grayscaleframes.png
AAA = [000 .. 039]
BB  = [01  ..  05]
CC  = [01  ..  08]
```

Figure 52: Dataset Folder Tree

Firstly, there are two main folders: `FixedMode` and `LoopMode`. The `FixedMode` folder contains solely "Comma Separated Values (csv)" files with Event Clouds, whereas `LoopMode` accommodates both Event Clouds and Grayscale Frames. However, despite having the same tasks, the data on these folders were recorded separately. That is to say, that the subjects had to record everything twice, one for each camera modality.

Secondly, the structure inside the two main directories has the subjects listed (userAAA), followed by the tasks (taskBB) and, lastly, each recording instance (recordingCC) for each task. More detailed information is provided by tables 5 and 7. Table 5 describes each and every activity performed by the subjects, whereas table 7 reveals which activity was done on each recording of each task.

Table 5: Task Activity Description

Activities	Description
A1	Read Nursery Rhyme A
A2	Read half of the Nursery Rhyme A
B1	Read Nursery Rhyme B
B2	Read half of the Nursery Rhyme B
C	Pronounce Full Name
D	Odd videos for spoofing attempts[10]
E	People mouthing the others' names[11]

Regarding activity D, to be more specific than the remark on ([10]), user000 did not wear glasses, while user001 wore user000's glasses. Meanwhile, concerning activity E, going beyond the remark on ([11]), for the general cases of user000, user003, user006 and user008 the correspondences between recording number and the number of the user's name pronounced go as follows in table 6.

Table 6: Recording Number to User Number Correspondence

user000		...	user003		...	user006		...	user008	
Rec. No.	User. No.		Rec. No.	User. No.		Rec. No.	User. No.		Rec. No.	User. No.
1	1		1	0		1	0		1	0
2	2		2	1		2	1		2	1
3	3		3	2		3	2		3	2
4	4		4	4		4	3		4	3
5	5		5	5		5	4		5	4
6	6		6	6		6	5		6	5
7	7		7	7		7	7		7	6
8	8		8	8		8	8		8	7

[10] Only user000 and user001 completed this activity.

[11] This was only done between the first 9 subjects of the dataset.

Table 7: Task Layout Throughout the Recordings

		Task Number				
		task01	task02	task03	task04	task05
Recording Number	recording01	A1	B1	C1	D	E
	recording02	A2	B2	C2	D	E
	recording03	-	-	C3	-	E
	recording04	-	-	C4	-	E
	recording05	-	-	C5	-	E
	recording06	-	-	-	-	E
	recording07	-	-	-	-	E
	recording08	-	-	-	-	E

Lastly, one final remark goes to the point cloud files of each modality. On one hand, for the `FixedMode` event clouds, the event information goes as follows:

```
[timestamp, x, y]
```

Also, the `timestamp` column does not start at zero and the events are not sorted. On the other hand, for the `LoopMode` clouds, the event information has one extra parameter:

```
[frame_number, timestamp, x, y]
```

The `frame_number` column links each event to a grayscale frame, which synchronizes both data channels. For instance, the event point represented by `[10, t, x, y]` occurred somewhere after frame `10` and before frame `11`. In addition, just like the case of `FixedMode`, `LoopMode` clouds are not sorted by `timestamp`. Nonetheless, the `frame_number` column is sorted.

II Nursery Rhymes

The Portuguese Nursery Rhymes used for the first two tasks of the subject task protocol for the NVSFD Dataset were extracted from the website `https://www.slideshare.net/MaraPinto2/lengalengas-5967731`. The first task used the one named "Gato Maltês", meanwhile the second used "O que está na varanda?".

Gato Maltês

Era uma vez
Um gato maltês
Tocava piano
E falava francês
Queres que te conte outra vez?

Era uma vez
Um gato maltês
Saltou-te às barbas
Não sei que te fez
Queres que te conte outra vez?

Era uma vez
Um gato maltês
Tocava piano
Falava francês
A dona da casa
Chamava-se Inês
O número da porta era o 33!
Queres que te conte outra vez?

Era uma vez
Uma galinha perchês
E um galo francês
Eram dois
Ficaram três...
Queres que te conte outra vez?

O que está na varanda?

O que está na varanda?
Uma fita de ganga.
O que está na panela?
Uma fita amarela.
O que está no poço?
Uma casca de tremoço.
O que está no telhado?
Um gato malhado.
O que está na chaminé?
Uma caixa de rapé.
O que está na rua?
Uma espada nua.
O que está atrás da porta?
Uma vara torta.
O que está no ninho?
Um passarinho.
Deixa-o no morno.
Dá-lhe pãozinho.