Face and Facial Landmark Detection for Event-based Imaging

by

Tomiris Rakhimzhanova

Submitted to the Department of Robotics and Mechatronics
in partial fulfillment of the requirements for the degree of

Master of Science in Robotics

at the

NAZARBAYEV UNIVERSITY

Apr 2023

Author .....
Department of Robotics and Mechatronics
Apr 29, 2023

Certified by .....
Huseyin Atakan Varol
Full Professor of Robotics
Thesis Supervisor

Accepted by .....
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

Face and Facial Landmark Detection for Event-based Imaging

by

Tomiris Rakhimzhanova

Submitted to the Department of Robotics and Mechatronics
on Apr 29, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science in Robotics

Abstract

Computer vision, an essential component of robotics, is an expanding field of research. While substantial advancements have been made in visual camera technology, conventional cameras still exhibit limitations, such as motion blur and low dynamic range, owing to their image acquisition and output format as 2-dimensional arrays. Event-based imaging is addressing these bottlenecks. Consequently, the utilization of event-based cameras has been gaining traction in the realm of robotics. These cameras asynchronously capture each pixel, providing numerous possibilities. Nevertheless, as a novel technology, many applications remain unexplored, such as utilizing event cameras for face detection and facial landmarks.

Although there has been a surge of research into face detection using event cameras, the lack of a comprehensive, annotated dataset of face bounding boxes and facial landmarks in event streams has impeded progress in this field. This thesis endeavors to bridge this gap by introducing the pioneering Faces in Event Streams (FES) dataset, which covers 689 minutes and is specifically designed to detect faces and facial landmarks for direct event-based camera output.

To showcase the efficacy of the FES dataset, 12 models were developed and trained to predict bounding box coordinates and facial landmarks with an mAP50 score exceeding 90%. Furthermore, during the course of the thesis research, efforts were made to demonstrate real-time face recognition using an event camera with the aid of one of our pre-trained models. The published dataset and pre-trained models are publicly available for further study at https://github.com/IS2AI/faces-in-event-streams.

Thesis Supervisor: Huseyin Atakan Varol
Title: Full Professor of Robotics

Acknowledgments

First and foremost, I would like to express my profound and heartfelt gratitude to Dr. Huseyin Atakan Varol for his transformative influence on my academic experience at Nazarbayev University. As the founding director of the Institute of Smart Systems and Artificial Intelligence (ISSAI), Dr. Varol provided our group with the motivation, inspiration, knowledge and resources to delve into the world of cutting-edge artificial intelligence research. Under his guidance and unwavering support, which recognized my potential, I was given the opportunity to be a member of ISSAI. This experience enabled me to grow exponentially as a specialist in a short period, significantly enhancing my professional skills beyond my initial expectations.

I would also like to convey my sincere appreciation to all the dedicated data scientists, computer engineers, researchers, and administrative staff of ISSAI. Together, they have fashioned an unparalleled research environment that any graduate student would dream of. Special recognition is extended to my project partner, Ulzhan Bisarinova, with whom I have had the honor of cooperating closely from the very inception of our project.

I am deeply indebted to Nazarbayev University for offering me a top-notch education and exceptional faculty members. My sincerest thanks go to the Department of Robotics and all its esteemed professors for their invaluable guidance, expertise, and engaging lectures, which rendered the learning experience truly unforgettable. I would like to particularly acknowledge Prof. Tohid Alizadeh for his leadership, for addressing my inquiries, and for providing unwavering support throughout the thesis submission process. I also would like to express my sincere thanks to Prof. Adnan Yazici. It is an honor to have a world-renowned scientist like him as a committee member. Furthermore, Nazarbayev University and its community consistently strive to create an environment conducive to academic success, as evidenced by the numerous resources available, including reading rooms, recreation areas, cafeterias, gym, swimming pool, and much more. I wish to mention warmly my fellow robotics classmates for making this journey enjoyable and for the creation of cherished, lifelong

memories.

Special appreciation is reserved for my dear husband and parents, whose unconditional love, motivation, confidence, and support I have felt throughout my academic journey.

This extraordinary academic experience has inspired me to further pursue scientific research in the field of artificial intelligence, ultimately leading me to aspire to become an academician in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer vision is an interdisciplinary field of research aimed at enabling computers to understand and interpret visual data. Its goal is to create methods and algorithms that can automatically analyze digital images and videos to extract relevant data, detect objects and actions, track changes, and create realistic representations of the world [87]. With applications across many industries and disciplines, computer vision has become an important area of research in computer science and engineering in recent decades. It has several applications in robotics, autonomous driving, security, and imaging in medical fields, among other fields. Deep learning techniques have enabled the creation of models that extract information more accurately, leading to significant advances in computer vision in recent years.

Recently, research in the development of various subfields of computer vision, such as action recognition, semantic segmentation, object detection and classification, including face detection and recognition, and others, has been gaining popularity. The state of the art literature will be analyzed in the following sections.

1.1 Action recognition

One of the areas of computer vision is action recognition, which solves the problem of extracting information about human movements from video. In this direction, examples are articles on the study of the application of various algorithms and the

study of improving the accuracy of recognition of an action. For example, such articles are [95],[60] where the authors study and propose methods for determining the area of human movement on a frame in spatio-temporal space.

In comparison, several papers have approached the problem of motion classification by focusing on temporal aspects. For instance, the authors in the article [101] propose to use a network of structured segments, and in a similar article [12] a new approach is proposed using a temporal context network to improve the accuracy of classifying human movements. One of the sub-directions of this topic is the study of specific aspects of motion recognition, which include the detection of the completion of motion in temporal space. So, such are the articles [96], [93], where the authors propose a method that uses temporal models to localize the completion of an action in a video.

Also recently, there has been a growing number of articles on the application of various architectural models of deep learning and their research for the classification of human actions. An example is the article [43], where instead of using two separate neural networks, such as CNN and RNN, to extract spatio-temporal characteristics, the authors propose a single method using the Temporal Convolutional Network, which reduces the computational cost of action recognition. In addition, in the article [14], the authors propose, by replacing the layers of the middle pool with a weighted attention pool in ResNet architectures, to increase the accuracy of recognizing people's actions on popular data sets. In addition, the article [8] proposes a context-sensitive architecture using LSTM neural networks to predict actions based on fully extracted information from past frames to improve classification accuracy.

1.2 Object detection and classification

A rapidly developing field of computer vision that has made significant progress in recent years is the identification and classification of objects [7]. This topic includes the development of algorithms and models that can automatically detect and find objects in images and video streams. These algorithms are used in many industries

such as augmented reality, self-driving cars and security systems.

One of the main tasks of object detection is to improve the accuracy of object detection based on visual information. To solve this problem, various neural networks are actively used, which are divided into one-stage and two-stage neural networks. One-stage object detection algorithms are faster but less accurate than two-stage neural networks because they directly predict the class and location of objects in a single pass through the network [7]. Examples are YOLO [73], which predicts bounding boxes, class probabilities, and confidence scores for each grid cell after gridding the input image, and Single Shot Detector (SSD) [53], which is another single-stage object detection algorithm that uses multiple layers to detect objects of different size and aspect ratio. On the other hand, the accuracy of two-stage object detection algorithms is higher, but the processing time is longer, since they first create region proposals and then refine them to find objects [87]. One of the representatives of such an algorithm is the Faster R-CNN model developed by Ren et al. [74], which is a well-known object detection model based on deep learning. This method creates potential areas of interest in an image using a network of area suggestions and then uses a convolutional neural network to categorize and fine-tune those areas.

1.3 Semantic segmentation task

Assigning semantic labels to each pixel in an image is the goal of a computer vision task known as semantic segmentation. The goal is to divide an image into meaningful chunks that represent real objects or parts of real objects such as streets, buildings, and people [20]. Semantic segmentation offers a dense pixel-by-pixel classification of the entire image, unlike object detection, which only provides bounding boxes around objects.

For a more detailed overview of the application of image segmentation for driving autonomous vehicles, see [20], where the authors describe segmentation methods and existing datasets for use in autonomous driving. Other articles discuss performance and accuracy improvements to semantic image segmentation.An example of this is the

article [32], in which the authors solve the problem of data loss at the stage of feature extraction using an edge detection network, while improving the performance and accuracy of object segmentation. The related article [54] suggests using attention modules to solve the problem of semantic segmentation in cases of complex image scenarios. In addition, in [99], the authors explore the application of real-time image segmentation and propose an EADNet network that improves segmentation accuracy using minimal parameters.

1.4 Pose estimation

Pose estimation is the most important task of computer vision, the purpose of which is to identify and classify the nodes of an object or person in an image or video.

One promising approach to pose estimation is the detection and tracking paradigm, which combines object detection and tracking to achieve efficient and accurate results. This paradigm is used in [24], where the authors used Mask R-CNN to predict human key points on short clips, followed by prediction of key point locations using the proposed 3D extension system.

Another method for detecting pose keypoints is using partial similarity fields (PAFs), which predict the location of parts of the human body and link different parts of the image. This approach was used in [9] for estimating poses in multi-person images and in [10], where the authors released an open source code and real-time human pose detection system.

1.5 Description of used cameras in computer vision

One of the most important elements of computer vision is the camera, which is used to capture images and videos for analysis. Various cameras are used in computer vision applications, the best known being RGB cameras, depth cameras, and thermal cameras.

RGB cameras are widely used in computer vision, robotics applications due to

their affordability and affordability. These cameras take pictures under visible light conditions, which allows them to analyze colors and detect patterns and contours in the environment [7]. Red, green, and blue are typically represented as three separate channels in RGB images, which are then combined to create a full color image. In computer vision, RGB cameras are used for various tasks such as object detection and recognition, tracking, and segmentation. For example, in [18], the authors used an RGB-D camera that includes RGB images and depth information to make robots detect and avoid pedestrians.

The camera that provides information about the environment provides information about the temperature difference in the scene, called a thermal imaging camera, which is widely used in computer vision and robotics. They do this by capturing the thermal radiation emitted by objects in the scene and turning it into an image or video that can be explored in a variety of ways. Thermal imaging cameras are often used in computer vision to detect and track objects in low light conditions where regular RGB cameras can perform poorly [79]. They can also be used for temperature estimation, crowd analysis, and people counting. For example, in the article [37], the authors use a thermal imaging camera to navigate a quadrocopter in very low light conditions. Also in the article [79], the authors developed a mobile robot for face recognition and determining the temperature of a person.

An excellent application in robotics for tasks of navigation and manipulation, where an accurate perception of the environment is important, has found a depth camera [36]. Such a camera provides information about the depth of objects. In comparison with RGB cameras, depth cameras measure the distance between the camera and objects in a scene using various methods such as time-of-flight or structured light, allowing the creation of 3D images. For example, in the article [36], the authors developed a robot that tracks and recognizes the movement of a person and follows him along the path in real time using depth camera.

1.6 Output of traditional cameras

Any camera used in computer vision, including thermal, RGB, and depth cameras, can output an image or video frame. An image or frame is created by a camera by collecting light rays reflected from nearby objects and focusing them on a light-sensitive surface such as a sensor or film [18]. Thus, consecutive frames form one video stream.

However, in order to solve the problem of slow capture rate, in 2008 the first event-based camera appeared on the market [4], which presents the output as packets of events in space, resulting in sparse and efficient results [66]. Instead of capturing entire frames, as conventional cameras do, they can simply record the changes that occur in a scene, resulting in faster processing and less bandwidth usage (see Fig.1-1). However, the use of these cameras is limited by the relatively recent appearance of cameras on the market and the new approach to information processing, which makes it impossible to apply existing algorithms used for processing frame-based camera output [62],[23]. Consequently, a small number of studies have placed a limitation

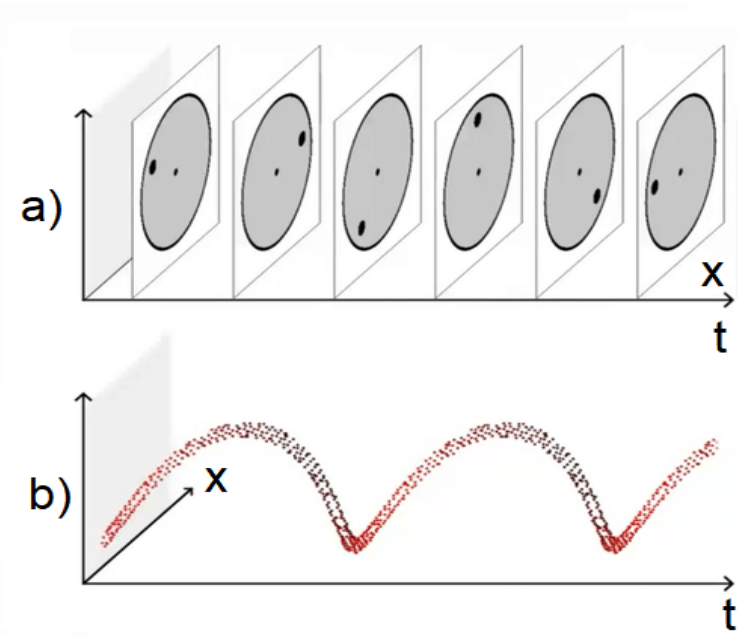

Figure 1-1: Comparison between outputs: a) conventional camera output, b) event-based camera output. Adapted from [28].

on the use of cameras on par with traditional ones. Therefore, this research aims to explore the use of event-based camera output to determine bounding box coordinates and facial landmarks of the face.

Chapter two takes a close look at the key ideas and distinguishing characteristics of event-based cameras compared to frame-based ones. This chapter discusses in detail how event cameras work, including a mathematical explanation of event detection and processing. The chapter also discusses the benefits of using event-based cameras and highlights the challenges, and it provides a review of the literature on the use of event cameras in robotics, as well as an overview of research on the creation and use of face detection and facial landmark detection technologies, emphasizing their importance for computer vision. Finally, it concludes by outlining the significant contribution that the current thesis work brings to the development of event-based camera technology for face and facial landmarks field.

Chapter three contains a detailed description of the process used to collect data for a dataset. The chapter discusses various aspects of data collection, including how the data is annotated, experiments performed, and statistics. In addition, the chapter explains the structure of the dataset as well as the splitting process that was undertaken to prepare for further machine learning processes.

Chapter four clarifies the detailed description of the model architecture with exploring the blocks of architecture. This chapter also discusses the cost function, problem formulation and notation to provide a solid understanding of the underlying methodology used in conducting experiments.

Chapter five contains a comprehensive analysis of the experimental results, including determining the optimal accumulation time, testing DFES models on unseen data, and comparing different models. In addition, the chapter describes the results of calculating the inference time and provides a detailed discussion of the overall results.

Chapter six provides a conclusion with possible future work description.

Chapter 2

Literature review

Despite advances in traditional cameras, they still have limitations in capturing all environmental information in certain situations. These limitations include low dynamic range, dependence on illumination, immobility of objects, and time delay in creating a frame [42], [38].

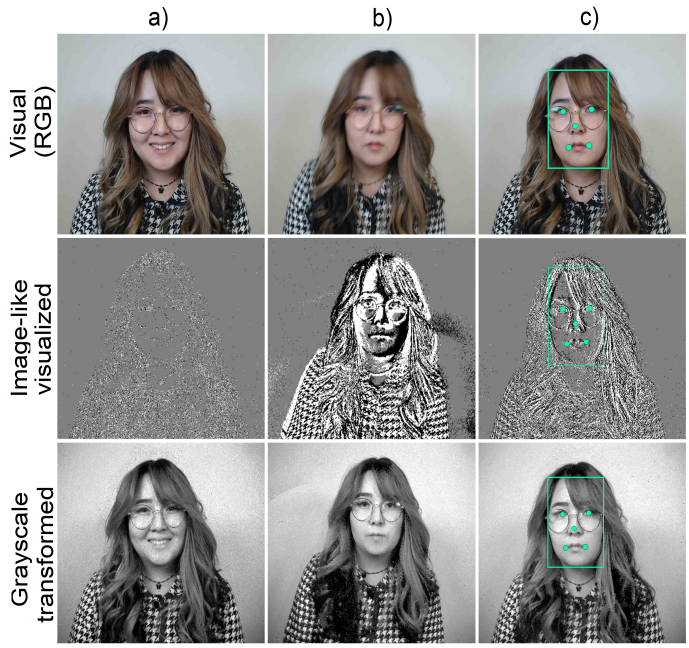

Figure 2-1: Facial RGB images and the corresponding image-like visualized and grayscale transformed event-based images: a) static pose, b) moving camera causing blur in the visual image, c) facial image with annotated bounding box and five-point facial landmarks. The event-based facial images are generated from the raw event streams using the Metavision software (https://www.prophesee.ai/).

To solve these problems, the researchers tried to turn to neuromorphic engineering, which is a field of research aimed at developing systems that can mimic the behavior of biological systems, in particular the human brain [89].

2.1 Motivations for event-based camera development

The direction of neuromorphic engineering was coined by Carver Mead around 1990. This direction studies biological features in nature for their application in technologies. Thus, inspired by the work of the human retina, work began on the creation of devices with independently working pixels [66]. Thus, in 1992, the Address Event Representation protocol was introduced for the operation of neuromorphic chips [15]. Then, in 2008, the first event video camera with a dynamic vision sensor was presented, which gave impetus to the development of a new direction of research in applied engineering - the study of event cameras for various purposes [48].

Table 2.1: Comparison between two cameras. Adapted from [4]

Characteristics	**Frame-based camera**	**Event camera**
Update rate	syncronous	aynschronous
Latensy	yes	≈ 0
Dynamic range	53 db	>120 db
Motion blur	exist	absent
Temporal resolution	low	high

Event-based cameras, also known as neuromorphic or spiking cameras, represent a new paradigm in visual sensing that departs from traditional frame-based imaging techniques [21],[83]. Unlike conventional cameras that acquire images at a fixed frame rate, event-based cameras operate by asynchronously detecting changes in brightness at each pixel location, and reporting these changes as independent events [42], [6]. This approach enables event-based cameras to operate at very high speeds (up to millions of frames per second) [21] while consuming very little power (on the order of microwatts) [50] and small motion blur, as can be seen in Fig.1 [23]. In addition, such

biosensors can read the image regardless of the intensity of the light source [70], which makes their use more powerful than using conventional cameras with a small amount of light [5]. Thus, this class of new cameras, due to its merits, has great potential for making them particularly suitable for real-time, low-latency applications such as robotics, autonomous driving, and augmented reality.

2.2 How the event camera works?

The principle of operation of event cameras differs from standard cameras and is similar to the retina of the human eye [42]. Also, it is based on capturing changes in the relative intensity of the scene at the pixel level [47], [21].

Structurally complex and important part of the visual system, which plays a huge role in the transmission of information from the eye to the brain, is the retina, which contains photoreceptor cells. This is a tiny layer of tissue that lines the back of the eye and is designed to convert light into electrical signals that are sent to the brain.

As shown in the figure 2-2, the light signal passes through the pupil and the eyeball, reaching the back wall, where the retina is located. The light is then picked up by photoreceptors, which come in two varieties: rods, which detect light in low light conditions, and cones, which are designed for precise vision in good light. Thus, photoreceptors absorb light, forming a chemical reaction that generates an electrical signal [35].

The signal then passes through the bipolar cells, which process and modify the electrical signal, after which the signal is turned on or off of the ganglion cells. "On" and "off" ganglion cells respond to light stimuli in opposite ways, so when "on" cells are activated, it results in an increase in the amount of light reaching the retina, and when "off", it results in a decrease in the amount of light reaching the retina. At the output after many ganglion cells, the optic nerve is formed, through which information enters the brain [35].

The operation of event-based camera is similar to the described retina. As an illustrative example, we can describe the principle of operation of one of the pixels in

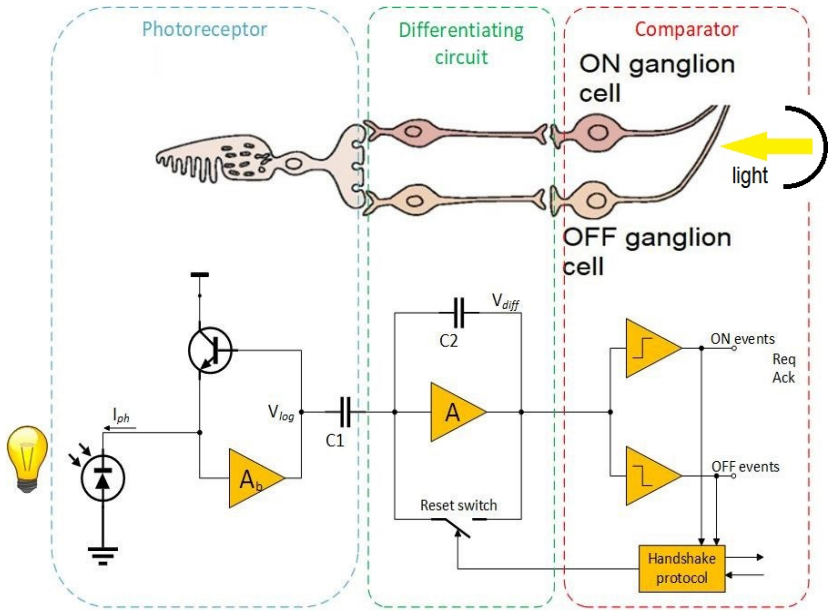

Figure 2-2: Pixel technical diagram of DAVIS event-based sensor. Adapted from [66]

the electrical circuit of a 128 x 128 pixel array of an event camera with a dynamic and active vision sensor (DAVIS) (Fig. 2-2) [66]. As can be seen from the figure, the light first hits the photoreceptor of the pixel, where the photocurrent is monitored and converted into an electrical signal. Each peak event is then processed in a bipolar cell where the current is differentiated. Next, the signal voltage values are compared by the comparators in the third step, thereby separating the ONN and OFF events [65].

The mathematical representation of the actuation of reading information by a pixel looks like a comparison of the difference between the logarithms of the intensity of the previous recorded event $log(I_{x,y,t+\Delta t})$ and the present one $log(I_{x,y,t})$ with the product of the polarity p and the contrast threshold T [86]. So, when this difference exceeds the threshold, an event is generated at time t:

$$log(I_{x,y,t+\Delta t}) - log(I_{x,y,t}) \geq pT \tag{2.1}$$

The polarity of an event is a binary value that describes whether the amount of light in the corresponding pixel increased (ON) or decreased (OFF) during the event. By comparing the light intensity of a pixel at the time of appearance with a threshold value, the polarity is established. Polarity is set to OFF if the intensity falls below

the threshold, and to ON if the intensity exceeds the threshold. Let $I(x, y, t)$ be the intensity value in the pixel with coordinates (x, y) at time t, so $T(x, y)$ is the threshold value in the pixel [5]. Then the polarity of the event p in the pixel and the time t is defined as [44],[48],[62]:

$$p(x, y, t) = \begin{cases} \text{ON if} & I(x, y, t) - T(x, y) > 0 \\ \text{OFF if} & I(x, y, t) - T(x, y) < 0 \end{cases} \tag{2.2}$$

In other words, if the difference between light intensity and threshold is positive, the polarity is ON, and if it is negative, the polarity is OFF. If the difference is zero, no event is generated.

Events from the pixel array are redirected to the camera chip, from where they are transmitted to the device via a common digital input using the event address representation protocol [62], [26] Thus, the output of an event camera is a sequence of events, where each event represents a change in brightness. Each event is represented as 4-tuples (x, y, p, t), where x and y are the coordinates of the pixel where the event occurred, $p \in (0, 1)$ [30] represents the polarity and is equal to 0 (ON events) to reduce and 1 (events OFF) to increase the change in pixel intensity, t is the time of occurrence of the event [23].

2.2.1 Benefits and challenges of using event-based sensors.

Due to the unique principle of operation of event sensors, they have a number of advantages over conventional traditional cameras:

1. High dynamic range: Event-based cameras can operate without overexposure or underexposure in a variety of lighting conditions, including strong sunlight and low light. This is because each pixel only responds to changes in brightness and not to the absolute value of brightness, which allows the camera to capture fine details in both bright and dark areas of the scene [89].

2. Low latency: As opposed to frame-based cameras, which have delay in the millisecond range, event-based cameras have very low latency, often on the

scale of microseconds [34].This is because each pixel only generates an event when there is a significant change in brightness, which can be detected almost instantaneously [85]. Because of this, event-based cameras are appropriate for applications like robots and autonomous vehicles that need quick and precise visual feedback [11].

3. Low power consumption: Event-based cameras consume much less power than traditional cameras because they only activate pixels when the brightness changes [1], [81]. This makes them good for battery powered applications such as drones or mobile devices .

4. Motion blur resistance: Event-based cameras are less susceptible to motion blur than traditional cameras because they only capture changes in brightness that occur within a short amount of time [51]. This allows them to capture sharp images of fast-moving subjects even in difficult lighting conditions.

The novelty of event-based cameras presents several disadvantages. Firstly, the technology has not been comprehensively studied, and several areas remain underdeveloped. Secondly, the event-based camera's unique approach to processing pixels renders it incompatible with conventional camera algorithms and approaches. As a result, implementing pre-existing algorithms and approaches designed for conventional cameras is not feasible [23].

2.2.2 Application in robotics field

Despite the novelty of using event sensors, due to the advantage of such cameras, they have become actively used in the field of robotics, where fast reaction time is critical.

One of the attractive applications of these cameras in robotics is navigation, in particular for visual inertial odometry. This is due to the high dynamic range and the ability to use the neuromorphic sensor in lighting complexity mode. However, robot movement can be estimated in real time with very low latency and excellent

temporal resolution using event-based cameras that capture changes in the scene. In the article [55], the authors open a new direction of using event vision for quadcopter navigation in Mars exploration missions. They also developed a visual-inertial odometry algorithm called EKLT-VIO for more accurate navigation than other algorithms and navigation using traditional cameras. Thus, with the help of this algorithm and the use of an event sensor, quadcopters will be able to pass hard-to-reach places on missions in low light conditions. Another example is the article [92], where the authors developed an evaluation pipeline that takes advantage of three sensors: IMU, traditional and event cameras. Thus, by using this pipeline to control a quadcopter, the authors have shown that it is possible to achieve precise navigation in complex scenarios in this way, which was not possible in the past.

For robot navigation, an important area of research is the development of methods for using event cameras to estimate the optical flow, which is an important task for calculating the movement of an object. Thus, some studies propose to use neural networks to estimate the optical flow in event streams. An example of this is the article [44], where event-based camera data and spike neural networks are used to estimate the optical flow. Also in the article [100], the authors experimentally proved that the use of event cameras allows the robot to perform more dynamic movements at low computational costs, which makes it possible to surpass existing systems for estimating the optical flow using traditional cameras. Another approach to estimating optical flow is to use spatial filters to filter out the noise in the event stream and a correlation filter to estimate the optical flow between two event streams accumulated over equal time intervals. For example, in the article [61], a method for estimating the position of a moving object with 6 degrees of freedom using a real-time Kalman filter based on event data was developed. The paper experimentally shows the effectiveness of the method when piloting a drone at high speed and in difficult-to-pilot scenarios.

Another rapidly developing area of using event-based camera is motion estimation systems and tactile systems. This is because conventional cameras produce a continuous stream of images that need to be analyzed [62] to determine the movements of the robot, which can be computationally and time-consuming. Robotic arms, for

example, can use event-based cameras to detect changes in the location of the object being manipulated and then change its position as needed. In the article [88], the authors used event vision in conjunction with a tactile sensor to classify liquids in jars, where the camera allows the robot to provide additional visual information about the liquid and the shape of the jar for classification. In another article [45], the authors used event vision to capture the robot. In this study, the authors declare that there is no database for training a neural network and publish their dataset, which contains event streams of 91 objects. In addition, they created a gripping system that uses event stream data about an object to calculate the angle at which the robot will grab it. Similarly, this topic was developed in [77], where the authors use an event camera attached to the hand to capture and localize objects in real time. Using this system, the authors experimentally proved the effectiveness of using event vision for capturing, localizing an object and moving an object in real time.

2.3 Face and facial landmarks detection as a field

Face detection and localization of facial landmarks are important tasks in computer vision and machine learning. There are many applications for face detection and facial landmarks in areas such as surveillance, security, human-computer interaction, and entertainment. Facial localization refers to the process of identifying and locating certain facial features such as the eyes, nose, mouth, and eyebrows [78]. Face detection is the process of detecting the presence of a human face in an image or video [64]. These tasks are needed in various fields such as security, entertainment, healthcare, and social media. However, due to differences in position, lighting, occlusion, and expression, as well as individual differences in appearance, these tasks are challenging [30].

Face detection and facial landmark localization have significant importance and application in various domains. For example, in the field of security, face detection is used for identity verification [56], surveillance and crowd analysis [3], since face detection is an important step towards the further applying of face analysis meth-

ods [13]. For example, the article [22] reveals the importance of accurate localization of facial landmarks in high-quality photographs to improve the security of biometric verification. Another example of application for emergency situations are articles [16] and [68], where the authors developed a definition of the state of drowsiness of motor vehicle drivers based on face detection and facial expressions to prevent and reduce the number of accidents. Also, the articles [27], [97] indicate the importance of identifying facial landmarks for identifying cognitive, need and emotional states of a person during the interaction of a computer and a person.

This field became especially popular during the COVID-19 period, when one of the important criteria for preventing the spread of the virus was the wearing of masks. Thus, methods for detecting face and facial landmarks were actively used in research to determine whether a person wears a mask or not [91], [49], [72]. In addition, it is used to detect various diseases in medicine by analyzing facial features [76].

Despite a long study and many different datasets for traditional cameras to detect face and facial landmarks, this topic is still relevant. In [41], a solution was proposed to facilitate the annotation of large-scale face datasets. It uses datasets from many domains (such as people, animals, and cartoons) to train one of the facial identification models already in use. The results of the experiment show that the model can generalize animal, cartoon, and artistic paintings to learn agnostic facial traits. In another paper [40] a dataset was published and ,also [39],the model was trained to classify faces and facial landmarks using a thermal imaging camera. The use of these cameras can facilitate determine face area in low light conditions. Also in articles [94],[19], the conversion of images from depth sensors into RGB images for subsequent use in face detection is studied.

Other areas of research are exploring ways to improve the accuracy and performance of face detection models. For example, the article [69] proposes a new algorithm based on the CNN neural network to improve the performance of the face detection model. In another article [98], the improvement of the accuracy of determining the landmarks of the face and the position of the face for the classification of the main emotions of a person is investigated.

2.3.1 The situation in the field of face detection in event-based research

On the other hand, despite the above research and the popularity of this field, the use of event cameras for face detection is a relatively undeveloped topic, which is explained by the novelty of the camera. Article [67] proposes a method for face detection using event cameras using Forced Kernel Correlation Filters (BKCF). One way to hide the boundaries between the regular and event cameras is image reconstruction, which is used in [80] to determine facial landmarks in images. However, such methods do not take advantage of event-based cameras and do not improve system performance.

Another way to detect faces in these events is to determine the area of the face by blinking the eyes, because blinking is a natural and always present human movement. Such examples are the articles [77],[44], where algorithms were proposed that, by determining the position of the eyes, presumably calculate the position of the bounding box of the face coordinates. However, these studies depend on the resolution of the event data and the relatively small distance from the cameras.

One of the possible reasons for the underdevelopment of this direction is the lack of a large dataset with different facial poses in different conditions, this problem was also noted by the authors in the articles [66],[77]. The Face Pose Alignment dataset [66] is made up of 108 clips with a combined duration of 10.2 minutes that were recorded under conditions of moderate and vigorous head motion. In this dataset, bounding boxes are used to mark the mouth and eyes. The dataset has some limitations, including the small number of participants (30), the brief period of the collected data,

Table 2.2: Face datasets developed using event cameras

Dataset	**Duration (min)**	**Participants**	**Environment**	**Camera**	**Resolution**	**Bounding box**	**Facial Landmark**
Lenz et al. [44]	13.5	10	controlled, wild	–	640 × 480	–	2
Face Pose Alignment [2]	10.22	30	controlled	ATIS	304 × 204	–	3
Faces in Event Streams	689	73	controlled, wild	Prophesee PPS3MVCD	480 × 360	✓	5

the poor resolution of the sensor (304×204), and the restricted variety of the face positions and camera angles. Lenz et al. [44] provided a different dataset in which they captured facial event streams for the purpose of detecting eye blinks. 50 videos total in this dataset, 25 of which have face and eye position annotations. Nevertheless, this dataset does not address the issue of having a significant amount of data for training deep neural networks because of the short duration of the recorded films (13.5 minutes) and the small number of participants. The fact that both of these datasets offer event streams as reconstructed grayscale films is another drawback. However, a dataset that contains the real events in four-column tuples and is saved in the original event-based, uncompressed format is necessary to deal with direct asynchronous event-streams.

In our task, it is possible to use other datasets designed to classify other objects, such as N-Cars [82] and the DAVIS [57] dataset, since people's faces are visible in the event output, but these datasets do not imply the presence of annotations and their use for face detection. In addition, the authors of [63], [58] propose another proposed way to compensate for the absence of a dataset, which propose the transformation of frame-based dataset images into images similar to those obtained from an event-based neuromorphic sensor. However, imaging event streams in this way does not quite correspond to the actual output of the event sensor, and the spatio-temporal component is lost, since the data was obtained from frames[46]. Therefore, as far as we know, only 2 datasets can be used for the task of detecting a face and facial landmarks, which are presented in Table 2.2.

2.4 Contribution

This dissertation focuses on the development of a face and facial landmark detection system that uses event-based output directly (see Fig.). As follows from the literature review, research on event chambers is becoming an increasingly popular topic because their advantages over conventional cameras make them attractive for various applications. However, a review of the literature concluded that the area of face detection

and face landmarks is still not explored, and models and datasets for face detection and face landmarks are not exists. Therefore, this dissertation will study the application of this camera in the field of face detection and facial landmarks, which is important in the direction of communication between a computer and a person. So, the contribution includes:

- In this research was created and published the first rich and structured dataset of 689 minutes of machine learning-transformed event streams. Meanwhile, the dataset includes annotations with facial landmarks and face boundinb box coordinates that were annotated with an accumulation time of 33ms, which is about 1.6 million annotated faces.
- Also in this study, for the first time, 12 research-based DFES models were created and trained for face and landmark detection that use outputs based directly on events.
- Experiments and comparative analysis of DFES models with a combination of various feature extractors were carried out. An experiment was also conducted on calculation of inference time and the use of a model for real-time detection.

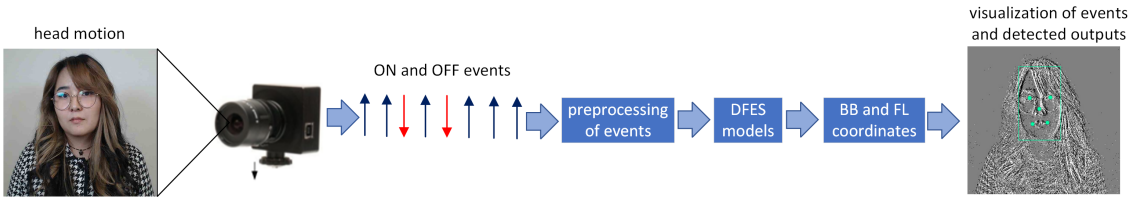

Figure 2-3: A visual scheme for demonstrating the developed system for detection faces and facial landmarks on event-based output. Camera photo retrieved from https://www.prophesee.ai/

As a result, with widely available models and a large data set, this research could stimulate the development and subsequent research into the application of event cameras towards face detection, or could be used as a basis for more advanced steps such as face recognition, emotion classification, and others.

Chapter 3

The process of creating a dataset

In order to solve the problem of the lack of an extensive and large dataset for face detection with bounding box and facial landmarks coordinates, work was carried out in this study to publish the dataset to stimulate the development of the use of the camera in this direction and to use it in a further part of the training neural network.

3.1 Data collection algorithm

The "Faces in Events" dataset created in this part is the first database for an event camera with a rich set of participants, including different nationalities and the presence of facial accessories, as well as a number of experiments, examples of which can be seen in Fig. 3-1. There were 73 participants in the data collection, of which 31 were women and 42 were men. In addition, the average age of the participants was 25.3 years. The Institutional Research Ethics Committee of Nazarbayev University approved the data collection project. Also, each participant was informed and agreed to the terms of the experiment and the use of data for research and publication.

Dataset event-streams were recorded using a Prophesee PPS3MVCD event camera with 480×360 resolution, $> 120\ dB$ dynamic range with pixel size equals to $15 \times 15\ \mu m^2$, $1\ \mu s$ temporal resolution, and a 70 field of view angle.

Event streams were collected using the Metavision program (https://www.prophesee.ai/)

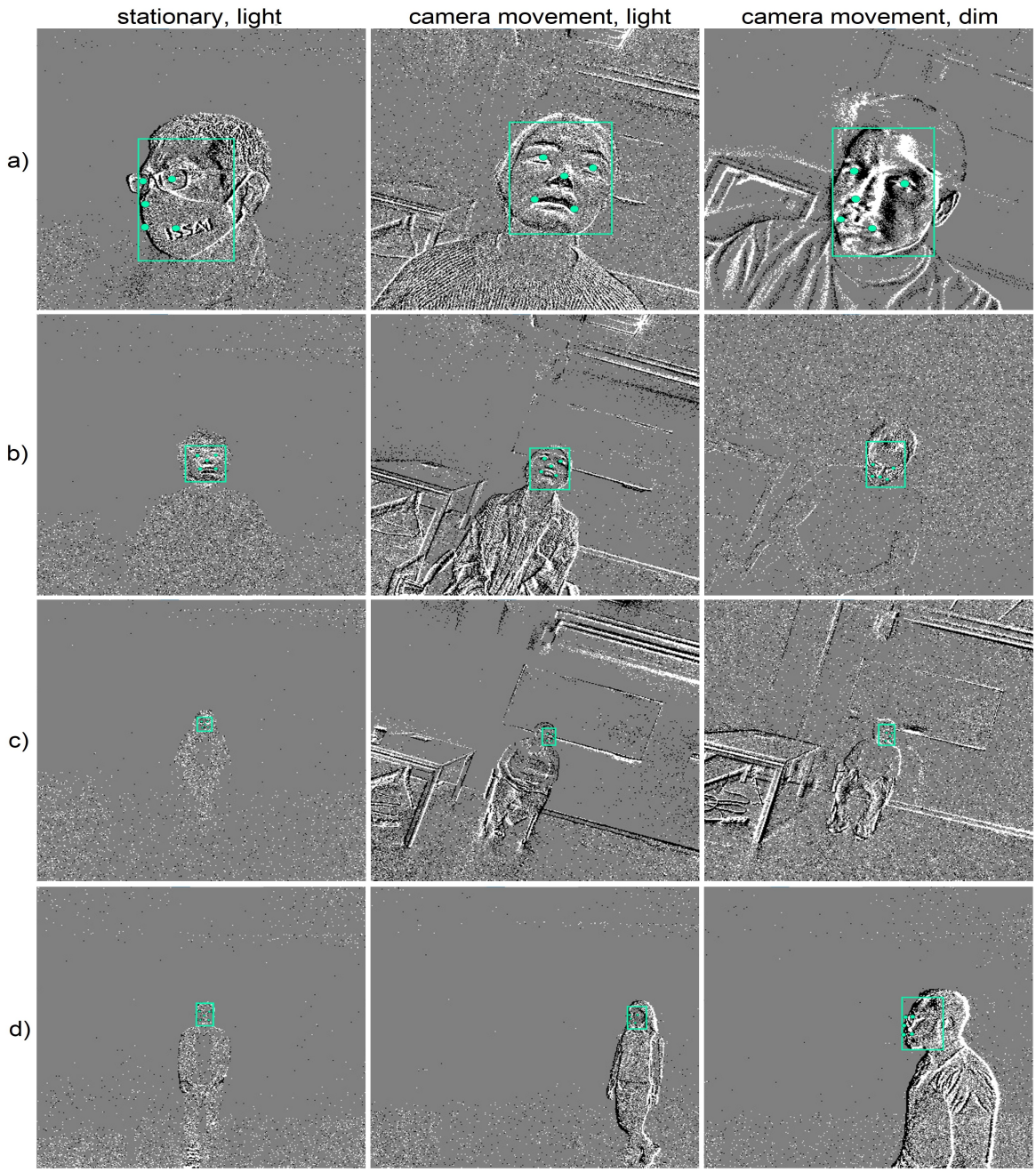

Figure 3-1: Screenshots of the event streams of the FES dataset laboratory part with bounding box and facial landmarks annotations, which include participants of two genders, different ages,faces from different angles at different distances and the presence of accessories such as glasses, masks, etc. a) a face at a distance of 50 cm from the camera; b) a face at a distance of 150 cm from the camera; c) a face at a distance of 400 cm from the camera; c) event flows 56, 58, 59 of the experiment, details can be seen in Table 3.1

installed on a desktop workstation (Intel Core i5-8500, 24 GB DDR4 memory, Ubuntu 18.04 Linux operating system). The recorded event stream data was saved to the hard disk as ".raw" files.

The data collection process was divided into two parts. The first part was recorded in a university office, where participants performed movements according to a pre-prepared scenario under the guidance of a researcher. This part of the experiment included 60 identical motion scenarios for each participant. For a more detailed explanation:

- On 0-47 recorded event streams, participants perform such head movements as up-down, right-left, in a circle and motionless with pronunciation of counting numbers, performing movements at different distances from the event camera

Table 3.1: Experiment protocol for controlled/laboratory part of data collection

Experiment number	**Camera movement movement**	**Participant's movements**	**Light condition**	**Distance from the event camera**
0-3	stationary	head movement: left-right, up-down, circular and counting	bright	50 cm
4-7	moving		bright	50 cm
8-11	stationary		dim	50 cm
12-15	moving		dim	50 cm
16-19	stationary		bright	150 cm
20-23	moving		bright	150 cm
24-27	stationary		dim	150 cm
28-31	moving		dim	150 cm
32-35	stationary		bright	400 cm
36-39	moving		bright	400 cm
40-43	stationary		dim	400 cm
44-47	moving		dim	400 cm
48-51	stationary	the participant in the mask moves head left-right, up-down, circular and counting	bright	50 cm
52-55	moving		bright	50 cm
56	stationary	far-near walking	bright	different
57	stationary	zigzag walking, starting on the right side	bright	different
58	stationary	zigzag walking, starting on the left side	bright	different
59	stationary	sideways walking	bright	different

and under different lighting conditions;

- Experiments 48-55 include movements with a mask on the face, the same as in the previous experiments at a distance of 50 cm from the camera;
- The experiments for event-streams with number 56-59 involve recording streams of events as the participant moves in a zigzag, diagonal, straight, and right-left direction.

The complete experimental protocol, which describes data collection in controlled environment can be seen in Table 3.1.

The second part of the dataset includes the uncontrolled or wild part of the dataset, which includes event streams recorded indoors in spaces such as university halls, lobbies, cafes, and classrooms. This part of the data set included experiments in which several participants moved freely in front of the camera without instructions (see Fig. 3-4).

3.2 The process of annotating a dataset

3.2.1 Visulisation of recorded event-streams

Event cameras produce different output than frame-based cameras because the events produced by the camera are spatially sparse and asynchronous. Thus, the event itself cannot provide complete information for a person to understand that a face has been detected. Therefore, for annotations and more readable visualization, the stream of events accumulated over a certain short period of time, called the accumulation time. Thus, during each accumulation time, the collected event data was visualized using different colors for the ON and OFF events and the background color (see Figure 3-2).

3.2.2 Description of dataset annotations

Also, for visualization, the optimal value of the accumulation time was chosen between a small amount of information about events, but, as a result, inaccurate boundaries

of the faces, and a long period of accumulation time, but the appearance of blurring and fuzziness of the data delay. As can be seen in Fig. 3-2, with an accumulation period of 200 μ s, not enough events appear during this period of time, so a person cannot see enough information to understand the full picture. Despite the accuracy of rendering faces in the foreground with an accumulation period of 2 ms, this is still not enough to distinguish faces in the distance. At 33 and 100ms, the full image is visible, but at 100ms there is a motion blur effect, which can be noise for further face detection task using a neural network. Thus, an accumulation time of 33 ms was used for the annotation.

Four annotators worked on the annotation task for nine months under the supervision of researchers using the free CVAT toolkit (https://cvat.ai). Annotations are tied to the timeline, that is, to each accumulation time period. As a result, over 1.6 million accumulated event streams were manually annotated. Also, for some videos where faces are not clearly visible event streams for placing annotations, the event-streams data has been converted to grayscale video for more accurate annotations, as

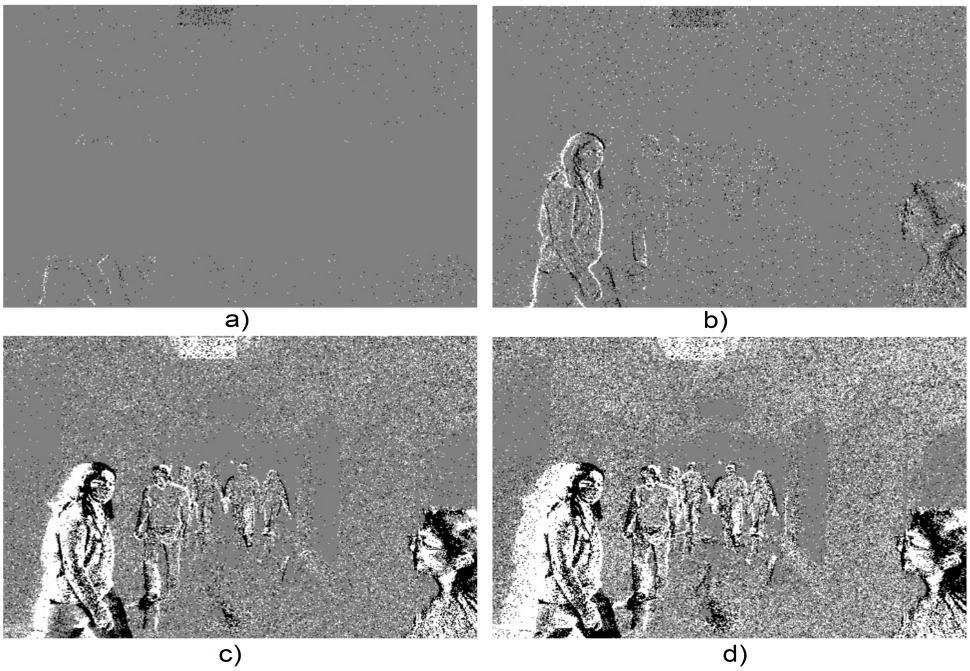

Figure 3-2: Data visualization of event streams at different accumulation times: a) 200 μs, b) 5 ms, c) 33 ms, and d) 100 ms.

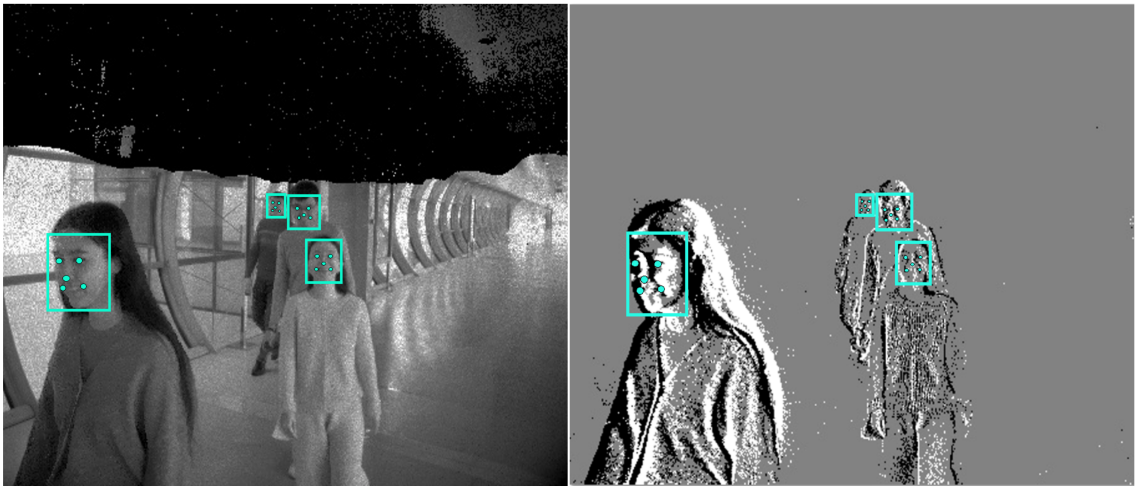

Figure 3-3: Screenshots of the event streams of the FES dataset wild part with bounding box and facial landmarks annotations: a) visualization of converted event streams into a grayscale version; b) visualization of recorded event streams.

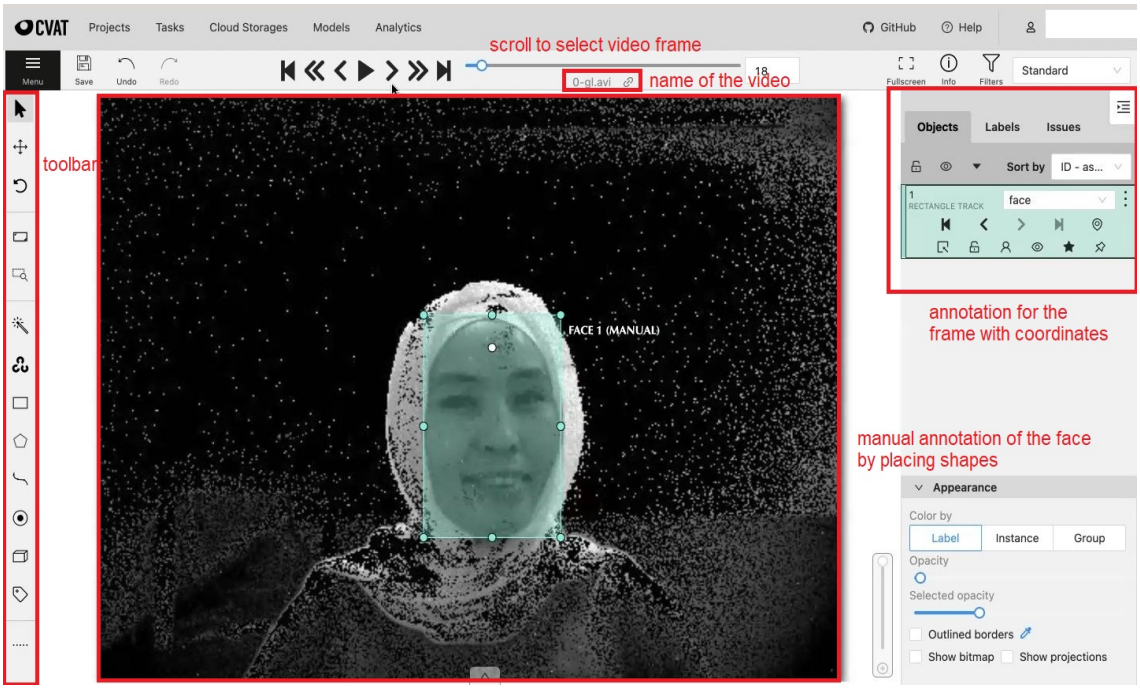

Figure 3-4: Screenshots of the free CVAT toolkit (https://cvat.ai). The moderators on the left side selected the desired shapes and annotated each frame in grayscale video, then the program saved the coordinates of the bounding box and facial points in xml format at the output.

it was shown in Fig. 3-4.

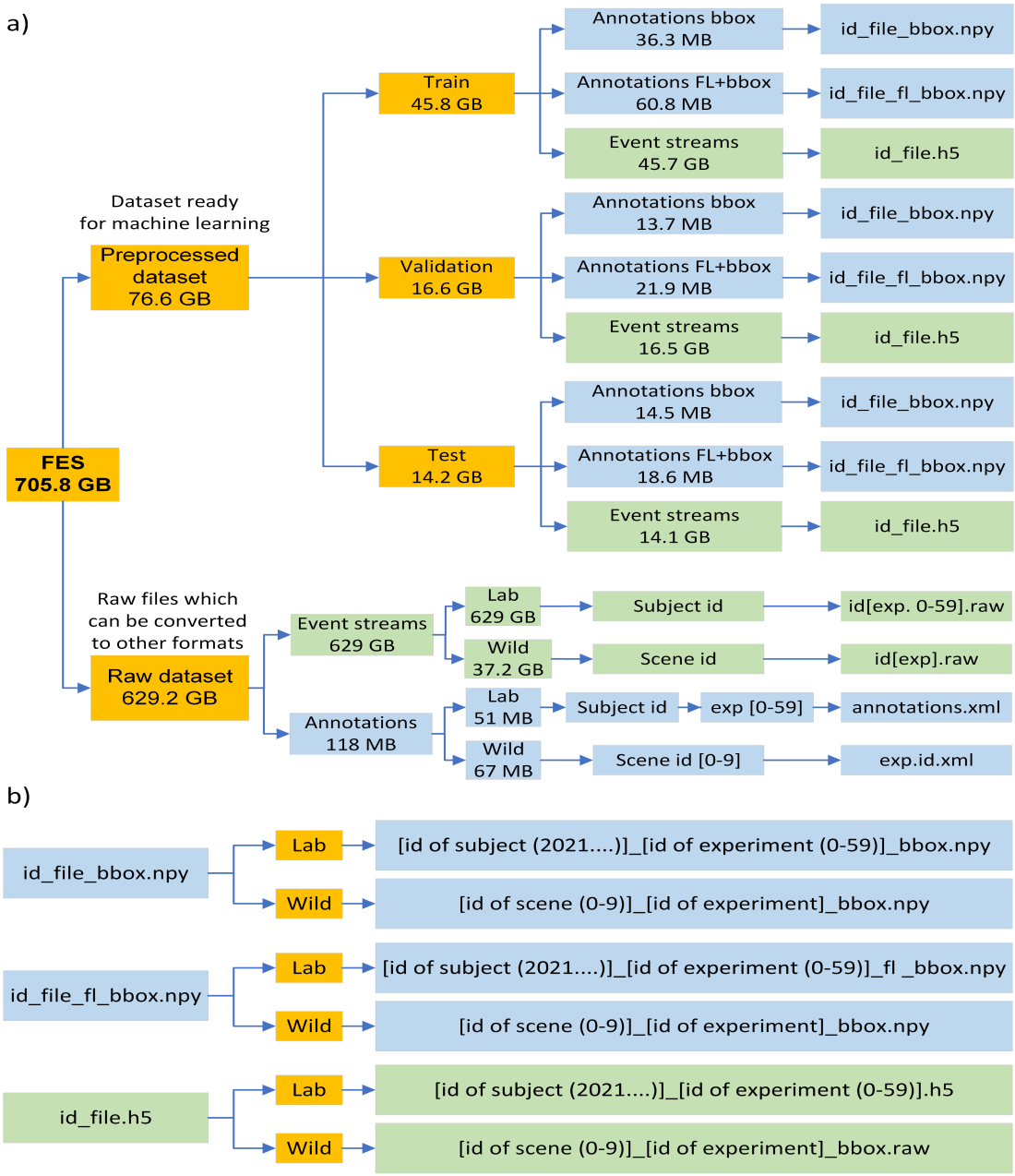

Figure 3-5: File structure of the FES dataset, with orange representing folders, green representing an event stream and blue representing annotations: a) The preprocessed data are divided into three folders, with each folder containing only bounding box annotations, both bounding box and facial landmark annotations, and event streams in the h5 format. The raw dataset contains lab and wild folders with raw videos and annotations. b) Each controlled experiment (Lab) file has an individual subject ID and an experiment ID. Each file in the uncontrolled (Wild) dataset contains a scene ID that provides information about the location of a recording and the number (ID) of an experiment.

3.2.3 Gray-scale transformation

Reconstruction of event streams into gray-scale images occurs using the method proposed in the article, where the authors developed a system for event-streams reconstruction [71]. The goal of the system is to mapp the grayscale image to the streams of events. To do this, the authors use a recurrent neural network consisting of the UNet module similar to presented in the article [75]. Thus, the input events are accumulated over a certain period of time, after which the sequence of event-streams of event-streams is fed to the input of the neural network. At the output of the neural network, each event is mapped to the corresponding pixel value with a gray level value. The resulting grayscale image is formed by summing the pixel values over time, as it was shown in Fig. 3-4.

3.2.4 Description of dataset annotations

The final version of the dataset includes the original raw extension videos obtained immediately after the participants were filmed and the converted Python “h5” binary format to interact with the event stream data as an array for use in training the selected neural network. Annotations with bounding box and facial landmark coordinates for "raw" extension event streams are stored in the popular Pascal Voc "xml" machine learning format and were later converted to the "npy" format for further research.

3.2.5 Dataset splitting into sets

For further study, the transformed dataset was divided into three parts for training, validation, and testing. The laboratory part was divided in such a way that the participant IDs were divided into three sets and the same IDs did not occur in each set to eliminate the bias for the neural network, and the wild part of the data set was divided in the same way.Specifically, the training, validation, and testing sets contain 60%, 22%, and 18% of the dataset, respectively.

To evaluate the accuracy of determining the boxes bounding faces, faces were di-

Table 3.2: Statistics for the FES dataset

Category	**Train**		**Valid**		**Test**		**Total**		
	Lab	**Wild**	**Lab**	**Wild**	**Lab**	**Wild**	**Lab**	**Wild**	**Both**
# of subjects	40	15	12	14	12	11	64	18	73
# of images (thousands)	720.0	57.3	216.0	19.1	216.0	11.0	1,152.0	87.4	1,239.4
# of labeled faces (thousands)	715.4	357.9	210.5	82.4	215.0	37.7	1,140.9	478.0	1,618.9
# of recordings	2,400	32	720	10	720	7	3,840	49	3,889
Duration (minutes)	400	28	120	10	120	11	640	49	689
Mean duration per record (seconds)	10	60	10	63	10	52	10	59	10.62
Mean # of events per record (millions)	27.5	180.4	28.7	184.6	29.2	82.7	28.5	149.2	37.3
Mean # of ON events per record (millions)	16.5	99.2	17.8	119.9	16.9	40.52	17.1	86.54	22.0
Mean # of OFF events per record (millions)	11.0	81.2	10.9	64.6	12.26	42.17	11.4	62.66	15.0

Note. The sum of the number of subjects in the lab and wild conditions does not equal the total number of subjects, as some subjects were recorded for both sets of experiments.

Table 3.3: Face bounding box size statistics for the FES dataset

	$<$ 35 pixels	**35–90 pixels**	**$\geq$ 90 pixels**
Wild	0.2%	7.9%	91.9%
Laboratory	41.1%	29.9%	29.0%
Overall	39.5%	29.1%	31.4%

vided into three groups depending on the distance of the face from the camera—large, medium, and small, as shown in Table 3.3. Since the videos were recorded based on an experimental protocol, large faces were those whose bounding box height was greater or equal to 90px; medium faces were those whose height was between 35px and and 90px, and small faces were of a height below 35 px.

Chapter 4

Methodology

4.1 Deep Faces In Event-Streams (DFES) Architecture

This part will describe the architecture of training models called Deep Faces in Event Streams (DFES) for determining the coordinates of facial landmarks and bounding boxes.

The models generated in this study are based on the paper [17], where the authors used this architecture to detect pedestrians, cars, and other vehicles for use in autonomous vehicles. For this study, this architecture was used, since the architecture algorithm of the author's model is based on the use of direct camera event output and the use of information about previous event states. Thus, because of using this architecture, there is no need for additional stages of video reconstruction of the event-streams and the load on computing power, which allows to fully use the capabilities of event cameras.

4.1.1 Problem definition and notation

To formulate the research problem, we denote the event ej as an event in the relative plane with pixel coordinates $x_n \in [0, K-1]$, $y_n \in [0, L-1]$ and polarity $p_n \in [0, 1]$. The set e_n accumulates over a period of time $t_n \in [0, \infty]$ as successive events, which

represent a stream of events $\mathbf{E}$ and are formulated as $\mathbf{E} = \{e_n = (x_n, y_n, p_n, tn)\}$.

At each time interval t, annotations of the face region Fbb appear with a bounding box of coordinates in the form $F_{bb} = \{f_{bb} = (t, x, y, w, h)\}$, where x and y are the coordinates of the top left pixel, and w and h are the width and the height of the rectangle.Similarly, annotations of the location of facial landmarks F_{fl}, which are the eyes, nose, left and right corners of the mouth in time t with coordinates x and y respectively, are formulated as $F_{fl} = \{f_{fl} = (t, xe_1, ye_1, xe_2, ye_2, xn, yn, xm_1, ym_1, xm_2, ym_2)\}$.

Formally, the face detection problem can be expressed as a mapping from $\mathbf{E}$ to F_{bb} with $F_{bb} = D_{bb}(E_{tn<t})$, where the D_bb detector must predict the face's bounding boxes using past event data. Similarly, $(F_{bb}, F_{fl}) = D_{bb+fl}(E_{tn<t})$ can be used to determine the combined detection of bounding boxes and facial landmarks coordinates that appear at time tn.

4.1.2 Model architecture explanation

Therefore, according to the problem statement, in this study, the D_{bb} and D_{bb+fl} detectors were implemented using deep learning. Since the use of the detectors D_{bb} and D_{bb+fl} to analyze each event will be a load on computing power due to the rapid updating of information, the information about incoming events accumulated over a period of time $[t_k, t_{k+1}]$ is collected in an array with the accumulation time $\delta t = t_{k+1} - t_k$. Consequently, using the histogram preprocessing function, we correlate the x, y, and p spatio-temporal coordinates of the events in the respective 2d cells by mapping coordinates to pixel coordinate of cell. Thus, we transform the array of events over the accumulation time period into a tensor map, which has a dimension of $2 \times K \times L$, where K and L are resolution of the map. Then, the tensor map Hk is used to extract features fk by using the feature extractor.

Squeeze-excite layers were used in [17] to extract spatial information from features. The squeeze module and the excitation module are the two sub-modules that make up the Squeeze-excite layer. To decrease the dimensionality of each channel, the squeeze module executes a global average pooling operation along the feature map's spatial dimensions. The squeeze module's output is then subjected to the excitation module's

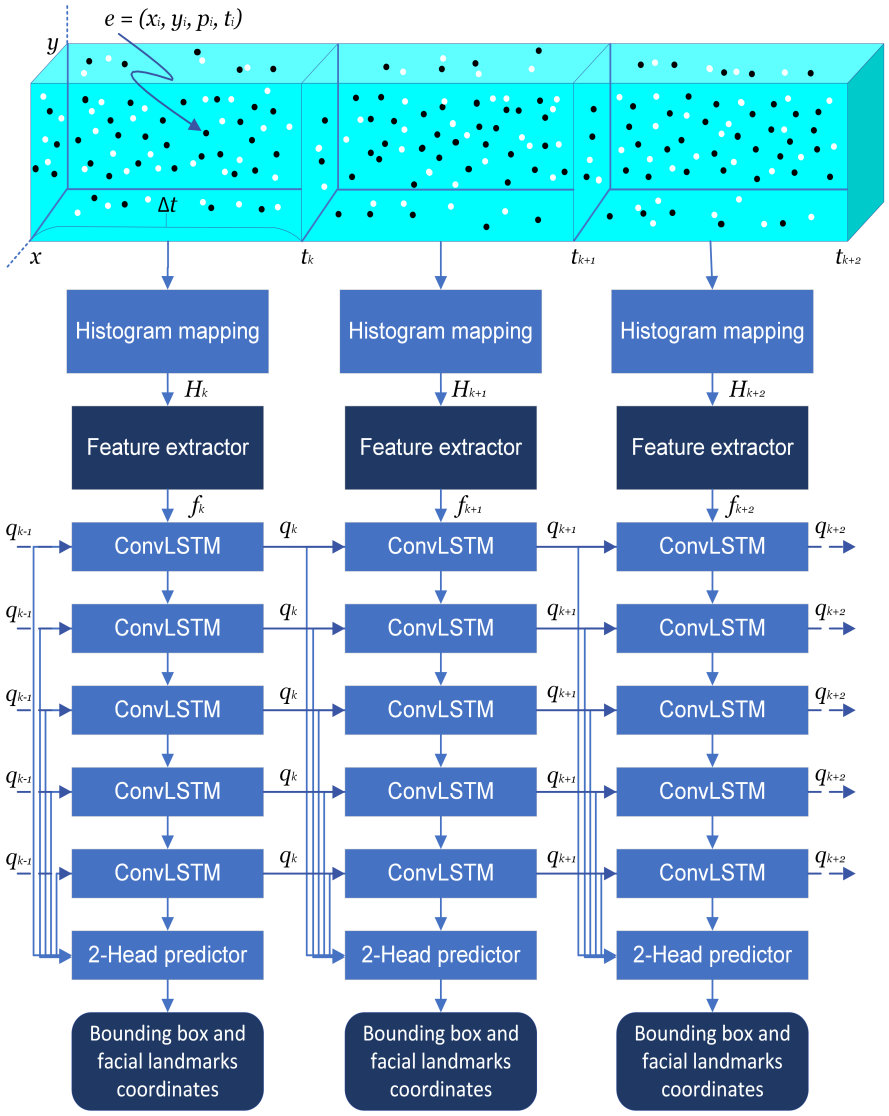

Figure 4-1: Model architecture of DFES for face detection and facial landmark extraction (adapted from [17]), where $q_0 = \mathbf{0}$.

application of a set of channel-wise weights, which highlights the key features [33].

However, due to the popularity of using ResNet networks to extract features from visual information, work has been done in this study to replace this feature extractor with various ResNet architecture options for further comparison. This was done with the motivation that the ResNet variants can improve accuracy and also have a deep network architecture [28]. The use of a residual block, which uses a pass-through connection to link two convolutional layers, is the main advantage of the ResNet network. So, in our models, we specifically used the ResNet18, ResNet-34,

Figure 4-2: Residual block implementation. Adapted from [28]

and ResNet-50 variants. Table 4 provides a summary of the architecture and settings of the feature extractors used for the DFES model.

The detectors D_{bb} and D_{bb+fl} depend on encoded information from the past as an internal state in addition to information currently accumulated, is used to exploit previous events. As a result, the internal state vector q_k was generated using a recurrent neural network design, which is a sort of network that manages sequential data. RNNs have feedback connections, in contrast to conventional feedforward neural networks, which enable information to last over time. A recurrent layer, which is the base of an RNN, works by processing sequential data one element at a time while keeping a "memory" of prior inputs. The network's hidden state is this memory, which is changed at each time step based on the input being used and the hidden state that came before it. The output at the current time step can then be predicted using the hidden state, or a series of outputs can be produced [59].

RNNs are susceptible to the vanishing gradient problem, in which the gradients used for training get smaller over time as they spread throughout the network. To decide this problem, it was introduced the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. So, to limit information flow inside the network and to selectively update or forget some elements of the hidden state, these models employ additional gating techniques.

As a recurrent neural network in our work it was decided to use LSTM with combination of Convolutional layer. The group of gates that control the flow of information to and from the memory cell in the LSTM governs the memory cell. The information flow of the network is adaptively controlled by these gates, which include an input gate, a forget gate, and an output gate [31]. So with combination of Convolutional

Table 4.1: DFES network's feature extractors variants

Original	**ResNet-50**	**ResNet-34**	**ResNet-34**
ConvLayer, 32	ConvLayer, 16	ConvLayer, 16	ConvLayer, 16
BatchNorm	MaxPool	MaxPool	MaxPool
ReLU	BatchNorm×2	[Resblock, 16]×3	[Resblock, 16]
Squeeze-excite block	ReLU	[Resblock, 32]×4	[Resblock, 32]×2
Squeeze-excite block	[Bottleneck, 16]×3	[Resblock, 64]×6	[Resblock, 64]×2
–	[Bottleneck 32]×4	[Resblock, 128]×3	[Resblock, 128]×2
–	[Bottleneck 64]×6	–	−−
–	[Bottleneck 128]×3	–	–

layer Conv-LSTM [84] can process the spatial and temporal features. Therefore, it was used 5 connected layers of convolutional long-term short-term memory (ConvLSTM), whose properties are to store past information q_{k-1} in addition to the extracted features f_k.

In the last layers, convolutional layers are used in a two-head regression predictor as described in [17] to predict face detection bounding boxes and five-point face landmarks based on the output of each ConvLSTM layer. The SSD algorithm uses anchor boxes to apply a default set of bounding boxes to each location in a feature map generated by a convolutional neural network. Then it predicts the class label of the object enclosed in each anchor block, as well as the offsets needed to resize the block to better fit the object. The set of convolutional layers that are added to the CNN is used to make these predictions [52].

4.1.3 Cost function

In this study, a smooth loss l_1, referred to in this study as the cost function $\mathcal{L}_{reg}$, was used to find the coordinates of the bounding boxes and landmarks of the face, which is a regression problem. To solve the classification problem between the background and face classes, the softmax focal loss function [90], denoted as $\mathcal{L}_{cls}$, was used.

So, to calculate the final cost function, we used the summation of the output of two loss functions to train our neural network:

$$\mathcal{L}_t = \mathcal{L}_{reg} + \mathcal{L}_{cls} \tag{4.1}$$

Let F^*_{bb} and F^*_{fl} denote the ground true values, and F'_{bb} and F'_{fl} denote both the predicted values for the face bounding boxes and the five-point face landmarks, respectively, to characterize the smooth $l1$ loss in our model. We can define a smooth loss function $l1$ as in formula 4.2 [25] if we combine the values of the bounding box and facial landmarks for ground truth and predicted values as $F_{gt} = \{F^*_{bb},\ F^*_{fl}\}$ and $F_{pr} = \{F'_{bb},\ F'_{fl}\}$.

$$\begin{aligned} \mathcal{L}_{reg}(\mathbf{F}_{gt}, \mathbf{F}_{pr}) &= \frac{1}{N}\sum_i \mathcal{L}_{reg}(F^i_{gt},\ F^i_{pr}) \\ \mathcal{L}_{reg}(F^i_{gt}, F^i_{pr}) &= \begin{cases} 0.5(F_{pr} - F_{gt})^2/\beta & \text{if } (F_{pr} - F_{gt}) < \beta \\ |F_{pr} - F_{gt}| - 0.5 * \beta & \text{otherwise} \end{cases} \end{aligned} \tag{4.2}$$

where β is a tunable parameter.

4.2 Methodology of the experiments

To begin with, as described earlier, the sets are divided into three parts: training, validation and test sets. The training sets were used to train the neural network and the validation set was used to evaluate intermediate results between epochs and select the best model. Meanwhile, a test set was used to evaluate the performance and accuracy of the model.

The code for determining the architecture of the model was written using the PyTorch tool, which allowed the use of a multiprocessor parallel model training mode. So, we used an Nvidia DGX-2 server with V100 GPUs for model training (GPUs). Each model goes through 40 epochs training of Adam optimizer with a learning rate equals to 0.0001. Each model required about two days of training. After each epoch,

the training results of the model and its performance on the validation set were analyzed. The generated models was saved as a checkpoint at the end of each epoch. In further experiments, checkpoints of models were used with the filter on best results for face area detection on unseen event-streams data.

The first experiment was conducted to select the best accumulation time, at which the model could detect the face with the best results. For this, 6 different models were trained, of which three $DFES_{BB}$ models were trained to determine only face bounding box coordinates, including models that were trained with accumulation times of 33 ms, 50 ms, and 100 ms. In the same way, 3 more $DFES_{FL+BB}$ models were trained, but to determine not only the coordinates of the bounding box of the face, but also the facial landmarks.

After selecting the accumulation time to work with, 6 more models were trained with various combinations of model architecture and feature extraction parameters, including: ResNet18, ResNet-34 and ResNet-50. Half of them were trained for cases where only the coordinates of the bounding box were determined, and the other half were trained for the FL+BB case. Subsequent experiments tested these models on test event streams and compared the results, which will be described in the next chapter. As a result, a comparison will be made of the performance and influence of different models with different feature extractors on the results of determining the coordinates of the BB and FL+BB.

To compare different models, the Mean Average Accuracy Threshold 50% (mAP_{50}) metric was used to evaluate the accuracy of coordinate prediction only for the bounding boxes of the faces for $DFES_{BB}$ and $DFES_{of the FL+BB}$ models. Also, for the $DFES_{FL+BB}$ models, to assess the accuracy of predicting the location of facial landmarks, the NME metric was used, which was calculated by the formula [29]:

$$NME = \frac{1}{NZ} \sum_{Z=1}^{Z} \frac{||\hat{x}_Z - \hat{x}_Z||}{D_Z} \tag{4.3}$$

where $||x_Z - \hat{x}_Z||$ is the Euclidean distance between predicted x_Z and ground-truth $\hat{x}_Z$, N is a constant equal to five, which is the number of facial landmarks, Z is the

number of predictions, D_Z is the distance between the centers of the eyes for the given sample. In addition, to compare models, an estimate will be made of the time it took the model to analyze the invisible part of the data and produce a prediction, known as inference time.

The last experiment was to evaluate the performance of the model in detecting the face and face landmarks in real time. For this experiment, a new version of the event-based camera EVK-4-HD was used, which was connected to a computer with the following specifications: Intel Xeon L3403, 64 GB DDR3 memory, and an Nvidia GeForce RTX 2080 Ti GPU running Windows 10 system. This part of the experiment was recorded on a RGB camera for later demonstration and as evidence that the models can run in real time without delays since they use event output directly.

Chapter 5

Results and Discussion

5.1 Determination of the Optimal Accumulation Time

In this experiment, we trained models on a FES dataset with different accumulation times, that is, the time period during which events are accumulated before being processed by the algorithm. The optimal accumulation time for the event camera should be chosen carefully, as it directly affects the efficiency and accuracy of the model. The lower accuracy may be the result of the algorithm skipping certain events if the accumulation time is too short, and on the other hand, the model may be slow and less efficient with long accumulation times when processing real-time events. In Table 5.1 and Table 5.2, we can see the results of bounding box detection for $DFES_{BB}$ and $DFES_{BB+FL}$ models using the original feature extractor for 33 ms, 50 ms, and 100 ms on the test dataset. Higher average accuracy at 50% intersection compared to pooling (mAP_{50}) was achieved using a model trained on event streams with an accumulation time of 50 ms. Thus, for this model, the total $mAP50$ for the bounding box detection model was 0.93, and for the combined face landmark and bounding box detection model it was 0.918.

Meanwhile, Table 5.1 and Table 5.2 shows that the model with an accumulation time of 33 ms gave worse results, which may be due to the fact that a small accumulation time could not provide a sufficient number of events for accurate face detection. Conversely, a longer accumulation duration of 100 ms could collect an excess number

Table 5.1: Results for face bounding box detection on FES laboratory and wild testing set

Model	Feature extractor	Δt	mAP_{50} Lab				mAP_{50}Wild			
			Large	Medium	Small	Overall	Large	Medium	Small	Overall
$DFES_{BB}$	Original	33 ms	0.375	0.4	0.328	0.353	0.57	0.13	0.06	0.1
$DFES_{BB}$	Original	50 ms	**0.99**	**0.978**	**0.97**	**0.978**	**0.919**	0.273	0.138	0.146
$DFES_{BB}$	Original	100 ms	0.989	0.973	0.964	0.977	0.8	0.231	0.133	0.15
$DFES_{BB}$	ResNet-18	50 ms	**0.99**	0.974	**0.97**	**0.978**	0.83	0.3	**0.149**	0.165
$DFES_{BB}$	ResNet-34	50 ms	0.989	0.962	0.952	0.965	0.794	**0.436**	0.17	**0.182**
$DFES_{BB}$	ResNet-50	50 ms	0.988	0.964	0.9	0.957	0.73	0.12	0.05	0.1
$DFES_{FL+BB}$	Original	33 ms	0.371	0.397	0.38	0.37	0.599	0.443	0.26	0.252
$DFES_{FL+BB}$	Original	50 ms	0.989	**0.978**	**0.871**	**0.973**	0.728	**0.782**	0.482	0.528
$DFES_{FL+BB}$	Original	100 ms	0.989	0.976	0.7	0.937	0.64	0.7	**0.645**	**0.653**
$DFES_{FL+BB}$	ResNet-18	50 ms	**0.99**	0.969	0.8	0.96	0.72	0.75	0.47	0.5
$DFES_{FL+BB}$	ResNet-34	50 ms	**0.99**	**0.978**	0.869	0.966	**0.789**	0.75	0.498	0.54
$DFES_{FL+BB}$	ResNet-50	50 ms	0.985	0.928	0.75	0.925	0.184	0.282	0.124	0.138

of events in the face area, which would lead to a blur effect. To address these issues, an accumulation time of 50 ms was chosen for later experimentation with event streams, as it strikes a balance between capturing enough events for detection and avoiding blur effects caused by over-accumulation.

5.2 FL and BB Detection Results

As described earlier in this experiment, with already existing knowledge of the optimal accumulation time, the neural network was trained with various combinations of feature extractors, in particular ResNet and original neural networks with one combination of short detectors. To provide a more detailed analysis of the performance of the model, the test data was divided into various subsets, including lab, wild, and test sets. In addition, the detection results were evaluated for faces of different sizes, in particular for large, medium, small and general (all) faces. This approach allows a more detailed assessment of the model's ability to generalize faces of different sizes and environmental conditions.

Table 5.1, Table 5.2 and Table 5.4, Table 5.4 contain the results of evaluating the

Table 5.2: Results for face bounding box detection on FES overall testing set

Model	**Feature extractor**	**Δt**	**mAP$_{50}$Overall Test Set**			
			Large	**Medium**	**Small**	**Overall**
DFES$_{BB}$	Original	33 ms	0.375	0.4	0.328	0.353
DFES$_{BB}$	Original	50 ms	**0.99**	**0.976**	0.8	0.93
DFES$_{BB}$	Original	100 ms	0.989	0.964	0.8	0.927
DFES$_{BB}$	ResNet-18	50 ms	**0.99**	0.97	**0.827**	**0.936**
DFES$_{BB}$	ResNet-34	50 ms	**0.99**	0.969	0.8	0.931
DFES$_{BB}$	ResNet-50	50 ms	0.988	0.96	0.715	0.884
DFES$_{FL+BB}$	Original	33 ms	0.369	0.393	0.325	0.347
DFES$_{FL+BB}$	Original	50 ms	0.989	**0.97**	0.7	**0.918**
DFES$_{FL+BB}$	Original	100 ms	0.989	0.949	0.575	0.868
DFES$_{FL+BB}$	ResNet-18	50 ms	**0.99**	0.96	0.7	0.9
DFES$_{FL+BB}$	ResNet-34	50 ms	**0.99**	**0.97**	**0.72**	0.912
DFES$_{FL+BB}$	ResNet-50	50 ms	0.984	0.873	0.52	0.8

Table 5.3: NME results for bounding box and facial landmarks detection models on FES laboratory and wild testing set

Model	**Feature extractor**	**Δt**	**NME Lab Test Set**				**NME Wild Test Set**			
			Large	**Medium**	**Small**	**Overall**	**Large**	**Medium**	**Small**	**Overall**
DFES$_{FL+BB}$	Original	33 ms	**0.335**	**0.298**	**0.577**	**0.394**	16.69	15	15.87	15.74
DFES$_{FL+BB}$	Original	50 ms	0.398	0.342	0.6	0.44	16.09	**12.01**	**13.8**	**13.5**
DFES$_{FL+BB}$	Original	100 ms	0.57	0.45	0.83	0.61	**9.965**	14.23	14.72	14.73
DFES$_{FL+BB}$	ResNet-18	50 ms	0.414	0.373	1.276	0.656	16.8	12.7	15.9	15.3
DFES$_{FL+BB}$	ResNet-34	50 ms	0.383	0.325	0.6	0.42	17.9	12.5	14	13.7
DFES$_{FL+BB}$	ResNet-50	50 ms	0.84	1.98	3.03	1.8	16.54	14.23	15.46	15.32

performance of the model in terms of Mean Average Accuracy ($mAP50$) for bounding box detection for laboratory and wild sets, overall set, and Normalized Mean Error (NME) for face landmark detection, respectively. So, mAP_{50} scores were found to be generally higher for large faces, indicating difficulty in detecting faces further from the camera. In addition, the wild part of the test set scored lower compared to the laboratory test set, which can be explained by the presence of multiple faces with uncontrolled postures in the event streams.

The model with the ResNet-18 feature extractor and 50 ms accumulation time showed the best results for face bounding box detection as shown in Table 5.1 and

Table 5.4: NME results for bounding box and facial landmarks detection models on FES overall testing set

Model	**Feature extractor**	**Δt**	**NME Overall Test Set**			
			Large	**Medium**	**Small**	**Overall**
$DFES_{FL+BB}$	Original	33 ms	**0.358**	1.5	5.387	2.52
$DFES_{FL+BB}$	Original	50 ms	0.432	1.44	4.85	2.338
$DFES_{FL+BB}$	Original	100 ms	0.6	**1.35**	**3.74**	**1.99**
$DFES_{FL+BB}$	ResNet-18	50 ms	0.45	1.615	5.9	2.786
$DFES_{FL+BB}$	ResNet-34	50 ms	0.414	1.638	4.79	2.365
$DFES_{FL+BB}$	ResNet-50	50 ms	0.92	3.281	6.98	3.65

Table 5.2. This model achieved an mAP_{50} score of 0.978 on the lab test set, 0.165 on the wild. test set and 0.936 mAP_{50} on the total test set.

In the combined $DFES_{FL+BB}$ models for face landmark and bounding box detection, the highest level of mAP_{50} in the wild test set was achieved using a model with a 100 ms accumulation time and an initial feature extractor. However, the best overall results were obtained using a model with an initial feature extractor and an accumulation time of 50 ms. This model achieved an mAP_{50} score of 0.973 on the lab test set, 0.528 on the wild part, and 0.918 on the total test set. Some test samples processed by this model are shown in Fig. 5-1, where the model accurately determines the bounding box and landmarks of most of the faces, although there are some discrepancies between the predicted results and the actual data. Notably, the $DFES_{FL+BB}$ model performed better than the $DFES_{BB}$ model in detecting a wild bounding box, indicating that face landmarks provide additional information to guide the model training process. Despite high performance, the model still struggled to detect small faces on the wild test set.

Table 5.3 and Table 5.4 shows the results of determining the orientation of the face, measured by the normalized mean error (NME). The model with an accumulation time of 100 ms and the original feature extractor achieved the lowest NME score of 0.394 on the entire test set, which may be due to motion blur. Among the models with 50 ms accumulation time, the ResNet-34 feature extractor model and the original

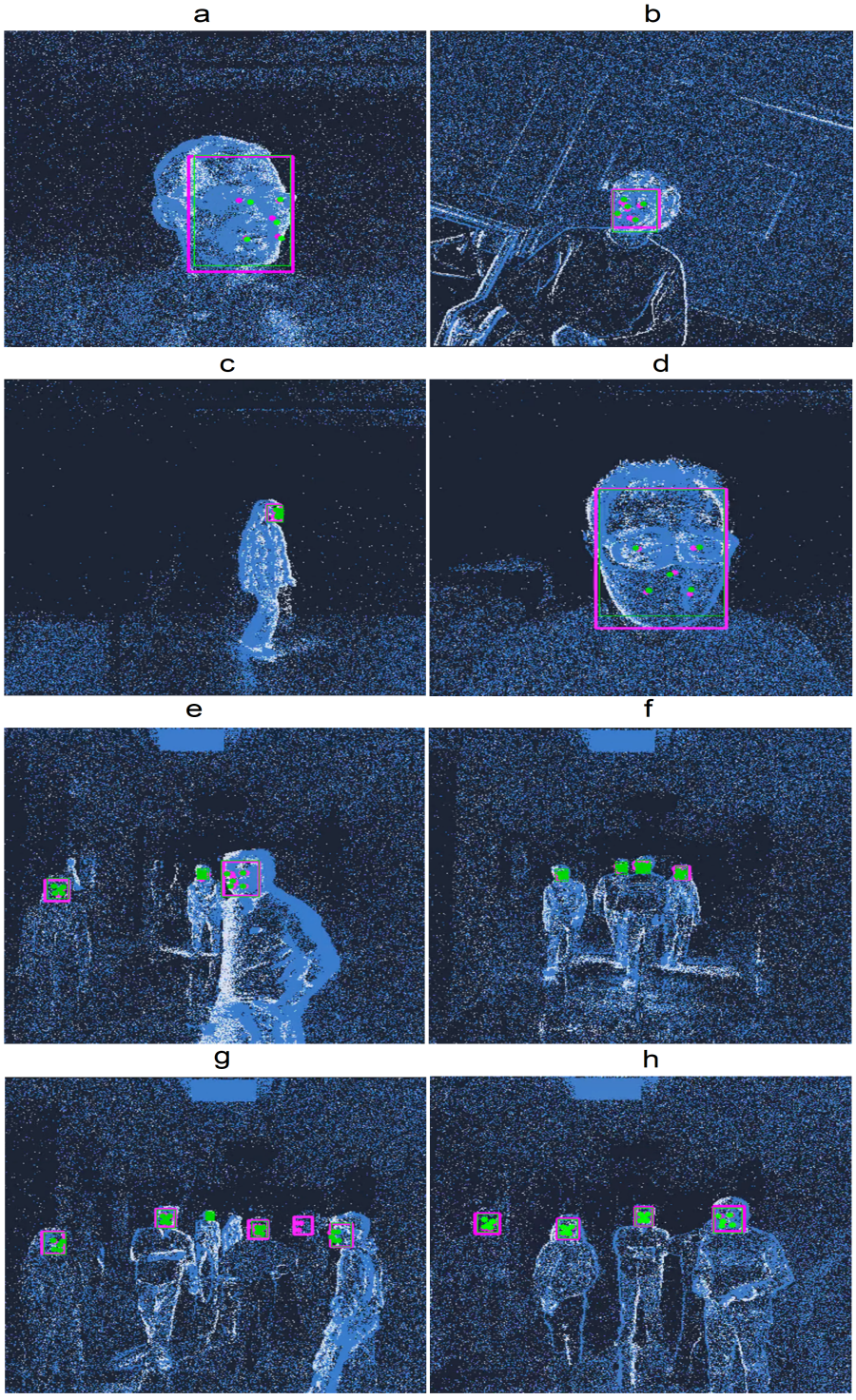

Figure 5-1: Samples of the predicted versus the ground truth bounding box and facial landmarks for the model with the ResNet-34 feature extractor from the controlled (a-d) and wild (e-h) environments. The green color denotes the ground truth, and the magenta color denotes the predictions.

feature extractor model had the same NME scores. However, the original feature extractor model had a slightly lower NME score across the entire test sets. Fig. 5-1 shows that the model predicted facial landmarks better in the lab portion of the test set, but as with the bounding box, the difference between the predicted facial landmarks and the resulting facial landmarks was small.

5.3 Inference Time and Real-time Detection Experiment

The amount of time it takes for a model to analyze and provide a prediction for a single input is measured by an important parameter known as inference time. For real-time applications like ours, where fast and accurate predictions are essential to ensure efficient performance, inference time is important. We can determine the effectiveness and potential of a model for use in real-time applications by evaluating the inference time. In addition, we can determine which model is the most efficient for a particular task by comparing the inference time of many models with different combinations of feature extractor and single detector.

In this experiment, I analyzed the inference time required for the model to predict the face bounding box coordinates and determine the face landmarks on a single GPU (Tesla V100) for a single frame. It has been found that the time taken to predict the bounding box coordinates and facial landmarks increases somewhat as the accumulation time increases. Specifically, for accumulation times of 33 ms, 50 ms, and 100 ms, the models predicted test outputs of 10.3 ms, 10.5 ms, and 10.7 ms, respectively. It was also founded that the ResNet feature extraction model has a longer inference time than the original model. The ResNet-18 feature extractor model predicted faces in 11.9 ms, the ResNet-34 feature extractor model in 13.5 ms, and the ResNet-50 feature extractor model in 15.8 ms. There was no significant difference in the inference times between the $DFES_{BB}$ and $DFES_{FL+BB}$ models. It was also noted that the inference time was less than the event accumulation time, which demonstrates

the possibility of using models for real-time face detection.

Chapter 6

Conclusion

The research presented in this thesis focused on the area of computer vision in robotics, with the introduction of event-based cameras as a solution to the limitations of conventional frame-based cameras. Event-based cameras, which is a new type of retinomorphic sensors, have shown significant advantages over conventional cameras and have found various applications in computer vision. However, due to the relative novelty of event-based cameras, there has been a shortage of datasets and models to apply deep learning to event streams.

To address this gap, this thesis introduced the FES dataset, which is the first large and diverse event-based camera dataset for face and facial landamrks detection. The dataset contains 689 minutes of raw event streams recorded in both wild and controlled environments, with multiple face poses and distances. Additionally, the dataset provides accurately annotated bounding box and facial landmark coordinates, making it an ideal resource for the application of machine learning and other algorithms. Furthermore, 12 different models were created to detect faces and facial landmarks in real-time, using a modified architecture of feature extractor part with a combination of SSD. These models were able to detect faces and facial landmarks with high accuracy.

To encourage further research in this area, the dataset, codes, and trained models are shared on https://github.com/IS2AI/faces-in-event-streams under the MIT license. As the detection of faces and facial landmarks is only the first step towards

the detection of advanced face features such as emotions, gender, age, and face recognition, future work could focus on using these models to create software that can detect these features directly from event-based output. Therefore, some potential direction for further research should be using created models identify the key points of the face and determine the characteristics of the face from them, such as the classification of emotions, human identification. Also, future directions may be research to improve the accuracy of face detection from event-streams, so it can be the study of other architectures of neural networks and the use of attention mechanisms.

This work using these models can be applied to the field of autonomous driving, so using direct event-based output with the fastest response and our models, we can quickly respond to driver falling asleep while driving. Also, the developed models can be used for robotics, so using event cameras, you can supplement the model so that robots can quickly respond to human emotions and actions or identify a person in poor lighting conditions.

As a result, this research has contributed significantly to the development of computer vision and robotics, and the future of event-based cameras looks promising with the availability of resources such as the FES dataset and models for face and facial landmarks detection.

Bibliography

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, and et al. A low power, fully event-based gesture recognition system. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017.

[2] Savran Arman and Bartolozzi Chiara. Face pose alignment with event cameras. *Sensors*, 20:7079, 2020.

[3] Emre Avuçlu and Fatih Başçiftçi. An interactive robot design to find missing people and inform their location by real-time face recognition system on moving images. *Journal of Ambient Intelligence and Humanized Computing*, 13(9):4385–4396, 2022.

[4] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[5] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan. Direct face detection and video reconstruction from event cameras. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016.

[6] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Leng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014.

[7] Ying Bi, Bing Xue, Pablo Mesejo, Stefano Cagnoni, and Mengjie Zhang. A survey on evolutionary computation for computer vision and image analysis: Past, present, and future trends. *IEEE Transactions on Evolutionary Computation*, 27(1):5–25, feb 2023.

[8] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. *Computer Vision – ECCV 2018*, page 781–797, 2018.

[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 43(1):172–186, 2021.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 172–186, 2017.

[11] Dong-il "Dan" Cho and Tae-jae Lee. A review of bioinspired vision sensors and their applications. *Sensors and Materials*, pages 447–463, 2015.

[12] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S. Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5793–5802, 2017.

[13] Kirti Dang and Shanu Sharma. Review and comparison of face detection algorithms. *2017 International Conference on Cloud Computing, Data Science amp; Engineering - Confluence*, pages 629–633, 2017.

[14] Bappaditya Debnath, Mary O'Brien, Swagat Kumar, and Ardhendu Behera. Attentional learn-able pooling for human activity recognition. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13049–13055, 2021.

[15] Tobi Delbruck, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429, 2010.

[16] Wanghua Deng and Ruoxue Wu. Real-time driver-drowsiness detection system using facial features. *IEEE Access*, 7:118727–118738, 2019.

[17] Perot Etienne, de Tournemire Pierre, Nitt Davide, Masci Jonathan, and Sironi Amos. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020.

[18] Francisco F. Sales, David Portugal, and Rui P. Rocha. Real-time people detection and mapping system for a mobile robot using a rgb-d sensor. *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, pages 467–474, 2014.

[19] Matteo Fabbri, Guido Borghi, Fabio Lanzi, Roberto Vezzani, Simone Calderara, and Rita Cucchiara. Domain translation with conditional gans: From depth to rgb face-to-face. *2018 International Conference on Pattern Recognition (ICPR)*, pages 1355–136, 2018.

[20] Di Feng, Christian Haase-Schutz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multimodal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2021.

[21] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, and et al. 5.10 a 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline. *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, pages 112–114, 2020.

[22] Javier Galbally, Sebastien Marcel, and Julian Fierrez. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE Transactions on Image Processing*, 23(2):710–724, 2014.

[23] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, and et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:154–180, Jan 2022.

[24] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 350–359, 2018.

[25] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[26] Arren Glover and Chiara Bartolozzi. Event-driven ball detection and gaze fixation in clutter. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2203–2208, 2016.

[27] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[29] Yang Heng, Jia Xuhui, Loy Chen, Change, and Robinson Peter. An empirical study of recent face alignment methods. *arXiv preprint arXiv:1511.05049*, pages 1–12, 2015.

[30] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4–21, 2017.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[32] Hao Hu, Hua Cai, Zhiyong Ma, and Weigang Wang. Semantic segmentation based on semantic edge optimization. *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 612–615, 2021.

[33] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[34] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. V2e: From video frames to realistic dvs events. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1312–1321, 2021.

[35] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth. *Visual Processing by the Retina*, page 507–515. McGraw-Hill, 2012.

[36] Syamsiar Kautsar, B. Widiawan, Bety Etikasari, Saiful Anwar, Rosiana Dwi Yunita, and Mat Syai'in. A simple algorithm for person-following robot control with differential wheeled based on depth camera. *2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, 2019.

[37] Shehryar Khattak, Frank Mascarich, Tung Dang, Christos Papachristos, and Kostas Alexis. Robust thermal-inertial localization for aerial robots: A case for direct methods. *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1061–1068, 2019.

[38] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. Simultaneous mosaicing and tracking with an event camera. *Proceedings of the British Machine Vision Conference 2014*, pages 566–576, 2014.

[39] Marcin Kopaczka, Raphael Kolk, Justus Schock, Felix Burkhard, and Dorit Merhof. A thermal infrared face database with facial landmarks and emotion labels. *IEEE Transactions on Instrumentation and Measurement*, 68(5):1389–1401, 2018.

[40] Askat Kuzdeuov, Dana Aubakirova, Darina Koishigarina, and Huseyin Atakan Varol. TFW: Annotated thermal faces in the wild dataset. *IEEE Transactions on Information Forensics and Security*, 17:2084–2094, 2022.

[41] Askat Kuzdeuov, Darina Koishigarina, and Hüseyin Atakan Varol. Anyface: A data-centric approach for input-agnostic face detection. pages 211–218, 2022.

[42] Annamalai Lakshmi, Anirban Chakraborty, and Chetan S. Thakur. Neuromorphic vision: From sensors to event-based algorithms. *WIREs Data Mining and Knowledge Discovery*, 9(4):1–34, 2019.

[43] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, 2017.

[44] Gregor Lenz, Sio-Hoi Ieng, and Ryad Benosman. Event-based face detection and tracking using the dynamics of eye blinks. *Frontiers in Neuroscience*, 14:587, 2020.

[45] Bin Li, Hu Cao, Zhongnan Qu, Yingbai Hu, Zhenke Wang, and Zichen Liang. Event-based robotic grasping detection with neuromorphic vision sensor and event-grasping dataset. *Frontiers in Neurorobotics*, 14:1–14, 2020.

[46] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11, 2017.

[47] P. Lichtsteiner and T. Delbruck. A 64x64 event-driven logarithmic temporal derivative silicon retina. *Research in Microelectronics and Electronics, 2005 PhD*, 2005.

[48] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[49] Dongyun Lin, Yiqun Li, Yi Cheng, Shitala Prasad, and Aiyuan Guo. Masked face recognition via self-attention based local consistency regularization. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 436–440, 2022.

[50] Hongjie Liu, Diederik Paul Moeys, Gautham Das, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Combined frame- and event-based detection and tracking. *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2511–2514, 2016.

[51] Shih-Chii Liu and Tobi Delbruck. Neuromorphic sensory systems. *Current Opinion in Neurobiology*, 20(3):288–295, 2010.

[52] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. *Computer Vision and Pattern Recognition*, pages 21–37, 2016.

[53] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *European Conference on Computer Vision (ECCV)*, pages 21–37, 2016.

[54] YunXiang Liu, QianXun Guan, and XinXin Yuan. Research on complex scene recognition based on semantic segmentation. *2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS)*, pages 1–4, 2022.

[55] Florian Mahlknecht, Daniel Gehrig, Jeremy Nash, Friedrich M. Rockenbauer, Benjamin Morrell, Jeff Delaune, and Davide Scaramuzza. Exploring event camera-based odometry for planetary robots. *IEEE Robotics and Automation Letters*, 7(4):8651–8658, 2022.

[56] Salim Malek and Silvia Rossi. Head pose estimation using facial-landmarks classification for children rehabilitation games. *Pattern Recognition Letters*, 152:406–412, 2021.

[57] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso Garcia, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2018.

[58] Timothée Masquelier and Simon J Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, 3(2):247–257, 2007.

[59] Warren McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity (1943). *Ideas That Created the Future*, page 79–88, 1943.

[60] Pascal Mettes and Cees G. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1–10, 2017.

[61] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768, 2014.

[62] Daniel Neil, Shih-Chii Liu, and Delbruck Tobias. *Deep neural networks and hardware systems for event-driven data.* PhD thesis, 2017.

[63] Peter O'Connor, Daniel Neil, Shih-Chii Liu, Tobi Delbruck, and Michael Pfeiffer. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*, 7, 2013.

[64] Yael Omer, Roni Sapir, Yarin Hatuka, and Galit Yovel. What is a face? Critical features for face detection. *Perception*, 48(5):437–446, 2019.

[65] Christoph Posch, Ryad Benosman, and Ralph Etienne-Cummings. Giving machines humanlike eyes. *IEEE Spectrum*, 52(12):44–49, 2015.

[66] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck. Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102(10):1470–1484, 2014.

[67] Bharath Ramesh and Hong Yang. Boosted kernelized correlation filters for event-based face detection. *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 155–159, 2020.

[68] Muhammad Ramzan, Hikmat Ullah Khan, Shahid Mahmood Awan, Amina Ismail, Mahwish Ilyas, and Ahsan Mahmood. A survey on state-of-the-art drowsiness detection techniques. *IEEE Access*, 7:61904–61919, 2019.

[69] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2019.

[70] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. *International Journal of Computer Vision*, 126(12):1394–1414, 2017.

[71] Henri Rebecq, Rene Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–23, 2019.

[72] Sweety Reddy, Silky Goel, and Rahul Nijhawan. Real-time face mask detection using machine learning/ deep feature-based classifiers for face mask recognition. *2021 IEEE Bombay Section Signature Conference (IBSSC)*, pages 1–6, 2021.

[73] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 61–72, 2016.

[74] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[75] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, page 234–241. Springer International Publishing, Cham, 2015.

[76] Imaan Roomaney, Clement Nyirenda, and Manogari Chetty. Facial imaging to screen for fetal alcohol spectrum disorder: Anbsp;scoping review. *Alcoholism: Clinical and Experimental Research*, 46(7):1166–1180, 2022.

[77] Cian Ryan, Brian O'Sullivan, Amr Elrasad, Aisling Cahill, Joe Lemley, Paul Kielty, Christoph Posch, and Etienne Perot. Real-time face amp; eye tracking and blink detection using event cameras. *Neural Networks*, 141:87–97, 2021.

[78] Abd AL-BastRashed Saabia, TarekAbd El-Hafeez, and Alaa M Zaki. Face recognition based on grey wolf optimization for feature selection. In *Proc. of the International Conference on Advanced Intelligent Systems and Informatics*, pages 273–283. Springer, 2018.

[79] Safaa Najah Saud Al-Humairi, Ummar Idraqi Noh, and Wee Ying Ci. Raspberry pi based: Design an android-thermal surveillance robot. *2022 IEEE Conference on Systems, Process amp; Control (ICSPC)*, pages 7–11, 2022.

[80] Arman Savran and Chiara Bartolozzi. Face pose alignment with event cameras. *Sensors*, 20(24):7079, 2020.

[81] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, and et al. Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory–processing– learning–actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks*, 20(9):1417–1438, 2009.

[82] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[83] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, and et al. 4.1 a 640×480 dynamic vision sensor with a 9µm pixel and 300meps address-event representation. *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67, 2017.

[84] Linye Song, Irene Schicker, Petrina Papazek, Alexander Kann, Benedikt Bica, Yong Wang, and Mingxuan Chen. Machine learning approach to summer precipitation nowcasting over the eastern alps. *Meteorologische Zeitschrift*, 29(4):289–305, 2020.

[85] Lea Steffen, Daniel Reichard, Jakob Weinland, Jacques Kaiser, Arne Roennau, and Rüdiger Dillmann. Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms. *Frontiers in Neurorobotics*, 13:1–10, 2019.

[86] Ruolin Sun, Dianxi Shi, Yongjun Zhang, Ruihao Li, and Ruoxiang Li. Data-driven technology in event-based vision. *Complexity*, 2021:1–19, 2021.

[87] Richard Szeliski. *Computer vision algorithms and applications*. Springer Nature Switzerland AG, 2023.

[88] Tasbolat Taunyazov, Weicong Sng, Brian Lim, Hian Hian See, Jethro Kuan, Abdul Fatir Ansari, Benjamin Tee, and Harold Soh. Event-driven visual-tactile sensing and learning for robots. *Robotics: Science and Systems XVI*, pages 1–13, 2020.

[89] Mohammad-Hassan Tayarani-Najaran and Michael Schmuker. Event-based sensing and signal processing in the visual, auditory, and olfactory domain: A review. *Frontiers in Neural Circuits*, 15:1–15, 2021.

[90] Lin Tsung-Yi, Goyal Priya, Girshick Ross, He Kaiming, and Dollár Piotr. Focal loss for dense object detection. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[91] Nizam Ud Din, Kamran Javed, Seho Bae, and Juneho Yi. A novel gan-based network for unmasking of masked face. *IEEE Access*, 8:44276–44287, 2020.

[92] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[93] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. *Computer Vision – ECCV 2016*, page 20–36, 2016.

[94] Xiaoli Wang, Yinglin Zheng, Ming Zeng, Xuan Cheng, and Wei Lu. Joint learning for face alignment and face transfer with depth image. *Multimedia Tools and Applications*, 79(45):33993–34010, 2020.

[95] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv: Computer Vision and Pattern Recognition*, pages 1–14, 2017.

[96] Yanchun Wu, Jianqin Yin, Lei Wang, Huaping Liu, Qi Dang, Zhiming Li, and Yilong Yin. Temporal action detection based on action temporal semantic continuity. *IEEE Access*, 6:31677–31684, 2018.

[97] Biao Yang, Jinmeng Cao, Rongrong Ni, and Yuyu Zhang. Facial expression recognition using weighted mixture deep neural network based on double-channel facial images. *IEEE Access*, 6:4630–4640, 2018.

[98] Biao Yang, Jinmeng Cao, Rongrong Ni, and Yuyu Zhang. Facial expression recognition using weighted mixture deep neural network based on double-channel facial images. *IEEE Access*, 6:4630–4640, 2018.

[99] Qihang Yang, Tao Chen, Jiayuan Fan, Ye Lu, Chongyan Zuo, and Qinghua Chi. Eadnet: Efficient asymmetric dilated network for semantic segmentation. *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2315–2319, 2021.

[100] Yisa Zhang, Hengyi Lv, Yuchen Zhao, Yang Feng, Hailong Liu, and Guoling Bi. Event-based optical flow estimation with spatio-temporal backpropagation trained spiking neural network. *Micromachines*, 14(1):203, 2023.

[101] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 74–95, 2017.