Titre: Title:	A Platform for Enhancing the Vision of Patients Suffering from Age-Related Macular Degeneration Disease
Auteur: Author:	Nizar El Zarif
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	El Zarif, N. (2021). A Platform for Enhancing the Vision of Patients Suffering from Age-Related Macular Degeneration Disease [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/9974/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/9974/
Directeurs de recherche: Advisors:	François Leduc-Primeau, & Mohamad Sawan
Programme: Program:	Génie électrique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

A platform for enhancing the vision of patients suffering from age-related macular degeneration disease

NIZAR EL ZARIF
Département de génie électrique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie électrique

Décembre 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

A platform for enhancing the vision of patients suffering from age-related macular degeneration disease

présentée par **Nizar EL ZARIF**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Yves AUDET, président
François LEDUC-PRIMEAU, membre et directeur de recherche
Mohamad SAWAN, membre et codirecteur de recherche
Samuel KADOURY, membre
Jocelyn FAUBERT, membre externe

DEDICATION

This is dedicated to my wonderful mother, father and brothers, to my friends who encouraged me to continue working on my PhD despite all the issues and setbacks . . .

ACKNOWLEDGEMENTS

Thank you for Prof. Sawan for the original vision, supervision and support throughout the years. And thanks, to Prof. Leduc-Primeau who helped me focus in my research, refine my ideas, and better present the work. Thanks for NSERC and Intel for financial support and opportunity to improve my technical and research skills. I would also like to thank my colleague and friend Leila Montazeri for complementing her contribution to this work with custom designed micro-LED stimulation driven by a custom power driver to supply the LED in a tiny footprint.

RÉSUMÉ

La dégénérescence maculaire liée à l'âge est la principale cause de cécité en Amérique du Nord sans traitement médical fiable. Cette maladie progressive se manifeste par une déficience visuelle sévère de la vision centrale à ses étapes intermédiaires ou par une perte totale de la vision centrale dans ses étapes ultérieures en raison de la dégradation de la couche photoréceptrice de la rétine. Il n'existe actuellement aucun traitement fiable pour ces maladies et très peu de mesures préventives qui ralentissent leur progression. De nombreuses approches ont été proposées pour aider à atténuer ce problème. Une canne et des chiens-guides sont parmi les méthodes les plus fiables au monde pour aider les malvoyants. Les nouvelles approches visent à tirer parti de la puissance de la technologie pour offrir une meilleure expérience au patient grâce à des guides de navigation électroniques ou des aides visuelles.

Le travail présenté dans cette thèse est une plate-forme destinée à la stimulation optique pour la dégénérescence maculaire liée à l'âge, ceci en émulant le traitement rétinien se produisant dans la macula en temps réel et en envoyant un signal optogénétique à un dispositif de stimulation photo-électrique à la couche rétinienne souhaitée pour restaurer la vision dans une rétine endommagée. Ceci est réalisé avec un modèle de traitement d'image rapide, simplifié et ajustable de la rétine utilisant un filtrage spatial et temporel. Une capture vidéo en direct et des images à l'échelle sont ensuite utilisées pour adapter la prothèse visuelle à plusieurs résolutions. L'aspect temporel de la macula est émulé à l'aide d'un filtre temporel multi-images personnalisé et d'une intégration décroissante qui correspond à la vitesse de la voie temporelle dans la rétine, tandis que l'aspect spatial est modélisé comme une différence de gaussien. Ensemble, ils se combinent en un modèle simple et puissant qui représente la voie principale de la rétine humaine. Le modèle est mis en œuvre de manière à être extensible et efficace en termes de calcul pour obtenir les meilleurs résultats dans un budget de temps strict. Le modèle implémenté est suffisamment rapide pour fonctionner en temps réel sur les appareils Raspberry Pi 3 et Raspberry Pi 4.

La sortie de la plate-forme génère un signal à envoyer à un stimulateur micro-LED conçu par Leila Montazeri *et al.* Le stimulateur générera un signal lumineux focalisé qui agit sur les cellules ganglionnaires rétiniennes traitées avec des opsines photosensibles, dans le but de restaurer une certaine acuité visuelle dans la vision centrale. En raison des limites de fabrication, seule une matrice micro-LED 8x8 peut être produite. Cela signifie que seule une image 8x8 peut être envoyée au micro-stimulateur, et il est donc important de représenter les images par un très petit nombre de pixels. Nous avons choisi de nous concentrer sur l’expression faciale car elle est très importante dans la communication et a de nombreuses utilisations dans des domaines très différents, du divertissement à la médecine personnalisée et aux interactions homme-machine.

Deux algorithmes de reconnaissance automatique des expressions faciales en temps réel ont été mis en œuvre, capables de détecter rapidement les expressions faciales sous différentes résolutions avec plusieurs poses. Le premier algorithme utilise plusieurs descripteurs locaux basés sur des caractéristiques faciales géométriques et un ensemble de classificateurs qui reconnaissent les caractéristiques locales et déduit les caractéristiques globales de la reconnaissance locale. Le deuxième algorithme fusionne à la fois des caractéristiques géométriques et de texture rapides avec un petit réseau de neurones convolutif qui produit des résultats rapides et précis en une fraction de la taille et du temps des grands réseaux populaires, et avec une meilleure précision sur les résultats de validation croisée.

ABSTRACT

Age-related macular degeneration is the leading cause of blindness in North America with no reliable medical treatment. This progressive disease manifests itself with severe visual impairment in the central vision in its intermediate stages or with the total loss of central vision in its later stages due to the decay in the photoreceptor layer in the retina. There are currently no reliable treatments for these diseases and very few preventative measures that slow down their progression. There have been many approaches proposed to help mitigate this problem. A walking stick and guide dogs are among the most reliably used around the world to provide aid for the visually impaired. Newer approaches aim to leverage the power of technology to provide a better patient experience through electronic navigation guides or visual aids.

The work presented in this thesis is a platform intended for driving age-related macular degeneration optic stimulation by emulating retinal processing occurring in the macula in real-time and sending an optogenetic signal to a photo-electric stimulation device to the desired retinal layer to recreate vision within a damaged retina. This is achieved with a fast, simplified, and tunable image processing model of the retina using spatial and temporal filtering. Live video capture and scaled images are then used to fit the visual prosthesis with multiple resolutions. The temporal aspect of the macula is emulated using a custom-written multi-frame temporal filter and a decaying integration that matches the speed of the temporal pathway in the retina, while the spatial aspect is modeled as a difference of Gaussian. Together they combine into a simple and powerful model that represents the major pathway in the human retina. The model is implemented in a way to be expandable and computationally efficient to get the best results under a strict time budget. The implemented model is fast enough to work in real-time on Raspberry Pi 3 and Raspberry Pi 4 devices.

The output of the platform generates a signal to be sent to a micro-LED stimulator designed by Leila Montazeri *et al.* The stimulator will generate a focused light signal that works on retinal ganglion cells treated with light-sensitive opsins, with the aim of restoring some visual acuity in the central vision. Due to manufacturing limitations, only an 8x8 micro-LED matrix can be produced, hence it is important to represent images with a very small number of pixels. We chose to focus on facial expressions since they are crucial for human communication. In addition, facial expression recognition has many uses in several fields, such as entertainment, personalized medicine, and human-machine interaction.

Two automatic real-time facial expression recognition algorithms were implemented that are capable of fast detection of facial expressions under different resolutions with multiple poses. The first algorithm uses multiple local descriptors based on geometric facial features and an ensemble of classifiers that recognize local features and deduces global features from local recognition. The second algorithm merges both fast geometric and texture features with a small convolutional neural network that produces fast and accurate results in a fraction of the size and time of large popular networks with better accuracy on cross-validation results.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

AMD	Age-related Macular Degeneration
ASM	Active Shape Model
AVFD	Alternate Video Frame Detector
CCTV	Closed Circuit Television
CK+	extended Cohn-Kanade
CNN	Convolutional Neural Network
DoG	Difference of Gaussian
FACS	Facial Action Coding System
FER	Facial Expression Recognition
FERNet	Facial Expression Recognition Network
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
GIL	Global Interpreter Lock
GPU	Graphcal Processing Unit
LBP	Local Binary Pattern
LGN	Lateral Geniculate Nucleus
LIFO	Last In First Out
LoG	Laplacian of Gaussian
MEA	Micro-electrode Array
MSTM	Multiple Spatial-Temporal Model
MUG	Multimedia Understanding Group
REFER	Real-time Ensemble for Facial Expression Recognition
RGC	Retinal Ganglion Cells
RP	Retinitis Pigmentosa
SVM	Support Vector Machine
VEGF	Vascular Endothelial Growth Factor

CHAPTER 1 INTRODUCTION

1.1 The human eye

Humans are primarily visual creatures, which means that the eyes are the primary sense of the world. They are used to analyze an object, to identify friend from foe, to read books, and to explore the surroundings and gaze at the vast emptiness of the space. The human visual system evolved to become one of the most complex systems in the universe. From the eyes that capture the light and convert it to an electrical signal, to the ganglion cell that relays the image to the brain, to the visual cortex that decodes the image. It is a system that requires millions of photoreceptors and billions of neurons. Unlike cameras where it is designed in a rectangular grid, the photoreceptors whether cones or rods do not appear to have an arranged structure. Amazingly, we are able to perceive the world without distortion.

1.1.1 Visual impairment

It is estimated that 285 million people around the world suffer from a form of visual impairment [1]. These visual impairments range from moderate to severe to complete blindness. One of the more critical types of diseases that affect human vision are macular diseases. These diseases usually affect people with old age. And these diseases account for almost 6.6% of all blindness cases and about 3.1 % of moderate to severe visual impairment [2], and this number of blindness or severe visual impairment is likely to rise due to the shift in old age of the population. Per World Health Organization report [1], that 65% of the visually impaired are 50 years of age or older.

Retinal visual diseases can be classified into two types based on the effect that they have on the person: the first is a loss of visual acuity such as Age-related Macular Degeneration (AMD), the second type is the loss of field of view such as Retinitis Pigmentosa (RP). For some visual impairments, such as Myopia or Stigmatism, the error can be corrected using lenses since both are due to a fault in the human lens, not the retina. However, AMD and RP cannot be corrected with regular lenses, since in both cases the photoreceptors themselves are inactive, so traditional treatment fails.

Figure 1.1 shows the human visual system [3]. The eyes are the input of the system, the light goes through the optical lens that focuses the image on the retina which is filled with photoreceptor cells known as cones and rods. The cones are mostly concentrated in the fovea and while the rods are concentrated outside the fovea. Generally, there are three types of

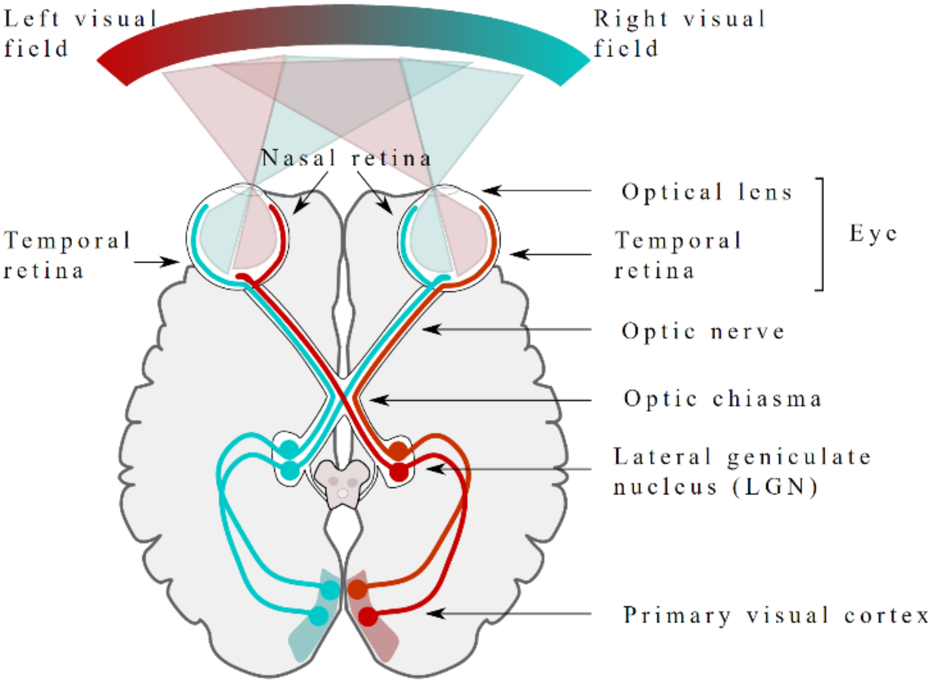


Figure 1.1 Simplified view of The Human Visual System [3]

cones known as S, M, and L, where the S sensitivity peaks in the blue-light wavelength, M near the green, and L near the red. Rods are sensitive to all types of light; hence the cones are best to extract color and detail from the scene, while rods are good for night and peripheral vision. The photoreceptors are connected to several layers in the retina, which does some basic image processing such as edge detection and movement detection. At the end of the retina, we find the retinal ganglion cells. These cells connect the retina to the rest of the visual pathway. Retinal Ganglion cells aren't photosensitive but they connect the retina to the rest of the visual pathway.

1.1.2 Layers in the retina

The retina itself is constructed out of 5 different layers which are:

1. **Photoreceptors**: Rods and cones, these cells receive light and transform it into an electrical signal. When the lights are off the photoreceptors are constantly broadcasting neurotransmitters. When a change in the scenery is detected the neurotransmitters stop firing.
2. **Horizontal Cells**: These cells are usually connected to several rods or cones and

connect to ganglion cells. It is thought that this layer is responsible for detecting edges and can inhibit other ganglion cells so that eye can focus more on the edges.

3. **Bipolar Cells**: This layer of cells in the retina works when it receives an inhibitory signal from the photoreceptors and sends an activation signal to the ganglion cells. This layer can be considered as a delayed inverter. There are at least 14 types of bipolar cells [4].

4. **Amacrine cells**: Not much is known about these types of cells types but they are thought to be essential in motion detection and spatial computation and orientation [4].

5. **Retinal Ganglion Cells (RGC)**: There are at least 30 types of RGC, such as On-surround and Off-surround [5]. The connection from this layer to the fovea can be one to one (one Photoreceptor to one RGC) in the macula and up to thousands to one in peripheral vision. The RGC combines input from the previous layers of cells and relays a response to the next part of the visual pathway which is the Lateral Geniculate Nucleus (LGN).

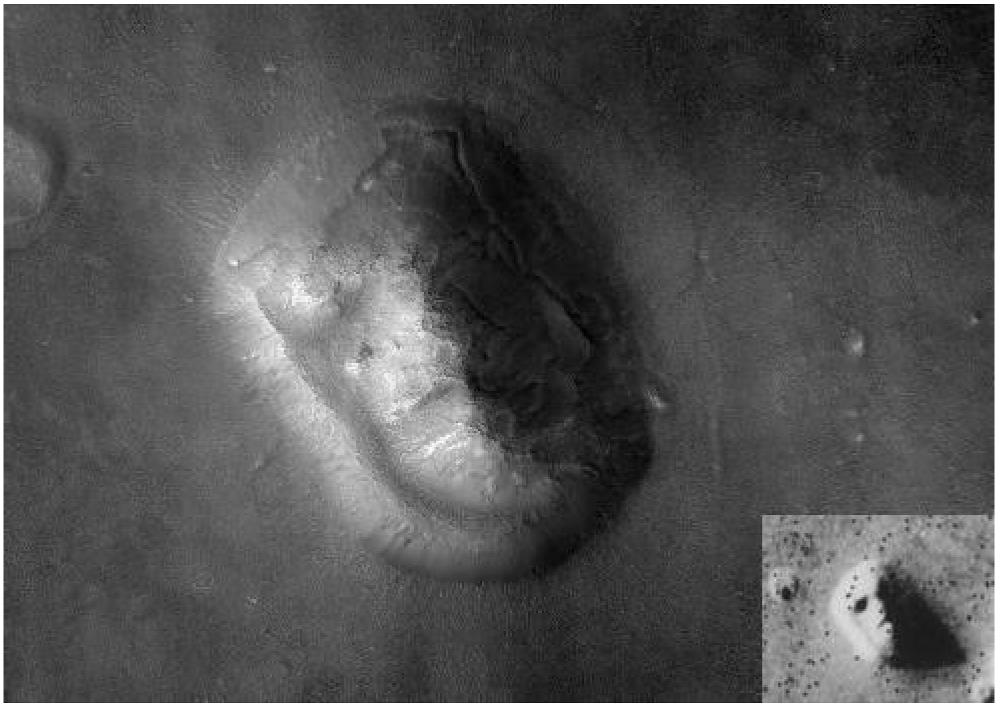


Figure 1.2 Face on Mars[10]

There have been several proposed mathematical models for each layer of the retina such as differential equations of each connection between two layers. However, the simplest model for the entire retina is the decaying exponential function [6].

1.1.3 Retinal stimulation

The first step to restoring visual acuity in a blind patient was via cortical stimulation. Brindley and Lewin [7] showed that phosphenes (white dot) can be created inside the visual field of the test subject by stimulating the part of the visual cortex via controlled electric stimulation. The frequency and length of the stimulation determine the size and the intensity of the created phosphenes. The location of the stimulation directly affects the position of the phosphenes in the visual field. This means that controlled phosphenes stimulation can potentially draw some basic images inside the visual field. Cortical stimulation can be dangerous since it might cause seizures, convulsions, and infections [8]. So other types of stimulation have been developed such as subretinal, epiretinal, optic nerve [9]. However, electric stimulation is not the best solution for retinal stimulation since the number of electrodes that can be planted inside the retina is limited, and power delivery for stimulation in the retina proves to be challenging. Additionally, electrode suffers from wear and tear, and constant electrical stimulation can be dangerous to the nerves cells. For these reasons, it is important to look for different types of stimulation that can create precise phosphenes without the previously mentioned limitations. While the human visual system has amazing capabilities to detect patterns and identify objects, it is easily fooled by shading and abnormal shapes. This means that humans can see patterns or faces when there are none, probably because human brains have been trained to detect patterns since birth. Figure 1.2 is the famous “face on mars” photo, which shows a mountain range that under certain lighting condition looks like a face (see the image on the bottom right of Figure 1.2 [10]. Another example of the interesting aspects of the human visual system is that it can easily fool our brain to see 3D stereoscopic images and videos on 2D display by splitting images between left and right.

1.1.4 Benchmarking human vision

Humans can classify thousands of objects in a matter of seconds. The process of object recognition is so complex that it is estimated that half of the nonhuman primate neocortex is devoted to visual processing [11]. This biological process includes object tracking, segmentation, obstacle avoidance, object grasping... The behavioral response of monkeys shows that it is capable of reacting to an image within 250 ms, while the same for humans is up to 350 ms for one image and 100 ms for a sequence of images, the actual object recognition hence takes even less time, Figure 1.3 shows the division of different part of the visual system. The basic scheme for a brain method to identify an object is equivalent to a simple neural network, with multiple input neurons and a single output neuron thus drawing a hyperplane that separates the category of the object as shown in Figure 1.4.

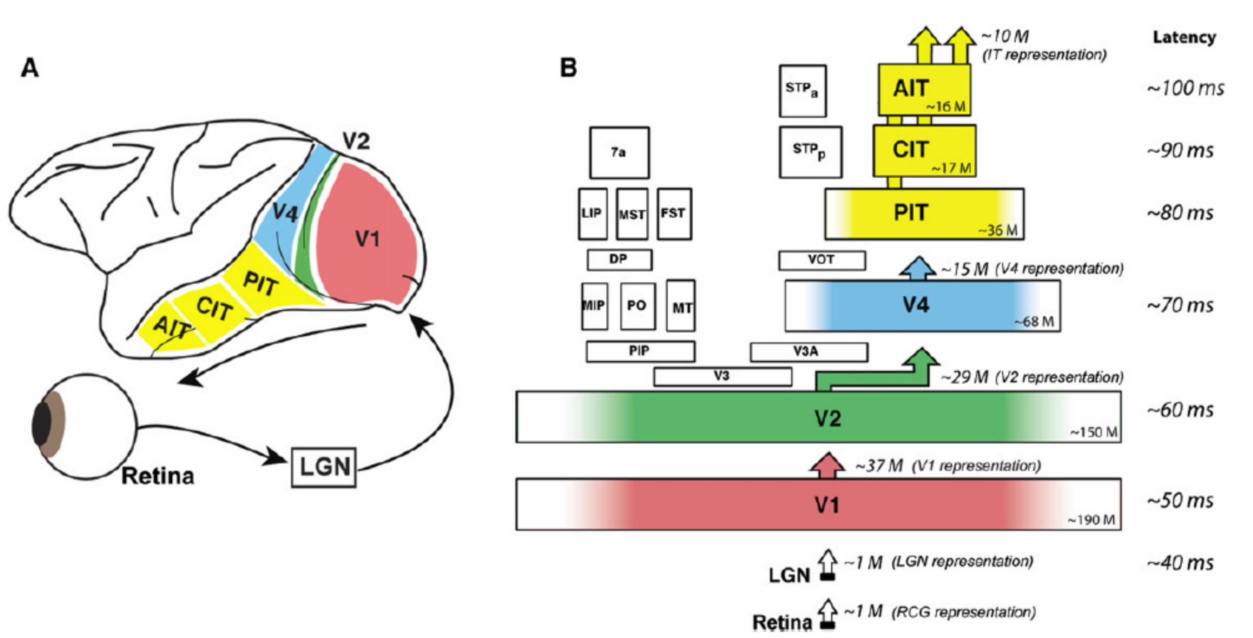


Figure 1.3 The Monkey Visual System: (A) Flow of the Visual Information, (B) The Flow of Visual Information Based on The Size of Each Part of The Visual Pathways [12]

1.1.5 The role of the remaining parts of the human visual system

The ventral visual stream plays a central role in invariant object recognition. The null hypothesis is that core object recognition is obtained by a largely feed-forward cascade of nonlinear filtering operations. Neurons allow moderate changes in object position, size, pose, illumination, and clutter. RGC and LGN processing help deal with important real-world issues such as variation in luminance and contrast across each visual image. Two theories explain how the brain process visual information. The first is a series of cascade forward-only manner (like an assembly line), the second theory is that of a feedforward system with negative feedback for optimization and stability. It is also theorized that there are some abstraction layers between each node of the process. Some authors theorized that local units can solve their task with global supervision [12].

Essentially the neural organization of the visual system is a largely feedforward, reflexively computed, cascaded scheme in which visual information is gradually transformed and retransmitted via a firing rate code along the ventral visual pathway, and presented for easy downstream consumption (i.e., simple weighted sums read out from the distributed population response) [11].

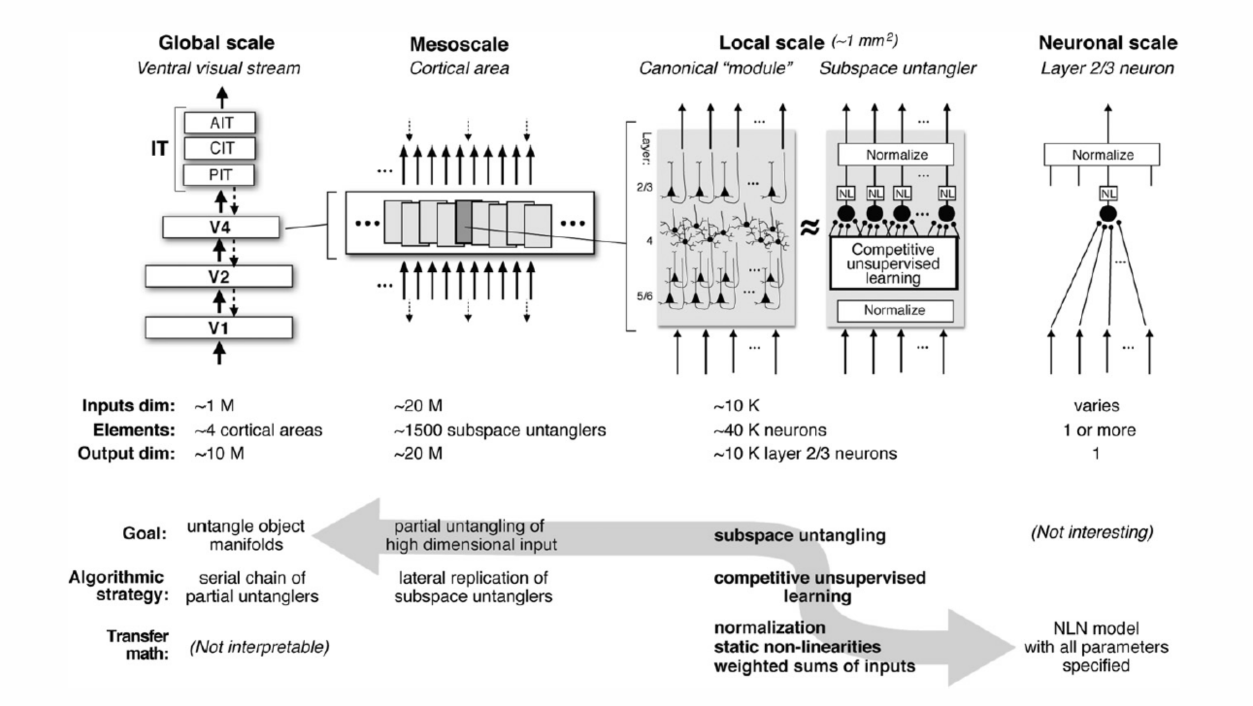


Figure 1.4 Human Neural Network Breakdown of Different Components and Connections on Different Hierarchical Scales [10]

1.2 Age-related macular degeneration

In Europe, the USA, and Australia, the leading cause of blindness is Age-related macular degeneration causing up to 50% of all blindness cases [13]. The center of the retina, the macula, is particularly prone to a process of degenerative changes known as AMD. The prevalence of AMD is up to 3% in all adults [14]. There are signs of AMD in 25.3% of the population over 60 years of age [15].

1.2.1 Types of AMD

Broadly speaking, there are two types of AMD, “dry” AMD and “wet” AMD. The former is caused by Retinal Pigment Epithelium (RPE) cells atrophy. RPE cells are responsible for light absorption, spatial ion buffering, epithelial transport, visual cycle, secretion and immune modulation, and phagocytosis. Wet AMD is attributed to abnormal growth of blood vessels underneath the macula. Dry AMD is characterized by loss of central vision and tends to progress gradually, while wet AMD presents itself suddenly, with metamorphosia (change in the size of the object), distortion, and loss of central vision. Late phases of AMD patients usually retain their peripheral vision even when the patient is identified as legally “blind”. In the US, the term legally blind refers to a person with less than 20/200 vision in the better eye or less than 20-degree field of view on the central vision. Figures 1.5 and 1.6 shows the images of healthy retina compared to damaged retina [16]

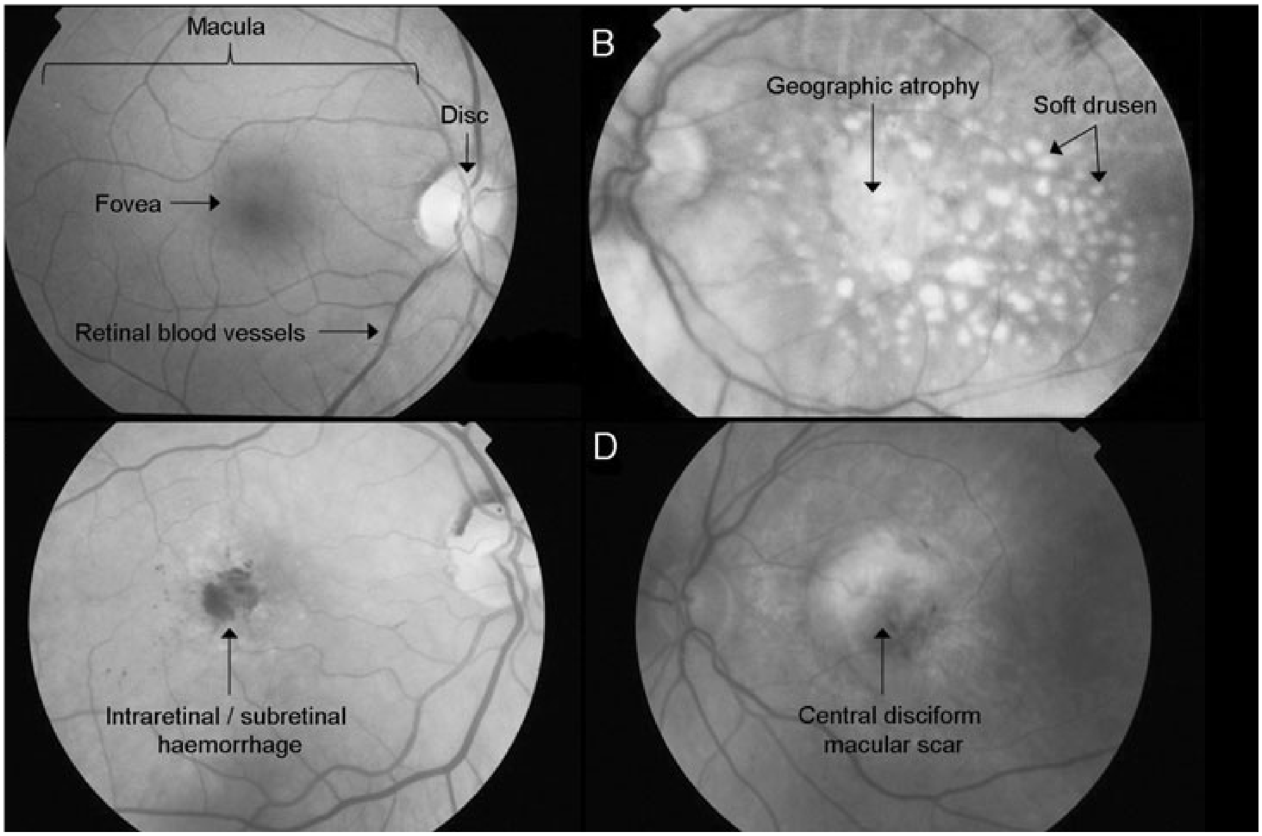

Macula
Disc
Fovea
Retinal blood vessels
B
Geographic atrophy
Soft drusen
Intraretinal / subretinal haemorrhage
D
Central disciform macular scar

Figure 1.5 Comparison of Healthy Eye vs Sick Eyes. (A): healthy, (B) Geographic atrophy, (C) Subretinal Hemorrhage, (D): Macular Scar [16]

There is currently no reliable preventative measure for either dry or wet AMD or treatment of lost sight due to AMD, but there are some modern intravitreal injections to stabilize the condition and sometimes partially restore vision in wet AMD. In later stages of AMD, the patient might suffer from severe vision impairment which can lead to legal blindness. Some aids are given such as magnifying aids, Closed Circuit Television (CCTV), telescopes, and screen reader software. Some patients even reported benefits from inserting magnifying intra-ocular lenses.

A suggested treatment for wet AMD based on human embryonic stem cells transplantation in the RPE. Their efforts resulted in a slight improvement in visual acuity, but it is still not clear what the long-term effects are [17].

Dry AMD progress from small drusen and hyperplasia of RPE which have a small effect on vision to later stages which cause Geographical Atrophy (GA) or Choroidal Neovascularization (CNV). Evidence that smoking and genetic disposition might contribute to the development of AMD. There have been several proposed methods to stop or slow the progression of AMD like using a laser to dissolve drusen or anti-Vascular Endothelial Growth Factor (VEGF) injections for wet AMD. A recent study showed that the number of moderate to severe cases for AMD has dropped by 30% between 1990 to 2020 largely attributed to anti-VEGF injections [18]. For a while, wet AMD had an effective treatment which is the removal of the excessive fluid built up in the eye via surgery, this treatment can stop AMD progression, but it can also cause loss of visual acuity if the operation failed, it was later replaced with anti-VEGF injection. Dietary changes can also help reduce the progression of AMD, especially Zinc and antioxidants [19].

The introduction of anti-VEGF agents has revolutionized the management of wet AMD. In a rapidly changing field, the recommended treatment includes:

1. Pan-anti-VEGF blockade using ranibizumab. The use of Avastin remains controversial but potentially very cost-effective.

2. Argon laser ablative therapy restricted to small, well-defined extrafoveal CNV.

3. The role of PDT may be in combination therapy and is subject to further study at present.

4. Surgical macular translocation may be considered in selected cases including second affected eyes not suitable for anti-VEGF therapy (including RPE involving the macula).

5. Preventative measures including cessation of smoking and use of AREDS nutritional supplements.

There are two types of drusen, hard drusen, and soft drusen. Hard drusen are small lesions with sharp bodies and soft drusen which have no indistinct border. Hard drusen pose no adverse prognostic significance, on the other hand, soft drusen are considered precursors to advanced AMD.

1.2.2 Stages of age-related macular degeneration

Dry AMD progresses into three phases. The first dry phase of AMD is characterized as small drusens in the retina which are caused by waste accumulation under the RPE and slowing down of nutrient transport. In the second phase, the overlaying of receptors gets damaged.

The final phase is characterized by the progressive atrophy of RPE and photoreceptors. 80-90% of AMD diagnosed patients AMD fall in this stage, the progression from early to late dry AMD tends to be slow.

Wet AMD is due to the breaking of vessels under the RPE causing huge damage, this might cause the objects in the visual field to become distorted. The progression of this type of AMD is rapid and can lead to blindness if left untreated.

There are several imaging techniques used to detect if the patient has AMD. Fundus camera can be used to detect small drusens by capturing images of the internal layers of the eye. Fundus Auto Fluorescent (FAF) photography can detect fluorophore which is a by-product of photoreceptor degeneration. Different types of AMD can also be detected using spectral-domain optical coherence tomography (SD-OCT), which shows volumetric information on retinal fluids and drusen. Infrared images are also used to detect dry AMD along with sub-retinal structure, however, they are not good at detecting fluids hence not good for wet AMD.

We can identify GA or drusen with several methods, one such as texture segmentation by applying a Gabor filter or wavelet transform. This process can even be automated using classification techniques such as support vector machine (SVM). Thresholding is also used to detect drusen since they have distinct intensity values. Even though we have methods to detect the existence of drusen, it is hard to detect whether it is hard or soft by machine. Anti-oxidant and zinc supplement showed a significant slowdown in dry AMD progression for 25% of the patient [19].

From all the literature, we can see that there is no current reliable cure for AMD or other types of retinal degenerative disease. However, things are looking brighter in the field of computer vision and digital image processing. Since modern push towards electrical stimulation and controlled stimulation of the RGC shows great promise for recreating part of the vision.

There is a need to find a solution, preferably a non-invasive one. Electronic aids are usually much faster to research, develop and deploy, they are also more affordable and less risky than medical procedures. They have proven their worth before (e.g. insulin pumps, pacemakers, artificial limbs, and cochlear implants). For this reason, it is important to look at electronic prostheses to provide the people with the aid needed.

1.3 Objectives

Given what we know about the human visual system and the brain, what can be done for people with visual impairment to help them with their daily life? This research provides a platform of visual stimulation targeting Age-related Macular Degeneration (AMD). We can divide the objectives into long and short-term objectives.

1.3.1 Long-term objectives

The long-term goal of this research is to create a platform that can deliver optogenetic signals in real-time to a portable battery-powered device. The device we are aiming for should not be bigger than a typical Video Processing Unit (VPU) used in other visual prostheses.

1.3.2 Short-term objectives

To reach the long-term objective of this project we can split the long-term goal into other goals.

Aim 1: Create a custom real-time high-resolution model that emulates the pathways in the macula such as spatial, temporal filtering, and light adaptation that works well with optogenetic stimulation targeting the retinal ganglion cells. This goal was partially achieved in Article 1 and further improved upon in Article 2, which are presented in Chapter 3.

- Article 1: **N. E. Zarif**, L. Montazeri and M. Sawan, "Real-Time Retinal Processing for High-Resolution Optogenetic Stimulation Device," 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018, pp. 5946-5949, doi: 10.1109/EMBC.2018.8513692.

- Article 2: **N. El Zarif**, L. Montazeri, F. Leduc-Primeau and M. Sawan, "Spatio-temporal Retinal Processing Platform for Optogenetic Wearable Microstimulator Targeting Age-related Macular Degeneration." IEEE Access, to be submitted.

Aim 2: Because we were aiming to use a microLED stimulator developed by a colleague from Polystim Group (Leila Montazeri) that is limited by an 8 by 8 black and white LED matrix we propose using machine learning to get context from the image. We focused on facial expression recognition as it is important in day-to-day interaction. This was achieved in Article 3, which occupies a large part of Chapter 4.

- Article 3: **N. El Zarif**, L. Montazeri, F. Leduc-Primeau and M. Sawan, "Mobile-Optimized Facial Expression Recognition Techniques," in IEEE Access, doi: 10.1109/ACCESS.2021.3095844.

This work represents a part of a large project. Some additional publications related to the overall project:

- L. Montazeri, **N. El Zarif**, S. Trenholm and M. Sawan, "Optogenetic Stimulation for Restoring Vision to Patients Suffering From Retinal Degenerative Diseases: Current Strategies and Future Directions," in IEEE Transactions on Biomedical Circuits and Systems, vol. 13, no. 6, pp. 1792-1807, Dec. 2019, doi: 10.1109/TBCAS.2019.2951298.
- L. Montazeri, **N. El Zarif**, T. Tokuda, J. Ohta and M. Sawan, "Active Control of μLED Arrays for Optogenetic Stimulation," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-5, doi: 10.1109/ISCAS.2018.8351272.
- L. Montazeri, **N. E. Zarif** and M. Sawan, "Optical Control of Neural Dynamics Using LED Array," 2018 16th IEEE International New Circuits and Systems Conference (NEWCAS), 2018, pp. 96-99, doi: 10.1109/NEWCAS.2018.8585524.

1.4 Thesis organization

In this chapter, we first described the prevalence of worldwide visual impairment, and how it is related to AMD. We also presented a high-level description of the human retina layers. We also explained the different phases of AMD with possible treatments and the limitations of these treatments. Thus indicating a strong need for prostheses to help with this type of visual impairment.

The rest of this article-based thesis is organized as follows. Chapter 2 will dive into the literature and discuss in depth the cause and effect of AMD and the possible treatments. Then, Chapter 3 presents the proposed retinal model, which will be the starting point for future retinal stimulation based prostheses. Part of this work was published in Article 1. We have also submitted a journal article with improvements to the model (Article 2). Chapter 4 presents the two proposed facial expression recognition systems, which would be helpful in prostheses targeting scene enhancement or replacement for possible future prostheses. This work was published in Article 3. Finally, Chapter 5 presents the overall system-wide tests, Chapter 6 summarizes possible limitations of each subsystem, and Chapter 7 concludes the thesis.

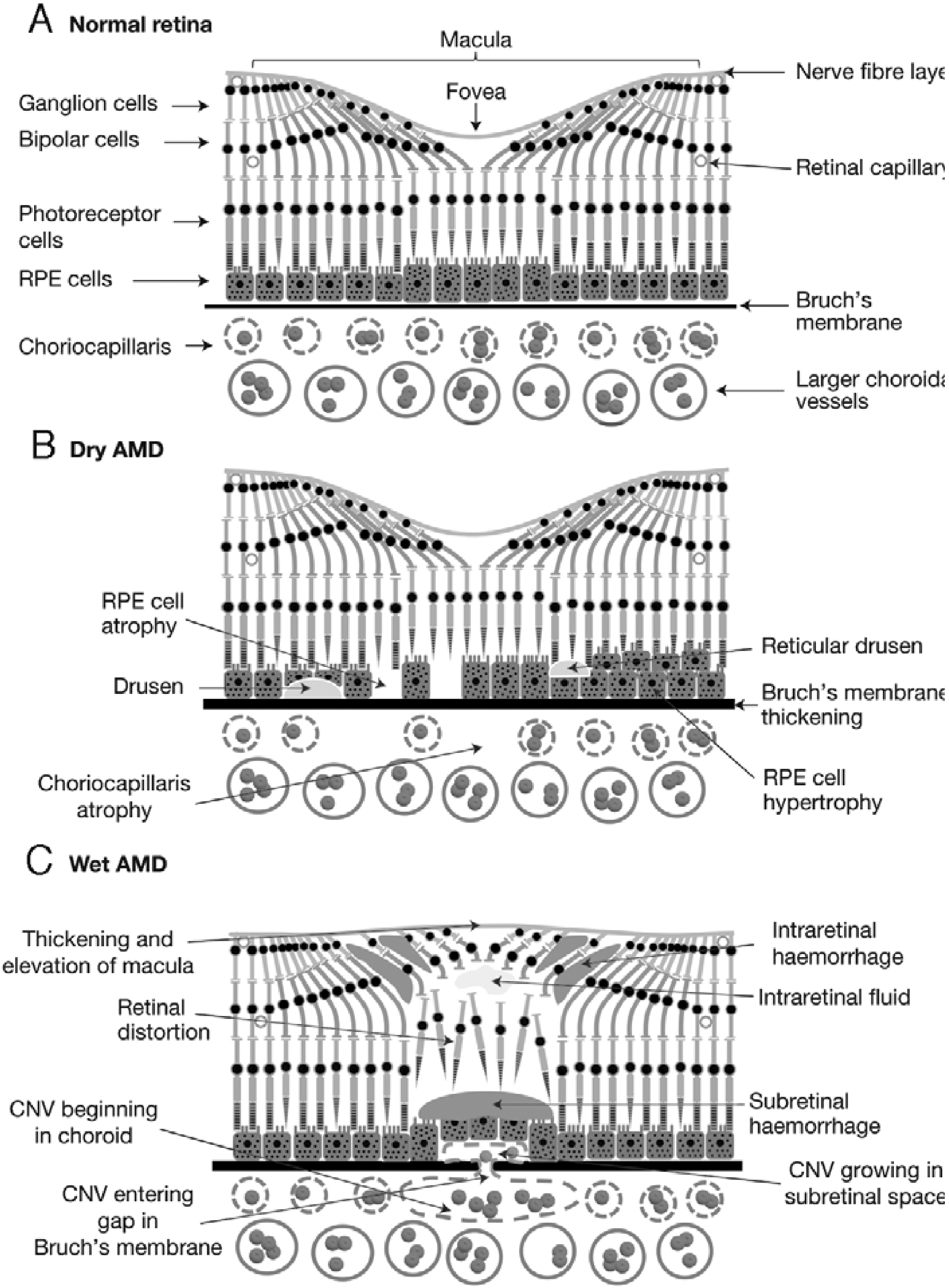


Figure 1.6 Constitution of the Retina. (A) Healthy, (B) Dry AMD, (C) Wet AMD [16]

CHAPTER 2 LITERATURE REVIEW

The main goal of a prosthesis is to replace a malfunctioning organ with synthetic devices that mimic its main functionality. Such as a replacement limb that restores the capability to walk for people with injured libs, cochlear implants that can restore hearing for some patients who lost their hearing, speech synthesizer to help people that cannot speak. Visual prostheses are meant to help the patient by mimicking the functionality of the visual system. These devices can either provide a more personalized image for people with visual defects. To other types of non-direct non-visual aid such as obstacle detectors, text readers, personal navigators...

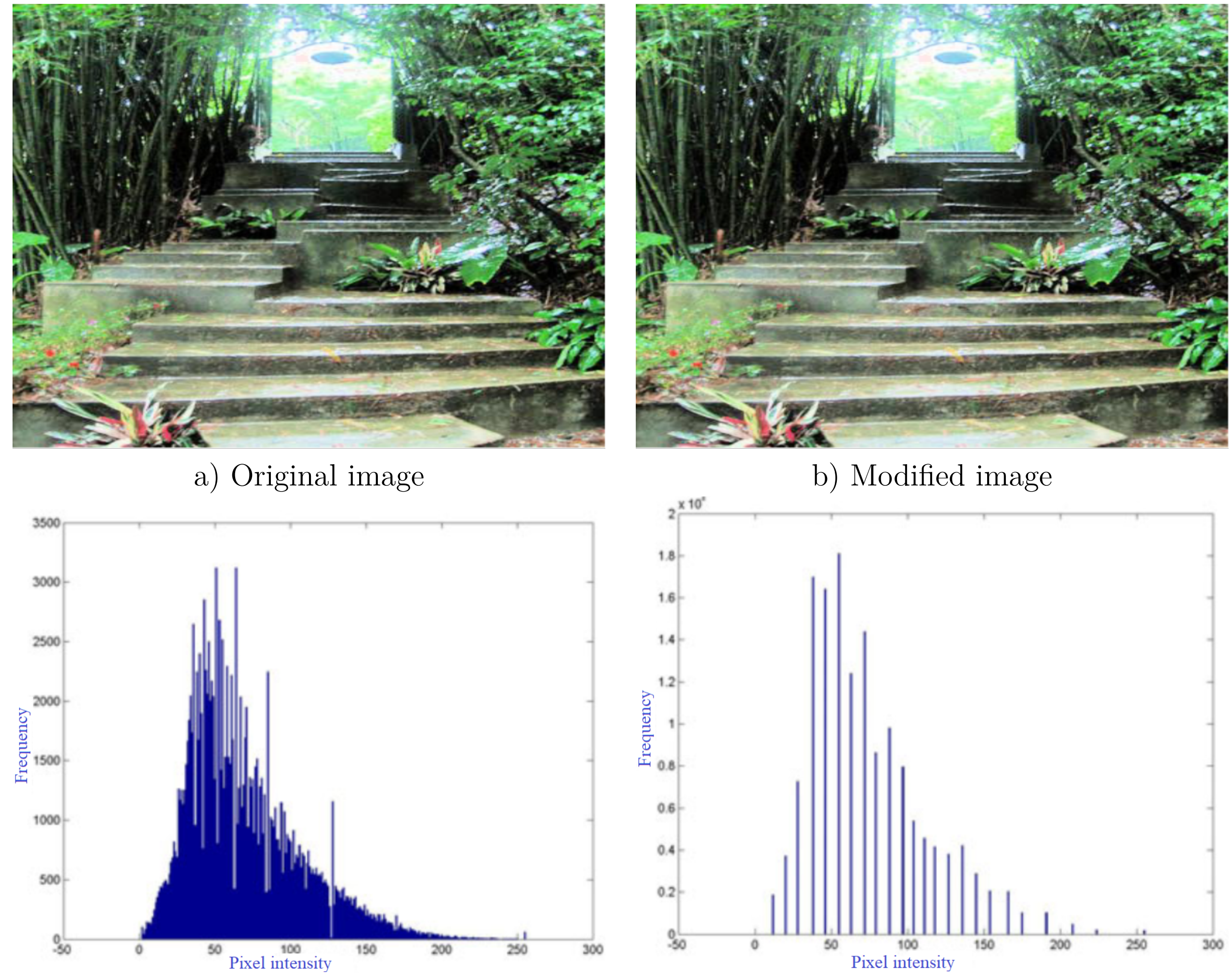
There are plenty of types of visual prostheses aimed to help people with visual impairment. We are classifying the prostheses on the type of feedback that they return to the user. There are visual prostheses with visual feedback, which means that they modify the inputs image and give back a tailored image that works best for the patient. The second type of feedback is non-visual-based feedbacks. Those types of feedback are usually either aural, haptic, or a mixture of both. Instead of a modified picture, these devices usually provide a sound or a vibration to notify the user of a potential obstacle, point of interest, or other predefined objects. Finally, prostheses based on stimulation are currently under research and look promising. The goal behind stimulation-based prostheses is to recreate a low-resolution image inside the visual field using controlled stimulation, the most common types of stimulation are retinal stimulation and intracortical stimulation. These types of stimulation can be electric or optic based. At the heart of visual prostheses is digital image processing. The obvious use is to detect and classify obstacles, detect patterns, scan texts, and many other applications. However, it can be used for people with remaining usable vision to create a more recognizable image by contrast enhancement, enhancing high-frequency elements in the image tends to exaggerate edges which makes sharp objects more visible. Some image processing enhancement techniques work better on different types of displays (LCD, CRT) [19].

2.1 Visual feedback

Visual prostheses with visual feedback are meant to improve the original image by replacing it with an image that is designed to work for specific visual impairment. For these types of prostheses to work the patient should still have some remaining usable vision and not be blind. Such people with color blindness, degenerative eye diseases, or diseases that restrict the field of vision or reduce the detail of objects. Lei et al. [20] introduced several algorithms for low vision people to compensate for the visual defects using Head Mounted Display (HMD) as seen in Figures 2.1a and 2.1b . For example, correcting for color blindness by modifying the color hue and the saturation of the video stream, which all test subjected found easier to distinguish colors. Or by enlarging the video or image to make it more readable, all testers had a speed increase in reading tests. However, their system was only helpful in very specific situations such as mentioned above and not portable. AMD cannot be simply corrected with post-processing effects since the damage is on the photoreceptors rather than the perception.

In [21], the authors created a system that displays distances on HMD and guides the users on an obstacle course as seen in Figure 2.2. The obstacle course was tested on regular sighted people. The visually impaired were tasked to search for bright objects in their vicinity. They found that the users were quickly adapted to the system. A method for improving text readability was presented in [22] by image segmentation followed by image enhancement for text using Otsu's algorithm [23], but with adaptive scalable window size. This can be helpful if implemented on a mobile HMD, which can automatically detect text, zoom and enhance it and make it more readable, but any system that can affect the field of vision should be designed to be user-friendly. Lin et al. in [24] found that HMD display generally performs better than standard CCTV to detect objects and text readability for people with low vision [24]. This is expected since in HMD the display is in direct contact with the eyes of the user.

The authors of [25] implemented a Field-programmable gate array (FPGA) design of a Gabor filter on an HMD device by enhancing edges and contrast based on individual contrast sensitivity. They reported a 25% reading speed. A system with tracks the pupil and moves the blind area in the visual field to another using Bresenham Algorithm introduced in [26]. Al-Atabany et al. [27] designed a retinal processing technique to help patients with AMD and RP called Tinted reduced outlined nature (TRON). They showed that TRON and edge overlaying are the most effective for the visually impaired subject for image and video respectively as seen in Figure 2.3, where (a) is a low contrast image, (b) extracted edges, (c) Gaussian filter, (d) anisotropic diffusion filters, (e-g) Superimposed edges from the original image, simplified image by Gaussian and anisotropic diffusion filters respectively, (h) Is the overlaid on the cartoonized image. The authors of [28] created an algorithm for parameterized active contouring solving the two main issues which are initialization and concave contouring. The method is called gradient vector flow, there were successful in finding and drawing contours on unknown edges. Using Contouring along with superimposing or enhancing edges can make objects more recognizable.



a) Original image

b) Modified image

c) Histogram of original image

d) Histogram of modified image

Figure 2.1 Example of of modifying an image to make it more visible [20], where a) represents the original image, b) is the modified image, c) is the histogram of the original image, and d) is the histogram of the modified image

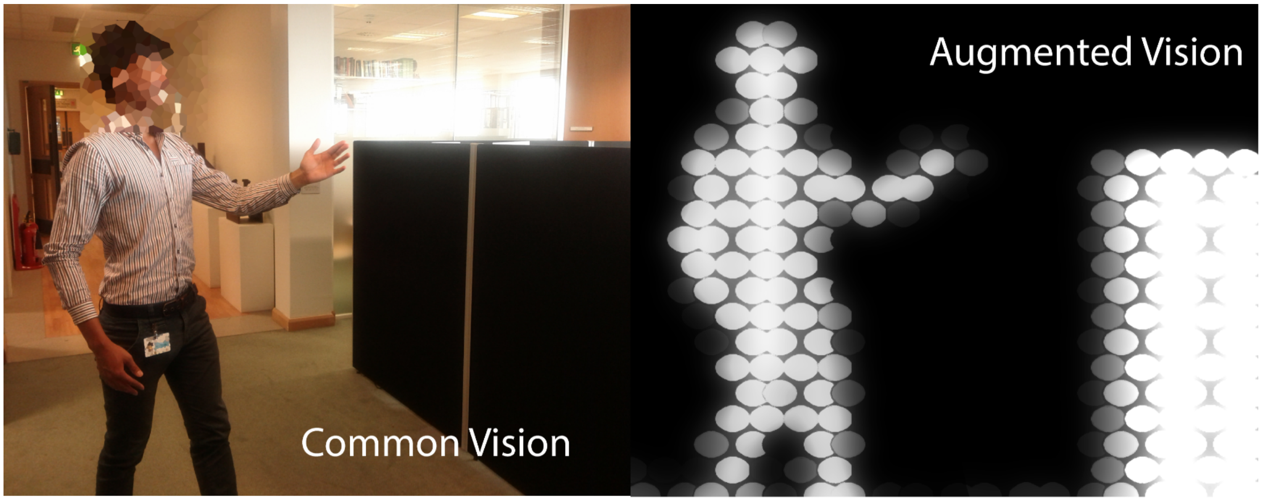


Figure 2.2 Common vision vs augmented vision [21]

2.1.1 Stimulation based feedback

Another area of research that aims for partial vision restoration for the visually impaired is using direct stimulations either to the retina or the visual cortex. Some preliminary tests show that an electrical stimulation produces a phosphene in the visual field (bright white spot), changing the frequency and the intensity of the stimulation affects the phosphene size and brightness. Traditionally, stimulation is achieved with an implanted electrode that provides an electric shock directly to the targeted cells. The stimulation needs to gradually increase to keep the same level of excitation, this can cause cell atrophy. Another drawback the size of the current electrode only allows a few dozens to be implanted inside the visual cortex or the retina.

A new approach for stimulation involves using optical stimulation to activate neurons, but since neurons are typically not sensitive to light we must inject a photosensitive protein to make it more sensitive. This hot new area of research is called optogenetics and it is a new method to allow the stimulation of the nervous cells using controlled light signals. The main advantage of stimulation-based techniques is that they can partially restore vision acuity for people with low to no remaining vision. One of the systems for retinal stimulation currently being sold and used is called Argus II from a company called Second Sight. This device provides basic stimulation on ganglion cells of the retina to create a small number of phosphenes inside the visual field. Note that the restored vision does not replace the original, usually the patient can see a pattern of objects, and upon use and training the user can learn to interpret the visible patterns.

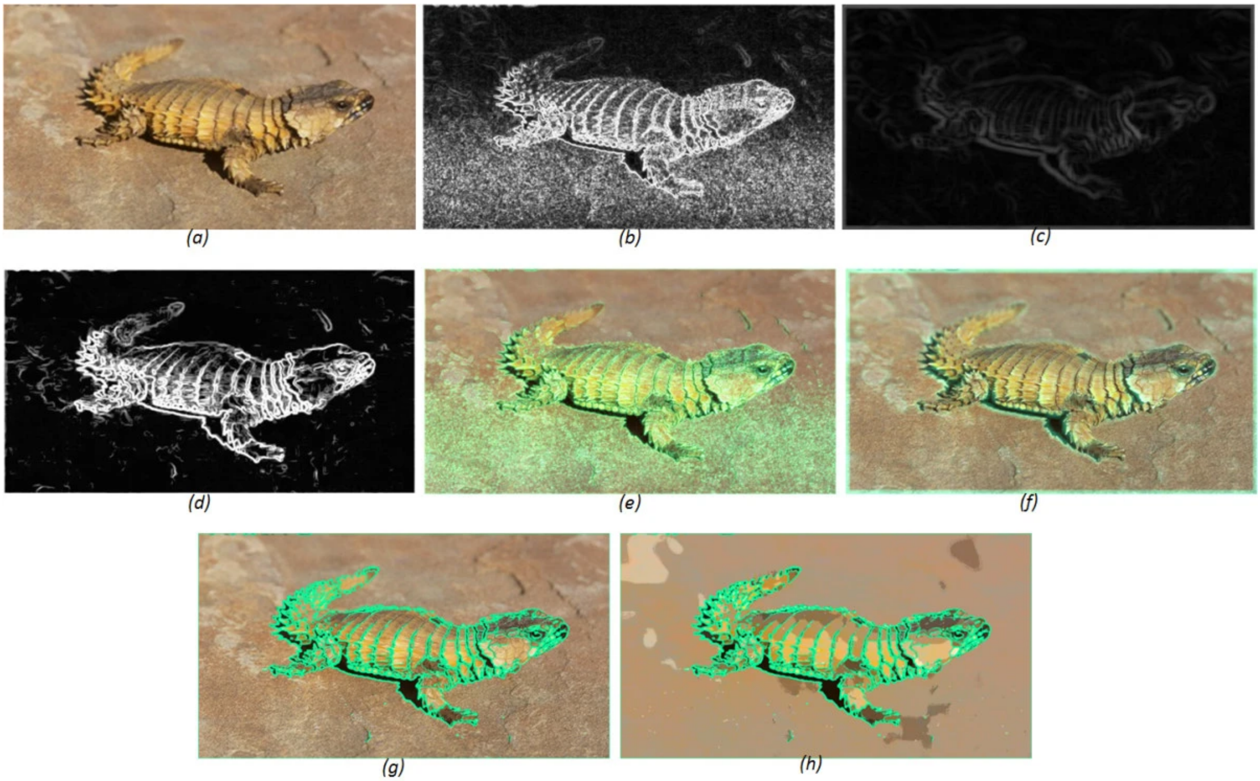


Figure 2.3 Image enhancements[27]

Retinal stimulation has been shown to provide partial vision restoration with controlled stimulation by using a small stimulation current to an electrode array planted in various locations in the visual system [29]. Some of the more successful ones include [30] which used epiretinal stimulation with 4 by 4 electrodes. Users, we able to detect the presence and absence of light, recognize basic shapes, and track bright objects in dark rooms. a clinical trial with 20 stimulation electrodes resulted in visual acuity of 20/8397 from Snellen chart with suprachoroidal stimulation with location consistency and ability to track and locate bright objects in dark rooms for blind patients [31]. There are currently two commercially available prostheses. Argus II from Second Sight used epiretinal prostheses with 60 electrodes that achieved a visual acuity of 20/1262 [32]. Alpha IMS has the highest electrode density on any commercially available prostheses with 1500 electrodes achieved acuity of 20/400 [33].

The stimulation-based approach still needs a lot of development but it is promising since it is trying to eliminate the visual defect problem by bypassing the damaged layers. The inspiration of using stimulation to bypass these layers comes from the very successful cochlear implants, which bypass the damage from the inner ear and target directly the cochlea that sends the electric signal directly to the brain. But it is important to be able to simulate the functions of layers of the brain or retina we intend to target, as many studies of awareness indicated that a significant part of the eye functions is not aware, such as pupil dilation or even object tracking [34]. Damage to parts of the visual system does not necessarily mean that the entire visual system is no longer usable, so it is reasonable to assume that we can replace the functionality of the damaged part with emulation and stimulation.

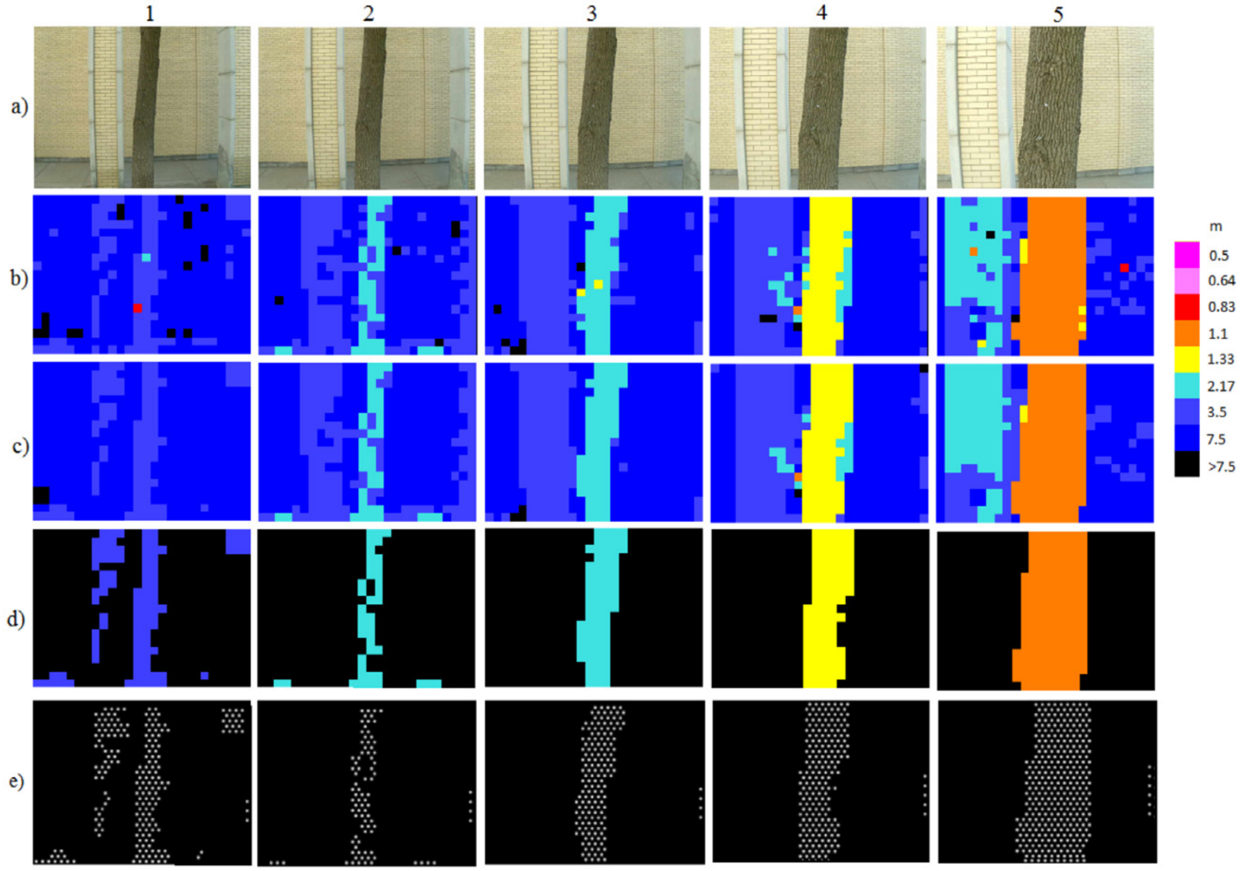


Figure 2.4 Stimulation vision based on distance[25]

The stimulation technique should work with many different types of patients. It can work for blind people to people with remaining functional vision. For example, the authors of [35] created a method to recognize objects within 7.5 m from the subject and alert the user. The authors introduced a low complexity image processing algorithm that reduces the size of an image to the available number of phosphene by using pixel averaging and contouring Figure 2.4 shows the input and output stimulation images based on distance, where (a) Input image, (b) Range imaging picture, (c) Range imaging picture after noise removal, (d)

Range imaging picture after background removal, (e) Simulated prosthetic vision, and m (meter) is the unit of distances, the columns of 1 to 5 refer to the input and result images in five consecutive viewpoints. A platform to test non-invasive retinal prosthetic based on stimulation of RCG using light was shown in [36]. That platform had a webcam connected to a PC that captures the images via Matlab, then transferred to a PC to process the image using Laplacian convolution filtering (based on spatial filtering). They delivered the light using microlenses to focus the light. In [37], the authors proposed a combination of higher resolution optics to deliver the data and the encoder to model the work of the retina layers. Their test showed that the recorded neural response from the mouse retina was similar for encoded images and normal vision images. The authors of [38, 39], created a filter that acts like the layers of the retina and simplifies the picture by removing noise and high-frequency texture as seen in Figure 2.5. This is achieved by using image encoding, scene simplification, spatial scene compression, and micro-LED pulse coding. The first trials using optogenetic for partial vision restoration with a patient suffering from RP were reported in [40]. They validated their finding with patient tests and cortical recording with and without the RP goggles.

2.1.2 Non-visual feedback

Non-visual feedback refers to visual prostheses that relay a piece of information with a sense other than vision. The most common type of non-visual feedback such as aural feedback and haptic feedback. The goal of this type of prosthesis is to detect relevant objects and inform the user with feedback that is not based on vision. These types of prostheses can be helpful for blind people or assist people with poor vision. This can be a mobile application that gives the user some aural information regarding his surrounding based on GPS data such as in [41], where the authors tested several configurations such as automatically giving information when close to points of interest or on-demand information. They found that automatic information was overwhelming to the user, it can also be misleading since the databases are not 100% accurate. Visually impaired individual mostly use their hearing and focus to localize, trying to provide additional information through the same channel might lead to confusion. The authors of [42] found similar results with their application. A hand wearable micro camera with the capability of reading text and books was shown in Figure 2.6 with audio and haptic feedbacks [43], however, such device was less preferred by testers than a regular screen reader if available as it requires less effort in tracking.

In [44], Mascetti et al. designed an algorithm that can detect traffic signals using a smartphone camera (shown in Figure 2.7), then the smartphone would respond by a tactile re-

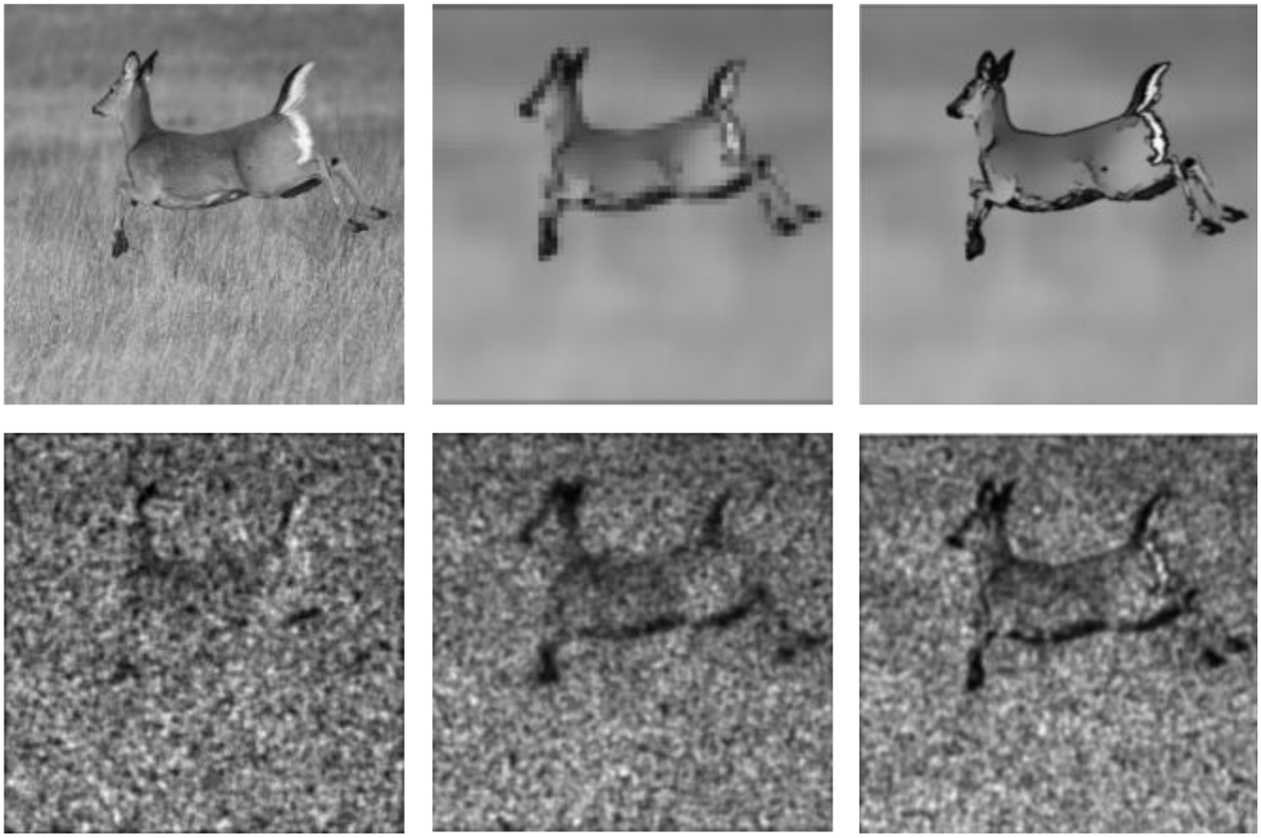


Figure 2.5 Above: original image, cartoonized 64*64, and cartoonized 256*256, Below: correspondent images with 30% of random cone cell atrophy [38,39]

sponse (vibration) or by aural responses. They were capable of correctly detecting traffic signals under different light conditions. The visually impaired that used the app stated that the aural feedback felt sometimes distracting or was ignored. The tactile feedback on the other hand was distinguishable and clear. This is because the visually impaired already uses their hearing to localize and identify their surrounding (such as the sound of cars, fountains, traffic, machinery ...). Using aural feedback for any system to provide additional information is distracting since we are adding more information for the brain to process while already occupied with other tasks.

In [45], the authors designed a system of visual aids including a simulated retinal stimulator alongside a vest with vibration motor feedback, they found that people using these systems had lower incidents of collisions with obstacles than people using the white cane, but they were also significantly slower. This means that even a device basic as a white cane gives a better feeling for spatial awareness than aural aid, even though that they stumbled on more obstacles. This is because the visually impaired used the white cane as an extension of the

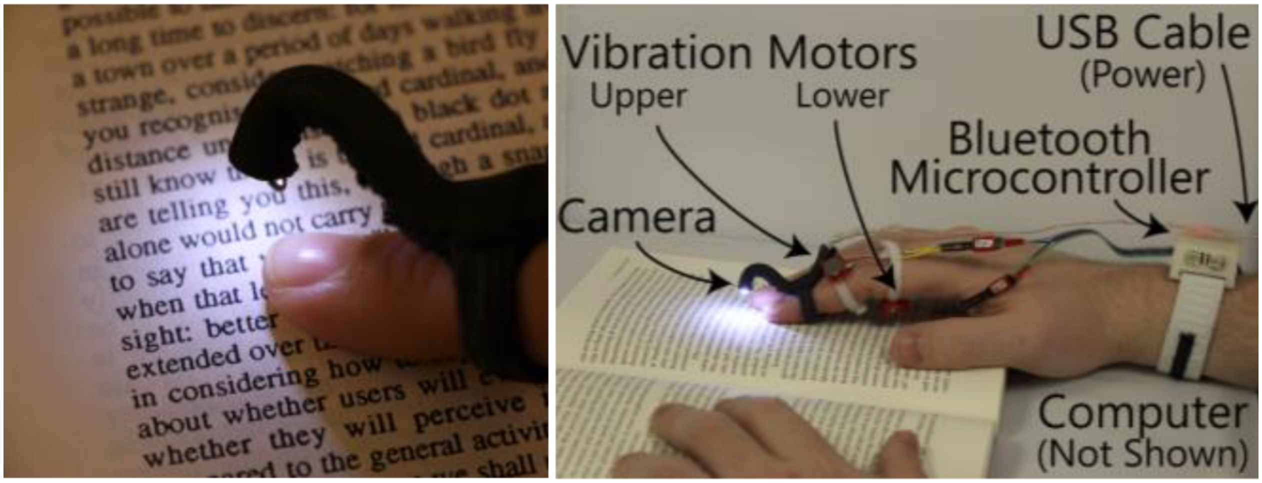


Figure 2.6 Hand wearable camera [43]

tactile feeling of their hands [46]. Doing so allows them to easily create a mental image of their surroundings. Besides that, tactile feedback provided by the white can feel more real than any aural feedback from a device. Another consideration is that the human tactile sense tends to be more urgent than the aural because it indicates direct contact. Another interesting approach is to use 3d virtual environment to introduce the visually impaired to a brand new location. An example of this can be seen in [47] where the authors trained the visually impaired to move in a 3d virtual space using a Wiimote controller. They found that some people responded well to navigating a virtual environment using Wiimote, but others found the experience confusing and disorienting. The idea is interesting; but it is hard for a blind individual to estimate how much he/she moved in virtual space using only a controller since an essential part of moving in real-world or even in virtual is to localize, that is essentially missing in their experiment.

The classification of these visual prostheses by the type of feedback helps establish the targeted audience of each type of prostheses. However, some work has been done on improving prostheses or technology that can help visual prostheses but do not function on their own. These approaches and techniques will be classified as others.

2.1.3 Other approaches to aid the visually impaired

It can be helpful to have a system that can localize an object or an obstacle at a point in space and can notify the visually impaired person of potential hazards in their vicinity like a pothole or road construction. There are a variety of approaches designed to aid the

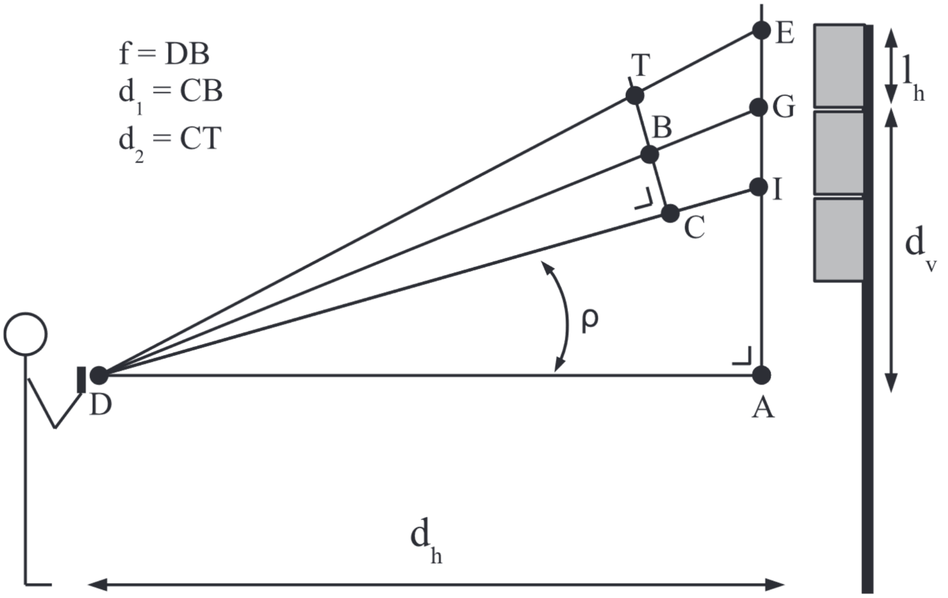


Figure 2.7 Mobile system to detect traffic signals [44]

visually impaired this can be a system to detect tactile pavement surfaces with high accuracy (up to 93% accuracy) based on Grey Level Covariance Matrix (GLCM) and decision tree as in [48]. Or an approach of traversable area detection on street with audio feedback based on possible routes such as the work presented in [49]. In [50], the authors implemented a system of localization, obstacle detection, and obstacle classification using optical flow. Their system was able to detect doors and stairwells most of the time, using Euclidean distance along with Connected Component Labeling (CCL) algorithm for obstacle detection, CCL with color information for obstacle classification, Hungarian algorithm for camera tracking. Such an approach can help detect any dangerous obstacle but didn't provide how to use the information to be useful for the visually impaired.

The aid to the impaired individual doesn't have to be locked to navigation only. As an example, Mittal et al. created three modules to help detect currency [51]. The first one correctly detects color, the second detect the value of an Indian currency, and the third detect and extract text from paper. Each had of these modules had over 80% accuracy. This can be helpful in currencies with a lack of Braille marking. The authors of [52] created a simulator for several types of visual defects such as diabetic retinopathy, macular degeneration, hemianopsia, retinitis pigmentosa. Their simulators found that a significant reduction in reading speed for central vision loss and a huge reduction in reading speed for peripheral vision loss. Such models can be helpful to model and test visual protheses' effect on several types of visual impairment. However, the accuracy and reliability of the simulator remain to be proven.

Algorithms for detecting image disparity in real-time can be very helpful in systems that are trying to detect obstacles. An example can be seen in [53], where the authors implemented a disparity image calculator using FPGA that as the sum of absolute differences and the zero-mean sum of absolute differences matrices. Their implementation manages to compute both matrices at 640*360 at 30fps. The disparity image can be used to estimate distances. Another use for disparity can be depth detection. This can be beneficial in systems that read sign language [54]. We can also use depth sensors to help with 3d image recognition by training a decision forest classifier for fast detection and recognition of joints and limbs based on distance data which can help with human-computer interaction [55]. Tracking pedestrians in real-time can help with collision avoidance [56]. Using virtual reality (VR) or augmented reality (AR) can be very helpful for low vision by identifying obstacles and pointing them out to avoid collisions. AR can be used to directly impose edge and contrast enhancement to make objects more visible and more recognizable. This is helpful for visually impaired people with residual vision.

The brain has incredible plasticity, experiment on people with recent eye damage tend to use the part of the brain traditionally used to decipher visual information into extracting information from sounds [57]. This method is called echolocation, and it is very similar what bats do to detect objects. Since our brain can pick up cues from a time delay of an echo, changes in the spectrum of sound, differences in sounds coming to the two ears, and differences in the reverberation pattern. While this talent is found in a regularly sighted individual, it is significantly more developed in a visually impaired individual. Virtual space has been used to teach a blind subject to navigate a real space using aural cues [58]. Virtual sounds show great advantages for guidance. This was tested on people who are born blind, recently blind, and blindfolded individuals to navigate a corridor while adding some familiar sounds (footsteps, elevator, etc.). These sounds were meant to help the individual to locate himself with respect to the sound source. Footstep noise and reverberation noise (with finger snap) were found to be helpful to construct a mental image of the surrounding without being there. Exploiting the brain capability and elasticity to build a customized prosthesis have large potentials. Techniques such as countering, echolocation, object recognition, and 3d mapping can help make prostheses more reliable.

2.2 Facial expression recognition

Due to the limitation of the currently available retinal stimulation techniques, we can propose using artificial intelligence to understand the context of the scene. We chose to focus on facial expression as it is a critical aspect of daily communication. There are a plethora of facial expression recognition techniques, and these can be divided by feature types, such as geometric, texture-based, or neural networks.

2.2.1 Geometric features

Geometric features use face geometry to compute distances between facial points such as the distance between the mouth and eyes. These distances are used to train a machine learning model, such as support vector machines, decision trees, or neural networks. A common method for geometric feature extraction is the active shape model which detects several landmark points around the face. From these points, a custom descriptor is calculated and used for training and then inference. Some examples can be seen in [59] which used ASM for feature extraction and SVM for recognition, the authors of [60] used a similar approach but further improved it by tuning parameters using genetic algorithms to reduce feature set and improve accuracy. Geometric features' main advantage is low memory and computational requirement and good accuracy for new data.

2.2.2 Texture features

Texture features use pixel intensity levels to detect small changes unique to specific facial expressions. Some of the popular feature extraction techniques are Gabor wavelets, local binary patterns, and the histogram of Gaussian. Texture features need pre-processing to work such as denoising and intensity normalization as these features are very sensitive to illumination variation. An example of this approach is in [61] where the author used local binary patterns to extract features around the nose and mouth and used that information to train and detect facial expression. These approaches can be very accurate, but it tends to be computationally expensive and requires lots of memory. They are best used for familiar data.

2.2.3 Deep neural networks

Neural networks were traditionally used just for classification, recent advancements in both model and hardware allowed neural networks to take over the role of feature extraction, as certain layers can take the role of feature extraction and data selection in the form of convolution layer and max-pooling layers respectively. Recurrent layers can help with detecting temporal features from data. Finally, fully connected layers are used for classification. Some of the earlier approaches of using deep neural networks for both feature extraction and classification were in [62]. Since then many architectures have been popular in facial expression recognition such as ResNet 18/34/50, and VGG such as [63]. Typically neural networks are prone to overfitting the data (where the model learns the data too closely and cannot generalize a solution that works on newer data), but that can be solved in dropout of normalization layers. Neural networks tend to require a large amount of data, but that can be solved with data augmentation. Neural networks can be implemented as a wide computational graph which makes it ideal for Graphical Processing Unit (GPU) acceleration or a vector processor. This allowed neural networks to run efficiently on a variety of modern hardware. Note that additional approaches are discussed in article 2 of this thesis.

2.3 The future of visual prostheses

From all the approaches presented above, we can notice that many visual prostheses or system attempting to help the visually impaired does not perform as hoped. Only a few tend to be successful at what they aim to do, even fewer end up as actual products. To design a working visual prosthesis, it is essential first to understand why some fell short and others succeeded. But first, it is essential to understand the difficulties that the visually impaired face and how they adapt around them. This will make it easier to predict how the patient would react to a prosthesis.

Spatial learning is the process in which we -as humans- build a cognitive local map of the area. To get familiar with a new area, first the blind usually asks for a verbal description of the area, then try to get familiar in that area with a friend or family [10]. The blind then uses these descriptions to build a mental map of the local area. Other senses can help as well, such as smell to identify coffee shops or audio such as the sound of a fountain... Talking GPS can help with learning a new environment. However, this is not an easy task, as snow can muffle the sounds and rain amplify it, thus altering the aural description of the area and making navigation much harder. Learning new places is always a challenge as it might feel unsafe. On the other hand, local interactions such as smell, audio, or hazards such as construction might lead the blind to gather more information about a new area.

One approach to designing a visual prosthesis with visual feedback is to use mobile HMD along with scene enhancement to technique to target specific and personalized visual impairment, like a color space replacement for the color blind, edge enhancement, and contouring to make images or videos more visible. Text detection and automatic zooming and enhancement will make reading signs and text much better. Such approaches can help people with low vision. The difficult part is designing these in a mobile system that is fast, accurate, and simple to use and to customize based on personal preferences. Recent developments in Mobile VR and AR seem to be the best approach at this moment. The previous generations of HMD where bulky with slow response times, which can lead to headaches, and low frame-rates, which makes movements appears choppy and unnatural. Smartphones based VR seems to be a good way to develop vision algorithms since they have an AMOLED display that solves the problem of latency, persistence, and new fast SoC have large processing capability not only for general-purpose computing but also for graphics and imaging with dedicated hardware such as GPU and ISP, such as Occulus Quest 2 from Meta which a uses a smartphone SoC that can run virtual reality software for 2-3 hours. Modern smartphones even come equipped with a camera that can be used to capture an image and display it automatically. The downsides that most smartphones do not capture a stereoscopic vision, and second most smartphone cameras lenses are not wide enough to capture the field of view of the human vision, this means that the scale of object and field of view might not be correct. Recent developments in AR such as with Microsoft HoloLens 2 might solve this issue. However, using an HMD for an extended period of time can lead to fatigue and strain, so it is important to have a lightweight and comfortable system that does not lead to fatigue after a short period of use.

When it comes to no visual feedback, aural feedback is sometimes helpful, but in most cases, they tend to be obtrusive, confusing even in some cases overwhelming, in other cases aural feedback simply underperform compared to the haptic feedback. That is why we believe that the best approach in designing a practical prosthesis is with tactile controls, like pressing a button, and any feedback should also be tactile like motor vibration if the user has arrived at his destination or to grab his attention for a point of interest. Because the visually impaired would already be using their other senses for localization by hearing the traffic to determine if they are near an intersection, smell to determine if they are near a restaurant, coffee shop, or any area with a distinct smell. This localization helps the visually impaired build a local mental map of the area. Any visual prosthesis must be designed to be either hands-free or give a sensation like something the visually impaired is familiar with such as a white cane or walking dog. These protheses should be incredibly easy to use and be instantaneously familiar, comfortable, and safe. Complicated or obtrusive devices can be abandoned quickly.

While the first two approaches (scene alteration and aural feedback) should help people to mitigate the problem or help with navigation, they can be relatively quick and inexpensive to build since they use mass-produced hardware. A stimulation prosthesis is still a long way from becoming a reality since it faces many technical challenges such as optimizing stimulation delivery, adapting retinal processing to form a comprehensible image within the visual field [64], and legal challenges in form of government regulations. The biggest advantage of stimulation-based retina processing is that it attempts to treat the problem rather than mitigate it. This approach requires directly affecting either the human eye, retina, or visual cortex, hence, extensive animal testing should be done first, before testing on humans. One of the main technical difficulties in this approach is that any input signal should be encoded and modulated before stimulating the targeted organ. For example, if we want to design a retinal prosthesis that targets the ganglion cells, we must encode the image with the processing model of the different retinal layers to get the correct image at the retinal ganglion cells.

Another option is to skip the retina altogether and directly stimulate the visual cortex. Doing so with direct stimulations is orders of magnitude more complex since it requires encoding the entire visual pathway up to the visual cortex and it is significantly more dangerous as it requires exposure between the stimulation device and some of the brain tissues which can cause severe complications. It is suggested that the basic scheme for a brain method to identify an object is equivalent to a simple neural network, with multiple input neurons and a single output neuron [12]. This structure is equivalent to a simple artificial neural network painting a hyperplane that separates the category of the object, however, the brain neurons tend to be much more parallel than any artificial neural network we have today. While direct cortical stimulation is possible, it is much more dangerous since it requires exposing part of the brain for stimulations and requires additional pre-processing that occurs in the optical nerve and LGN, so it is better to use retinal stimulation when possible. There have been a limited number of stimulators that have begun limited human trials, like the Argus line of retinal stimulators from Second Sight Medical Products [32]. Such trials usually have a handful or dozens of participants, and the devices are prototypes that are not necessarily meant as long-term solutions [40].

2.4 Vision and contributions

In the previous chapter, we presented both the long-term and short-term objectives of this thesis. Here we clarify our vision.

There are plenty of limitations for creating prostheses for the visually impaired suffering from AMD. Optogenetic stimulation has proven itself capable of creating prostheses for RP [40], and MEA has proven itself to be able to help with restoring some visual acuity in the vision [32]. We believe that optogenetic can help with creating the next generation of visual prostheses for people suffering from AMD.

Performing constant optogenetic retinal stimulation on the RGC layer in the fovea for a prosthesis require finding a simple model to emulate the major functionality of all previous layers. Such a system also has to be able to run in real-time matching the retinal processing speed occurring in the major pathway in the retina. The model we created for these next-generation prostheses is discussed in Chapter 3. Note that this model is not meant to be a full recreation of the retina processing in the macula, but a close approximation of the major pathways that form the central vision. The contribution of the proposed model is that it takes into account the temporal aspect of the retina, something that is traditionally ignored in prostheses due to its difficulty.

The technology of creating high-resolution optogenetic stimulation is also limited by the technology of delivering the optogenetic signal to the correct region of the retina. We envision wearable glasses that use a matrix of micro-LEDs assembled on a custom power delivery and control chip. The microLED design and the chip design are published in [65,66]. Since these chips are meant as a proof of concept, it is designed as an 8x8 LED matrix design. This limits how many pixels from our model we can deliver, thus we created a system that can read facial expressions. We chose facial expressions as they are an essential means of communication and social interaction.

Being able to read facial expressions and replace the facial expression with an optogenetic representation is critical for human interaction, and should be done reliably and in real-time under various poses. In Chapter 4 we present our approach for creating a system of real-time facial expression recognition that can detect facial expressions in real-time and under different poses. We also show an example of the system running our algorithm in real-time and outputting to an off-the-shelf 8x8 LED matrix with the detected facial expression.

CHAPTER 3 ARTICLE 1: SPATIO-TEMPORAL RETINAL PROCESSING PLATFORM FOR OPTOGENETIC WEARABLE MICROSTIMULATOR TARGETING AGE-RELATED MACULAR DEGENERATION

Nizar El Zarif[1], Leila Montazeri[1], François Leduc-Primeau[1], and Mohamad Sawan[1,2]
[1]Polystim Neurotech Lab, Department of Electrical Engineering, Polytechnique Montreal
[2]CenBRAIN, School of Engineering, Westlake University, Hangzhou, Zhejiang, China,
310024, IEEE Access, submitted on October 20th,2021.

This chapter presents the first objective of the thesis, which is to create an image processing model of the macula that works with high-resolution stimulation. The model focuses on the two major pathways in the macula (spatial and temporal pathways) and adds a simple light adaption model. The model starts at the photoreceptors layer up to the retinal ganglion cells and adds to it a radial shift to emulate the ring structure of the retinal all in real-time. All these steps are necessary to able to drive optogenetic-based prostheses into the retina. Most work in the literature only uses spatial filters. As far as we know, this is the first work to add a temporal element to the filter, light adaptation, compensation for the ring structure of the macula. A preliminary version of this work has been published in EMBC 2018 [67]. This version is to be submitted soon to IEEE access.

3.1 Abstract

We present the Multiple Spatial-Temporal Model (MSTM), a computationally efficient model that can perform real-time image processing equivalent to midget cell spatio-temporal retinal processing with geometric compensation to emulate the midget pathway in the human macula. This simplified model is intended to deliver customized signals to a wearable optogenetic stimulator system. The model emulates retinal processing occurring in major pathways in the macula such as spatial, temporal filtering, and light adaptation while compensating for the radial shift of ganglion in the human retina. The proposed model is efficient enough to run on mobile hardware with a battery-powered device in real-time and it is ideal for high-resolution optogenetic stimulation devices that target the ganglion cells in the retina. The implemented model achieves up to 98 images processed per second on Raspberry Pi 4.

3.2 Introduction

Nowadays, the number of people around the world suffering from visual impairment is estimated to be around 285 million [1]. Age-related macular degeneration (AMD) is the leading cause of blindness in developed countries where it counts for around 6.6% of all cases. This number is expected to rise as people are living longer, and this disease primarily affects the elderly [2]. In Europe, North America, and Australia, over 66% of adults over 80 years old have a form of AMD [68].

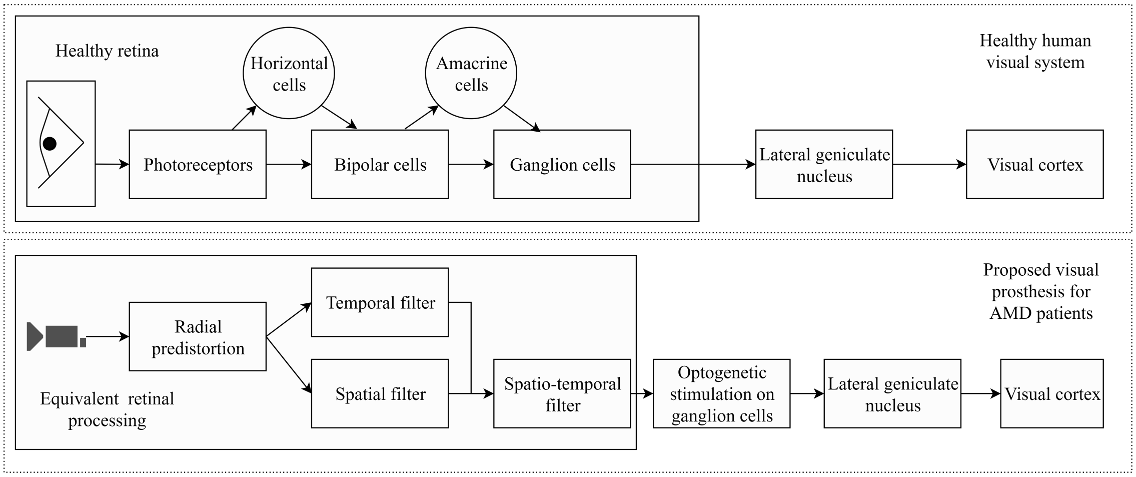

Healthy retina
Horizontal cells
Amacrine cells
Healthy human visual system
Photoreceptors
Bipolar cells
Ganglion cells
Lateral geniculate nucleus
Visual cortex
Temporal filter
Radial predistortion
Proposed visual prosthesis for AMD patients
Spatial filter
Spatio-temporal filter
Optogenetic stimulation on ganglion cells
Lateral geniculate nucleus
Visual cortex
Equivalent retinal processing

Figure 3.1 Simplified arrangement of different layers in the retina (top) vs proposed prosthesis for AMD treatment

The top image shows the retinal processing flow for a healthy retina, the bottom shows our proposed optogenetic stimulation system aims to bypass non-functioning layers and directly stimulate the RGC before propagating the visual information to the next layers.

AMD is characterized by a progressive loss of visual acuity in the central vision until it goes completely dark. At this point, the person is considered blind even if some peripheral vision remains. AMD is one type of progressive disease that affects the retina, another well-known type is retinitis pigmentosa (RP) which causes degeneration of rods in the retina and affects both peripheral vision and field of view, which makes object identification much more difficult [21], this is because rods are extremely sensitive to light even as low as a single photon [69], hence important for measuring fine luminance variation in the scene.

The human retina has 5 different layers. Photoreceptors receive the light and send it to bipolar cells. The bipolar cells then invert the signal and relay it to retinal ganglion cells (RGC) which perform weighted sum averaging [12]. The horizontal cells perform basic edge detection, while amacrine cells are essential for motion detection. It is estimated that there are at least 14 types of bipolar cells [4], and more than 30 types of RGC [5]. In the case of AMD, the cones in the fovea are progressively being eroded due to waste accumulation in the capillaries under the photoreceptors. The other layers of the retina remain functional but inactive since there is no stimulus from the cones to cause activity in Phase 1 of the retinal degeneration. Activating these layers with direct stimulation such as a current via Micro-electrode Array (MEA) has been shown to produce phosphene inside the visual field.

There is no reliable medical treatment for AMD. Stem cells transplantation [17], anti-vascular endothelial growth factor (VEGF), and gene therapy can slow down the progression of AMD but do not treat it [19].

As the retinal degenerative disease progress, Phase 2 of retinal remodeling occurs in the retina due to photoreceptor loss accompanied by glial remodeling of the outer nuclear layer, the third phase is characterized by neuronal morphologic change and migration, network rewiring resulting in altered ganglion cell connection [70]. This can inhibit rescue attempts to restore full vision where targeting bipolar or amacrine cells. Phase 3 is the final phase where most layers of the retina have been damaged with a significant number of RGC remaining with altered connections [71].

To create a reliable visual prosthetic, we must have a relatively accurate and efficient model of the visual retina. That model is then used to supplement any missing information via stimulation. Even if the stimulation is missing some information we can count on the brain's remarkable adaptability and plasticity to fill in some of the gaps [72]. The brain can even repurpose part of itself to perform other tasks when needed. For example, the brain can transform the visual processing part into hearing and other sensory processing after a sudden vision loss [57].

To recreate a theoretical 20/20 vision from the Snellen chart we should have an accuracy of 1 arc minute of stimulation per degree. That means 1 electrode every arc minute of 1° for at least 20°, that is 1200 electrodes per axis, or 1200^2 in total for two-axis or 1.44 million electrodes in the macula, which has a diameter of 5.5mm [73, 74]. This is not possible for electrodes in today's technology. However, partial vision restoration is already available, and several electrical stimulation techniques have been introduced. The highest density for retinal stimulation achieved 1500 electrodes on an $11^\circ \times 11^\circ$ [33] which allowed the users to read large letters and locate bright objects in dark rooms with an effective acuity of 20/400. The volunteers reported a vision similar to a blurred black and white television image.

The proposed model focuses on creating a custom image optimized for optogenetic stimulation, which has shown the ability to restore some visual acuity inside the retina without the need to implant MEA [32, 37]. Optogenetic is the process of transforming normally non-light-sensing cells such as RGC or bipolar cells into light-sensitive cells by infecting the retina with microbial or viral opsins which are harmless light-sensitive proteins. The benefits of this approach over traditional implants are enormous. While regular implants are effective and provide real benefits for compatible patients [75], they can suffer from biocompatibility, trauma, degradation of electrodes over time, might require medical intervention, and can cause infection during their use even with commercially approved prostheses such as in Argus II by second sight [32]. Optogenetic avoids most of these problems, while also having the ability to operate at a higher spatial resolution as it can activate one neuron at a time, unlike electrodes which can activate hundreds or even thousands due to large surface and the high current required. An earlier version of optogenetic stimulation used a type of blue light that interacts with Channelrhodopsin. Recent advancements allowed the use of red light, which has lower frequency and energy with different opsins. This makes it safer to use on tissues or inside the eye as the heat generated from high-frequency sources can damage the tissues over time [76]. The viral delivery of opsins allows selective targeting of a specific type of neural cells for example selective targeting for ON cell and OFF cells for depolarization and inhibition with specific opsins [77]. Another alternative is to use animal opsins that work with a larger frequency range and lower energy requirement [78]. To stimulate RGC for example. we must compensate for the retinal processing for all previous layers.

This paper aims to recreate a simplified midget retinal processing model up till the RGC as shown in Figure 3.1. Another approach was to target ON-bipolar cells due to their simplicity and reduced processing requirements. However, targeting RGC has several benefits over bipolar. In vivo tests on rats showed that targeting retinal ganglion cells under the same conditions resulted in a significantly better light response and more visual function on rats when compared to bipolar ON cells [79]. The light sensitivity in rats is similar to that in humans which makes them ideal candidates to test stimulation approaches [80]. Retinal remodeling in phase 3 makes stimulating bipolar cells nearly impossible, however, a good portion of the RGC is viable for partial vision restoration in phase 3.

The focus here on midget pathways instead of the parasol is to target AMD without affecting peripheral vision. The LED circuits and controls are described in our previous work [64–66] should work well with our simplified model and previously designed algorithm to detect facial expressions and replace it with a representation of the scene [81].

3.2.1 Previous Work

There has been some limited amount of work that replicates the entire function of the macula in real-time. Authors in [26] designed a system that tracks the pupil and shifts the blind pixels from the visual field to nearby pixels. Our approach focuses on shifting pixels to accommodate for the image distortion due to the ring structure of the RGC layer. In [27], authors implemented a scene simplification and edge overlaying for people with retinal diseases to improve the visibility of photos displayed on monitors without any implants. Our model is meant to complement stimulation hardware, and aid in partial vision restoration in real-time. Authors in [36] implemented a system to drive a LED-based device that can be used in an optogenetic platform. Reported work in [82] presented an image processing technique to improve the appearance of objects by increasing contrast and removing details from images. Their work was focused on images while our approach is designed to work in real-time with video. The authors of [83] introduced wearable glasses that used a cartoonization algorithm optimized for RP that can process video up to 30 FPS on Raspberry Pi 3. Finally, in [84], a system was designed for implant tracking and used radial distortion to emulate retinal processing in bipolar cells. Their test is done using a pre-stored video rather than a live capture. This work focus on real-time processing on embedded systems with reliable outputs. We are also targeting the RGC, which is more complex than bipolar cells. We also tested on both live video to ensure real-time performance and stored video to measure maximum possible performance.

The proposed model has three main contributions. The first is that our proposed model simplifies the spatial and temporal processing in the macula and skips fewer contributing pathways in the retina, if we combine that with an optogenetic stimulation it should be able to restore some visual acuity in the central vision without affecting peripheral vision. Most previous work only emulates spatial processing, leaving out temporal processing, and other processing pathways in the macula. Secondly, the model performs scene simplification which is a critical step in any stimulation approach [9]. Another strong point over other works is that this work is both fast and scalable and can run on a low-powered and low-cost mobile processor in real-time while providing consistent results. This model can then be used to drive an image to a wearable micro-stimulator that can produce Phosphene in the human visual field. Thus helping in partial vision restoration.

The rest of the paper is organized as follows. Section 3.3 describes the different models of the retina. Section 3.4 describes the processing algorithm and our model tailored for use with a high-resolution stimulation device. We describe our software implementation and performance considerations in Section 3.5, followed by the testing setup and results in Section 3.6. Finally, the conclusion and planned future work are presented in Section 3.7.

3.3 Different Retinal Models

A major difficulty of emulating the RGC is that there are at least 20 types of RGCs [85], some recent studies even suggest 32 or more in the mouse retina [86]. Midget channels in the RGC are represented with a connection between a single photoreceptor connected to a single bipolar cell which is in turn connected to a ganglion cell. However, ganglion cells can be connected to several bipolar or amacrine cells and these connections can either help in depolarization or with hyper-polarization. RGCs also exhibit additional weak properties, for instance, some cells are very sensitive to a specific direction [87], at the same time are weakly sensitive to other forms of visual stimulus [72]. Under some conditions, even some ON RGC can act as a weak OFF [88]. The type of RGC can be deduced from size and shape, hence, its functionality [89]. An earlier model estimated the receptive field of RGC in the retina as the difference of gaussian (DoG) [90, 91]. Lowe proved that the DoG filter can be approximated as Laplacian of Gaussian (LoG), which reduces the computation time [92, 93].

There are plenty of different types of retinal models that aim to replicate the functionality of the retina. Some use biomedical and biochemical models that can estimate the output of neural spikes and voltages based on the measurement of ion concentration in the retina, such as sodium, potassium, and glutamate. From the concentration, we can figure out if the membrane potential is enough to cause hyperpolarization or inhibition of neural cells in the retina. There are also functional models where the goal is to emulate functionality in hardware and software in more computationally efficient ways.

3.3.1 Physical models

A physical model of the retina is obtained by measuring the output spikes from a set of input and fitting the data into a function, the measurement of output spikes is done with MEA [94]. This measurement can be taken in vivo or in vitro. The authors in [95] found that in macaque monkeys, L cones act as a simple Gaussian curve while M cones act as two curves. One is acting as a weak luminance band-pass filter and the other as a chrominance low-pass filter. In [96], the authors created a linear-non-linear model with Poisson spiking based on recording from a macaque monkey, their model also took into account the Spatio-temporal effect for nearby cells in the retina. Light adaptation to background visual stimuli was discussed in [97] which was found to be an essential part of the temporal response in the retina. The authors showed that a functional image can be recreated from neural recording in the retina [98].

3.3.2 Functional models

The functional models are more concerned with recreating some of the functionalities of human vision with hardware or software models such as [84]. This approach has three different purposes. It can help with creating a low-level image processing algorithm that enhances images for patients [99]. It can be used to visualize the functionality of different parts of the human visual system. Furthermore, it can be used to drive visual prostheses.

Pre-distortion of images makes objects and images of objects more visible [100]. There are also many approaches to aid the visually impaired by image enhancement: such as contrast enhancement can aid with identifying objects, enhancing high-frequency elements in the image tends to exaggerate edges which makes sharp objects more visible, same for background attenuation and scene simplification, however, these approaches can only work on displays and will not work on all visual types of visual impairment [101]. Velazquez in [52] created a simulation model of different types of visual defects, such models can help with testing new visual compensation algorithms for people with visual defects. The authors in [102] created an event-based real-time system that emulates the output of RGC cells to an 8 by 8 light-emitting diode (LED) matrix. In [103], the authors created an artificial neural network to generate retina images based on *pulse2perception* to create a retinal model for patients with RP. pulse2perception is a computationally expensive framework that emulates retinal processing in peripheral vision

3.4 Proposed Model

Most of the available visual prostheses do not consider the temporal aspect for retinal processing. This paper accounts for both the spatial and temporal aspects of this work by an LoG filter and our multi-frame temporal filter. The proposed algorithm is summarized as follows:

1. Capture video frames using camera and convert image from colored to grayscale
2. Run radial shift to compensation for ring arrangement in the RGC layer
3. Spatial filter is implemented with an LoG filter.
4. Run our new computationally efficient multi-frame temporal filter
5. The spatial and temporal filters are combined to create equivalent Spatio-temporal filtering in the retina.

The algorithm we described above should compensate for retinal processing for the midget pathway in the macula. We call this approach the multiple spatial-temporal model (MSTM). Figure 3.2 shows the high-level steps of the proposed model. The goal of this model is to represent simple and fast implantation of the major pathway in the macula, which should be enough to restore partial visual acuity when used with a stimulation device. The high spatial and temporal resolution achievable with this model makes it ideal to drive optogenetic stimulation in the macula. We omit minor midget pathways and weak properties as their behavior is unpredictable and non-uniform. The steps in MSTM are described in the following subsections.

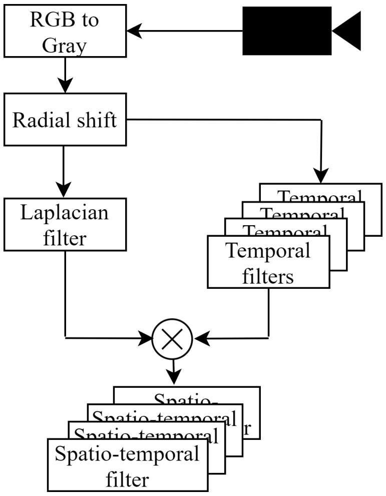


Figure 3.2 The high steps of MSTM

The high steps of MSTM. First we get the camera frames and convert them from color to gray, then we perform the radial shift. The output is then passed by a spatial and multiple temporal filters. The output of the spatial filter is then multiplied by the output of every temporal filter. The result should indicate which pixel should be used in optogenetic stimulation to have spatial and temporal filtering similar to retinal processing in the human retina.

3.4.1 Capture and convert

For real-time image processing for video frames, we first capture colored video frames from the camera and place their output into a memory buffer. We then convert each colored frame from the captured three color channels (red, green, and blue) into a single channel grayscale frame. This reduces computation time and makes the next steps faster and easier. Using a grayscale image is a logical step as most stimulation devices today can only provide output into a single channel and with a small number of intensity levels. Converting the image from color to grayscale is done by using the widely used transformation equation:

$$I(i,j) = 0.299 \times I_r(i,j) + 0.587 \times I_g(i,j) + 0.114 \times I_b(i,j), \tag{3.1}$$

where i and j represent the horizontal and vertical position respectively, I is the grayscale image, I_r is the red channel, I_b is the blue channel and I_g is the green channel from the image.

3.4.2 Radial shifting and circular interpolation

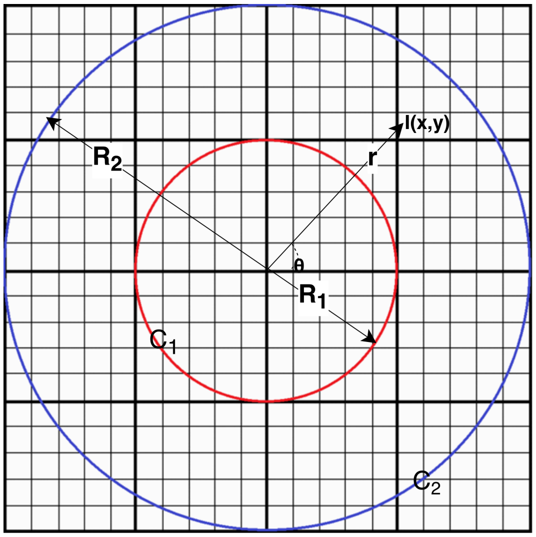


Figure 3.3 Radial shift

Radial shift, all the pixels inside the outer circle C_2 are scaled and refitted into the region between C_1 and C_2 radially, based on the distance between the center and every pixel multiplied by the ratio of the outer circle over the inner circle.

In the human macula, the cells' pathways are radially shifted, hence we need to compensate for the ring structure of retinal cells in the macula since the optogenetic signals projected from outside need to reach the radially shifted ganglion cells, not the damaged photoreceptors [104, 105]. Note that due to the vertical stacking of ganglion cells [106], some resolution will be lost when projecting a signal from outside since we can only project a two-dimensional signal on a three-dimensional structure.

Figure 3.3 shows how the circular interpolation is implemented. We propose that the pixel outside the outer circle C_2 to remain the same and the pixels inside the inner circle C_1 to be pushed to the area between the C_1 and C_2, while the pixel already existing between C_1 and C_2 to be rescaled to make room for the pixels shifted from C_1. In other words, we rescale the pixel of C_1 and C_2, into the area described by the difference between C_2 and C_1. We denote as x and y the horizontal and vertical pixel position, by R_1 the radius of C_1, and by R_2 the radius of C_2. To get the distorted image I' from I in several steps:

At first, we get the position of the distortion center x_c,y_c, then we compute the distance from the center to each pixel following this equation:

$$r(x, y) = \sqrt{((x - x_c)^2 + (y - y_c)^2)}. \tag{3.2}$$

Then we get the distance between the inner radius of inner circle and actual distance of the pixel from the center using the following equation:

$$d(x, y) = r(x, y) - R_1. \tag{3.3}$$

The distorted image I' is generated according to three cases:

1. in case $r(x, y)$ is inside the inner circle C_1 then the output pixel is zero
2. in case $r(x, y)$ is bigger then outer circle C_2 then the output remain the same as input image
3. in case of $r(x, y)$ is between C_1 and C_2 we compute the relative pixel distance between inner circle and outer circle and find the corresponding pixel in the original image

The equation describing these three cases is:

$$I'(x, y) = \begin{cases} 0, & d(x, y) > r(x, y), \\ I(x, y), & r(x, y) > R_2, \\ I'(x'.y'), & R_2 > r(x, y) > R_1. \end{cases} \tag{3.4}$$

In the last case we can compute the corresponding pixel position x',y' using the following steps: we first get the sin and cos of the angle formed with the axes.

$$\sin\theta = \frac{y - y_c}{r(x, y)}, \quad \cos\theta = \frac{x - x_c}{r(x, y)}. \tag{3.5}$$

Then we use that difference in distance to compute the relative position of the shifted pixel position.

$$x' = \frac{R_2}{R_1} \times d(x, y) \times \cos\theta. \tag{3.6}$$

$$y' = \frac{R_2}{R_1} \times d(x,y) \times \sin\theta. \tag{3.7}$$

Finally, after obtaining the relative pixel position we find the values from the original image. In some cases the computed pixel position falls between 4 pixels as shown in Figure 3.4, hence we use bilinear interpolation to estimate the real value based on the equation below.

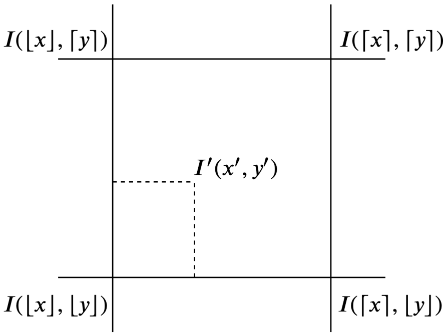

$I(\lfloor x\rfloor, \lceil y\rceil)$
$I(\lceil x\rceil, \lceil y\rceil)$
$I'(x', y')$
$I(\lfloor x\rfloor, \lfloor y\rfloor)$
$I(\lceil x\rceil, \lfloor y\rfloor)$

Figure 3.4 Bilinear interpolating, the obtained pixel value is average between the 4 closets pixel scaled based on distance.

$$\begin{aligned} I'(x,y) = I(\lfloor x\rfloor, \lfloor y\rfloor) \times (\lceil y\rceil - y') \times (\lceil x\rceil - x') \\ + I(\lceil x\rceil, \lfloor y\rfloor) \times (\lceil y\rceil - y') \times (x' - \lfloor x\rfloor) \\ + I(\lfloor x\rfloor, \lceil y\rceil) \times (y' - \lfloor y\rfloor) \times (\lceil x\rceil - x') \\ + I(\lceil x\rceil, \lceil y\rceil) \times (y' - \lfloor y\rfloor) \times (x' - \lfloor x\rfloor), \end{aligned} \tag{3.8}$$

where $I'(x,y)$ is the output of radial shift of image I by pushing and refitting the central pixel in a circular means to emulate the structure of neurons in the retina, $\lfloor x\rfloor$ denotes the largest integer smaller or equal to x, $\lceil x\rceil$ denotes the smallest integer larger or equal to x.

3.4.3 Spatial filtering

As mentioned earlier the first models of the retina proposed were based on DoG [90], this is due to ON ganglion cells being inhibited by nearby cells. This inhibition can block ON cells from depolarizing when the lights are ON, this serves in reducing information transmitted to the Lateral Geniculate Nucleus (LGN) and the visual cortex when presented with locally uniform stimuli. While this model can accurately represent the spatial properties of the retina it does not incorporate the temporal aspects. The 2D DoG is calculated by:

$$DoG(I') = I' * g_{(1)} - I' * g_{(2)} = I' * (g_{(1)} - g_{(2)}). \tag{3.9}$$

where g_1 and g_2 are the Gaussian blur under different kernel sizes, and DoG is the difference of Gaussian image. The 2D Laplacian of Gaussian is define by:

$$LoG(I') = \Delta(I' * g) = I' * \Delta^2 g, \tag{3.10}$$

where I' is the input image and g is the Gaussian kernel. The proof of approximating DoG as an LoG can be found in [92] and [93].

An example of the LoG can be seen in Figure 3.6d, where an input image is passed by a two-dimension convolutional LoG filter. This filter detects the location of variation in local intensity. It is used in image processing to detect edges of objects, the human eye also tends to focus more on edges and geometry rather than texture details as edges have high-frequency information. In our case, we used a kernel size of 3. The computational complexity is proportional to the size of the image and size of the kernel meaning $O_{LoG}(I \times n^2)$ where n is the kernel size.

3.4.4 Multi-Frame Temporal Filtering

To emulate the temporal response of the macula we first must understand its origin. A neural response is usually transmitted after a stimulus in the form of glutamate. This causes either hyper-polarization or de-polarization based on the type of the cell. For ON-type cells operating in the dark, the membrane of a photoreceptor cell is open, causing a constant flow of sodium into these cells and the release of neurotransmitters. When the photoreceptor is exposed to a light source, a chain reaction occurs in the photoreceptors causing the sodium channel to close and the cell membrane potential to drop resulting in hyperpolarization which stops the release of neurotransmitters. The cells exhibit a refractory period after hyperpolarization or depolarization which affects the response to new stimuli. This process is slow compared to digital transistors and clock cycles, the biological process is also non-uniform, meaning that some cells exhibit de-polarization faster than others. This is especially true when comparing rods vs cones. This means to replicate the temporal filtering in the macula we should take into account the refractory period. The faster the camera and the processing frame rate, the better time resolution for a more accurate representation of the temporal functionality of the macula.

Our algorithm is designed to emulate the temporal processing in the macula as efficiently as possible while remaining relatively accurate. It has been suggested that the human eye cannot distinguish between two consecutive stimuli occurring within 40 ms of each other, resulting in the two stimuli appearing simultaneously [107]. Two rapid stimulation of the same color causes the stimuli to appear simultaneously [108]. Rapid consecutive stimulation of two disks with different colors gives the illusion of movement with a trajectory with color changes in the middle [109]. More recently, models have suggested that the human retina have two pathway for processing temporal information, a fast conscious pathway where even a subtle variation in information can be detected in 3 ms mostly in parasol, and a slower unconscious pathway that process detailed information within 40 ms and 300 ms [110] in midget cells. This put the needed frame rate between 3.33-25 frames per second since we are only interested in emulating midget cells.

In our previous work, we approximated the gaussian mixture model as the equivalent model for primate retina [67, 91, 111]. In this work, we simplified the technique and allowed more flexibility in terms of the number of frames integrated, and achieved significantly better speed while yielding more accurate results.

To compute the temporal information in the vision we introduce the following equations:

$$F_{i,j} = I'_{i,j} - \sum_{n=1}^{N} \frac{B_{i,j,n}}{2^n}, \tag{3.11}$$

$$T_{i,j} = \begin{cases} 0, & F_{i,j} < \dfrac{1}{s}, \\ 1, & \text{otherwise}, \end{cases} \tag{3.12}$$

where F is a measure of the difference in pixel intensity in the last N frames, T is the temporal filter output, B is the stored frame buffer, n is the frame number in the queue, and s is the luminance sensitivity.

To run the multi-frame temporal filtering, we first read the input received from the radial shift. Then, we create a fixed memory buffer with a size of $I \times N$ to store every new frame until it is full. Once it is full we subtract the value of every new frame with the leaky integration per pixel for every correspondent pixel in the last N frames. We use leaky integrator because visual information tends to do drop exponentially in the macula as time pass [112, 113]. Note that we are using a first-in-first-out frame buffer, hence the leaky integration only accounts for the last N frames and drops the rest. To remove the resulting noise from the frame we compare it to a threshold of $1/s$. This threshold can be tweaked to get the desired luminance sensitivity where a high value of s indicates high sensitivity to light variation and a low s means a low sensitivity to luminance variation in the scene. From Equation 3.11 and 3.12 we can deduce that the computational time is proportional to $1 + N$ for every pixel, therefore the computational time for a temporal filter with a frame depth of N is $I(1+N)$, meaning that this filter has a computational complexity of $O_{temporal}(I \times N)$. If we want to create several temporal filters with different time-depth we simply have to change the value of N, the integration does not have to be recomputed in entirety, we can simply add the frame information to the integration scaled with the corresponding decay factor.

3.4.5 Output Frame Combinational Filter

The final step is the combine both the spatial and the temporal aspects of the retina. Depending on the adaptation state of the retina (mostly due to absolute light level) there will be more or less contribution from the rods. At high light levels, the rods will be completely inhibited, and at low levels, their signal will be much stronger than that of the cones. The rods are absent in the foveola. In the case of optogenetic the logarithmic compression properties of the photoreceptor and bipolar cell stage are lost, hence we can consider the optogenetically altered ganglion cells will respond almost linearly with light intensity [114]. At the ganglion cell level, there is no separate rod pathway left. Hence, we can model the effect of brightness as a combined linear amplitude. At high intensity, we put more emphasis on the cones while, and at low intensities, we can place a heavier weight on the delayed frames.

To get the final output combinational filter we multiply the output of both the spatial and temporal filter pixel by pixel. when we first start the camera the output of the temporal filter is null, so only the spatial filter information will propagate. As time passes, the temporal effects kick in, it causes inhibition of the redundant information obtained from the spatial filter. The equation for the combinational filter is shown below:

$$I_{avg} = \frac{\sum_{j=0}^{H} \sum_{i=0}^{W} I_{(i,j)}}{H \times W \times 256}, \tag{3.13}$$

$$L_{i,j} = LoG_{i,j} \times \frac{1 + I_{avg}}{255}, \tag{3.14}$$

$$C_{i,j} = L_{i,j} \times T_{i,j}, \tag{3.15}$$

where I_{avg} is the average image intensity, H and W are the image height and width, $L_{i,j}$ is the light adaptation, and $C_{i,j}$ is the output of the combinational filter.

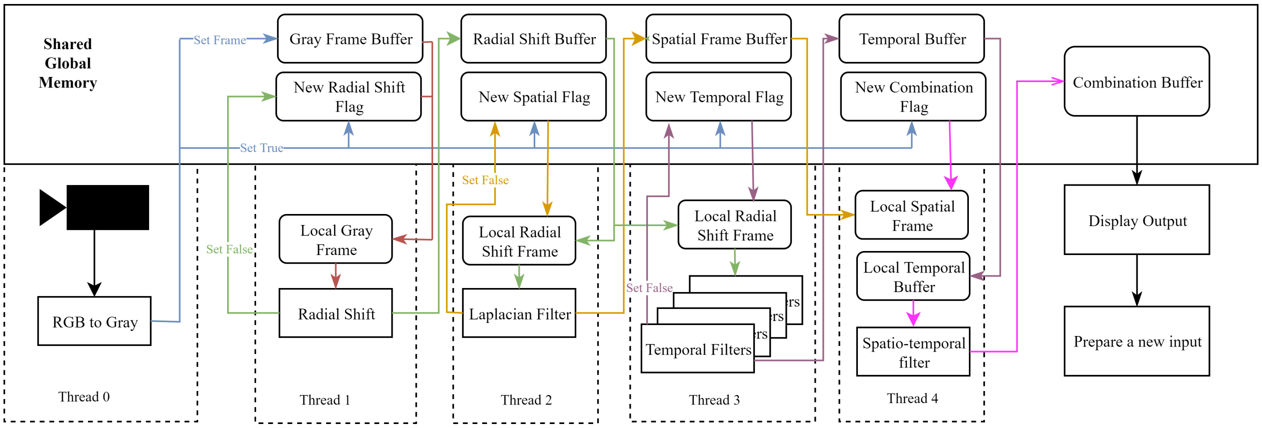


Figure 3.5 Multi-threaded optimized model of the retina

Multi-threaded optimized model of the retina. Memory location and variable are placed in a rounded rectangle, while functions are placed as a regular sharp rectangle, the dotted rectangle encapsulates the content of the thread. The blue arrow corresponds to the processes related to the grayscale input, the green arrow correspond to the process related to the radial shift, the orange arrows correspond to the spatial filtering process, the purple arrows correspond to the temporal process, and finally, the pink arrows correspond to the combinational process. Each colored process is set with its own thread, the shared memory is necessary to share data between different threads, while the flags are placed for synchronization.

3.5 Software Implementation

Since the model needs to run on an embedded platform and provide a signal to an optogenetic stimulation device in real-time, the implementation must be able to reach above 25 FPS to mimic the temporal aspect in the macula, which might be challenging on low-power performance limited devices. Modern SoCs available in smartphones, micro-controllers, and single-board computers use multiple cores to achieve desired performance as power efficiency scales poorly with high voltages and frequencies. Hence, we used C++ based code and implemented a software pipeline that uses the CPU cores effectively.

3.5.1 Threading structure

Figure 3.5 shows the workflow of the algorithm. We divided the pipeline into five stages, each running on its separate thread. Each thread is fixed at creation time to handle one function. Thread 0 is only capable of reading the camera input, Thread 1 is responsible for performing radial shift, Thread 2 is implementing spatial filtering by using LoG, Thread 3 is emulating temporal filtering using a multi-frame averaging filter, and finally, Thread 4 is combining both the spatial and temporal filters to produce the output. To ensure communication and synchronization between each step, we created four different flags to signal when it is safe and efficient to perform the filtering.

When the code first starts it initializes the camera, the shared variables (flags and frame buffers), and threads. At first, Thread 0 reads an image from the webcam and converts the images from color to gray, and places the output into the shared global memory pool, then it will set all the flags to $true$ to signal other threads that a new frame is ready for reading. Thread synchronization is important after every shared variable is updated to make sure that other threads are reading the correct value.

Thread 1 will pull on the new radial shift flag by checking if is $false$ or $true$, If the flag is $false$, then the thread will sleep for $1ms$ and then check again until it becomes $true$. It will then clone the gray frame information from the shared global memory to the local private one, then it performs radial shift on the clone, and finally, it will copy the output to the radial shift buffer in the global memory, furthermore, it will set the radial shift flag to $false$. Thread 2 and 3 will take the same steps as Thread 1 with their corresponding flags, but will instead use the global radial shift frame as input and global spatial and global temporal filter as output respectively. This also applies to Thread 4, but with both global spatial and global temporal as input, and the output of the combinational filter is sent to the optogenetic stimulation device.

Other approaches such as dividing each frame into quadrants would use more memory and require more synchronization between threads, which reduces performance and increases latency. Another approach for parallelism is to have a shared queue and each thread handling one frame through start to finish, but such an approach requires more memory since every thread needs a clone for every frame buffer and requires an input and output queue with a proper locking mechanism, and an output sort to ensure that frames are displayed in order, and finally, the biggest problem is the additional synchronization overhead needed to ensure that the temporal filter in each thread is being filled and maintained across all threads since each thread can only see a subset of frames in that approach.

3.5.2 Optimization

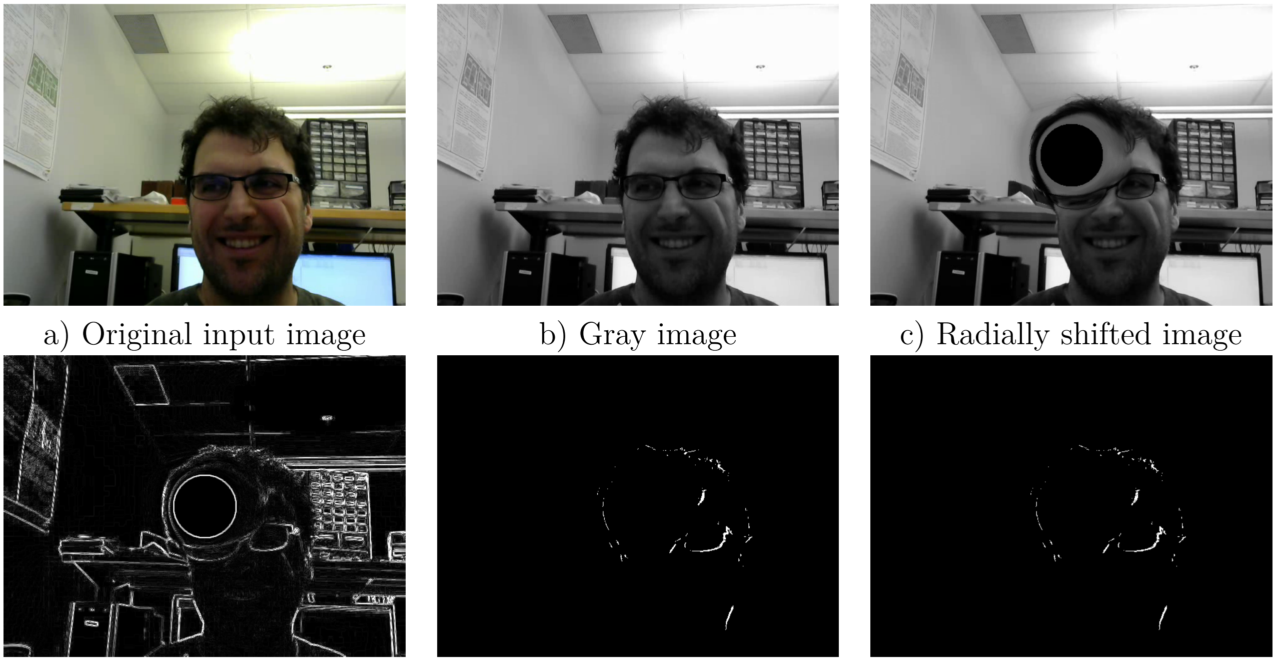
The computational complexity can be computed for calculated for each stage as such. To get the computational complexity of RGB to gray conversion we have to iterate for each pixel in the image once to get the value, each per pixel conversion to gray will always perform the same constant number of multiplication, therefore the computational complexity can be represented as O($H * W$) for image conversion. Similarly, combinational filter will also require O($H * W$) computation. Radial shift the computational complexity is O(${R_2}^2$) since we only operate on the selected outer circle C_2. The temporal filter will have O($H * W * N$) complexity as we have to perform the leaky integration for pixel for the last N pixel. Finally, the spatial filter complexity should be O($H * W * n^2$). Since R_2 is always smaller than H or W we can assume that the radial shift isn't the most computationally demanding aspect of the algorithm. The overall computational complexity will be either O($H * W * n^2$) or O($H * W * N$) based on the size of the spatial filter length n or temporal filter depth N. As for memory complexity, the temporal filter will always require the most memory of any step of the algorithm with O($H * W * N$).

Practically, the most demanding aspect of the algorithm will be the temporal filtering, since it is not cache friendly, since the access pattern will requires large memory strides for each computation. This also makes vectorization like using Intel's advanced vector extensions (AVX) practically impossible, The temporal filter is also not memory efficient as we have to store several large images in the memory.

To ensure the best running performance several optimizations are made to ensure sufficient performance. Since C++ and OpenCV is a row first language, meaning we represent arrays as matrix[row][column], indicates that the column values are stored consecutively in memory, for example A[0][0] is stored next to A[0][1], next to A[0][2], etc. Hence, it is better when writing custom implementation to access column information first and then rows when the column overflows. The reason is that the caches are not large enough to store entire images. Looping through rows would result in larger cache misses since we would be jumping between large memory offsets. Form our test using visual studio profiler, we saw that the time spent in the temporal filter is reduced from 27% to 15% of total time just by looping through column first processing. We verified that using performance counters when processing 1000 frames resulted in a decrease in last level cache miss by 25% when processing column first compared to processing row first. The CPU-time also dropped by 10%. Another performance consideration is to store common computation in a small lookup table such as the leaky integration factors, we can also reduce computation time by transforming some divisions into multiplication when possible. Creating arrays in the power of 2 helps make the code more cache-friendly. Organizing the code to reduce thread synchronization and spinlocks required can also reduce latency and wait times.

These approaches combined provide a significantly improved performance with minimal latency impact compared to running the code using a single thread. After cache and thread optimization. We found that the most time-consuming step in the algorithm is temporal filtering which occupies 15% of total CPU time, followed by the radial shift at 12%, then reading a new frame by 7.96% and finally spatial filtering at 3.22 %, the rest are divided between various general tasks, involving system and library calls, thread synchronization, and memory copy.

3.6 Experimental Setup and Results


a) Original input image b) Gray image c) Radially shifted image

d) Laplacian filter output e) Temporal filter output f) Combinational filter

Figure 3.6 Example output for each of the four steps

The input in (a) is obtained from a webcam. In this case, the output of the combination filter greatly resembles the image in c. This is because temporal information has suppressed the spatial information due to very limited movement from previous frames hence very little information to transmit.

The code we wrote to implement the model uses C++ with OpenCV and OpenMP for portability between different systems and different OS. The algorithm was tested with two different systems. The first system is a laptop computer with an i7-8750h clocked at 2.2 GHz base clock and 4.1 GHz boost with 32 GB of dual-channel DDR4 memory. We used the visual studio 2019 compiler along with OpenCV 4.1 to compile the code with compiler optimization turned on. The second system is Raspberry Pi 4 which has a quad-core A72 processor clocked at 1.5 GHz with 4 GB of LPDDR4 RAM, running on Raspbian Buster OS with OpenCV 4.1.1. We used the GCC compiler along with OpenCV 4.1 with compiler optimization ON to compile and run the code. We verified that code has generated single instruction, multiple data (SIMD) instructions (NEON for ARM and AVX for Intel) by looking at the object dump of the compiled binaries. The same webcam used for both systems is the Logitech HD c270 running at 640 × 480 at 30 FPS. All systems were tested after a fresh reboot of the OS with no background application running. Note that the camera we used has auto-brightness adjustment and auto-focus at the driver level, so we didn't need to implement these pre-processing steps here.

For these experiments, the depth of the temporal filtering is fixed at 4 frames since after 4 the residual effect of the decaying integrator is very small and the typical refractory period of the eye. The spatial filter size is set to 3 for two reasons, the first is that we don't expect far away neurons to have an effect on the image as adjacent neurons, and using a filter size of 3 gets faster results. The luminance sensitivity is set to 0.0625, this is based on visual feedback that reduces random noise appearing in the image temporal filter. However, these parameters can still be tweaked by users' preferences.

We repeated each test 4 times and measured the number of frames per second (FPS) that each system achieved. Every run lasted at least one minute to have a consistent and stable output. Figure 3.6 shows the output of each step of the algorithms. In Figure 3.6a we see the original input, Figure 3.6b shows the resulting color to gray image, Figure 3.6c hows gray image after radial shift, while Figure 3.6d shows the result of the spatial filter. We can see the visual information from the center has been shifted to the side, we can also see the redundant spatial details have been removed, this technique allows the removal of low-frequency texture information and keeps the high-resolution information. 3.6e shows the result after temporal filtering and 3.6f shows the output of combining both spatial and temporal filtering. In this case, the camera was fixed and the person in the foreground was not exhibiting any conscious movement. The result is the output of the temporal filter is kept to a minimum which inhibits some of the output of the spatial filter.

The laptop ran the algorithm with an FPS of 29.6 with low CPU utilization of 4-5% at a frequency of 0.9-1.0 GHz, this indicates that the laptop has a lot of headroom to implement more complex features in the retina. On Raspberry Pi 4 the algorithm ran at 29.6 FPS with average utilization of 41% giving a decent headroom to implement more visual channels in the future. This system can also run on batteries, making it truly portable with a weight of around 130 grams for the Pi and the battery. Also, the algorithm presented in this chapter can easily be extended by adding more threads with additional retinal functionalities such

Table 3.1 Comparison table of visual prostheses used for different retina degeneration diseases

Paper	FPS	Portable	Main contribution
(this work)	98-250	yes	real-time midget cell processing for AMD patient
[27]	N/A	no	static image enhancement for RP patients
[82]	N/A	no	scene simplification for optogenetic stimulation
[83]	25	yes	Image enhancement for RP patients
[84]	50	no	simple retinal bipolar model with geometric compensation
[115]	N/A	no	retinal model for peripheral vision

as light adaptation and directional movement detection.

To measure the maximum performance achievable we recorded a 4-minute video with the same resolution of 640 × 480 and saved it to local storage. We then modified the code to read directly from the stored video instead of a camera with no wait between frames and measured the average image processed per second. The Raspberry Pi 4 achieved on average 98 images processed per second with a CPU time of 86%. Running the code and video on the laptop resulted in 250 images processed per second with 45% utilization, showing the algorithm scale well beyond 30 FPS and would work well with high-speed cameras.

3.7 Conclusion and Future Work

In this paper, we proposed MSTM, an efficient functional model of midget retinal processing which can emulate the high-level retinal processes in real-time. We have shown that we were able to emulate major pathways in the retina in real-time under the typical speed we normally find in the retina. In the long run, we hope that the approach we proposed can be used to create a retina stimulator that can restore some visual acuity in the central vision similar to how cochlear implants can restore hearing for some deaf patients. Full vision restoration for AMD using optogenetic is still impossible due to the gaps in knowledge when it comes to minor visual processing pathways in the macula, especially in the amacrine cells, retinal remodeling in late phases of retinal degeneration, and accurate stimulation delivery to the ganglion cells.

This work can be expended in the future by connecting the Raspberry Pi to a custom-designed optogenetic stimulator and validating the functionality in vitro on neural cells and later on in vivo on test mice retina. Another improvement that can be made to this model is adding additional channels in the retina, to account for the non-uniform latency, and achieve sub 3ms latency to match the fastest pathways in the macula.

Acknowledgment

The authors would like to acknowledge the financial sup-port from Natural Sciences and Engineering Research Council of Canada (NSERC). We would also like to thank Prof. Gislin Dagnelie from Johns Hopkins University School of Medicine for his invaluable feedback.

CHAPTER 4 ARTICLE 2: MOBILE-OPTIMIZED FACIAL EXPRESSION RECOGNITION TECHNIQUES

Nizar El Zarif[1], Leila Montazeri[1], François Leduc-Primeau[1], and Mohamad Sawan[1,2]
[1]Polystim Neurotech Lab, Department of Electrical Engineering, Polytechnique Montreal
[2]CenBRAIN, School of Engineering, Westlake University, Hangzhou, Zhejiang, China, 310024, IEEE Access, published on July 9^{th},2021.

This chapter answers the second objective of the thesis, where we develop two new real-time algorithms that can detect facial expressions in real-time. Due to the current limitation of the custom microLED matrix we designed and wanted to test, a high-resolution image downscaled to a low-resolution 8x8 LED matrix would fail to produce anything visible. Hence we need to represent what is in the image with a limited number. We first want to detect what is in the image. We chose facial expression as it is ubiquitous yet there is very little work in the literature that can detect facial expression reliably and in real-time. In this chapter, we presented two new algorithms capable of performing real-time facial expression recognition, under different poses, while using affordable mobile hardware with the Raspberry pi 4 system. We targeted the Raspberry pi, not only for our affordability but also for easy expendability, as it exposes its serial interface with direct general purpose pins, making it an ideal candidate to connect to custom build or off-the-shelf microLED matrix. This work has been published in the IEEE access [81].

4.1 Abstract

This chapter presents two novel facial expression recognition techniques: the real-time ensemble for facial expression recognition (REFER) and the facial expression recognition network (FERNet). Both approaches can detect facial expressions from various poses, distances, angles, and resolutions, and both techniques exhibit high computational efficiency and portability. REFER outperforms the existing approaches in terms of cross-dataset accuracy, making it an ideal network to use on fresh data. FERNet is a compact convolutional neural network that uses both geometric and texture features to achieve up to 98% accuracy on the MUG dataset. Both approaches can process 14 frames per second (FPS) from a live video capture on a battery-powered Raspberry Pi 4.

4.2 Introduction

There are seven essential and recognizable facial expressions that do not require translators and can be understood across cultures and languages. These seven emotions are anger, neutral, fear, disgust, happiness, sadness, and surprise. Even people who are blind from birth can express the correct facial expressions without ever seeing them [116]. While facial expressions can be understood relatively easily by humans, they are challenging for machines because a facial expression detection algorithm must be sensitive to small variations in the face while being robust to changing environmental conditions. In addition, humans tend to express emotions slightly differently, which makes the problem even more difficult. While other senses can be used to estimate emotions, such as hearing, sight is the most accurate indicator of emotion.

Automatic facial expression detection has many beneficial uses in human-computer interactions [117]. Its applications extend to the gaming industry, commerce, medical field, academics, or even personal use. For example, horror-based video games can provide challenges, such as spawning new enemies, based on the player's fear level, which is dynamically extracted from his/her facial expressions. In commerce, facial expression detection can also measure people's reactions to movies, food, advertisements, and clothing. In the medical field, facial expression detection can help in personalized medicine, such as pain measurement [118], or rehabilitation monitoring. It can also provide personalized stimulation to complement a system that emulates retinal processing for people with low vision [119]. In education, facial expression detection can aid teachers by measuring the engagement of online students in virtual classrooms. For personal use, facial expression detection can be extended to tailor music to listeners' emotions or measure the drowsiness or responsiveness of drivers to ensure their safety [120].

For facial expression recognition there are two high-level approaches. The traditional approach relies on feature extraction using specific descriptors. This can be either texture, geometric, hybrid, or generic features. Geometric features, such as [60, 121–124] have lower memory and computational requirements, which makes them ideal for real-time applications. Texture features are significantly more computationally expensive than geometric features, but they provide a high accuracy. Some of the more popular approaches can be seen in [61, 125–133]. Generic features [134, 135] use general-purpose descriptors. They are relatively easy to implement and use but are not as accurate or as fast as the geometric or texture approaches. Hybrid features [136] are less common since they require both geometry and texture at the same time, which increases the complexity. The second high-level approach to facial expression recognition is to use deep neural networks to skip feature extraction, since neural networks will learn the features from data, as seen in [137–164]. This paper proposes two approaches to facial expression recognition. The first is based on the traditional geometric approach with a custom scale and poses invariant descriptors with a fast implementation to ensure the best accuracy on fresh data with a high real-time throughput. The other approach is a neural-network approach with a simple feature extraction that ensures a high accuracy on familiar data with a very fast and low latency execution pipeline, which is ideal for real-time embedded applications.

The remainder of the article is organized as follows. We first review related works in Section 4.3. Then, Section 4.4 introduces the REFER technique, and Section 4.5 introduces the FERNet convolutional neural network approach. Section 4.6 presents the accuracy and complexity results for the two proposed algorithms and compares them to the state of the art. Finally, Section 4.7 concludes the paper.

4.3 Related Work

There are three steps in any facial expression recognition (FER) task: feature extraction, training, and testing. The most crucial part of traditional approaches in FER is feature extraction, as it determines which image points are the most distinct for classification while being robust to variations in the scene. A deep neural network can skip feature extraction since the hidden layers in neural networks can detect abstract features.

4.3.1 Geometric Features

The geometric features are based on the face's geometry. Some of these features include the distances and angles between different facial traits, the distance between the mouth and the chin, and the curvature of the mouth. Advances in computer vision have allowed for the creation of the active shape model (ASM) and the active appearance model (AAM) [121]. In both approaches, several points across the face are tracked in real time with a low overhead, such as in [122, 123]. The greatest advantage of the geometric features is that they tend to be computationally inexpensive and memory-efficient due to their limited number, which is important for real-time and embedded applications. However, they are sensitive to scale and face orientation, which makes them prone to registration errors [124]. Liu et al. [60] showed that using a subset of the features can improve the detection for one dataset, but at the cost of others. The two proposed approaches solve the issue of both orientation and scale, and greatly improve the detection of geometric features by performing scaling on distances in real time.

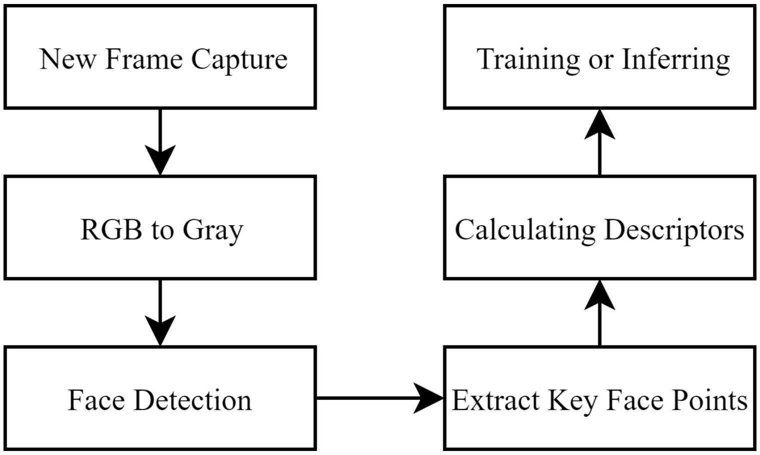


Figure 4.1 Typical workflow for automatic facial expression recognition algorithm using geometric features

Typical workflow for automatic facial expression recognition algorithm using geometric features. Initially, each frame is obtained from stored pictures or videos; then, the image is converted from RGB to gray. The face is then detected, and feature points are extracted. From these data, the descriptor is typically calculated based on the distance or angles. Finally, the descriptors are used either for training or to infer the facial expression

4.3.2 Texture Features

The second type of feature extraction is based on texture features, which use pixel intensity information of the face to classify expressions. These features can capture smaller face details, such as wrinkles and curves. Texture features tend to require considerable memory due to the need to store and process a large array of pixel data. In addition, textures are illumination-dependent; hence, they suffer from uneven lighting. The latter can be solved by using local texture features instead of global features [125], [126]. Local binary patterns [127] or Gabor wavelets [128] are two of the more popular approaches for texture-based feature extraction, but the latter can be memory and computationally expensive. In [129], the authors used a principal component analysis and template matching to achieve an accuracy that surpasses 99% on personal images. However, with no cross-validation and limited data, the method can be prone to overfitting and requires previous facial and emotional knowledge. To compute the local energy and distance based on the Symlet wavelet with optical flow, researchers in [130] created an unsupervised learning technique based on active contouring and moving features, which obtained an accuracy of 87% on the extended Cohn-Kanade (CK+) dataset [131]. In [61], the authors extracted the region around the nose and the eyes and used the LBP feature with weighted sum voting. They obtained up to a 90% accuracy on the CK+ dataset and 78% on the multimedia understanding group (MUG) dataset [165]. Generic features such as SURF have been used to describe facial features and perform FER with a good accuracy of up to 96% on the MUG dataset [134,135]. 2D linear discriminant analysis can also obtain a good detection rate [133].

4.3.3 Hybrid Features

The third approach to feature extraction is based on hybrid models, using both geometric and texture features. This approach should give more accurate results than each feature alone. In [136], the authors used hybrid features to obtain an accuracy between 85% and 95% using the JAFFE and CK+ datasets. However, execution speed is a considerable drawback to this type of descriptor, as it requires storing and processing both texture and geometric features and then efficiently combining them to produce accurate results.

4.3.4 Neural Networks

Neural networks are rapidly growing in popularity in machine learning since Alexnet won the ImageNet competition in 2012 [137,138]. Since then, many neural network-based approaches have been introduced.

The authors of [139] proposed a mixture of geometric and texture features by training both SVM and neural networks. Their results showed a similar performance; however, neural networks tend to overfit the model. As a result, neural networks can only classify the input dataset with a high accuracy, but will underperform with new types of data. Similarly, the authors of [140] used a convolutional neural network (CNN) for facial expression detection, with a high accuracy, but their network underperforms during cross-validation tests, which indicates overfitting. Recently, Sen et al. used angles and LBP as geometric and texture features to achieve accuracies of 78% and 91.85% using the MUG and CK+ datasets, respectively [141].

The authors of [142] used a CNN along with the ensemble method of voting to build and train the classifier. It resulted in a decent EmotiW challenge result with an average of 61.29% of the test set's detection. Researchers in [143] introduced a neural network for facial expression detection based on the difference in the geometric features and the difference in consecutive frames. They then combined both the temporal and geometric features with an accuracy rate of 45.5% on the CK+ dataset. Similarly, a CNN with different spatial and temporal features that computes features based on changes from the peak reaction was proposed in [144]. Using generic feature extraction such as SIFT before passing it to a neural network can give decent results, with an 80% accuracy on BU-3DFE [145]. Deep convolutional neural networks can lead to a high accuracy when validated on the same dataset [146].

The use of preprocessing image techniques, such as image segmentation and histogram equalization, can increase the accuracy by a few percentage points. Detection accuracy can be improved by concatenating several features, or several neural networks. The authors of [139] used autoencoders in the first two layers to concatenate the texture and geometric features in order to improve accuracy. In [147], the authors merged VGG with *ResNet 50* to produce an output slightly better than both. Data augmentation has also been shown to improve the detection rates [155–157]. For example, [157] improved the accuracy by using rotations, translations, and other transformations on the original images to augment the size of the datasets.

The residual network, proposed in [148], is one of the most popular networks for computer vision. Their network can avoid the vanishing gradient problem with the deep network. *ResNet 34* has been used in multiple works to predict facial expressions, such as in [149] and [150]. Other types of residual networks are also used to detect facial expressions, such as in [151], where a generative adversarial network is used to generate features based on input images combined with *ResNet 50* for classification, and [152], which uses a region attention network to extract features that are sent to a *ResNet 16* classifier. Most of the techniques based on *ResNet* result in marginally better results on average, with a trade-off in larger preprocessing steps. References [153] and [154] used the VGG neural network to detect facial expressions with a high accuracy, but VGG tends to be enormous and requires a large amount of memory, which is not optimal for performance- and memory-limited low-power devices.

Another approach to detecting facial expressions using deep learning is to create a custom neural network. Building a custom neural network can be beneficial since it allows us to tweak it to our intended use case, whether for speed, accuracy, or temporal stability. CNNs can be used to detect spatial features with a good accuracy, as seen in [158, 160–162], while smaller networks can trade in some accuracy to reduce the processing time [163, 164]. Another approach is to use a long short-term memory (LSTM) neural network to detect temporal features, such as in [159]. However, this requires training and testing on a sequence of images instead of one image at a time, and is more computationally expensive than a similarly sized neural network.

4.3.5 Other Features

Other features include advanced facial expression detectors that use specialized sensors, such as 3D and thermal sensors combined with normal cameras [166]. However, the reliance on specialized hardware to augment the capability of an RGB camera sensor makes the solution more expensive and less universal than a simple camera.

4.4 The REFER Algorithm

Unlike most of the previously mentioned approaches, the two proposed techniques work in real time and are capable of classifying emotions on-the-fly, not just in controlled environments, such as the popular CK+ dataset. To highlight the advantages of the proposed approaches, we ran them on our own dataset [167] (see Section 4.6.1 for details) and on three publicly available datasets: CK+, MUG, and KDEF [168]. Another drawback of the other FER techniques is that they are usually optimized to work on individual datasets only and do not generalize well to unfamiliar data sources [155]. The two proposed approaches obtain significantly better results than the state-of-the-art in that aspect.

REFER takes the geometric approach in facial expression recognition to interpret human emotion. It works in real time by using highly optimized computer vision libraries for face landmark detection and our own simple but powerful descriptors using logarithmic, distance, and orientation scaling. This approach allows our algorithm to work with virtually any resolution under different poses at variable distances from the camera. Our approach is fast enough to work in real time, even on low-cost mobile devices. Running in real time is very important for the type of application that requires instantaneous facial expression detection.

The typical automatic facial expression detection algorithm using geometric features is shown in Figure 4.1. The two most popular approaches for face detection are Viola-Jones using Haar cascades [169] and local binary pattern (LBP) cascades, with Viola-Jones offering a higher accuracy and LBP requiring less computational power. Then, the key face points are extracted from the detected face using ASM. After the key point positions are extracted, the face descriptor is calculated based on the distances between different key points. After this point, the descriptors are either used to train the model if the facial expression is known, or to infer if the model is already trained. To train the model, we can either directly use the expression or use the facial action coding system (FACS), which codes the movement of individual features in the face [170, 171].

References [59] and [172] used a similar structure to construct their facial expression recognition system. The reported maximum accuracy in [59] was 85%, while the reported maximum accuracy in [172] was 80.9%. Reference [59] reported a maximum speed of 2.4 FPS on mobile hardware. Reference [173] used an active appearance model with head pose estimation and normalization to detect faces at angles. REFER produces a higher accuracy rate in both cross-validation tests and is significantly faster.

4.4.1 Overview of REFER

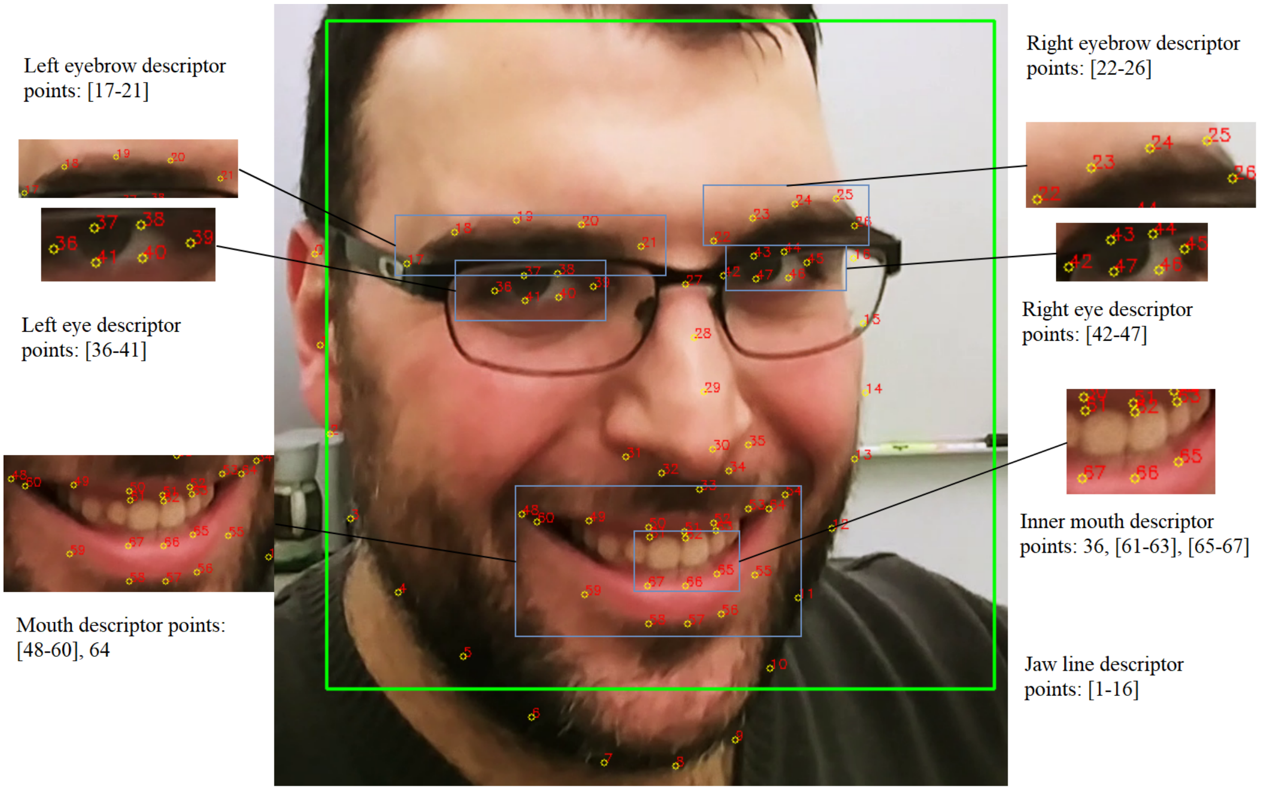
Algorithm 1: REFER processing for one video frame

```
Input: Red, green, blue pixel matrices $R$, $G$, $B$
Output: $P_1$,$P_2$,...$P_7$
$K \leftarrow 0.299 \times R + 0.587 \times G + 0.114 \times B$
$[F, (x_c, y_c)] \leftarrow \mathrm{VJH}(K)$
L = width(F)
$\{[x_0, y_0], [x_1, y_1], \ldots, [x_{67}, y_{67}]\} \leftarrow \mathrm{ASM}(F)$
$H = \frac{(x_0 - x_c)^2}{(x_{16} - x_c)^2}$,
$V = \frac{(y_{27} - y_{28})^2}{(y_{29} - y_{28})^2}$,
foreach local region $\mathcal{R}_i$ do                    // See Table 4.1
    $N = |\mathcal{R}_i|$
    $\tilde{x}_c \leftarrow \sum_{n=0}^{N} \frac{x_n}{N}, \tilde{y}_c \leftarrow \sum_{n=0}^{N} \frac{y_n}{N}$
    for $n \in \mathcal{R}_i$ do
        $d_n^2 \leftarrow \frac{(\tilde{x}_c - x_n)^2 + (\tilde{y}_c - y_n)^2}{L^2}$
        if $x_n \leq \tilde{x}_c$ then
            $\tilde{d}_n^2 \leftarrow d_n^2 \times H^2$
        if $y_n \leq \tilde{y}_c$ then
            $\tilde{d}_n^2 \leftarrow \tilde{d}_n^2 \times V^2$
        $s_n \leftarrow \log \tilde{d}_n^2$
    $\boldsymbol{p}_i \leftarrow \mathrm{pSVM}(\{s_n : n \in \mathcal{R}_i\})$
$V_p = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_8]$
$\boldsymbol{P} \leftarrow \mathrm{pSVM}(V_p)$
```

The overall REFER algorithm requires an input colored image K and outputs the detected facial expression. The REFER algorithm is described in Algorithm 1. From image K, we perform a few preprocessing steps described in 4.4.2. We then compute the local descriptors as shown in 4.4.3. After obtaining the local descriptors, we perform the first round of local facial expression detection, from which we compute the global descriptors and global output of the algorithm, as seen in 4.4.4. The result of the algorithm can be adjusted based on the previous frame to provide stability in the results in Section 4.4.5. Finally, in 4.4.6 we show how using AVFD works to provide significant performance gains on computationally limited devices.

4.4.2 Pre-processing

To train or detect faces using REFER, we first have to obtain the image frame, either from still images or from a live camera feed. The image is then converted from RGB to gray using the popular equation given on line 1, where K is the resulting gray image, and R, G, B are the red, green, and blue pixel matrices, respectively. We detect the face region using Viola-Jones with Haar cascade features, denoted by $[F, (x_c, y_c)] = \text{VJH}(K)$, which takes an image matrix K as the input and returns a new image matrix F containing only the face area, as well as the position (x_c, y_c) of the center of the face within the overall image.



Left eyebrow descriptor points: [17-21]

Right eyebrow descriptor points: [22-26]

Left eye descriptor points: [36-41]

Right eye descriptor points: [42-47]

Inner mouth descriptor points: 36, [61-63], [65-67]

Mouth descriptor points: [48-60], 64

Jaw line descriptor points: [1-16]

Figure 4.2 Volunteer descriptor example

Volunteer descriptor example: the green square is the detected face, the yellow points are the detected key points, [these two are automatically drawn in real-time], and the blue rectangles are manually added to visualize the different face descriptors.

4.4.3 Local Descriptors

Instead of using all ASM points as one set, we used these points to create several subsets. Figure 4.2 shows how the descriptors are extracted. The big green box is the face detected using Viola-Jones. The yellow dots with numbers on top are the face points detected using ASM with `dlib`. The smaller blue boxes are manually added to show what points each descriptor uses. We use 8 descriptors: one uses all the key points, while the others are partial descriptors that cover only one facial feature each. After calculating the distance between the key points, we scale the distance based on the detected face size and the perceived orientation of the face.

Table 4.1 The local region's ASM points

Feature	Key-Point Number
All points	$\mathcal{R}_0 = \{i : 0 \leq i \leq 67\}$
Jawline points	$\mathcal{R}_1 = \{i : 0 \leq i \leq 16\}$
Left eyebrow points	$\mathcal{R}_2 = \{i : 17 \leq i \leq 21\}$
Right eyebrow points	$\mathcal{R}_3 = \{i : 22 \leq i \leq 26\}$
Left eye points	$\mathcal{R}_4 = \{i : 36 \leq i \leq 41\}$
Right eye points	$\mathcal{R}_5 = \{i : 42 \leq i \leq 47\}$
Outer mouth points	$\mathcal{R}_6 = \{i : 48 \leq i \leq 60\} \cup \{64\}$
Inner mouth points	$\mathcal{R}_7 = \{61, 62, 63\} \cup \{65, 66, 67\}$

Unlike most approaches that use only one classifier, REFER uses a two-step ensemble of classifiers to detect facial expressions with a high accuracy under different head poses. The pseudo-code is shown in Algorithm 1. The variables will be discussed in the following subsections.

ASM is used to detect and track 68 key face points. From these 68 key points, we generate a global descriptor that uses all 68 key points, and we also generate local descriptors for each part of the face. The key points used for each descriptor are given in Table 4.1.

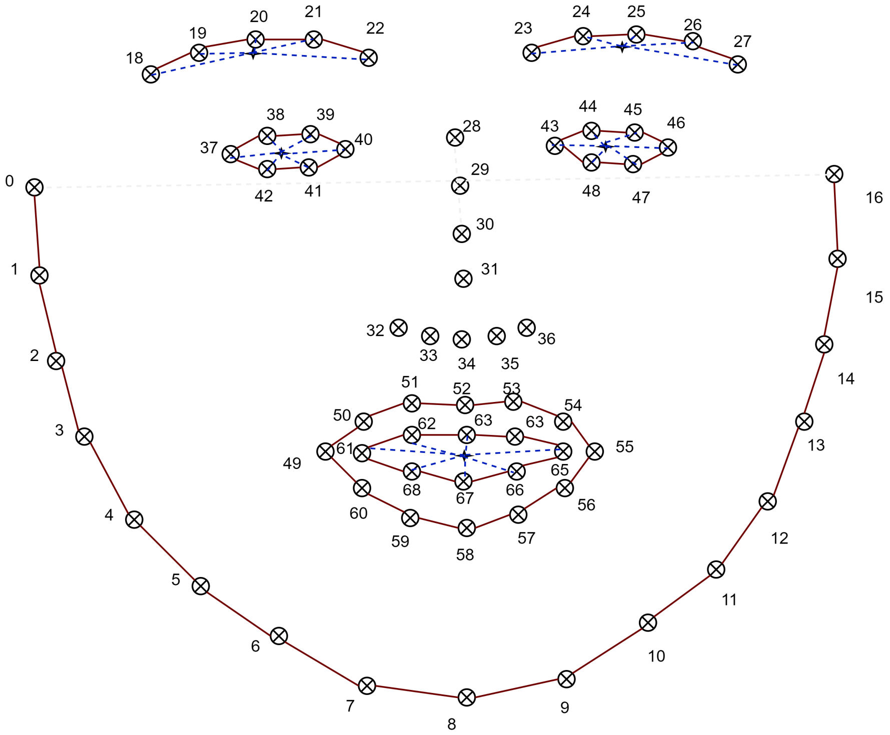


Figure 4.3 Location of ASM points on the face and the local descriptor subsets

Location of ASM points on the face and the local descriptor subsets. The red lines connect the ASM points that are part of the same set, while the blue dotted lines illustrate the distance measures used in our descriptors. Stars represent the mean points of each descriptor. Note that some subsets and mean points are not shown to improve clarity.

The proposed descriptors are designed to be robust to changes in pose and scale. This allows the facial expression to work better in the wild, where a camera could be pointed at almost any angle and distance. To achieve this, we scale the distance for both size and rotation, and finally, we apply a logarithmic kernel. The scaling is performed with respect to a virtual mean point $[\tilde{x}_c, \tilde{y}_c]$ for each local descriptor, which is obtained by averaging the location of the points in the descriptor (line 9). Figure 4.3 shows some of the sets and the location of the mean points for these sets. We only apply vertical scaling to the point above the calculated center and horizontal scaling to the point to the left of the calculated center. We compute the Euclidean distance of each key-point, as shown on line 11. This distance is normalized by the size of the face region, which helps provide invariance to the image resolution and to the distance between the subject and the camera.

To compensate for the horizontal and vertical orientations of the head pose, we multiply the scaled descriptors by an approximate ratio of the rotation. Horizontal scaling corresponds to lines 5 and 13 of Algorithm 1, where H is the horizontal scaling factor and $\tilde{d}$ is the scaled distance. Point 0 from the detected key points is the point at the top left of the face, and point 16 is the point at the top right of the face. We compute the ratio between these distances and use them to scale the left part of the face. These two points were chosen because they do not correspond with any moving muscle, meaning that they will not move with respect to facial expressions, making them good reference points for scaling. When $H > 1$, the left side of the face is closer to the camera than the right side, and when $H < 1$, the right side is closer than the left. Scaling is then applied to the left side to realign the distances in the face to be similar to the frontal view.

We then compute the ratio of the vertical scaling, which compensates for the face tilting back and forth. The vertical rotation ratio V computed on line 6 is based on the vertical positions of points 27, 28, and 29. These points are chosen since they are nose points that do not move with facial expressions, but they do move only when a face tilt is detected. These points are also equidistant in the full frontal view. The scaled distances are then computed using the equation on line 15. When $V > 1$, it means that the face is tilted forward or looking down on the camera. If $V < 1$, the face is tilted backward or looking up to the camera. We apply vertical scaling to the scaled distances for all key points above the central pixel to compensate for the tilt.

Finally, since the descriptors are based on relative distance, all the values are between 0 and 1. To increase the separation between the descriptors, we take the logarithm of $\tilde{d}$, as shown on line 16.

4.4.4 Ensemble SVM Classifier and Temporal Adjustment

After obtaining the multi-descriptors from each local facial feature, we train each classifier with a $pSVM$ classifier [174]; hence, every local feature will generate its probability prediction as an output probability vector. These probabilities are then used as training for the second-stage classifier, which takes all probabilities from all facial features as an input and builds a decision hyperplane based on these probabilities. Figure 4.4 shows the overall hierarchy of the descriptors and classifiers. The first stage of classifiers uses several parallel independent local descriptors, each with their key points. After obtaining the first descriptors, we predict the probability of the facial expression based on each classifier. This gives us an output probability vector of each facial expression for each local facial feature. These output probabilities are then used as input for the second-stage global classifier, which produces a global probability based on the local features. This will give a better prediction than the local classifiers. The output of the second-stage classifier is also represented in probability.

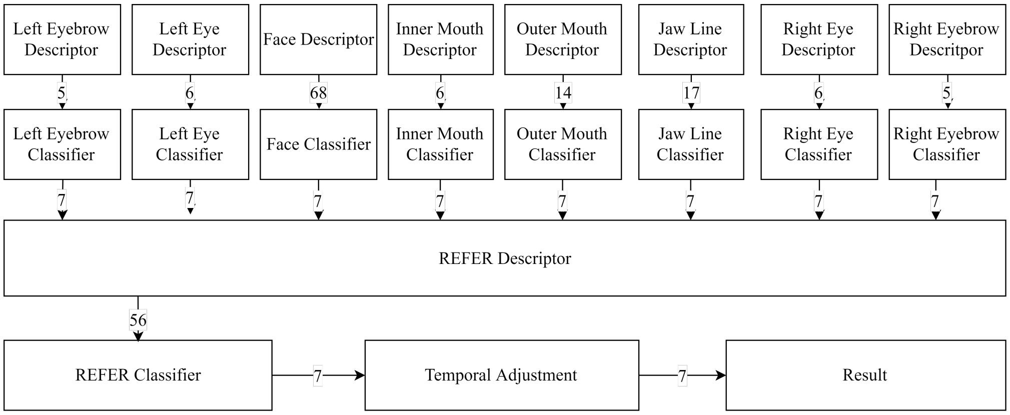


Figure 4.4 Two-stage classifier hierarchy

Two-stage classifier hierarchy: The arrows show the flow direction of classification, and the numbers represent the vector size for each step in the classifier. In the first stage, the classifiers are responsible for local-level predictions, while the second stage is responsible for global classifications.

4.4.5 Temporal Adjustment

Algorithm 2: Temporal adjustment of the classification

```
Input: $\boldsymbol{P}$
Output: $\boldsymbol{P'}$
$\boldsymbol{P'} \leftarrow \boldsymbol{P}$; $P'_{e^{(t-1)}} \leftarrow P_{e^{(t-1)}} + a$; $P'_{e^{(t-2)}} \leftarrow P_{e^{(t-2)}} + a$; $\boldsymbol{e}^{(t-2)} \leftarrow \boldsymbol{e}^{(t-1)}$;
  $\boldsymbol{e}^{(t-1)} \leftarrow \arg\max_i(P'_i)$ ;
```

Since facial expressions do not change quickly within a span of a few consecutive frames, we can assume that previous predictions would likely be similar to the current prediction. Therefore, to take advantage of this temporal property, we boost the current prediction probability vector with the result from previous predictions. This allows the result to be more stable. For example, when deciding between disgust and happy, both cases can include a partially open mouth, especially if the smile is not complete or transitioning from a different expression. In this case, we noticed that the prediction jumps back and forth between these two expressions, and adjusting the prediction probability based on previous predictions stabilizes the classification.

Algorithm 2 shows the steps of the temporal adjustment. It takes the probability vector $\boldsymbol{P}$ generated by Algorithm 1 as the input and outputs a corrected decision vector $\boldsymbol{P'}$. Initially, the prediction indices $e^{(t-1)}$ and $e^{(t-2)}$ are set to 2, the index of the neutral facial expression.

To avoid unnecessary computations, $\boldsymbol{P'}$ is not renormalized, and the final prediction is given by $\arg\max_i P'_i$. The constant a is selected empirically and set to $a = 0.4$.

4.4.6 Alternate Video Frame Detection

In our tests, running this algorithm on a PC from a live video capture resulted in 30 FPS, which is the maximum rate. However, running this on a Raspberry Pi 4, the FPS dropped to 6.3. This is 2.6 times faster than the ASM-based paper [59], which reported an average of 2.4 FPS, although they were using a different SoC. The process of running feature extraction and classification is largely serial. The performance results are discussed in depth in Section 4.6.5.

Most modern mobile devices offer multiple cores, even affordable ones such as the Raspberry Pi 4. To improve the performance on mobile devices, we can split the tasks between all 4 cores to maximize CPU utilization, increase performance, and reduce latency.

Normally, the process for facial expression recognition is a simple feed-forward technique, but due to the many steps required to complete the process, the execution time is relatively large, resulting in a low FPS. Using AVFD, the work can be divided among all the available cores. One core is assigned to read the camera input, convert it from color to gray, and put the frame buffer into a shared last in first out (LIFO) queue. The core can also retrieve the decision from the shared queue, sort, and apply the temporal adjustment. The three other threads each pop the last frame and run the remaining steps to REFER on their own assigned frame. Since Raspberry Pi only contains 4 cores, we will use three threads to compute REFER, but if more cores are available, we can easily divide the work into more cores. This technique can hide some of the latency of real-time facial expression recognition, since the processes of reading frames from webcams or stored video no longer wait for the previous prediction to finish.

Python has a limitation in the multithreaded workload due to the Global Interpreter Lock (GIL). The GIL effectively locks the interpretation of the Python script into machine code to only one thread; hence, it allows only one thread to run a Python command concurrently, so a multithreaded Python code can only use 1 core effectively. To address this limitation, we used an external C compiled library. Not only do these libraries run faster than a regular python expression, but they also run outside the locks of the GIL, thus effectively allowing the use of multiple cores. In our tests, we gained a 2.22 times speed increase, with a perceivable latency reduction (time between facial expression is performed and the result appears on the screen). The CPU utilization for a single-threaded execution was 30% on average with a memory utilization of approximately 200 MB. When using AVFD, the CPU utilization jumped to 85%, with memory allocation closer to 550 MB. This is because we had to keep a copy for the face detector, dlib, and our classifiers for each thread in memory; otherwise, we would have race conditions when threads are competing to access the classifiers that cause crashes or slowdowns.

There was no need for parallelization on the laptop PC, since we were already able to run the algorithm at the maximum frame rate of the camera. Thus, there would be no benefit to parallelize the code on the PC.

4.5 The FERNet Algorithm

As shown before, there have been many attempts to create a neural network dedicated to detecting facial expressions. The most popular approaches are *ResNet 34* and *ResNet 50*. These two networks are large and computationally expensive since they require approximately 20 and 23 million parameters, respectively. Thus, we propose FERNet, a small, computationally efficient and speedy neural network capable of detecting facial expressions from a wide range of angles with a relatively low number of computations with 325,479 parameters with a very minimal impact on accuracy.

The steps for FERNet are shown in Algorithm 3 and are described in the following subsections. REFER and FERNet share the first 7 steps in the algorithm.

4.5.1 Preprocessing and Face Detection

To perform automatic facial expression recognition, we have to initialize the camera, capture a new image, and transform it from a color image to a gray image (line 1). We then detect the face region in the image using the Viola-Jones algorithm.

4.5.2 Geometric Features Extraction

To obtain the most important geometric features, we use ASM to detect 68 pertinent points around the face. We then compute the cosine similarity for every pair of points using Line 3 in Algorithm 3, where u and v are any points within the 68 detected using ASM, and $C_{i,j}$ is the cosine similarity between two ASM points n_i and n_j. Since ASM gives 68 points, the resulting cosine similarity is a 68×68 matrix representing the cosine similarity from every ASM point to every ASM point. The benefits of using cosine similarity are that it is independent of the image scale and it does not exhibit a large variation in different poses.

Algorithm 3: FERNet processing for one video frame

```
Input: Red, green, blue pixel matrices $R$, $G$, $B$
Output: $Q_1, Q_2, ..., Q_7$
$K \leftarrow 0.299 \times R + 0.587 \times G + 0.114 \times B$; $F \leftarrow \text{VJH}(K)$; $\mathcal{P} \leftarrow \text{ASM}(F)$;
foreach $ASM\ point\ n_i \in \mathcal{P}$ do
    foreach $ASM\ point\ n_j \in \mathcal{P}$ do
        $u \leftarrow [x_{n_i}, y_{n_i}]$; $v \leftarrow [x_{n_j}, y_{n_j}]$; $C_{i,j} \leftarrow \frac{u.v}{\|u\| \times \|v\|}$
$Z \leftarrow \text{Imresize}(F, 68 \times 68)$; $B \leftarrow \text{LBP}(Z)$; $V_q = [C, B]$; $\boldsymbol{Q} \leftarrow \text{FERNet}(V_q)$
```

4.5.3 Texture Feature Extraction

Texture features can be good at detecting facial expressions, as they can detect ridges and wrinkles on the face, making them a good fit for facial expression recognition. However, analyzing texture features can be computationally expensive, such as when using a Gabor filter; hence, this work uses LBP for texture feature extraction, as it is effective, computationally inexpensive, and works in non-uniform lighting [127]. The use of LBP is denoted as LBP() in Algorithm 3. A size 3 ring is used for LBP, as we found it to be a good balance between performance and accuracy, meaning we only look at one pixel and the directly adjacent 8 pixels to obtain the LBP image. To compute the LBP on each pixel, we first resize the face image F to 68×68 using the function Imresize with bilinear interpolation, which results in a scaled face image Z. We then begin computing the LBP of the scaled face image Z by comparing the value of each pixel with its neighboring pixels. Based on which neighboring pixels are larger than the central pixel, we obtain a value between 0 and 255. The resulting values indicate the direction of intensity. *LBP* is explained in detail in [127].

4.5.4 FERNet

After obtaining both the texture and geometric features, we merge both features. Both features are represented by a 68×68 matrix, resulting in a concatenated matrix V_q of size 136×68. Matrix V_q is then given as input to the neural network FERNet shown in Figure 4.5. The FERNet architecture begins with a convolutional layer with rectified linear activation and 64 feature maps. We then implement a 2-by-2 max-pooling layer to reduce the size of the data to a fourth, followed by a dropout layer with a dropout parameter of 50%. As shown in Fig. 4.5, we repeat the same structure another time to further reduce the data, but this time with 32 feature maps. After that, we flatten the convolutional network to a dense layer of size 1088. We then add another dense layer of 256 neurons with a dropout of 50% and then another dense layer of 128 neurons with a dropout of 50%.

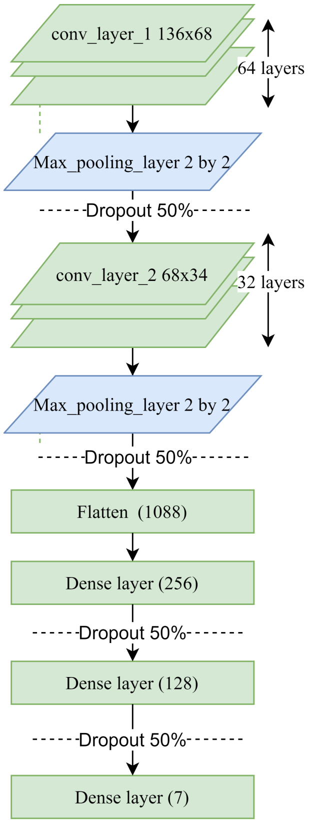


Figure 4.5 FERNet composition: the three-dimensional rectangles represent the convolutional layers, while the flat rectangle represents the dense flat layers

Finally, the output layer has 7 outputs, each providing the probability Q_i that the facial expression i was observed in the input.

4.6 Results

To verify the algorithms, we implemented the training code using Python on a Windows 10 laptop PC equipped with an i7-8750h Intel CPU and 32 GB of RAM. We used multiple libraries including `OpenCV` for frame capture and face detection, `dlib` to obtain ASM points [175], and `scikit-learn` to train the two-stage SVM model and test the results [176]. Tensorflow is used to train and test FERNet. For accuracy and performance testing, we used the same previously mentioned laptop PC, and we also tested the inference accuracy and performance using a Raspberri Pi 4 device with 4 GB of RAM.

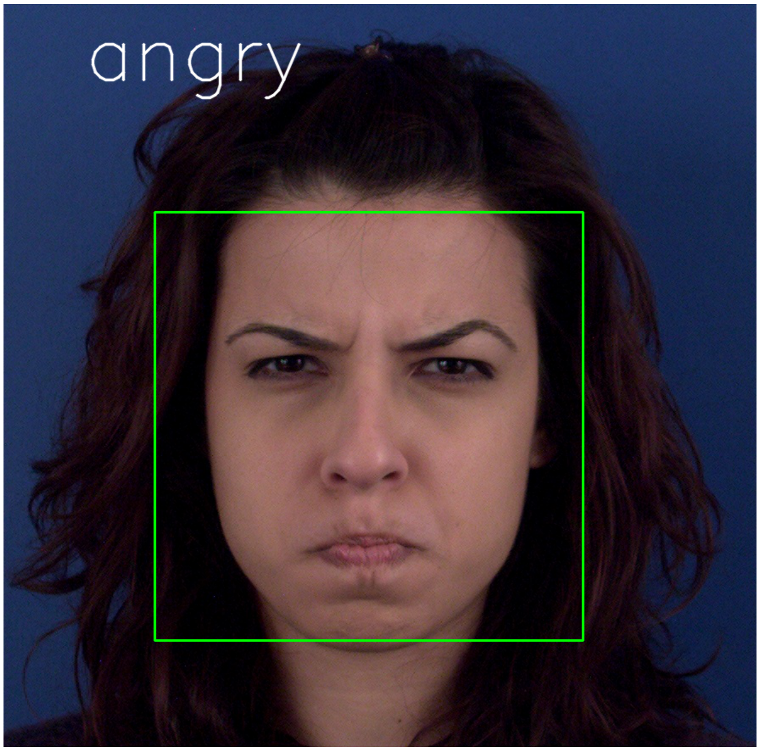


Figure 4.6 Example of correct classification from the MUG dataset with angry faces trained with CK+ datasets

4.6.1 Training and Testing Datasets

In our training and testing, we used four different datasets, all publicly available. We used three existing datasets: CK+, MUG, and KDEF. We also created a custom dataset called angled posed facial expression (APFE) [167]. This dataset contains approximately 15,000 frames of video filmed under different angles at peak expressions for 4 males with no acting experience aged between 25 and 30. The videos were shot at a resolution of 640x480 and at 30 FPS using a `Logitech c270` camera. The camera was fixed on top of a screen, and the volunteers were told to look at the camera and move their faces in concentric circles while performing facial expressions with different severities, to look up and down, and finally to make sure that the camera captured their faces from several angles. We also captured a 1080p video from an LG G6 smartphone to use for testing only in order to verify that the proposed algorithms work on different resolutions and different aspect ratios.

The CK+ dataset [131] contains 123 different subjects, both males and females, shot directly with a resolution of 640x490, for a total of 10,000 image frames. Each shot starts at neutral and increases until peak expression. The MUG dataset [165] contains images and videos for 52 different males and females shot at an 896x896 resolution at 20 FPS with 1032 videos and

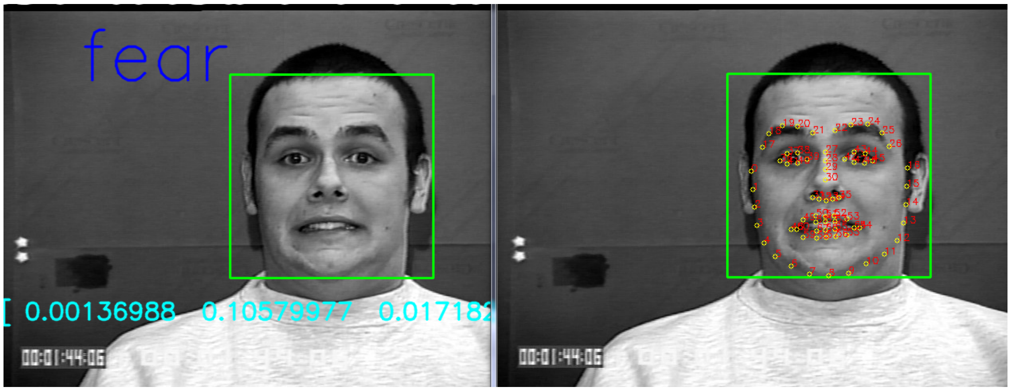


Figure 4.7 Example of correct classification from CK+ with fear faces

more than 100,000 images. Each video begins and ends with a neutral expression and peaks in the middle. The KDEF dataset contains 4900 photos of 70 subjects with 7 different facial expressions, each viewed from 5 different angles.

It is challenging to use *transfer learning* to train networks for facial expression recognition since the networks tend to overfit to the original data, making it difficult to target a new dataset [150]. Hence, in all our tests, we trained FERNet and *ResNet* from scratch.

4.6.2 Data Augmentation

Neural networks are well known to require a large amount of data to fine tune. Data augmentation methods that increase the size of the dataset can lead to an improved accuracy, as seen in [156], [157], and [159]. To see the effect of data augmentation on the detection accuracy, we implemented 5 types of augmentation for every image in each of the datasets. The 5 augmentations are image flip, random brightness adjustment, simultaneous flip and brightness adjustment, random positive rotation, and random negative rotation. In the image flip, we simply flip the image from left to right. In the random brightness adjustment, we convert the image from RGB to hue-saturation-value (HSV) and multiply the saturation and value components of each pixel by a random factor between 0.5 and 3 and convert it back to RGB. Finally, in the image rotation augmentation, we randomly rotate the image by a random angle between 0° and 30° for positive rotations or between -30° and 0° for negative rotations. The augmentation results in a dataset that is 6 times larger than the original dataset. Note that for all upcoming graphs and tables, the accuracy results are when using the data augmentations mentioned above, except for Table 4.2 where we only used the original dataset with no augmentation.

4.6.3 Accuracy of REFER

To test the accuracy of the proposed REFER algorithm, we trained REFER under different conditions to test for cross-validation accuracy and robustness for rotation.

First we tested for cross-validation accuracy by training with 90% of the data for every dataset independently and validating on 10% of the remaining data. We repeated each test three times and took the average of these runs. The results are shown in Table 4.3. While using REFER on the CK+ dataset, we obtained an accuracy of 80% in cross-validation when using data augmentation. Similarly, we obtained accuracies of 97.7%, 74.3%, and 97.7% for the APFE, KDEF, and MUG datasets, respectively, with the same 90-10 split between training and validation. The accuracy of predictions from the videos was measured frame by frame, meaning that we produce a classification for every frame and compare it to the actual facial expression.

Second, we tested the robustness to rotations by training each dataset with non-rotated images and testing on randomly rotated images. The results in Table 4.2 show the accuracy of REFER on all datasets for different ranges of uniformly distributed random rotations, which indicate that REFER is robust to rotations in wide ranges of angles.

These results show that this algorithm works well with the different camera systems, resolutions, apertures, distances, aspect ratios, and lighting, which are covered by the four considered datasets, and that it can also make accurate predictions at an angle. Figures 4.6, 4.7, 4.8a, 4.8b and 4.8c show some correctly predicted samples from the MUG, CK+, and APFE datasets.

Table 4.3 provides a comparison of the state-of-the-art literature for different approaches. Both REFER and FERNet exhibit some of the highest accuracies in cross-validation testing compared to other geometric- or neural-network-based approaches while having the fastest detection rate.

Table 4.4 shows the internal probability from classifiers in action when attempting to predict the facial expression from Figure 4.8c. We can see in Table 4.4 that individual features gave a different prediction without a high confidence. While the second stage classifier not only chose the correct facial expression, it also did so with a high confidence. REFER correctly predicted that “disgust” is the correct expression, as it has built a decision hyperplane that deduces which descriptors are more relevant for each expression.

For cross-dataset validation, we trained REFER on one dataset and tested the accuracy on the remaining datasets. The results are shown in comparison with FERNet and *ResNet 34* in Figure 4.10. Note that for graph readability, we omitted the results with KDEF since it

Table 4.2 Accuracy percentage when training from original images and testing with randomly rotated images

Angle range	**CKP**	**MUG**	**KDEF**	**APFE**
0°	75.74%	96.69%	73.44%	97.37%
[−15°,15°]	72.74%	95.84%	69.52%	94.81%
[−30°,30°]	71.07%	93.96%	66.55%	93.29%
[−45°,45°]	69.15%	91.11%	65.14%	93.17%
[−60°,60°]	68.68%	87.78%	61.8%	91.88%
[−75°,75°]	67.48%	85.25%	61.48%	92.19%

showed a similar trend. We also omitted the results of testing without data augmentation for graph readability. For example, training with CK+ and testing the accuracy on the MUG dataset, we obtained an accuracy of 70.15%, which increased to 72.1% when we used data augmentation. In all cases, we observed a small improvement when using data augmentation, which does not, however, have a bearing on the ranking of the solutions. We also used previously non-trained video sequences shot at 1920×1080 from the front-facing camera of LG G6 at different distances and locations while moving the camera around the face to see all angles while ensuring that all the faces were captured within the shot. We copied these shots to a PC and tested them with REFER, obtaining an accuracy of 97%.

While REFER achieves a good accuracy, there are several cases of failure in REFER that can be improved in the future. Figure 4.83 shows an instance where a face was not detected due to the high tilt angle. The Viola-Jones failed to detect a face failing the remainder of the algorithm. Figure 4.8d shows a misprediction, where REFER detected the face as a sad face while it is a surprised face. We can notice that the failure was in part due to a failure in ASM to accurately register points when at a large angle, the black beard and relatively dim lighting might have also negatively affected the accuracy of ASM and subsequently the remaining steps in the classification.

There are few instances where the detection fails, even with proper face registration due to the difference in how people's face moves while making facial expressions.

4.6.4 Accuracy of FERNet

Since FERNet is based on a CNN, we compared it with two popular approaches: *ResNet 34* and *ResNet 50*. We implemented all three networks using TensorFlow with the same dataset as REFER. We first detected the face in the image using Viola-Jones face detection. Then we resized the area of the detected face to fit the input layer of the network. Thus, we gave each network its ideal data shape to work with.

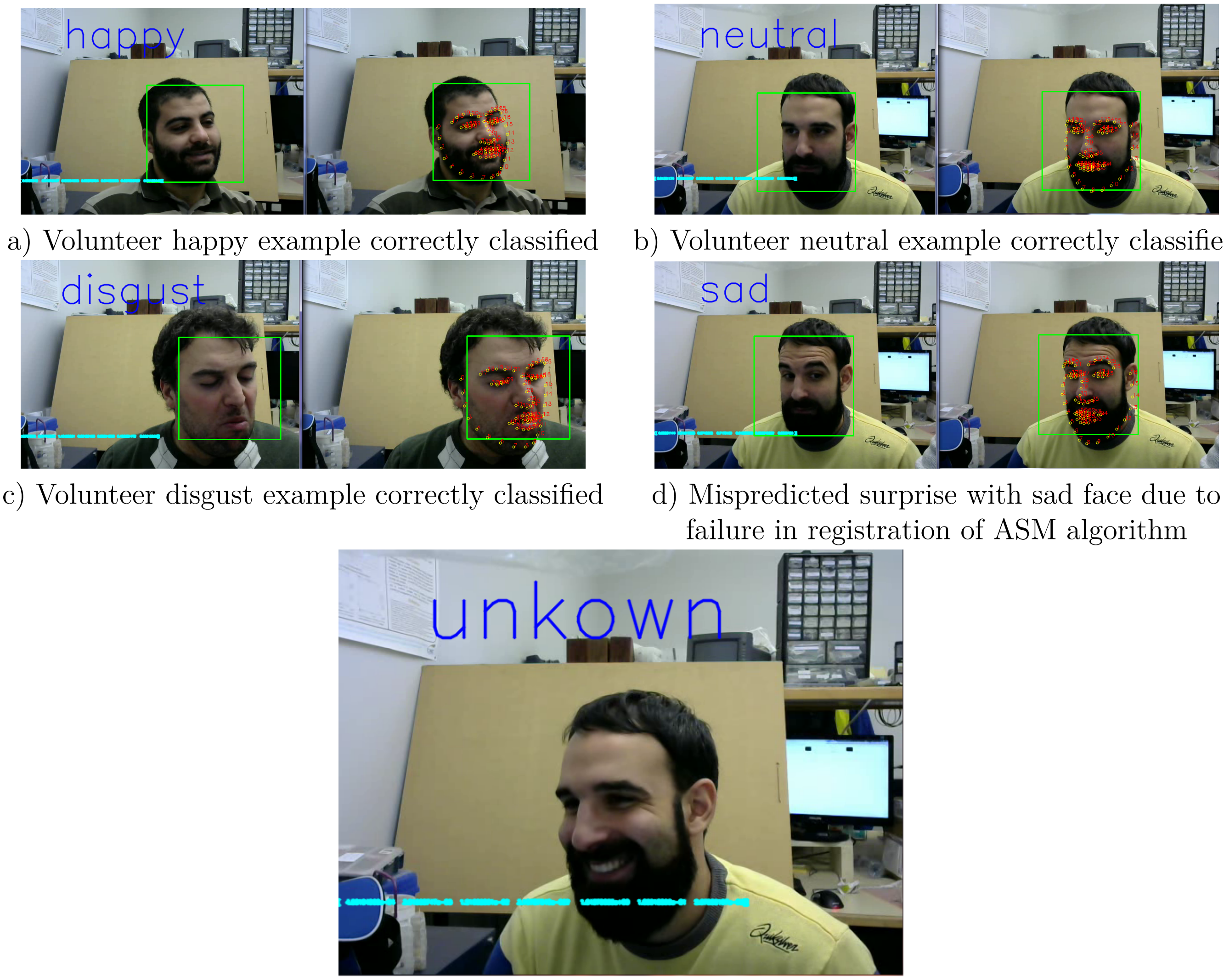
Table 4.3 Comparing REFER and FERNet with the state-of-the-art facial expression recognition techniques

Dataset	**Method**	**Accuracy**	**Real-time**	**Feature types**	**Classifier** (# of param)
CK+	ASM [59]	77.5%	2.4 FPS	geo	SVM
	2D-DCT [172]	77%	no	geo+texture	SVM
	GA-SVM [60]	88%	N/A	geo	SVM
	CNN [157]	97%	no	CNN	CNN (9.4M)
	GAN+CNN [151]	99%	N/A	GAN + CNN	CNN
	HoG+PCA [162]	82%	no	geo	DNN
	Resnet 34	43%	no	CNN	CNN (21M)
	Resnet 50	89%	no	CNN	CNN (23.6M)
	REFER	**80%**	**14 FPS***	**geo**	**SVM**
	FERNet	**83%**	**14 FPS***	**geo + CNN**	**CNN (0.3M)**
MUG	VGG-face [158]	94.5%	no	VJ+VGG-face	VGG-face (138M)
	GA-SVM [60]	88%	N/A	geo	SVM
	CNN [163]	87%	N/A	CNN	CNN
	HiNet [164]	88.6%	N/A	CNN	CNN (1M)n
	DWMV [134]	96%	N/A	VJ+SURF	Ensemble
	Resnet 34	55%	no	CNN	CNN (21M)
	Resnet 50	98%	no	CNN	CNN (23.6M)
	REFER	**97.7%**	**14 FPS***	**geo**	**SVM**
	FERNet	**98%**	**14 FPS***	**geo + CNN**	**CNN (0.3M)**
KDEF	CNN [157]	75.85%	no	CNN	CNN (9.4M)
	VGG. [154]	72%	no	VGG	VGG (138M)
	Resnet 34	28%	no	CNN	CNN (21M)
	Resnet 50	66%	no	CNN	CNN (23.6M)
	REFER	**74.3%**	**14 FPS***	**geo**	**SVM**
	FERNet	**76%**	**14 FPS***	**geo + CNN**	**CNN (0.3M)**
APFE	Resnet 34	47%	no	CNN	CNN (21M)
	Resnet 50	97%	no	CNN	CNN (23.6M)
	REFER	**97.7%**	**14 FPS***	**geo**	**SVM**
	FERNet	**99%**	**14 FPS***	**geo + CNN**	**CNN (0.3M)**

* Frames-per-second performance was measured on live video capture.

Table 4.4 Breakdown of the classification probabilities for each descriptor for the example of Fig. 4.8c

Descriptors	**Angry**	**Neutral**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**
Whole face	0.042	0.004	0.23	0.007	0.69	0.001	0.0107
Jaw line	0.137	0.196	0.161	0.172	0.029	0.206	0.095
Inner mouth	0.021	0.033	0.279	0.095	0.388	0.023	0.158
Outer mouth	0.015	0.036	0.406	0.308	0.007	0.0344	0.19
Left eye	0.057	0.056	0.223	0.062	0.425	0.163	0.011
Left eyebrow	0.279	0.104	0.27	0.054	0.103	0.172	0.015
Right eye	0.03	0.024	0.251	0.0429	0.585	0.057	0.006
Right eyebrow	0.12	0.215	0.141	0.122	0.133	0.228	0.037
REFER	0.0057	0.0026	0.975	0.004	0.004	0.002	0.003



a) Volunteer happy example correctly classified

b) Volunteer neutral example correctly classified

c) Volunteer disgust example correctly classified

d) Mispredicted surprise with sad face due to failure in registration of ASM algorithm

e) Failed to detect face due to large pose angle, no prediction can be made if face was not detected

Figure 4.8 Example of REFER in action

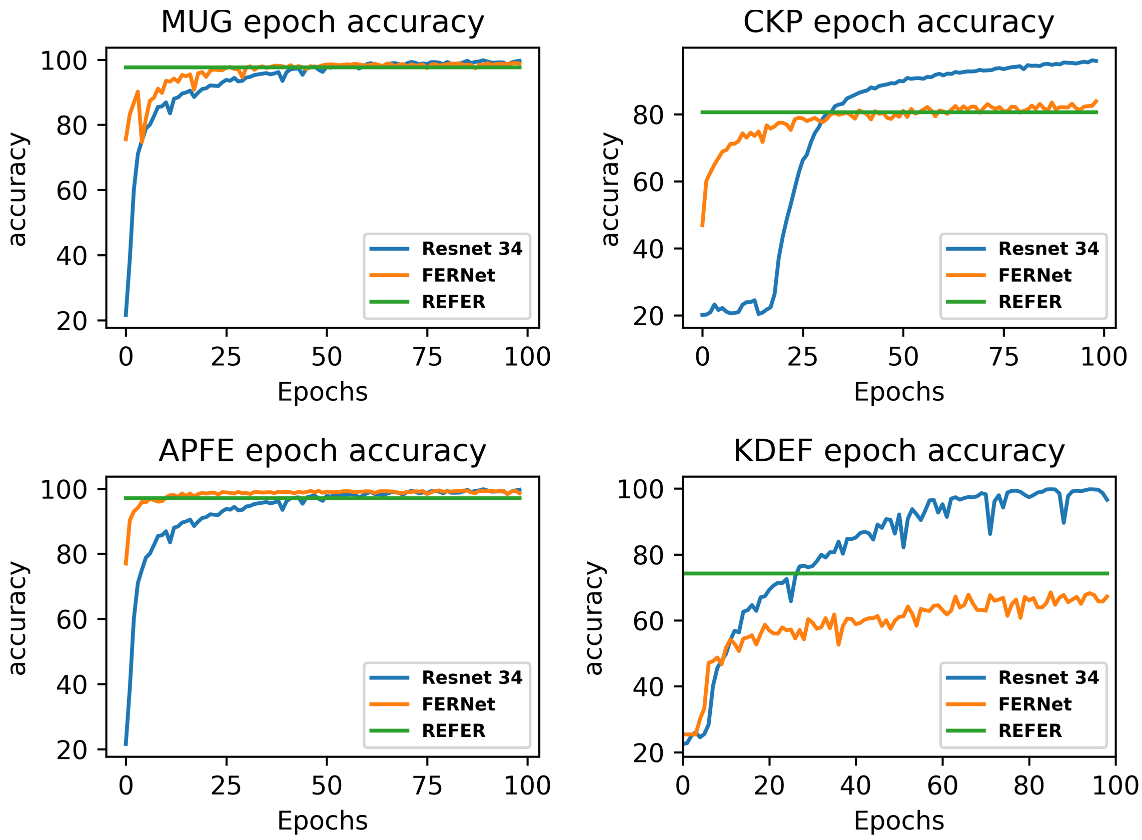


Figure 4.9 Accuracy for the first 100 epochs comparison between *ResNet 34*, FERNet, and REFER for each dataset

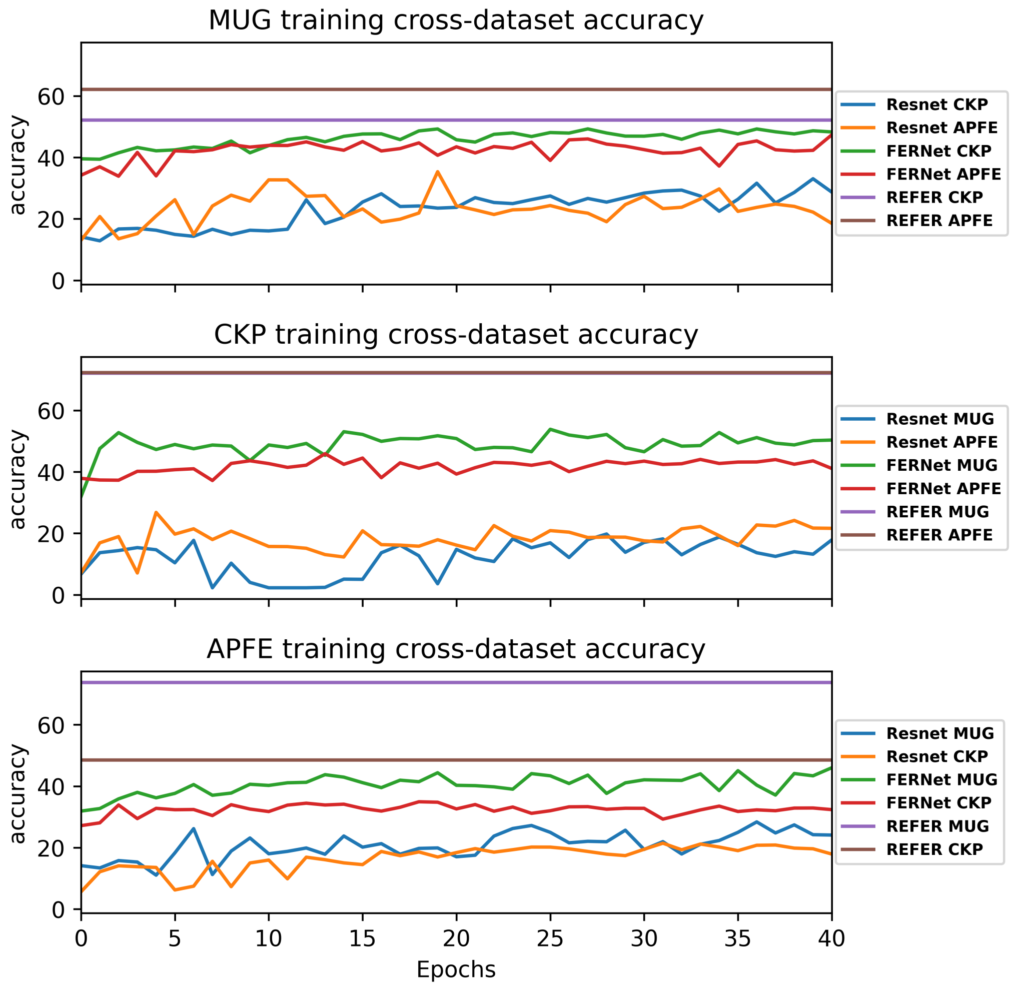


Figure 4.10 Cross-dataset validation comparison between *ResNet 34*, FERNet, and REFER as a function of the training epoch

The top is when training using MUG and validating using the CKP and APFE datasets, the middle is when training using CKP and validating using the MUG and APFE datasets, the bottom is when training using the APFE dataset and validating using MUG and CKP. The accuracy of REFER does not vary since there is no notion of epoch in SVM training

Figure 4.9 shows the epoch accuracy of FERNet compared to *ResNet 34* and to REFER. In these cases, the same data is used for both training and testing. Since REFER is an SVM-based approach, there is no need for epochs; hence, the value is constant. FERNet gained its accuracy much faster than *ResNet 34* with epochs and required a significantly lower number of epochs to settle. While *ResNet 34* achieved close to a 100% accuracy on the training set in all datasets, it performed poorly in the cross-validation tests, which indicates overfitting. FERNet training and cross validation accuracy were very close, indicating very little to no overfitting.

To compare the cross-validation accuracy, we used 90% of the data for training and 10% for validation with 499 epochs for all cases, except for REFER, where we used the same 90-10 split for training and testing but without having to use epochs, since it is based on SVM. The results are the bottom four of Table 4.3 for each dataset. This shows that FERNet outperforms *ResNet 34* in all tests and *ResNet 50* in three out of four datasets in cross-validation testing.

Figure 4.10 shows the results of cross-dataset validation accuracy. In these tests, we trained the network with one dataset and used the other two for testing. REFER has a significant lead in cross-dataset validation over other approaches, followed by FERNet. Both of our approaches performed significantly better than *ResNet 34* in all tests, with a better absolute accuracy of 20% or more.

ResNet 34 cross-dataset validation accuracy shows that there were no gains with the epochs; on the other hand, FERNet did show improvement with the cross-dataset training for every dataset. The combination of our smaller network with the large dropout rate ensures a better generalization outcome, and our simple descriptors give it a better starting accuracy, even at epoch 0. However, there is a large difference between the epoch accuracy and cross-dataset validation accuracy, indicating that current datasets do not have sufficient variations to allow for a generalized solution. Moreover, as seen in Figure 4.10, REFER performs better than the CNN-based approaches (FERNet and *ResNet 34*) in the cross-dataset validation. Therefore, it can be expected to perform better in less-controlled environments.

4.6.5 Performance Testing

One of the main benefits of using a smaller network is to have a network with a good real-time performance. Given that FERNet has 325,479 tunable parameters compared to 20 million parameters in *ResNet 34*, we expect the network to train faster and to run inference faster. To measure the performance, we used two systems. One laptop PC with an i7-8750h with 32 GB of RAM with an RTX 2060 GPU and a Raspberry Pi 4 with 4 GB of RAM. Training the network took 30 minutes for FERNet and 4 hours for *ResNet 34* when using the GPU to train the networks. Once those were trained, we ran the same network using a laptop . To run the networks on a Raspberry Pi 4, we saved the trained network to a file on a PC and then moved the file to the Raspberry Pi 4 and installed the required library, frameworks, and necessary system environment variables.

To measure the FPS, we ran a code that includes all the steps of automatic facial detection from start to finish, including opening the camera, obtaining a new frame, converting a color image to grayscale, detecting the face, running the processing and feature extraction, running the network, and finally obtaining the prediction. In the case of *ResNet 34*, we only need to detect the face and resize it to fit the input layer of the network and skip the remaining preprocessing steps. We measure the performance by counting the number of predictions divided by the number of seconds in the time elapsed. We ran the test for approximately 5 minutes of live video capture. Table 4.5 shows the performance summary when running the full stack of automatic facial expressions. On the laptop GPU, we found that running *ResNet 34* resulted in an average of 15 predictions per second, while the predictions when running FERNet are a stable 30 FPS, which is the maximum frame rate that can be achieved with the camera. On the Raspberry Pi 4 we ran three tests. Using *ResNet 34*, we determined that the average frame rate is 2.75 FPS. Running FERNet resulted in 11.5 FPS on the Raspberry Pi 4. Flattening the network and running it with TensorFlow-lite on the Raspberry Pi 4 resulted in 14 FPS.

The results show that, despite having more preprocessing steps and feature extraction than *ResNet 34*, FERNet ran up to 5 times faster with a very small detriment to the accuracy, even surpassing both *ResNet* networks, with the cross-validation accuracy indicating better results on the fresh data. Running REFER on the Raspberry Pi 4 results in 6.3 frames per second when using a single-threaded approach. Using AVFD on Raspberry Pi 4 resulted in 14 frames per second. Note that while AVFD does improve the throughput, the latency of detection will not improve since the detection pipeline still needs to go through all the steps before reaching a result, but it can reduce the latency of image capture since we do not have to wait for the current frame to finish to begin working on the next frame. Another thing to note is that both REFER while using AVFD and *ResNet 34* require more than 80% of the CPU time when running. Single-threaded REFER and FERNet require approximately 30% and 40% the CPU time, respectively.

Table 4.5 Number of frames per second for ResNet 34, FERNet and REFER

Network	**ResNet 34**	**FERNet**	REFER
Laptop	15	30	30
Raspberry Pi 4	2.75	11.5-14	6.3-14

From the performance and accuracy results, we can see that FERNet is the fastest approach with the lowest latency for detecting facial expressions with a high accuracy on familiar data, making it ideal for applications where we can tune the network before applying, while still working well on new data. REFER truly works well with fresh data, as shown in the cross-dataset validation, while being fast enough for many applications, while less accurate than FERNet or ResNet on familiar data.

4.7 Conclusion

This paper presented two novel approaches for real-time automatic facial expression recognition called REFER and FERNet. The first accurately predicts facial expressions at different angles and distances with up to a 97% accuracy in real time on the APFE dataset, even on mobile hardware, such as a Raspberry Pi 4.

REFER contains four improvements that can be used independently: custom descriptors based on geometric features that allow detection at different angles and scales, ensembles of SVM classifiers for a better global prediction from local predictions, temporal adjustments to improve consistency and accuracy over sequences of predictions, and AVFD for improving the speed of detection. With all of the previously mentioned improvements, it has been shown to have the best cross-dataset performance. On the other hand, FERNet is a compact neural network that achieves a similar accuracy to the state of the art with a significantly faster performance on the same datasets and a better accuracy on new data than *ResNet*.

CHAPTER 5 COMPLEMENTARY EXPERIMENTAL RESULTS

In the previous chapters, we discussed the retinal model to emulate retinal functions. While the developed model is ideal for future prostheses which can target high-resolution stimulation devices, we are currently limited by the number of stimulation points. Modern optogenetic platforms can target individual cells using gene therapy along with high precision focused lens. However, such an approach cannot yet be used in retinal stimulation due to the size and the power requirement for such devices. The goal of the project is to design portable and efficient prostheses that can provide real use for current stimulation technology. The full proposed system can be seen in Figure 5.1, the top image of this figure shows the retinal processing flow for a healthy retina, the bottom shows our proposed optogenetic stimulation system aims to bypass non-functioning layers and directly stimulate the RGC before propagating the visual information to the next layers..

In this chapter, we will present the testing platform for driving optogenetic prostheses. At the heart of the platform is the Raspberry Pi 4. We chose this device since it is an affordable, portable, well-supported, and low-power platform that can run complex programs in real-time. It is also capable of connecting to various outputs via a multitude of ports and interfaces. In the proposed platform Raspberry Pi 4 acts as a visual processing unit that can send signals to a stimulation device via serial protocol interface (SPI). This allows it to directly control stimulation devices with low latency and accuracy. It can also be powered by off the shelf components such as a lithium-ion battery which ensures long battery life and can be easily updated in the future to take advantage of better and more efficient models. Raspberry Pi 4 also can take advantage of USB accelerators such as Intel compute stick or custom USB-based accelerators.

Figure 5.2 shows the schematic of the test platform. The Raspberry Pi is connected to an off the self 8 by 8 LED matrix via MAX7219 driver. The interface is connected to Raspberry Pi 4 via SPI and is powered directly by the Raspberry Pi. The Raspberry Pi itself is powered by an off-the-shelf 3.7 Volt 3800 mAh battery connected directly to the Pi, hence it can completely run on battery for hours. hence, it is a completely self contain system. We used a Logitech c270 webcam to capture live images through a USB port. The Pi connects to the MAX7219 with 5 pins, pin 2 is used as a supply voltage (VCC), and pin 6 is ground (GND), these are both used to power the MAX7219 along with the LEDs, pin 19 is used on data input for the MAX7219 (DIN) it is referred by the Raspberry Pi as GPIO 10 or MOSI, pin 24 in chip select (CS) if we want to cascade multiple LED matrix or GPIO 8 (SPI CE0) as

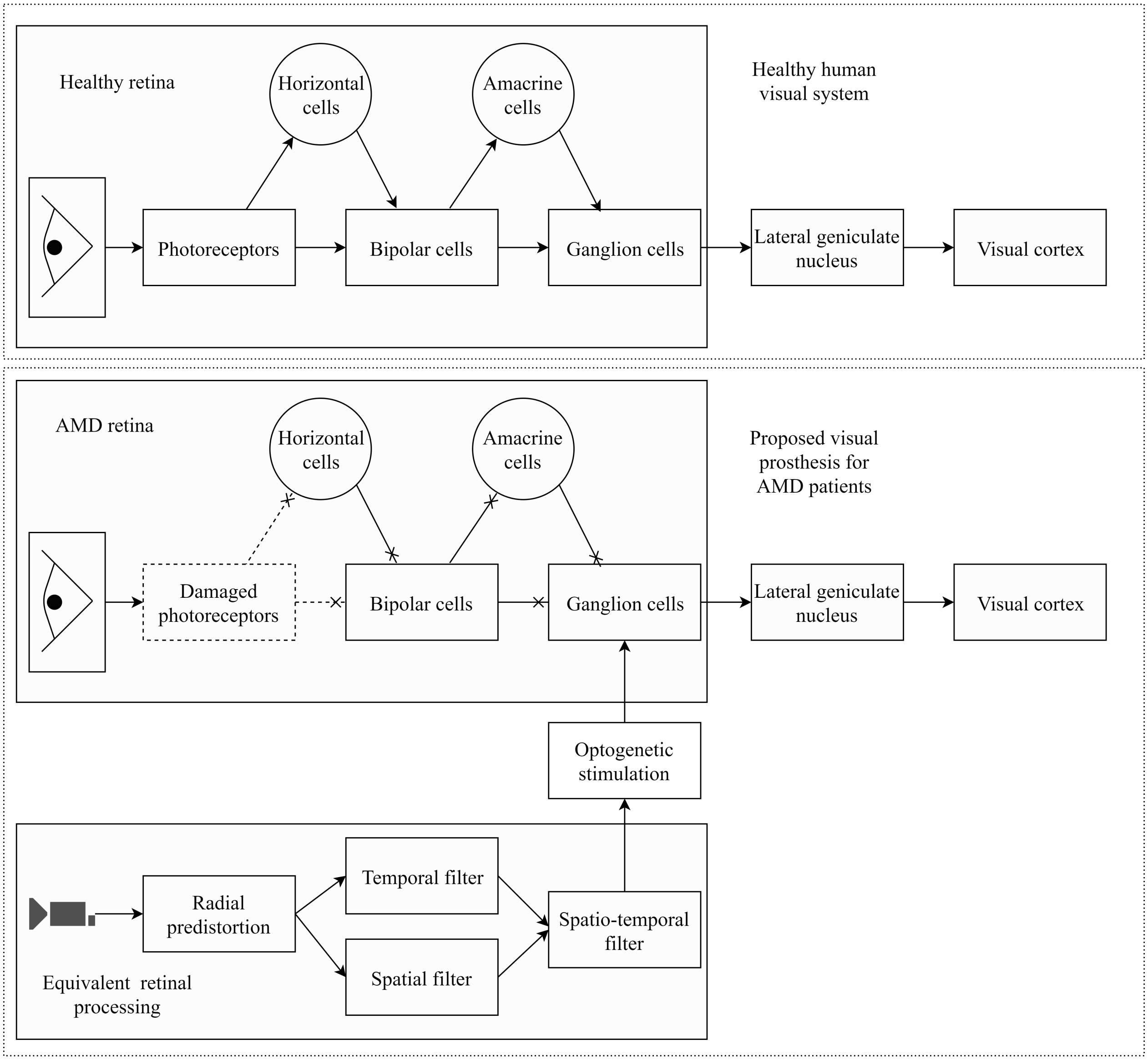


Figure 5.1 Simplified arrangement of different layers in the retina (top) vs proposed prosthesis for AMD treatment

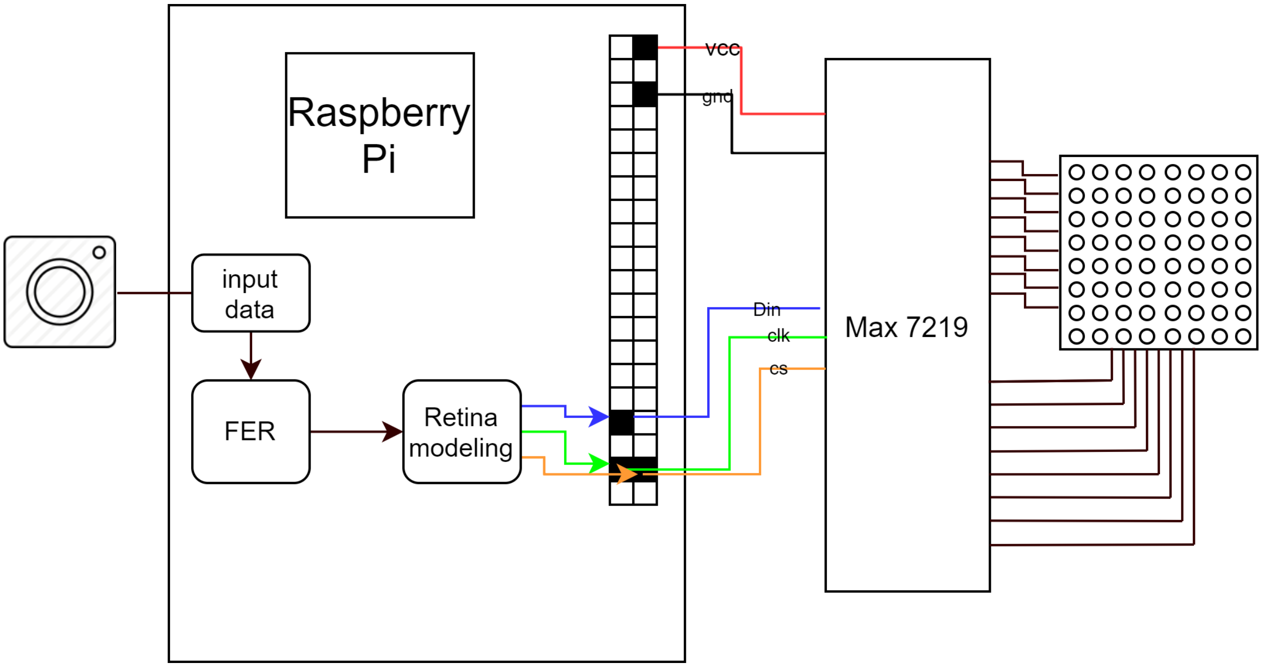


Figure 5.2 Test platform schematic

referred by the Raspberry pi, and pin 23 is the SPI clock (CLK). These three pins are used to communicate the desired stimulation pattern. Note that the LED matrix used here is a placeholder for the custom-designed LED matrix array that is used in [65, 66].

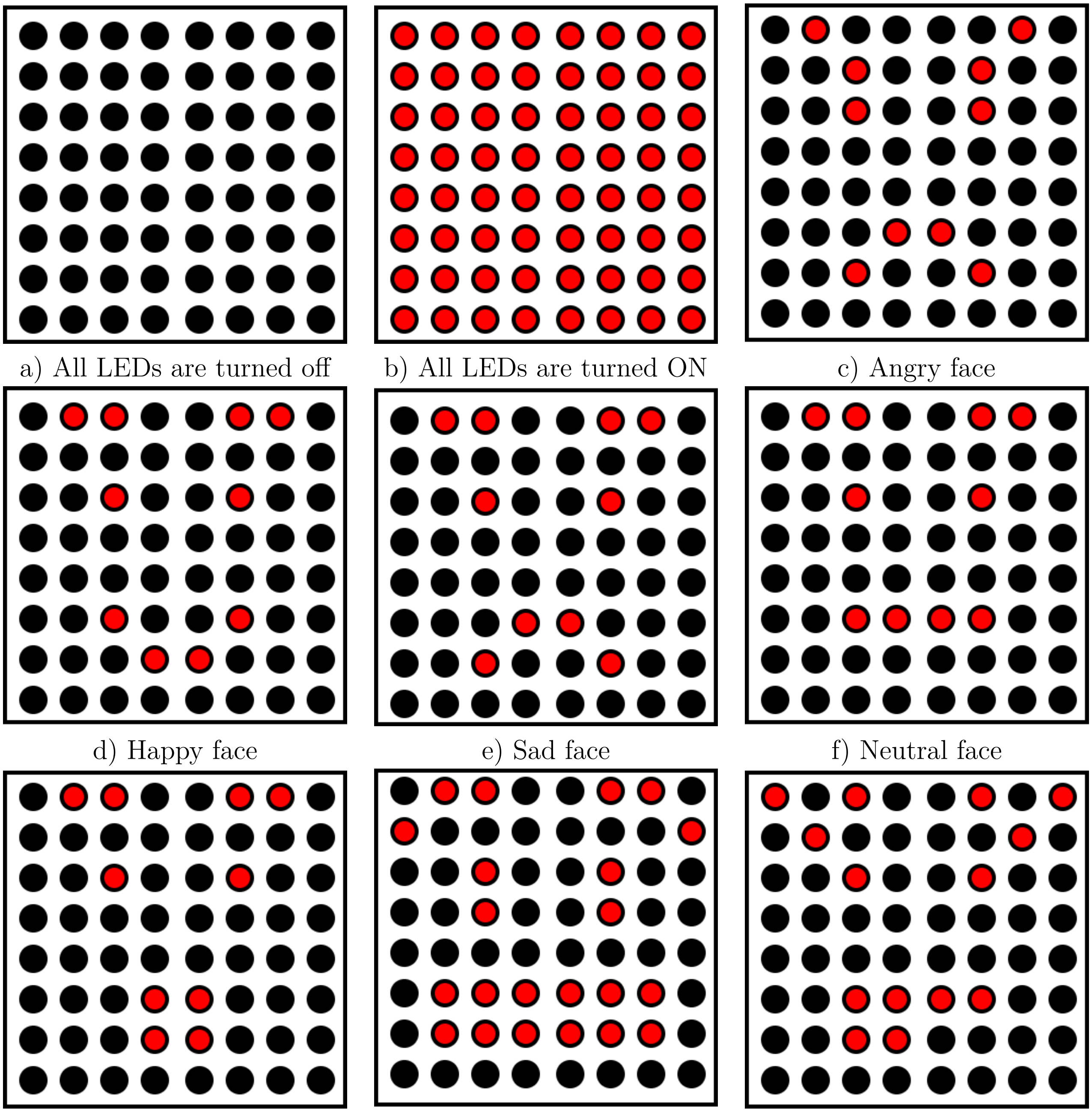
Figure 5.3 shows currently implemented faces in the platform. The goal is to build a distinct representation of the most important aspects of the world and send this representation to an 8 by 8 pixel matrix. But for now, we focused on facial expressions as it is a crucial part in communication. The faces shown were presented to a group of 4 different students and they all managed to guess the correct facial expression easily, except for fear and disgust, which are harder to convey with a limited number of LED lights.

Figure 5.4 shows an example of the entire system in action. Figure 5.4a shows an example of the powered-off system. With a press of the power ON button would turn on immediately. The system is assembled to work as an independent unit and does not need any additional input to start the desired programs. It is running Linux Raspbian OS. We modified the system boot sequence to allow us to launch our custom scripts without any user input. The code automatically starts and attempted to detect facial expression at boot (REFER or FERNet), once the facial expression is detected, the program sends the corresponding pattern of the detected face to the LED matrix directly via SPI. Since the system is based on a full version of Linux, there is about 1 minute until the system is fully booted. During that time the facial expression detection is OFF. Once the algorithm for facial expression detection starts the LED light on the camera starts

Figure 5.4a-5.4d shows three example of the platform detecting live facial expression. Figure 5.4b represent the surprised face, Figure 5.4c shows the happy face, and Figure 5.4d represent the neutral face. The system is powered by a 3.7 volt Lithium battery without any additional power source. The battery is connected directly to the micro-USB port in the Raspberry Pi system. Attached to the Raspberry Pi are a USB camera and the LED matrix with the MAX7219 driver chip. The scripts to run the algorithms are based on Python. The system is configured to run the necessary scripts automatically at boot, hence no need for external input to start the facial expression detection. Communicating over SPI is also done via Python. To run the code we can use either REFER or FERNet to detect facial expression in real-time, we also dedicate a thread to handle reading and writing to the LED matrix. Once we write to the matrix the facial expression stays until it is updated with the new value. To update the value of the LED matrix we have to initiate a clear command of the previous value and then update the value with the new input. The transition between two facial expressions is instantaneous, there is no perceived flickering for any LED light. When the system is working we can see the small LED light next to the camera would light up.

5.1 Conclusion

In this chapter, we presented a simple battery power platform that can load directly the desired software that reads the retinal model and facial expressions in real-time and outputs the signal into a placeholder LED matrix with the same dimensions without any input from the user making it extremely simple to use. The battery uses Lithium-ion cells and can be charged with a standard micro-USB cable.



a) All LEDs are turned off

b) All LEDs are turned ON

c) Angry face

d) Happy face

e) Sad face

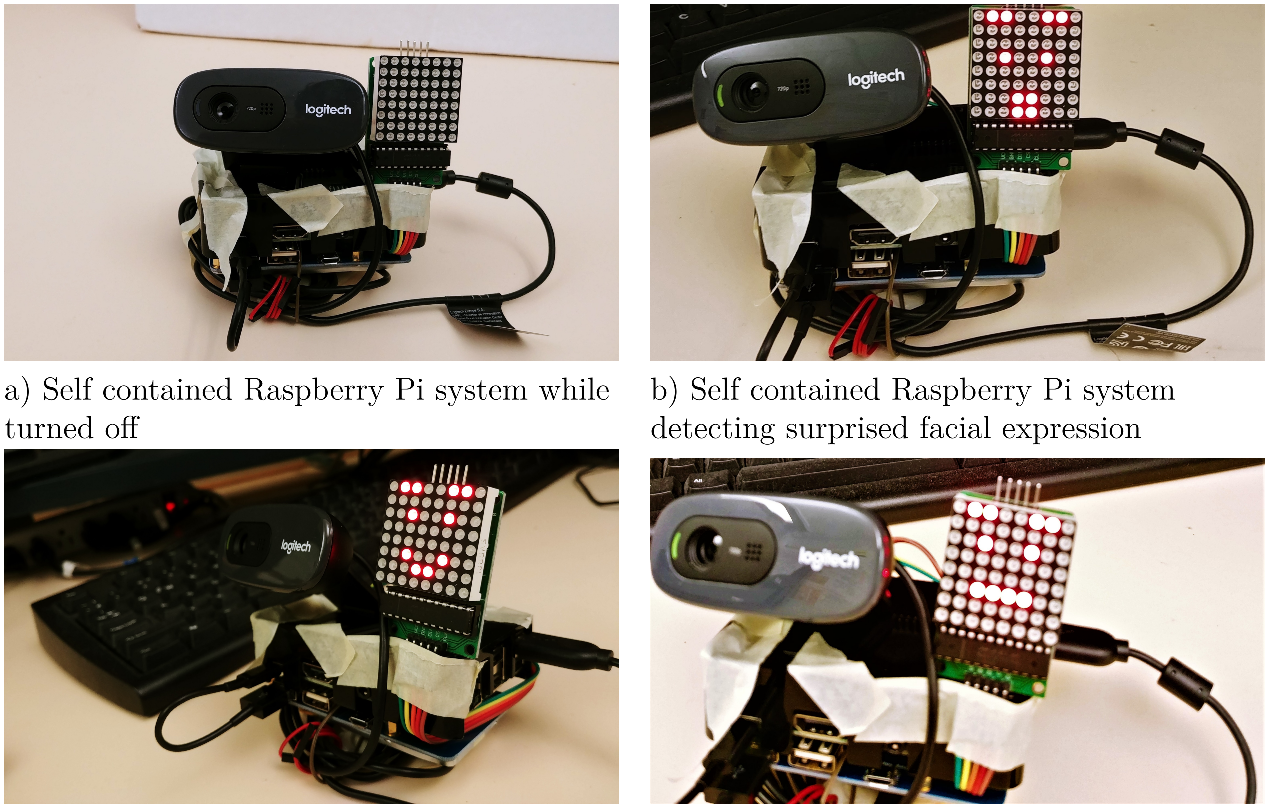
f) Neutral face

g) Surprised face

h) Fear face

j) Disgusted face

Figure 5.3 Representation of all type of output we implemented in an 8 by 8 matrix



a) Self contained Raspberry Pi system while turned off

b) Self contained Raspberry Pi system detecting surprised facial expression

c) Self contained Raspberry Pi system detecting happy facial expression

d) Self contained Raspberry Pi system detecting neutral face

Figure 5.4 Images from the entire system in action

CHAPTER 6 GENERAL DISCUSSION

This thesis is intended to be the starting step for prostheses to do for vision what cochlear implant did for hearing. Throughout this thesis, we attempted to provide a platform to drive visual prostheses. This starts by explaining the vision works and how AMD affects visions. We then try to put that understanding into a simplified model that mimics the retina and the effect of AMD on the retina (Objective 1), by doing so we can allow prostheses to more easily compensate for missing visual information in the macula using optogenetic stimulation. Since the technology for creating visual prostheses hasn't progressed to use a large number of stimulation points, we wanted to create something that works with current technology limitations. That is we developed a FER algorithm that can detect facial expression to facilitate communication (Objective 2), and it is an example of replacing what is in a scene with a simplified representation of what is in the scene.

6.1 Summary of contribution

Creating a reliable implant for retinal prostheses has always been difficult, from modeling to safety, to reliability, to effectiveness, and invasiveness. We believe in optogenetics has the potential to solve safety issues and the effectiveness issue without being an invasive procedure. Objective 1 attempts at creating a fast and effective retinal model that mimics the major pathways that should work with prostheses targeting the macula. We designed our model to take into account the latest research in the major pathways in the human macula and simplify them to work in real-time on a portable platform no larger than a typical VPU. While most real-time models only use spatial data, our proposed model uses both spatial and temporal aspects of the macula while accounting for light adaptation and simple radial structure for ganglion cells in the cornea, all in real-time on a portable device thus achieving Objective 1.

For Objective 2, we created two approaches, we created a facial expression recognition system that can detect facial expressions in real-time, under different poses, with a wide range of orientations. Both approaches are based on an active shape model with custom descriptors. The first uses cascaded pSVM along with custom-made descriptors with logarithmic scaling of distances while vertically and horizontally scaling the desired face. The second approach is using a mixture of cosine-distance for geometric scaling and LBP for texture feature extraction and merges the two descriptors into a relatively small network designed not to overfit the data. Our approach shows that is matching or beating state-of-the-art models in prediction accuracy in cross-validation accuracy yet show massive gains over the state-of-the-art in performance, and in cross-dataset accuracy due to the robustness of the pre-processing steps that reduce the need of deep layers to extract hidden information. We also showed that the platform can both drive run FER and drive off-the-shelves LED matrix in real-time as a placeholder for the custom-designed microLED used for optogenetic stimulation.

6.2 Drawbacks of the retinal model

Our model is based on the state-of-the-art model available in the literature, while we tried to be as accurate as possible, there are still a lot of missing pieces that prevent us from saying the model fully represents the macula. First of all, the retina layers are not fully known to us. Hence, it is impossible to have a fully accurate representation of the macula. But even from the data that we know about the macula, the spatial and temporal response of the human retina are not uniform due to the apparent randomness of the connection between the different layers, and the non-uniform latency of each cell. The proposed model is not meant to be completely accurate, rather a simplified model that should work well with prostheses. unfortunately, the model can't be completely validated before doing some basic human trials which are still far off in the future.

6.3 Possible Improvements for the FERNet model

Despite designing our model to maximize accuracy and minimize overfitting, it is still not as accurate with processing texture data as larger system networks. Despite having much better cross-dataset accuracy than state-of-the-art algorithms, there is still a big difference between cross-validation accuracy and cross-dataset accuracy which indicates that it is still room for lots of improvement, this can be helped with more datasets having a wider variety of poses and angles, and more datasets in the wild.

6.4 Platform Constraint

The goal of this thesis was to create a platform for optogenetic stimulation. Due to safety regulations and ethical concerns we cannot run these tests on any human. We wanted to first demonstrate that optogenetic stimulation is working in vitro by recording activity when the optogenetically treated cell line would be exposed to the light from the custom-designed microLED. Once that is successful we can try and create an in vivo test on mice, to see if they can distinguish between different stimulation patterns. We can then validate a working prototype of the system. Unfortunately, we ran into obstacles in the assembly of the microLED with the driver circuits which prevented further test on the custom-made microLED. So we used an off-the-shelve LED matrix to show that the system can run both FER and image processing in real-time while also driving a LED matrix-based system.

6.5 Future Considerations

Other limitations of the optogenetic visual prostheses need to be addressed before it becomes ready for testing on AMD patients. Optogenetic stimulation has already started shifting away from blue light stimulation to red light stimulation. This is because red light carries less energy than blue light and is safer to operate near human tissue which might suffer damage from extended use of blue light. A second consideration is the progression of the diseases. During later stages of AMD, retinal remodeling would make future rescue efforts more difficult. So prostheses need to be able to adapt to individual visual sensitivity, and to a changing visual acuity over time to ensure long-term operation. Finally, as human trials approach, we need to make sure that side effects can be minimized and controlled. There is no information for the long-term effect of optogenetic stimulation on human vision, and how the human brain would adapt to the new form of contentious stimuli in the retina, or how the stimuli might degrade or change over time.

CHAPTER 7 CONCLUSION AND RECOMMENDATIONS

In this thesis, we presented a portable platform that processes real-time images and provides an image for a wearable micro-stimulator device that is targeting visual enhancement for people suffering from AMD. To reach our objective we developed a simplified model of macular processing which can emulate the high-level midget cell pathway processing in real-time that can reach up to 100 images processed per second on Raspberry Pi 4 with lots of headroom for more complex solutions in the future. The model we presented uses both spatial and temporal and takes into account light adaptation which compensates for the ring structure inside the macula. This is ideal for future high-resolution optogenetic-based micro-stimulator.

We also presented two novel approaches for real-time automatic facial expression recognition. The first is called REFER and the second is FERNet based on multi-stage pSVM and CNN. Our proposed algorithm can detect facial expression with high accuracy, in real-time under different poses, angles, and scales, and it is ideal for human-computer interaction, or scene replacement in a low-resolution simulation platform with limited compute capability.

We also showed a prototype that is capable of running facial expression recognition and outputting to a LED matrix to simulate optogenetic signals in real-time. The prototype is fully automated, requires no user input, and can run for an extended period of time from a lithium-ion battery. The system is from easy to obtain off-the-shelve components. The prototype we have shown and the model we designed aren't meant for deployment directly in a medical trial, but it is meant as a first step in creating a platform where visual prostheses can be developed. Moving from a prototype to a usable device would still a lot of more research and many more limitation to be solved before what is shown in this thesis can be applied in medical trials.

7.1 Limitations

The proposed retinal model is based on compiling several state-of-art studies into one model, since it is too early for any sort of human trials, there is no way to empirically prove the accuracy of the model. However, since our model incorporates more elements and pathways of any used in previous visual prostheses, it should better than other approaches. Another limitation of the model is the missing gaps in knowledge in the retina itself, since the millions of neurons forming the macula are impossible to measure, model, and replicate. Current technology can't simply resolve that fine level of details needed for a fully accurate model.

7.2 Recommendations

Despite the mentioned limitation above, visual prostheses can still provide an improvement of visual acuity for patients that can be life-changing even if it doesn't completely restore full visual acuity. But to reach the level to allow for human testing some several steps and improvements can be made. For example, In our test platform, we used an off-the-shelf LED matrix for testing. A custom-designed microLED can produce a signal with better compatibility with optogenetic stimulation. Such a microLED array can be tested to stimulate cell lines treated with in vitro cells first to ensure full system functionality. This test can be followed up with in-vivo tests on rats to ensure that rats can distinguish different patterns from a predefined pool of LED representation. This can also be followed up by custom mazes that require the help of optogenetic stimulation to solve to show benefit. Such tests require a comparison with a blind group and a control group to show improvement and limitations.

Other improvements can be in the visual design of the prototype and the dimensions of the device. The prototype that we have shown earlier is made from off-the-shelf parts, while this is good for prototyping it can be improved upon by using custom-designed circuits, integrated cameras with custom drivers, and custom boards to reduce size.

REFERENCES

[1] D. Pascolini and S. P. Mariotti, "Global estimates of visual impairment: 2010," *British Journal of Ophthalmology*, vol. 96, no. 5, pp. 614–618, 2012.

[2] J. B. Jonas, R. R. Bourne, R. A. White, S. R. Flaxman, J. Keeffe, J. Leasher, K. Naidoo, K. Pesudovs, H. Price, T. Y. Wong *et al.*, "Visual impairment and blindness due to macular diseases globally: a systematic review and meta-analysis," *American journal of ophthalmology*, vol. 158, no. 4, pp. 808–815, 2014.

[3] W. Miquel Perello Nieto, "The human visual system," 2015, [Online; accessed 11-22-2021]. [Online]. Available: https://en.wikipedia.org/wiki/Visual_system#/media/File:Human_visual_pathway.svg

[4] R. H. Masland, "Neuronal diversity in the retina," *Current opinion in neurobiology*, vol. 11, no. 4, pp. 431–436, 2001.

[5] J. R. Sanes and R. H. Masland, "The types of retinal ganglion cells: current status and implications for neuronal classification," *Annual review of neuroscience*, vol. 38, pp. 221–246, 2015.

[6] A. A. Bharath and M. Petrou, *Next generation artificial vision systems: Reverse engineering the human visual system.* Artech House, 2008.

[7] G. S. Brindley and W. Lewin, "The sensations produced by electrical stimulation of the visual cortex," *The Journal of physiology*, vol. 196, no. 2, pp. 479–493, 1968.

[8] P. M. Lewis and J. V. Rosenfeld, "Electrical stimulation of the brain and the development of cortical visual prostheses: an historical perspective," *Brain research*, vol. 1630, pp. 208–224, 2016.

[9] K. Nowik, E. Langwińska-Wośko, P. Skopiński, K. E. Nowik, and J. P. Szaflik, "Bionic eye review–an update," *Journal of Clinical Neuroscience*, 2020.

[10] NASA/ JPL/ University of Arizona, "Face on mars," 2007, [Online; accessed 05-may-2020]. [Online]. Available: https://en.wikipedia.org/wiki/Cydonia_(Mars)#/media/File:Face_on_Mars_with_Inset.jpg

[11] D. J. Felleman and D. E. Van, "Distributed hierarchical processing in the primate cerebral cortex." *Cerebral cortex (New York, NY: 1991)*, vol. 1, no. 1, pp. 1–47, 1991.

[12] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?" *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[13] R. Klein, K. J. Cruickshanks, S. D. Nash, E. M. Krantz, F. J. Nieto, G. H. Huang, J. S. Pankow, and B. E. Klein, "The prevalence of age-related macular degeneration and associated risk factors," *Archives of ophthalmology*, vol. 128, no. 6, pp. 750–758, 2010.

[14] D. S. Friedman, B. J. O'Colmain, B. Munoz, S. C. Tomany, C. McCarty, P. De Jong, B. Nemesure, P. Mitchell, J. Kempen *et al.*, "Prevalence of age-related macular degeneration in the united states," *Arch ophthalmol*, vol. 122, no. 4, pp. 564–572, 2004.

[15] J. Q. Li, T. Welchowski, M. Schmid, M. M. Mauschitz, F. G. Holz, and R. P. Finger, "Prevalence and incidence of age-related macular degeneration in europe: a systematic review and meta-analysis," *British Journal of Ophthalmology*, vol. 104, no. 8, pp. 1077–1084, 2020.

[16] S. D. Schwartz, J.-P. Hubschman, G. Heilwell, V. Franco-Cardenas, C. K. Pan, R. M. Ostrick, E. Mickunas, R. Gay, I. Klimanskaya, and R. Lanza, "Embryonic stem cell trials for macular degeneration: a preliminary report," *The Lancet*, vol. 379, no. 9817, pp. 713–720, 2012.

[17] H. Cook, P. Patel, and A. Tufail, "Age-related macular degeneration: diagnosis and management," *British medical bulletin*, vol. 85, no. 1, pp. 127–149, 2008.

[18] J. D. Steinmetz, R. R. Bourne, P. S. Briant, S. R. Flaxman, H. R. Taylor, J. B. Jonas, A. A. Abdoli, W. A. Abrha, A. Abualhasan, E. G. Abu-Gharbieh *et al.*, "Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study," *The Lancet Global Health*, vol. 9, no. 2, pp. e144–e160, 2021.

[19] Y. Kanagasingam, A. Bhuiyan, M. D. Abramoff, R. T. Smith, L. Goldschmidt, and T. Y. Wong, "Progress on retinal image analysis for age related macular degeneration," *Progress in retinal and eye research*, vol. 38, pp. 20–42, 2014.

[20] C.-L. Lai and S.-W. Chang, "An image processing based visual compensation system for vision defects," in *2009 IEEE 13th International Symposium on Consumer Electronics*. IEEE, 2009, pp. 472–476.

[21] S. L. Hicks, I. Wilson, L. Muhammed, J. Worsfold, S. M. Downes, and C. Kennard, "A depth-based head-mounted visual display to aid navigation in partially sighted individuals," *PloS one*, vol. 8, no. 7, p. e67695, 2013.

[22] A. Belhedi and B. Marcotegui, "Adaptive scene-text binarisation on images captured by smartphones," *IET Image Processing*, vol. 10, no. 7, pp. 515–523, 2016.

[23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[24] C. S. Lin, H.-A. Jan, Y.-L. Lay, C.-C. Huang, and H.-T. Chen, "Evaluating the image quality of closed circuit television magnification systems versus a head-mounted display for people with low vision," *Assistive Technology*, vol. 26, no. 4, pp. 202–208, 2014.

[25] Y. Chang, Y.-G. Lee, and W.-K. Chao, "Head-mounted low vision aid," in *Proceedings of the 5th International Conference on Rehabilitation Engineering & Assistive Technology*, 2011, pp. 1–3.

[26] H. M. Fardoun, L. C. González, and A. S. Mashat, "Rehabilitation low vision algorithm for people with central or multiple losses of vision," in *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*. IEEE, 2013, pp. 339–343.

[27] W. I. Al-Atabany, M. A. Memon, S. M. Downes, and P. A. Degenaar, "Designing and testing scene enhancement algorithms for patients with retina degenerative disorders," *Biomedical engineering online*, vol. 9, no. 1, pp. 1–25, 2010.

[28] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE, 1997, pp. 66–71.

[29] S. Shim, K. Eom, J. Jeong, and S. J. Kim, "Retinal prosthetic approaches to enhance visual perception for blind patients," *Micromachines*, vol. 11, no. 5, p. 535, 2020.

[30] M. S. Humayun, J. D. Weiland, G. Y. Fujii, R. Greenberg, R. Williamson, J. Little, B. Mech, V. Cimmarusti, G. Van Boemel, G. Dagnelie *et al.*, "Visual perception in a blind subject with a chronic microelectronic retinal prosthesis," *Vision research*, vol. 43, no. 24, pp. 2573–2581, 2003.

[31] L. N. Ayton, P. J. Blamey, R. H. Guymer, C. D. Luu, D. A. Nayagam, N. C. Sinclair, M. N. Shivdasani, J. Yeoh, M. F. McCombe, R. J. Briggs *et al.*, "First-in-human trial of a novel suprachoroidal retinal prosthesis," *PloS one*, vol. 9, no. 12, p. e115239, 2014.

[32] M. S. Humayun, J. D. Dorn, L. Da Cruz, G. Dagnelie, J.-A. Sahel, P. E. Stanga, A. V. Cideciyan, J. L. Duncan, D. Eliott, E. Filley *et al.*, “Interim results from the international trial of second sight’s visual prosthesis,” *Ophthalmology*, vol. 119, no. 4, pp. 779–788, 2012.

[33] E. Zrenner, K. U. Bartz-Schmidt, H. Benav, D. Besch, A. Bruckmann, V.-P. Gabel, F. Gekeler, U. Greppmaier, A. Harscher, S. Kibbel *et al.*, “Subretinal electronic chips allow blind patients to read letters and combine them to words,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 278, no. 1711, pp. 1489–1497, 2011.

[34] M. Spering and M. Carrasco, “Acting without seeing: eye movements reveal visual processing without awareness,” *Trends in neurosciences*, vol. 38, no. 4, pp. 247–258, 2015.

[35] H. M. Mohammadi, E. Ghafar-Zadeh, and M. Sawan, “An image processing approach for blind mobility facilitated through visual intracortical stimulation,” *Artificial organs*, vol. 36, no. 7, pp. 616–628, 2012.

[36] K. Nikolic, N. Grossman, H. Yan, E. Drakakis, C. Toumazou, and P. Degenaar, “A non-invasive retinal prosthesis-testing the concept,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* IEEE, 2007, pp. 6364–6367.

[37] S. Nirenberg and C. Pandarinath, “Retinal prosthetic strategy with the capacity to restore normal vision,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 37, pp. 15 012–15 017, 2012.

[38] W. Al-Atabany and P. Degenaar, “A coding scheme for optoelectronic/optogenetic retinal prosthesis.”

[39] W. Al-Atabany, B. McGovern, K. Mehran, R. Berlinguer-Palmini, and P. Degenaar, “A processing platform for optoelectronic/optogenetic retinal prosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 3, pp. 781–791, 2011.

[40] J.-A. Sahel, E. Boulanger-Scemama, C. Pagot, A. Arleo, F. Galluppi, J. N. Martel, S. Degli Esposti, A. Delaux, J.-B. de Saint Aubert, C. de Montleau *et al.*, “Partial recovery of visual function in a blind patient after optogenetic therapy,” *Nature Medicine*, pp. 1–7, 2021.

[41] S. A. Panëels, A. Olmos, J. R. Blum, and J. R. Cooperstock, "Listen to it yourself! evaluating usability of what's around me? for the blind," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2107–2116.

[42] J. R. Blum, M. Bouchard, and J. R. Cooperstock, "Spatialized audio environmental awareness for blind users with a smartphone," *Mobile Networks and Applications*, vol. 18, no. 3, pp. 295–309, 2013.

[43] L. Findlater, L. Stearns, R. Du, U. Oh, D. Ross, R. Chellappa, and J. Froehlich, "Supporting everyday activities for persons with visual impairments through computer vision-augmented touch," in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, 2015, pp. 383–384.

[44] S. Mascetti, D. Ahmetovic, A. Gerino, C. Bernareggi, M. Busso, and A. Rizzi, "Robust traffic lights detection on mobile devices for pedestrians with visual impairment," *Computer Vision and Image Understanding*, vol. 148, pp. 123–135, 2016.

[45] J. D. Weiland, N. Parikh, V. Pradeep, and G. Medioni, "Smart image processing system for retinal prosthesis," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* IEEE, 2012, pp. 300–303.

[46] N. Banovic, R. L. Franz, K. N. Truong, J. Mankoff, and A. K. Dey, "Uncovering information needs for independent spatial learning for users who are visually impaired," in *Proceedings of the 15th international ACM SIGACCESS conference on computers and accessibility*, 2013, pp. 1–8.

[47] L. Evett, S. Battersby, A. Ridley, and D. Brown, "An interface to virtual environments for people who are blind using wii technology-mental models and navigation," *Journal of Assistive Technologies*, 2009.

[48] M. C. Ghilardi, R. C. Macedo, and I. H. Manssour, "A new approach for automatic detection of tactile paving surfaces in sidewalks," *Procedia computer science*, vol. 80, pp. 662–672, 2016.

[49] S. Deb, S. T. Reddy, U. Baidya, A. K. Sarkar, and P. Renu, "A novel approach of assisting the visually impaired to navigate path and avoiding obstacle-collisions," in *2013 3rd IEEE International Advance Computing Conference (IACC).* IEEE, 2013, pp. 1127–1130.

[50] M. Vlaminck, L. H. Quang, H. Van Nam, H. Vu, P. Veelaert, and W. Philips, "Indoor assistance for visually impaired people using a rgb-d camera," in *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI).* IEEE, 2016, pp. 161–164.

[51] I. Mittal, A. Mittal, and S. Indu, "A novel approach to a camera based assisstive aid for visually impaired," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT).* IEEE, 2016, pp. 448–452.

[52] R. Velázquez, J. Varona, and P. Rodrigo, "Computer-based system for simulating visual impairments," *IETE Journal of Research*, vol. 62, no. 6, pp. 833–841, 2016.

[53] V. C. Sekhar, S. Bora, M. Das, P. K. Manchi, S. Josephine, and R. Paily, "Design and implementation of blind assistance system using real time stereo vision algorithms," in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID).* IEEE, 2016, pp. 421–426.

[54] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Consumer depth cameras for computer vision.* Springer, 2013, pp. 119–137.

[55] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011.* Ieee, 2011, pp. 1297–1304.

[56] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," in *2014 IEEE international conference on robotics and automation (ICRA).* IEEE, 2014, pp. 5636–5643.

[57] A. J. Kolarik, S. Cirstea, S. Pardhan, and B. C. Moore, "A summary of research investigating echolocation abilities of blind and sighted humans," *Hearing research*, vol. 310, pp. 60–68, 2014.

[58] L. Picinali, A. Afonso, M. Denis, and B. F. Katz, "Exploration of architectural spaces by blind people using auditory virtual reality for the construction of spatial knowledge," *International Journal of Human-Computer Studies*, vol. 72, no. 4, pp. 393–407, 2014.

[59] M. Suk and B. Prabhakaran, "Real-time mobile facial expression recognition system-a case study," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 132–137.

[60] X. Liu, X. Cheng, and K. Lee, "Ga-svm based facial emotion recognition using facial geometric features," *IEEE Sensors Journal*, 2020.

[61] S. Agarwal, B. Santra, and D. P. Mukherjee, "Anubhav: recognizing emotions through facial expression," *The Visual Computer*, vol. 34, no. 2, pp. 177–191, 2018.

[62] B. Fasel, "Head-pose invariant facial expression recognition using convolutional neural networks," in *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces.* IEEE, 2002, pp. 529–534.

[63] X. Zhu, S. Ye, L. Zhao, and Z. Dai, "Hybrid attention cascade network for facial expression recognition," *Sensors*, vol. 21, no. 6, p. 2003, 2021.

[64] L. Montazeri, N. El Zarif, S. Trenholm, and M. Sawan, "Optogenetic stimulation for restoring vision to patients suffering from retinal degenerative diseases: current strategies and future directions," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 6, pp. 1792–1807, 2019.

[65] L. Montazeri, N. El Zarif, T. Tokuda, J. Ohta, and M. Sawan, "Active control of μLED arrays for optogenetic stimulation," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS).* IEEE, 2018, pp. 1–5.

[66] L. Montazeri, N. El Zarif, and M. Sawan, "Optical control of neural dynamics using LED array," in *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS).* IEEE, 2018, pp. 96–99.

[67] N. El Zarif, L. Montazeri, and M. Sawan, "Real-time retinal processing for high-resolution optogenetic stimulation device," in *Engineering in Medicine and Biology Society (EMBC), 2018 40th Annual International Conference of the IEEE.* IEEE, 2018.

[68] S. Khandhadia, J. Cherry, and A. J. Lotery, "Age-related macular degeneration," in *Neurodegenerative Diseases.* Springer, 2012, pp. 15–36.

[69] D. A. Baylor, T. Lamb, and K.-W. Yau, "Responses of retinal rods to single photons." *The Journal of physiology*, vol. 288, no. 1, pp. 613–634, 1979.

[70] R. E. Marc, B. W. Jones, C. B. Watt, and E. Strettoi, "Neural remodeling in retinal degeneration," *Progress in retinal and eye research*, vol. 22, no. 5, pp. 607–655, 2003.

[71] C. D. Eiber, N. H. Lovell, and G. J. Suaning, “Attaining higher resolution visual prosthetics: a review of the factors and limitations,” *Journal of neural engineering*, vol. 10, no. 1, p. 011002, 2013.

[72] R. A. da Silveira and B. Roska, “Cell types, circuits, computation,” *Current opinion in neurobiology*, vol. 21, no. 5, pp. 664–671, 2011.

[73] R. K. Shepherd, M. N. Shivdasani, D. A. Nayagam, C. E. Williams, and P. J. Blamey, “Visual prostheses for the blind,” *Trends in biotechnology*, vol. 31, no. 10, pp. 562–571, 2013.

[74] H. Kolb, E. Fernandez, and R. Nelson, “Facts and figures concerning the human retina–webvision: The organization of the retina and visual system,” 1995.

[75] E. Özmert and U. Arslan, “Retinal prostheses and artificial vision,” *Turkish journal of ophthalmology*, vol. 49, no. 4, p. 213, 2019.

[76] A. Sengupta, A. Chaffiol, E. Macé, R. Caplette, M. Desrosiers, M. Lampič, V. Forster, O. Marre, J. Y. Lin, J.-A. Sahel *et al.*, “Red-shifted channelrhodopsin stimulation restores light responses in blind mice, macaque retina, and human retina,” *EMBO molecular medicine*, vol. 8, no. 11, pp. 1248–1264, 2016.

[77] B. S. Henriksen, R. E. Marc, and P. S. Bernstein, “Optogenetics for retinal disorders,” *Journal of ophthalmic & vision research*, vol. 9, no. 3, p. 374, 2014.

[78] C.-J. Simon, J.-A. Sahel, J. Duebel, S. Herlitze, and D. Dalkara, “Opsins for vision restoration,” *Biochemical and Biophysical Research Communications*, 2020.

[79] Q. Lu, T. H. Ganjawala, A. Krstevski, G. W. Abrams, and Z.-H. Pan, “Comparison of aav-mediated optogenetic vision restoration between retinal ganglion cell expression and on bipolar cell targeting,” *Molecular therapy. Methods & clinical development*, vol. 18, p. 15, 2020.

[80] F. Naarendorp, T. M. Esdaille, S. M. Banden, J. Andrews-Labenski, O. P. Gross, and E. N. Pugh, “Dark light, rod saturation, and the absolute and incremental sensitivity of mouse cone vision,” *Journal of Neuroscience*, vol. 30, no. 37, pp. 12 495–12 507, 2010.

[81] N. El Zarif, L. Montazeri, F. Leduc-Primeau, and M. Sawan, “Mobile-optimized facial expression recognition techniques,” *IEEE Access*, 2021.

[82] W. Al-Atabany, B. McGovern, K. Mehran, R. Berlinguer-Palmini, and P. Degenaar, “A processing platform for optoelectronic/optogenetic retinal prosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 3, pp. 781–791, 2013.

[83] A. Soltan, P. Maaskant, N. Armstrong, W. Al-Atabany, L. Chaudet, M. Neil, and P. Degenaar, “Wearable glasses for retinal pigmentiosa based on optogenetics,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[84] A. Asher, W. A. Segal, S. A. Baccus, L. P. Yaroslavsky, and D. V. Palanker, “Image processing for a high-resolution optoelectronic retinal prosthesis,” *IEEE transactions on Biomedical Engineering*, vol. 54, no. 6, pp. 993–1004, 2007.

[85] O. S. Dhande and A. D. Huberman, “Retinal ganglion cell maps in the brain: implications for visual processing,” *Current opinion in neurobiology*, vol. 24, pp. 133–142, 2014.

[86] T. Baden, P. Berens, K. Franke, M. R. Rosón, M. Bethge, and T. Euler, “The functional diversity of retinal ganglion cells in the mouse,” *Nature*, vol. 529, no. 7586, p. 345, 2016.

[87] H. Barlow, R. Hill, and W. Levick, “Retinal ganglion cells responding selectively to direction and speed of image motion in the rabbit,” *The Journal of Physiology*, vol. 173, no. 3, pp. 377–407, 1964.

[88] S. Wienbar and G. Schwartz, “The dynamic receptive fields of retinal ganglion cells,” *Progress in retinal and eye research*, 2018.

[89] E. Famiglietti and H. Kolb, “Structural basis for on-and off-center responses in retinal ganglion cells,” *Science*, vol. 194, no. 4261, pp. 193–195, 1976.

[90] R. W. Rodieck, “Quantitative analysis of cat retinal ganglion cell response to visual stimuli,” *Vision research*, vol. 5, no. 12, pp. 583–601, 1965.

[91] M. J. McMahon, O. S. Packer, and D. M. Dacey, “The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-gabaergic pathway,” *Journal of Neuroscience*, vol. 24, no. 15, pp. 3736–3745, 2004.

[92] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[93] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision.* Springer, 2006, pp. 404–417.

[94] K. Reinhard, A. Tikidji-Hamburyan, H. Seitter, S. Idrees, M. Mutter, B. Benkner, and T. A. Münch, "Step-by-step instructions for retina recordings with perforated multi electrode arrays," *PloS one*, vol. 9, no. 8, p. e106148, 2014.

[95] B. B. Lee, B. Cooper, and D. Cao, "The spatial structure of cone-opponent receptive fields in macaque retina," *Vision research*, 2017.

[96] J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. Chichilnisky, and E. P. Simoncelli, "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, no. 7207, p. 995, 2008.

[97] J. Kleinschmidt and J. E. Dowling, "Intracellular recordings from gecko photoreceptors during light and dark adaptation." *The Journal of general physiology*, vol. 66, no. 5, pp. 617–648, 1975.

[98] N. Brackbill, C. Rhoades, A. Kling, N. P. Shah, A. Sher, A. M. Litke, and E. Chichilnisky, "Reconstruction of natural images from responses of primate retinal ganglion cells," *bioRxiv*, 2020.

[99] A. Benoit, A. Caplier, B. Durette, and J. Hérault, "Using human visual system modeling for bio-inspired low level image processing," *Computer vision and Image understanding*, vol. 114, no. 7, pp. 758–773, 2010.

[100] P. Chundi, M. Subramaniam, A. Muthuraj, and E. Margalit, "Estimating distortion parameters in simulated prosthetic vision," in *Healthcare Informatics (ICHI), 2013 IEEE International Conference on.* IEEE, 2013, pp. 421–430.

[101] H. Moshtael, T. Aslam, I. Underwood, and B. Dhillon, "High tech aids low vision: a review of image processing for the visually impaired," *Translational vision science & technology*, vol. 4, no. 4, pp. 6–6, 2015.

[102] M. L. Katz, C. Lutterbeck, and K. Nikolic, "An implementation of magnocellular pathways in event-based retinomorphic systems," in *2012 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Nov 2012, pp. 17–20.

[103] J. Steffen, G. Hille, and K. Tönnies, "Automatic perception enhancement for simulated retinal implants," 2019.

[104] C. A. Curcio, K. R. Sloan, O. Packer, A. E. Hendrickson, and R. E. Kalina, "Distribution of cones in human and monkey retina: individual variability and radial asymmetry," *Science*, vol. 236, no. 4801, pp. 579–582, 1987.

[105] A. Ahdi, H. Rabbani, and A. Vard, "A hybrid method for 3d mosaicing of oct images of macula and optic nerve head," *Computers in biology and medicine*, vol. 91, pp. 277–290, 2017.

[106] B. Roska and F. Werblin, "Vertical interactions across ten parallel, stacked representations in the mammalian retina," *Nature*, vol. 410, no. 6828, pp. 583–587, 2001.

[107] S. Exner, "Experimentelle untersuchung der einfachsten psychischen processe," *Archiv für die gesamte Physiologie des Menschen und der Tiere*, vol. 11, no. 1, pp. 403–432, 1875.

[108] R. Efron, "Conservation of temporal information by perceptual systems," *Perception & Psychophysics*, vol. 14, no. 3, pp. 518–530, 1973.

[109] P. A. Kolers and M. von Grünau, "Shape and color in apparent motion," *Vision research*, vol. 16, no. 4, pp. 329–335, 1976.

[110] M. H. Herzog, T. Kammer, and F. Scharnowski, "Time slices: what is the duration of a percept?" *PLoS biology*, vol. 14, no. 4, p. e1002433, 2016.

[111] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

[112] K. Amano, N. Goda, S. Nishida, Y. Ejima, T. Takeda, and Y. Ohtani, "Estimation of the timing of human visual perception from magnetoencephalography," *Journal of Neuroscience*, vol. 26, no. 15, pp. 3981–3991, 2006.

[113] V. Kornijcuk, H. Lim, I. Kim, J.-K. Park, W.-S. Lee, J.-H. Choi, B. J. Choi, and D. S. Jeong, "Scalable excitatory synaptic circuit design using floating gate based leaky integrators," *Scientific reports*, vol. 7, no. 1, pp. 1–13, 2017.

[114] A. Erofeev, E. Gerasimov, A. Lavrova, A. Bolshakova, E. Postnikov, I. Bezprozvanny, and O. L. Vlasova, "Light stimulation parameters determine neuron dynamic characteristics," *Applied Sciences*, vol. 9, no. 18, p. 3673, 2019.

[115] M. Beyeler, G. M. Boynton, I. Fine, and A. Rokem, "pulse2percept: A python-based simulation framework for bionic vision," *BioRxiv*, p. 148015, 2017.

[116] D. Matsumoto and B. Willingham, "Spontaneous facial expressions of emotion of congenitally and noncongenitally blind individuals." *Journal of personality and social psychology*, vol. 96, no. 1, p. 1, 2009.

[117] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image and vision computing*, vol. 27, no. 12, pp. 1743–1759, 2009.

[118] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. M. Prkachin, "Automatically detecting pain using facial actions," in *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd Int. Conf. on.* IEEE, pp. 1–8.

[119] N. E. Zarif, L. Montazeri, and M. Sawan, "Real-time retinal processing for high-resolution optogenetic stimulation device," in *2018 40th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5946–5949.

[120] E. Vural, M. Çetin, A. Erçil, G. Littlewort, M. Bartlett, and J. Movellan, "Automated drowsiness detection for improved driving safety," 2008.

[121] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[122] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image and Vision Computing*, vol. 47, pp. 3–18, 2016.

[123] W. T. Meshach, S. Hemajothi, and E. M. Anita, "Real-time facial expression recognition for affect identification using multi-dimensional svm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2020.

[124] W.-S. Chu, F. De la Torre, and J. F. Cohn, "Selective transfer machine for personalized facial expression analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 529–545, 2017.

[125] M. F. Valstar, T. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn, "FERA 2015 - second facial expression recognition and analysis challenge," in *2015 11th IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 06, pp. 1–8.

[126] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[127] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[128] Tai Sing Lee, “Image representation using 2D gabor wavelets,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[129] S. Agrawal and P. Khatri, “Facial expression detection techniques: based on viola and jones algorithm and principal component analysis,” in *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth Int. Conf. on.* IEEE, pp. 108–112.

[130] M. H. Siddiqi, R. Ali, A. M. Khan, E. S. Kim, G. J. Kim, and S. Lee, “Facial expression recognition using active contour-based face detection, facial movement-based feature extraction, and non-linear feature selection,” *Multimedia Systems*, vol. 21, no. 6, pp. 541–555, 2015.

[131] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conf. on*, pp. 94–101.

[132] M. F. Valstar, E. Sánchez-Lozano, J. F. Cohn, L. A. Jeni, J. M. Girard, Z. Zhang, L. Yin, and M. Pantic, “FERA 2017 - addressing head pose in the third facial expression recognition and analysis challenge,” in *2017 12th IEEE Int. Conf. on Automatic Face Gesture Recognition (FG 2017)*, pp. 839–847.

[133] Z. Sun, Z.-P. Hu, M. Wang, and S.-H. Zhao, “Discriminative feature learning-based pixel difference representation for facial expression recognition,” *IET Computer Vision*, vol. 11, no. 8, pp. 675–682, 2017.

[134] M. S. Zia, M. Hussain, and M. A. Jaffar, “A novel spontaneous facial expression recognition using dynamically weighted majority voting based ensemble classifier,” *Multimedia Tools and Applications*, pp. 1–31, 2018.

[135] M. Sultan Zia and M. Arfan Jaffar, “Facial expressions recognition using an ensemble of feature sets based on key-point descriptors,” *The Imaging Science Journal*, vol. 63, no. 3, pp. 160–167, 2015.

[136] S. L. Happy and A. Routray, “Automatic facial expression recognition using features of salient facial patches,” *IEEE Trans. on Affective Computing*, vol. 6, no. 1, pp. 1–12, 2015.

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

[138] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” in *2016 IEEE Winter Conf. on applications of computer vision (WACV)*, pp. 1–10.

[139] A. Majumder, L. Behera, and V. K. Subramanian, “Automatic facial expression recognition system using deep network-based data fusion,” *IEEE Trans. on Cybernetics*, vol. 48, no. 1, pp. 103–114, 2018.

[140] X. Chen, X. Yang, M. Wang, and J. Zou, “Convolution neural network for automatic facial expression recognition,” in *Applied System Innovation (ICASI), 2017 Int. Conf. on*. IEEE, 2017, pp. 814–817.

[141] D. Sen, S. Datta, and R. Balasubramanian, “Facial emotion classification using concatenated geometric and textural features,” *Multimedia Tools and Applications*, pp. 1–37, 2018.

[142] Z. Yu and C. Zhang, “Image based static facial expression recognition with multiple deep network learning,” in *Proceedings of the 2015 ACM on Int. Conf. on Multimodal Interaction*, pp. 435–442.

[143] K. Zhang, Y. Huang, Y. Du, and L. Wang, “Facial expression recognition based on deep evolutional spatial-temporal networks,” *IEEE Trans. on Image Processing*, vol. 26, no. 9, pp. 4193–4203, 2017.

[144] D. H. Kim, W. Baddar, J. Jang, and Y. M. Ro, “Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition,” *IEEE Trans. on Affective Computing*, 2017.

[145] T. Zhang, W. Zheng, Z. Cui, Y. Zong, J. Yan, and K. Yan, “A deep neural network-driven feature learning method for multi-view facial expression recognition,” *IEEE Trans. on Multimedia*, vol. 18, no. 12, pp. 2528–2536, 2016.

[146] V. Mayya, R. M. Pai, and M. M. Pai, “Automatic facial expression recognition using DCNN,” *Procedia Computer Science*, vol. 93, pp. 453–461, 2016.

[147] P. Dhankhar, “ResNet-50 and VGG-16 for recognizing facial emotions,” *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 13, no. 4, pp. 126–130, 2019.

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

[149] P. D. M. Fernandez, F. A. G. Peña, T. I. Ren, and A. Cunha, "FERAtt: Facial expression recognition with attention net," *arXiv preprint arXiv:1902.03284*, 2019.

[150] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Identity-free facial expression recognition using conditional generative adversarial network," *arXiv preprint arXiv:1903.08051*, 2019.

[151] F. Zhang, T. Zhang, Q. Mao, and C. Xu, "Geometry guided pose-invariant facial expression recognition," *IEEE Trans. on Image Processing*, vol. 29, pp. 4445–4460, 2020.

[152] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region attention networks for pose and occlusion robust facial expression recognition," *IEEE Trans. on Image Processing*, vol. 29, pp. 4057–4069, 2020.

[153] B. Yang, J. Cao, R. Ni, and Y. Zhang, "Facial expression recognition using weighted mixture deep neural network based on double-channel facial images," *IEEE Access*, vol. 6, pp. 4630–4640, 2017.

[154] M. V. Zavarez, R. F. Berriel, and T. Oliveira-Santos, "Cross-database facial expression recognition based on fine-tuned deep convolutional network," in *2017 30th SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, pp. 405–412.

[155] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Trans. on Affective Computing*, 2020.

[156] D. A. Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, "Enhancing CNN with preprocessing stage in automatic emotion recognition," *Procedia computer science*, vol. 116, pp. 523–529, 2017.

[157] S. Umer, R. K. Rout, C. Pero, and M. Nappi, "Facial expression recognition with trade-offs between data augmentation and deep learning features," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15, 2021.

[158] S. M. González-Lozoya, J. de la Calleja, L. Pellegrin, H. J. Escalante, M. A. Medina, and A. Benitez-Ruiz, "Recognition of facial expressions based on CNN features," *Multimedia Tools and Applications*, pp. 1–21, 2020.

[159] Z. Yu, G. Liu, Q. Liu, and J. Deng, "Spatio-temporal convolutional features with nested LSTM for facial expression recognition," *Neurocomputing*, vol. 317, pp. 50–57, 2018.

[160] K. Sikka, G. Sharma, and M. Bartlett, "Lomo: Latent ordinal model for facial analysis in videos," in *Proceedings of the IEEE Conf. on computer vision and pattern recognition*, 2016, pp. 5580–5589.

[161] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "From facial expression recognition to interpersonal relation prediction," *International Journal of Computer Vision*, vol. 126, no. 5, pp. 550–569, 2018.

[162] H. I. Dino and M. B. Abdulrazzaq, "Facial expression classification based on svm, knn and mlp classifiers," in *2019 Int. Conf. on Advanced Science and Engineering (ICOASE)*. IEEE, 2019, pp. 70–75.

[163] A. Fathallah, L. Abdi, and A. Douik, "Facial expression recognition via deep learning," in *2017 IEEE/ACS 14th Int. Conf. on Computer Systems and Applications (AICCSA)*. IEEE, 2017, pp. 745–750.

[164] M. Verma, S. K. Vipparthi, and G. Singh, "Hinet: Hybrid inherited feature learning network for facial expression recognition," *(IEEE) Letters of the Computer Society*, vol. 2, no. 4, pp. 36–39, 2019.

[165] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *Image analysis for multimedia interactive services (WIAMIS), 2010 11th Int. workshop on*. IEEE, pp. 1–4.

[166] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero, "Survey on RGB, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1548–1568, 2016.

[167] N. E. Zarif, "Angled posed facial expression dataset," 2020. [Online]. Available: http://dx.doi.org/10.21227/awgd-3t83

[168] E. Goeleven, R. De Raedt, L. Leyman, and B. Verschuere, "The karolinska directed emotional faces: a validation study," *Cognition and emotion*, vol. 22, no. 6, pp. 1094–1118, 2008.

[169] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[170] E. Friesen and P. Ekman, “Facial action coding system: a technique for the measurement of facial movement,” *Palo Alto*, 1978.

[171] S. Han, Z. Meng, A.-S. Khan, and Y. Tong, “Incremental boosting convolutional neural network for facial action unit recognition,” in *Advances in neural information processing systems*, 2016, pp. 109–117.

[172] D. Kim, “Facial expression recognition using asm-based post-processing technique,” *Pattern Recognition and Image Analysis*, vol. 26, no. 3, pp. 576–581, 2016.

[173] O. Rudovic, M. Pantic, and I. Patras, “Coupled gaussian processes for pose-invariant facial expression recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1357–1369, 2013.

[174] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” *ACM Trans. on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[175] D. E. King, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

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