**T.C.**
**DOKUZ EYLÜL UNIVERSITY**
**ENGINEERING FACULTY**
**ELECTRICAL & ELECTRONICS ENGINEERING**
**DEPARTMENT**

# Methodology

## Research Project Report

*by*

**Rabia DOĞAN**

*Advisor*

**Ph.D. Özgür TAMER**

January,2020
IZMIR

# ABSTRACT

The project is that Vestel Elektronik A.Ş. Interactive Personalized Smart Mirror project carried out and supported by TEYDEB 3170688 code. It is aimed to give the Smart Mirror controllable features with gestures. In this way, the product will be able to appeal to the upper customer segments and increase its market share. The main goal here is to enable users to use most of the smart mirror applications with hand movements, to eliminate the need to touch the mirror, thus to eliminate the contamination in the touched parts of the mirrors and therefore to possible dissatisfaction with the product and to ensure that the product can be used even in cases where the users cannot touch the mirror with their hands for various reasons, such as the bathroom. Although camera-based systems are generally used for gesture recognition, it will not be welcomed by many users to have a camera in a personal use area such as the bathroom. Therefore, passive infrared sensor arrays will be preferred for gesture recognition in our project. They can be preferred in application areas where cameras are relatively weak because they work with the infrared radiation emitted by living creatures. We aim to detect hand gesture and then the movement of the hand gesture. In this way, gesture distinction can be made with a simpler electronic design with less processing load and lower cost than a standard camera.
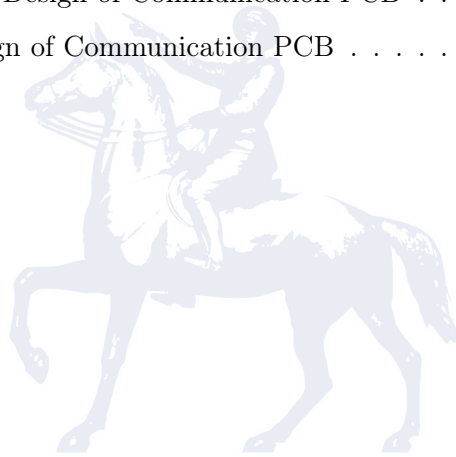
# ÖZET

Proje Vestel Elektronik A.Ş.taarafından TEYDEB 3170688 kodu ile yürütülen ve desteklenen Etkileşimli Kişiselleştirilmiş Akıllı Ayna projesine jest ile kontrol edilebilme özellikleri kazandırılması hedeflenmektedir. Bu sayede ürün üst müşteri segmentlerine de hitap edebilecek, pazar payını artırabilecektir.Burada temel hedef kullanıcıların akıllı ayna uygulamalarının birçoğunu el hareketleri ile kullanabilmesini sağlayarak, aynaya dokunma ihtiyacını ortadan kaldırmak böylece aynaların dokunulan bölgelerindeki kirlenmeyi ve dolayısıyla da ürün ile ilgili olası memnuniyetsizliği ortadan kaldırmak ve banyo gibi kullanıcıların çeşitli nedenler ile elleri ile aynaya dokunamayacakları durumlarda dahi ürünün kullanılabilmesini sağlamaktır.Jest tanıma için genellikle kamera temelli sistemler kullanılmakla beraber, banyo gibi kişisel kullanıma dönük bir alanda kamera bulunması birçok kullanıcı tarafından hoş karşılanmayacaktır.Bu nedenle projemizde jest tanımlama amacıyla pasif kızılötesi sensör dizileri tercih edilecektir.Bu tip sensörler oldukça düşük çözünürlüklerde görev yapmasına karşın, nesne ya da canlıların yaydığı kızılötesi ışıma ile çalıştıkları için kameraların görece zayıf kaldığı uygulama alanlarında tercih edilebilmektedirler.Pasif kızılötesi sensörler ile öncelikle kullanıcının jestini ve sonrasında da jestin hareketini algılamayı amaçlamaktayız.Bu sayede standart bir kameraya göre daha az işlem yükü ve daha düşük maliyet ve daha basit bir elektronik tasarım ile jest ayrımı yapılabilecektir.

İZMİR-1982

# Contents

# List of Figures

# List of Tables

# 1. Introduction

The general purpose of the project is to add a gesture detection system to the Smart Mirror at the cheapest cost.

In the project solution, the HTPA32x32d Thermopile Array sensor has been chosen to minimize the cost. It is a system that measures infrared radiation, developed with the thermocouple method dating back to 150 years. "A thermocouple consists of two different materials which are connected at one end, while the other two ends are attached to a voltage meter" [2]. If an absorber is attached to the junction and exposed to the effect of IR radiation from an object, the thermocouple junction heats up due to the incoming radiation while the absorber collects incoming heat [2]. The thermocouple material in turn converts the temperature difference voltage indicated by the voltmeter. Hence, the voltmeter reading is a direct measure for object temperature. This method is simple, does not require any mechanics and can detect static signals accurately [2]. HTPA32x32d sensor 32x32 pixel, operates between -10 and 70 degrees, provides I2C communication, has an internal EEPROM and provides an 8-bit data set. EEPROM data contains calibration data for each pixel of the sensor.

Before Smart Mirror integration, it was decided to use an external microcontroller and it was designated as Raspberry Pi. The sensor PCB to be developed is designed to allow the most efficient sensor to work. We can provide our sensor, which does not need an external communication circuit, to communicate directly with Raspberry Pi.

For the development of the software part, The sensor data is to read using one of the python libraries, the I2C python-periphery library [9]. In the acquired image, all objects in the background and the human body emit infrared radiation. For this reason, the sensor will detect these objects as well. This problem can be solved with the thresholding methods in the literature. The Simple Thresholding method gives the same threshold value for each pixel. Pixel values smaller than the threshold value are changed to 0 [9]. Adaptive threshold, It is the method by which the threshold value is calculated for smaller regions and therefore there will be different threshold values for different regions [9]. The most suitable method for the project is Otsu's method.

Also, it is the type of Adaptive threshold. The reason we chose this method is that we want the environmental conditions to change and assign a different threshold value each time [3].

After isolating the image from the background, The gestures are divided into two according to the state of the image. These are static and dynamic gestures. There are several methods to make these two distinctions. With the Barycenter method, it is possible to adopt a similar situation of planetary movements for gestures [16]. Another method and preferred method is MHI (memory History Image) [14].
8 Static and dynamic gestures are defined in the methodology chapter. Using MHI method, defined gestures can be distinguished.

The next step, classification methods should be performed to match the acquired image with the previously specified commands. Some methods are as follows; Hybrid Method (the combination of Haar-Like and HOG features ) is a feature extraction method. Using this method after, multi-class SVM(Support Vector Machine ) is used for final hand gesture recognition [5]. HOG(Histogram of Oriented Gradient) features is one of the feature extraction methods. After applying this method, hand recognition is to find the matched image in the database. Using nearest-neighbour algorithms, this step is carried out. [6].k-NN(k-nearest neighbour); It is one of the most popular machine learning algorithms because it is resistant to old, simple and noisy training data. However, it also has a disadvantage. Overlap, it is necessary to get up-to-date information required for big data that we store in the long term. There are database sets. Then, by using the fall detection algorithm, the object in front is distinguished with the other objects. Finally, k-NN classification is made using the data in the database [7]. We decided to use that CNN. The reason is that the CNN transaction is very fast. Realization at this speed will minimize my propagation delay time. some paper are available about using CNN classification [15] [11] [16]. More detailed information about CNN will be included in the methodology section.

All that remains is the communication between Raspberry Pi and the Smart Mirror. Due to the fact that it was a team dealing with the project before, the gesture detection system was worked on on the smart mirror side, but a problem occurred due to the sensor. Therefore, we do not have an extra workload on the smart mirror side, after the OTG connection settings between Raspberry Pi and the Smart mirror, our system is expected to work properly.

# 2. Methodology

In this section, the methods to be applied in the project are included.
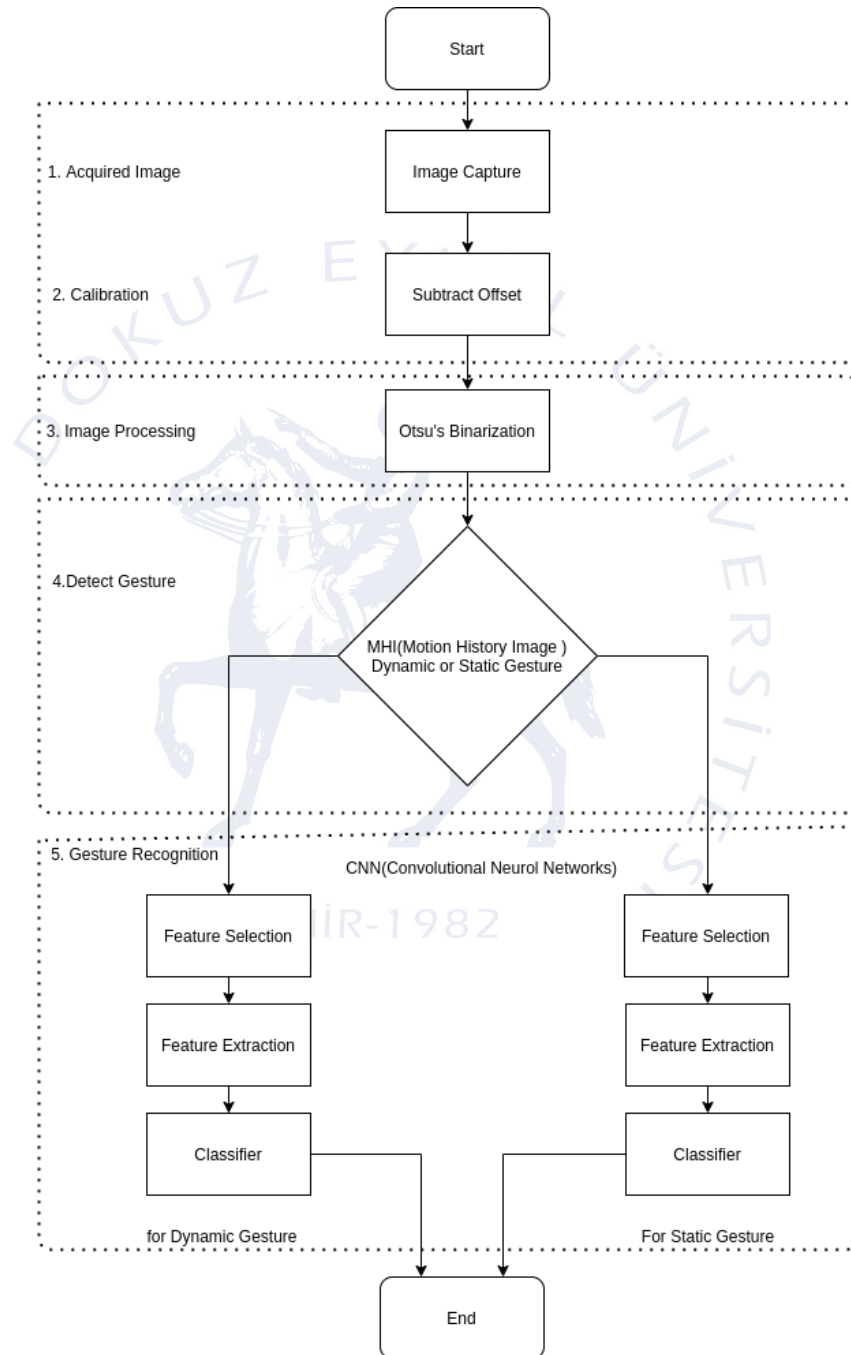


Figure 2.1: Software flowchart

## 2.1 Capture Thermopiles Image

The chip uses 7-bit I2C address 0x1A for configuration and sensor data and 7-bit I2C address 0x50 to access internal EEPROM. The address byte is followed by a W / R bit and an 8-bit command.

To read data from the chip first the address and read command must be sent. After the last ACK, a new start-bit (repeated start) and the address with a set read-flag initiates the read sequence. There can be bytes read as many as required. The last byte must be denoted by a not-acknowledge.



Figure 2.2: Read command

The sensor is divided into two parts (Top and Bottom Half), which are also divided into 4 blocks. The reading order is shown below for different blocks. When a conversion is initiated, the X Block of the upper and lower half are measured simultaneously. Each block consists of 128 Pixels sampled entirely in parallel. The reading order in the lower half is mirrored compared to the upper half so the center lines are always read last.

Table 2.1: Read Data 1 Command (Top Half of Array)

| Addr/CMD | 0x1A (7 Bit!) / 0x0A | | | | | | |
|---|---|---|---|---|---|---|---|
| Read Data | 7 | 6 | 5 | 4 | 3 | 2 | 1 \| 0 |
| 1. Byte / 2. Byte | PTAT 1 MSB / LSB or Vdd 1 MSB / LSB | | | | | | |
| 3. Byte / 4. Byte | Pixel (0+BLOCK*128) MSB / LSB | | | | | | |
| 5. Byte / 6. Byte | Pixel (1+BLOCK*128) MSB / LSB | | | | | | |
| ... | | | | | | | |
| 257. Byte / 258. Byte | Pixel (127+BLOCK*128) MSB / LSB | | | | | | |

Table 2.2: Read Data 2 Command (Bottom Half of Array)

| Addr/CMD | 0x1A (7 Bit!) / 0x0B | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Read Data | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1. Byte / 2. Byte | PTAT 2 MSB / LSB or Vdd 2 MSB / LSB | | | | | | | |
| 3. Byte / 4. Byte | Pixel (992-BLOCK*128) MSB / LSB | | | | | | | |
| 5. Byte / 6. Byte | Pixel (993-BLOCK*128) MSB / LSB | | | | | | | |
| ... | | | | | | | | |
| 65. Byte / 66. Byte | Pixel (1023-BLOCK*128) MSB / LSB | | | | | | | |
| 65. Byte / 66. Byte | Pixel (1023-BLOCK*128) MSB / LSB | | | | | | | |
| 67. Byte / 68. Byte | Pixel (960-BLOCK*128) MSB / LSB | | | | | | | |
| 69. Byte / 70. Byte | Pixel (961-BLOCK*128) MSB / LSB | | | | | | | |
| ... | | | | | | | | |
| 129. Byte / 130. Byte | Pixel (991-BLOCK*128) MSB / LSB | | | | | | | |
| 131. Byte / 132. Byte | Pixel (928-BLOCK*128) MSB / LSB | | | | | | | |
| ... | | | | | | | | |
| 257. Byte / 258. Bytes | Pixel (927-BLOCK*128) MSB / LSB | | | | | | | |

Complete sensor data must be read at once. If communication falls somewhere in between, all successive data will be corrupted. Reading can be stopped anywhere by pausing the clock. A newly started reading progress through this stopped byte, continuing the clock, but the index is reset when a new conversion is started. If the bit for electrical offsets (Bit 1 in Config 0x01) is set, the electrical offsets are sampled and can be read similar to the active pixel.

After reading the data in blocks, there are a few more steps we need to take.Object and ambient temperature can be calculated from sensor output and stored calibration data. The figure below shows an overview of the EEPROM in Figure 2.2.

| 32x32d | 0x00 | 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 | 0x08 | 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | PixCmin [float] | | | | PixCmax [float] | | | | gradScale | | | TN as 16 bit unsigned | | epsilon | | |
| 0x0010 | | | | | | | | | | | MBIT(calib) | BIAS(calib) | CLK(calib) | BPA(calib) | PU(calib) | |
| 0x0020 | | | Arraytype | | VDDTH1 | | VDDTH2 | | | | | | | | | |
| 0x0030 | | | | | PTAT-gradient (float) | | | | | | PTAT-offset (float) | | PTAT (Th1) | | PTAT (Th2) | |
| 0x0040 | | | | | | | | | | | | | | | VddScGrad | VddScOff |
| 0x0050 | | | | GlobalOff | | GlobalGain | | | | | | | | | | |
| 0x0060 | MBIT(user) | BIAS(user) | CLK(user) | BPA(user) | PU(user) | | | | | | | | | | | |
| 0x0070 | | | | DeviceID | | | | | | | | | | | | NrOfDefPix |
| 0x0080 0x0090 0x00A0 | DeadPixAdr as 16 bit unsigned values | | | | | | | | | | | | | | | |
| 0x00B0 | DeadPixMask | | | | | | | | | | | | | | | |
| 0x00C0 | DeadPixMask | | | | | | | | free to use | | | | | | | |
| 0x00D0 ... 0x0330 | free to use | | | | | | | | | | | | | | | |
| 0x0340 ... 0x0530 | VddCompGradᵢⱼ stored as 16 bit signed values | | | | | | | | | | | | | | | |
| 0x0540 ... 0x0730 | VddCompOffᵢⱼ stored as 16 bit signed values | | | | | | | | | | | | | | | |
| 0x0740 ... 0x0F30 | ThGradᵢⱼ stored as 16 bit signed values | | | | | | | | | | | | | | | |
| 0x0F40 ... 0x1730 | ThOffsetᵢⱼ stored as 16 bit signed values | | | | | | | | | | | | | | | |
| 0x1740 ... 0x1F30 | Pᵢⱼ stored as 16 bit unsigned values | | | | | | | | | | | | | | | |

Figure 2.3: EEPROM overview 32x32d

Unless otherwise specified, all values are stored as unsigned 8-bit values. Small endian format is used for larger values. Fields marked in gray are used during calibration or for future use and the Heimann Sensor is reserved.

The ambient temperature (Ta) is calculated from equation 2.1

$$Ta = PTATav \cdot PTATgradient + PTAT_{o}ffset \qquad (2.1)$$

where:

**PTATav** $= \dfrac{\sum\limits_{i=0}^{7} PTATi}{8}$ is the average measured PTAT value.

**PTATgradient** and **PTAToffset** are from EEPROM.

The thermal offset of sensor has to be subtracted for each pixel to compensate for any thermal deviation.

$$Vij_{C}omp = Vij - \frac{ThGradij \cdot Ta}{2^{gradscale}} - ThOffsetij \qquad (2.2)$$

where:

**ij** represents the row (i) and column (j) of the pixel.

**VijComp** is the thermal offset compensated voltage.

**Vij** is the raw pixel data (digital), readout from the RAM.

**ThGradij** is the thermal gradient, stored in the EEPROM from 0x740 to 0xF3F.

**ThOffsetij** is the thermal offset, stored in the EEPROM from 0xF40 to 0x173F.

**gradScale** is the scaling coefficient for the thermal gradient stored in the EEPROM.

Electrical stabilization is used to compensate for changes in supply voltage. This compensation is only one subtraction, so it can be done before or after thermal offset compensation. We chose to do it later.

The compensation for the top half is done by using the following formula:

$$VijComp^* = VijComp - elOffset[(j + i \cdot 32)]\%128] \tag{2.3}$$

The bottom half analogue with this formula:

$$VijComp^* = VijComp - elOffset[(j + i \cdot 32)]\%128 + 128] \tag{2.4}$$

where:

**ij** represents the row (i) and column (j) of the pixel and electrical offset.

**VijComp**$^*$ is the thermal and electrical offset compensated voltage.

**VijComp** is the **elOffset[ij]** thermal offset compensated voltage.

**elOffset[ij]** is the electrical offset belonging to Pixel ij.

A supply voltage compensation called VddComp is used to provide supply voltage changes. To use this compensation, the sensor's supply voltage (Vdd) must be measured by the sensor, adjusting the configuration record and averaging from time to time. $Vdd_1$ and $Vdd_2$ result in Vdd.Average Voltage is calculated by equation 2.5.

$$VDDav = \frac{\sum_{i=0}^{7} VDDi}{8} \tag{2.5}$$

Compensation for the Top half is done using the formula:

$$V_{ij-VDDComp} = V_{ij-VDDComp}^* - \frac{\frac{VddCompGrad[(j+i\cdot32)\%128]\cdot PTATav}{2^{VddScGrad}} + V_{ddCompoff}[(j + i \cdot 32)\%128]}{2^{VddScOff}}$$
$$\cdot (VDDav - VDD_{TH1} - (\frac{VDD_{TH2} - VDD_{TH1}}{PTAT_{TH2} - PTAT_{TH1}}) \cdot (PTATav - PTAT_{TH1})) \tag{2.6}$$

The bottom half analogue with this formula:

$$V_{ij-VDDComp} = V^*_{ij-VDDComp} -$$

$$\frac{\frac{VddCompGrad[(j+i\cdot 32)\%128+128]\cdot PTATav}{2^{VddScGrad}} + V_{ddCompoff}[(j+i\cdot 32)\%128+128]}{2^{VddScOff}}$$

$$\cdot (VDDav - VDD_{TH1} - (\frac{VDD_{TH2} - VDD_{TH1}}{PTAT_{TH2} - PTAT_{TH1}}) \cdot (PTATav - PTAT_{TH1})) \quad (2.7)$$

where:

**ij** represents the row (i) and column (j) of the pixel.

$\mathbf{V_{ij-VDDComp}}$ is the Vdd compensated voltage.

$\mathbf{V_{ij-VDDComp}}^*$ is the thermal and electrical offset compensated voltage.

**VddCompGrad[ij]** is the VddComp gradient belonging to Pixel ij.

**VddCompOff**[**ij**] is the VddComp belonging to Pixel ij.

**VDDav** is the average measured supply voltage of the sensor in Digits.

**VddScGrad** is a scaling coefficient and stored in the EEPROM 0x4E.

**VddScoff** is a scaling coefficient and stored in the EEPROM 0x4F.

$\mathbf{VDD_{TH1}}$ is the supply voltage during calibration 1 stored in the EEPROM 0x26, 0x27.

$\mathbf{VDD_{TH2}}$ is the supply voltage during calibration 2 stored in the EEPROM 0x28, 0x29.

$\mathbf{PTAT_{TH1}}$ is the PTAT value of calibration 1 stored in the EEPROM 0x3C, 0x3D.

$\mathbf{PTAT_{TH2}}$ is the PTAT value of calibration 2 stored in the EEPROM 0x3E, 0x3F.

All we need to do to get the thermal image and separate it is to do the above given mathematical operations and steps. For this we will use a python library periphery-python. After reading the data from our sensor, after applying the steps of thermal offset, electrical offset and Vdd Compensation, the data will appear in all its simplicity. **NOTE:All steps to acquired the image are made with reference to the datasheet [13].**

## 2.2   Hand Thermal Image Isolation

Our biggest advantage in the project is that our sensor is infrared, so only the objects that emit heat are detected. The main purpose in this section is to separate the hand from the back body. The hand is in front of the body will cause us to perceive the hand warmer. For this reason, the body with hand will appear as two different objects.

The second step is to pull the hand off the background. There are many methods for this today. Among these methods, we decided to use OTSU's Method. According to this method, "it approaches the feasibility of directly evaluating the "goodness" of the threshold and automatically selecting the most suitable threshold" [1].

You can see an example of an image with Otsu's method applied in the figure 4.1.

Formuluzation: the gray level histogram is normalized and regarded as a probability distribution:

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=1}^{L} p_i \tag{2.8}$$

Where:

$N = n_1 + n_2 + ... n_{1024}$ total number of pixel

Let's divide the pixels into two class as CO and C1 with a threshold, where CO represents the background while C1 represents the foreground. Suppose we call our Threshold Value k.The set C0 implies the background pixels with a gray level of [1, ...,k], and C1 means those pixels of foreground object with a gray level of [k + 1, . . . , L]. The probabilities of gray level distributions for the two class are the following: $\omega_0$is the probability of the background and $\omega_1$ is the probability of the object [10].



Figure 2.4: Application of Otsu's method

$$_0 = P_r(C_0) = \sum_{i=1}^{k} p_i = \omega(k) \tag{2.9}$$

$$\omega_1 = P_r(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k) \tag{2.10}$$

and

$$\mu_0 = \sum_{i=1}^{k} i P_r(i|C_0) = \sum_{i=1}^{k} \frac{i p_i}{\omega_0} = \frac{\mu(k)}{\omega(k)} \tag{2.11}$$

$$\mu_1 = \sum_{i=k+1}^{L} iP_r(i|C_1) = \sum_{i=k+1}^{L} \frac{ip_i}{\omega_1} = \frac{\mu_r - \mu(k)}{1 - \omega(k)} \tag{2.12}$$

where:

$$\omega_k = \sum_{i=1}^{k} p_i \tag{2.13}$$

$$\mu_k = \sum_{i=1}^{k} ip_i \tag{2.14}$$

cumulative of the zeroth- and first-order cumulative moments of the histogram:

$$\mu_T = \mu(L) = \sum_{i=1}^{L} ip_i \tag{2.15}$$

is the total average level of the original picture. For any choice of k, we can easily verify the following relationship: [1]

$$\omega_0 \mu_0 + \omega_1 + \mu_1 = \mu_T, \qquad \omega_0 + \omega_1 = 1 \tag{2.16}$$

calculated the class variance:

$$\sigma_0^2 = \sum_{i=1}^{k} (i - \mu_0)^2 P_r(i|C_0) = \sum_{i=1}^{k} (i - \mu_0)^2 \frac{p_i}{\omega_0} \tag{2.17}$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} (i - \mu_1)^2 P_r(i|C_1) = \sum_{i=k+1}^{L} (i - \mu_1)^2 \frac{p_i}{\omega_1} \tag{2.18}$$

Between-class variance is:

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0 \omega_1 (\mu_0 - \mu_1)^2 \tag{2.19}$$

Within class Variance is:

$$\sigma_W^2 = \omega_0 \mu_0^2 + \omega_1 \mu_1^2 \tag{2.20}$$

Total Varience:

$$\sigma_T^2 = \sigma_W^2 + \sigma_B^2 \tag{2.21}$$

Calculated Otsu threshold:

$$\sigma_B^2 k = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \tag{2.22}$$

and optimal threshold $k^*$ is:

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \tag{2.23}$$

After these procedures, we have a threshold value. When we ignore the parts below this threshold and take the parts above, we will see a clear hand shape.

If we cannot perceive the hand clearly as a result of the Otsu method, then the Using the Neighbor contour detection algorithm, we can make the image what we want [10].

## 2.3 Hand Gesture Recognition

Motion history image (MHI) helps in understanding the template location and path as motion progresses in a static image. In MHI, temporal motion information is narrowed down to a single image pattern where intensity is a function of the novelty of motion. Hence, the MHI pixel density is a function of motion history at that location where brighter values correspond to a newer motion [4].Using MHI method in the project, the information will be obtained whether the gestures are static or dynamic. Take two continuous frames. Some noise, background image, moving object information for the captured images in functions.

$$I(x, y, t) = b_t(x, y) + m_t(x, y) + n_t(x, y)$$
$$I(x, y, t + 1) = b_{t+1}(x, y) + m_{t+1}(x, y) + n_{t+1}(x, y) \tag{2.24}$$

where,

$b_t(x, y)$: Static background for t th frame.

$m_t(x, y)$: Moving objects for t th frame.

$n_t(x, y)$: Background noise for t th frame.

$$diff(x, y, t) = I(x, y, t + 1) - I(x, y, t)$$
$$diff(x, y, t) = b(x, y) + md(x, y, t) + nd(x, y, t) \tag{2.25}$$

where,

b(x, y, t): Overlapped area in consecutive frames.

md(x, y, t): Motion region.

nd(x, y, t): Noise (which can be ignored based on situations).

If there is no movement, it will be defined as no movement, since this difference will be 0 [4].

### 2.3.1 Static Gesture Recognition

Deep learning techniques have emerged recently and advances in convolutional neural networks (CNN) surpass the classical approach to hand gesture recognition as it eliminates the need to derive complex handcrafted features from images [12].CNN's automate the feature extraction process by learning high-level abstractions in images and capturing the most distinguishing feature values using the hierarchical architecture. Thus, it solves the disadvantage of obtaining inconsistent property descriptors when working with large numbers of motion classes with very small cross-class variations [17]. Convolution layer,"Each image of a picture is processed by convolution, and the width and length of the image are com-pressed to obtain deeper image information" [12]. Pooling layer,"study found that every time when the convolution, convolution may be missing information, pool layer can solve this problem. No compression when the convolution aspect at the same time, to retain more information,compressed work on completion of the pool floor" [12].
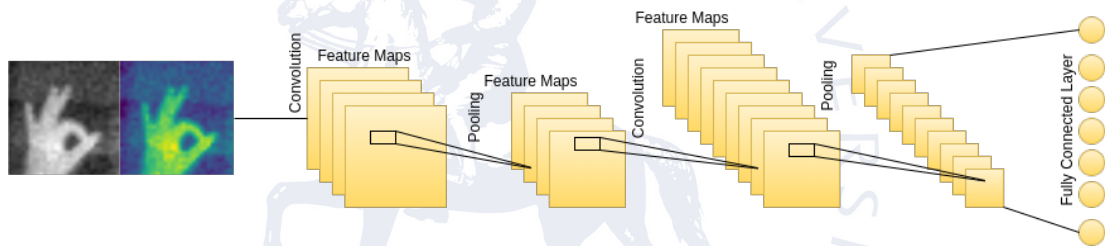


Figure 2.5: The architecture of a typical CNN [17].

"Deep learning is an extension of the neural network architecture that automates the process of extracting features from meta data by operating through a series of hierarchical layers" [17]. The project aims to use such a deep learning architecture that uses a convolutional neural network to recognize static hand gestures. Convolution layers contain units called feature maps, each of which is linked to local patches in the previous layer via filter sets [8]. The same filterbank is used in all units of a feature map, and different filter banks are used in different feature maps in one layer. This architecture allows us to easily identify local patterns distinctive from images, even if they are in different parts of the image [8].
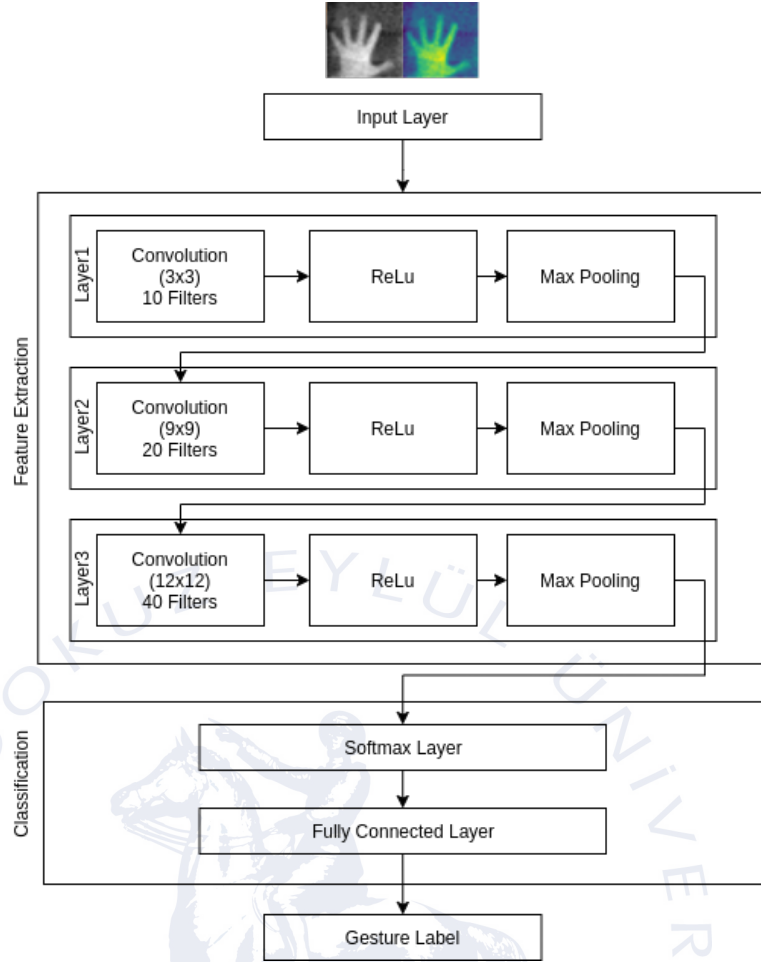
Figure 2.6: The Schematic Representation of the Proposed CNN model for hand posture recognition [17].

The locally weighted sum obtained by the filtering process is filtered and filtered through a nonlinear function called ReLu (Flattened Linear Unit) to stabilize the convolved results. The pooling process is included in the CNN structure to group semantically similar properties from the convolution layer. Thus, the architecture of a CNN includes two or three convolution layers, then more convolutional layers with linear activation and pooling layers, followed by pooling and activation, and a fully connected final layer that performs classification [17].

### 2.3.2 Dynamic Gesture

defined CNN algorithm for dynamic gesture

## 2.4 Matching Gesture to Commands of Smart Mirror

A number of scenarios have been prepared for command mapping for smart mirror. The scenarios prepared are as follows;

For Static Gestures:

We have 4 fixed gestures. It is the ability to Open, Close, Touch and Return to the Home Page.



Figure 2.7: Gestures of Open,Close,Touch,Return to the Home Page

For Dynamic Gestures:

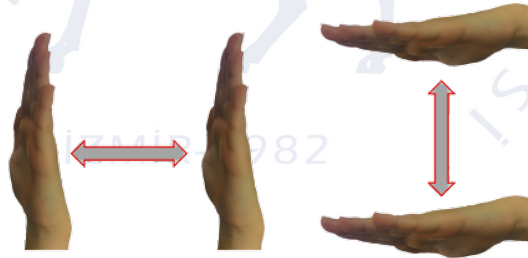There are two mobile situations. Down-Up, Right-Left properties.



Figure 2.8: Gestures ofDown-up,Right-Left

In total, we activated 8 features of the smart mirror with gesture.Afterthat, by activating the Raspberry Pi OTG communication channel, intelligent Mirror and MCU communication is provided.

## 2.5 Sensor and MCU Communication PCB Design

A sensor PCB design will be made with the selection and supply of the passive infrared sensor. In the PCB design in question, a schematic design will be made first and the necessary components and connections will be made schematically. After this part, the schematic design will be turned into a printed circuit. In the design of the printed circuit board, along with the schematic design, the physical properties of the place where the sensor will be mounted will be taken into consideration in the current smart mirror design. After this part, the designed sensor PCB will be produced in an external company and its assembly will be carried out in the company.

The pcb and schematic drawings of the communication PCB are as shown in the figures below. Schematic and PCB design will be done using KiCad EDA - Schematic Capture & PCB Design Software. In the next step, support will be received for printed circuit operations. After soldering the printed cards, our sensor communication PCB will be ready and the test phase will begin.
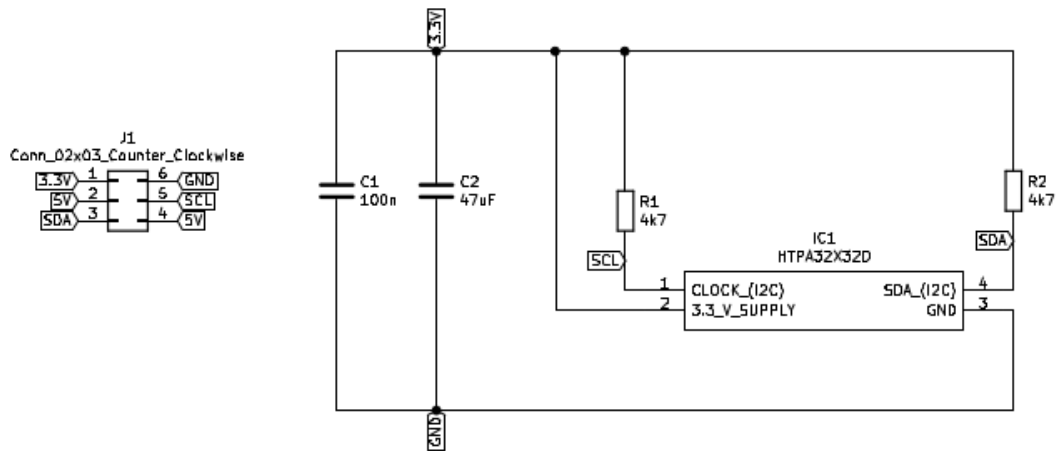


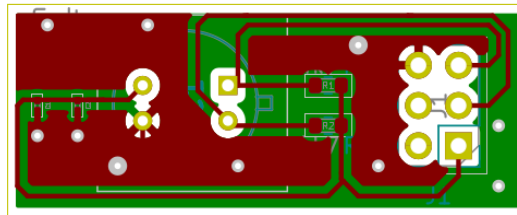Figure 2.9: Schematic Design of Communication PCB



Figure 2.10: PCB Design of Communication PCB

# 3. Work Plan and Work Packages

| Worker:Rabia DOĞAN | Time: 7 Weeks (2020.10.01-2021.01.15) |
|---|---|
| Work Package Name:Literature Review and Material Supply | |
| Inputs:<br>• Past studies.<br>• Articles on internet.<br>• Thesis on internet.<br>• Books. | |
| Steps:•<br>Searching for articles Gesture Recognition with Infrared array sensor.<br>•Searching the OpenCV library and Python programming language.<br>•Searching convenential neural network for gesture recognition<br>•Searching Thermopil Array sensors<br>• | |
| Outputs:<br>•Determining Thermopil Array Sensor.<br>•Determining methods to be used. | |
| Status:Complated. | |

| Worker:Rabia DOĞAN | Time: 3 Weeks (2021.01.11-2021.02.03) |
|---|---|
| Work Package Name:Sensor PCB Design | |
| Inputs:<br>•HTPA32x32d Application Notes | |
| Steps :<br>•Drawing Schematic Design<br>•Drawing PCB Design<br>•Testing Sensor PCB | |
| Outputs:<br>•Schematic and PCB Design of Sensor PCB. | |
| Status:In Progress | |

| Worker:Rabia DOĞAN | Time: 9 Weeks (2021.01.03-2021.03.15) |
|---|---|
| Work Package Name:Gesture Detection | |
| Inputs:<br>•Methods of Gesture Recognition<br>•HTPA32x32d Datasheet<br>•Books about of Image Processing | |
| Steps :<br>•Reading sensor data with the help of the python-periphery library.<br>•Making calibrations with EEPROM calibration data.<br>•Removing the hand from the background with the Herbaceous Method.<br>•Hand condition detection with Motion History Image method.<br>•Defining gesture with CNN | |
| Outputs:<br>• Capture sensor Image<br>•Detected Gesture Types<br>•Tracking Gesture Recognition<br>•Matching Gesture to Commands of Smart Mirror<br>• | |
| Status:In Progress | |

| Worker:Rabia DOĞAN | Time: 3 Weeks (2021.03.16-22.03.30) |
|---|---|
| Work Package Name:Smart Mirror and Sensor PCB Integration | |
| Inputs: <br> •WP1,WP2,WP3 outputs | |
| Steps : <br> •Connectting Raspberry pi and Smart Mirror with OTG Communication | |
| Outputs: <br> •Complated the project | |
| Status:In Progress | |
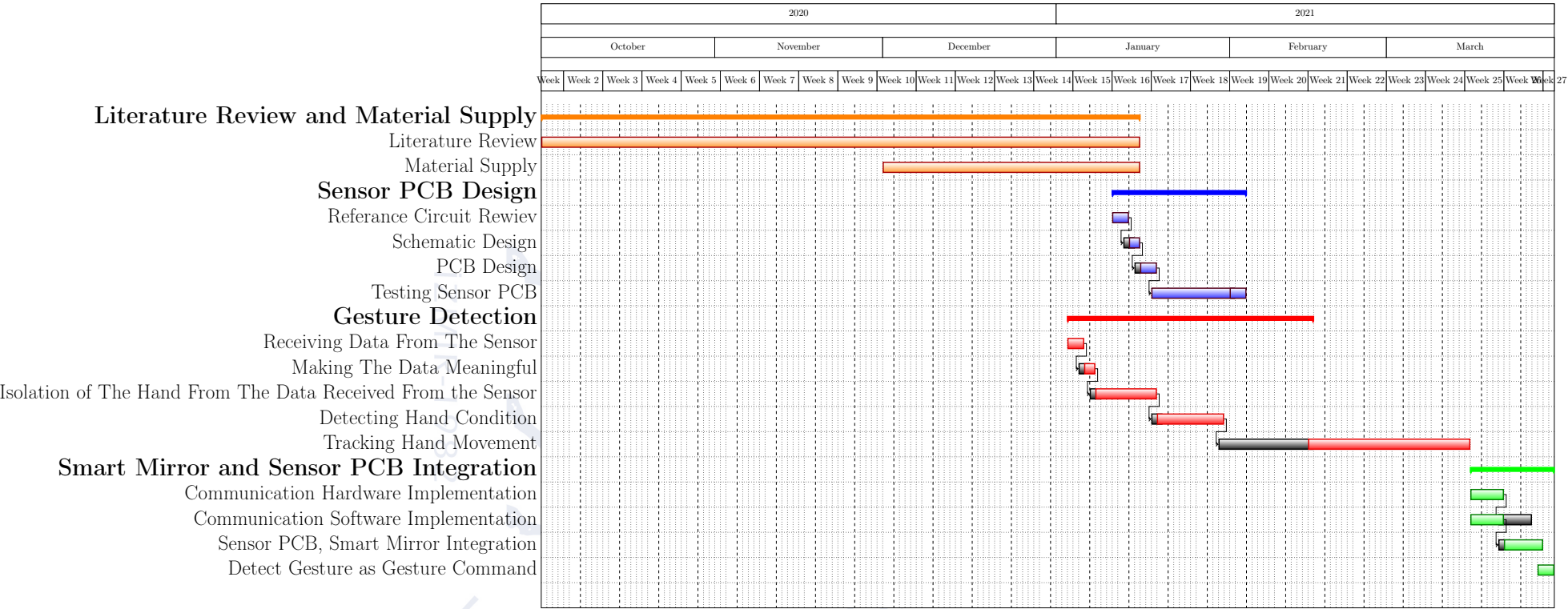
# 4. Cost Analysis

## 4.1 Economical Costs

In this project, the economic cost is only in the hardware part. A Raspberry Pi 4B and Thermopil array sensor are used in our project. The market price of Raspberry Pi 4B is $55. The thermopil array sensor, on the other hand, has been chosen the most suitable for our project and has been researched for the market. The price of the HEIMANN HTPA32x32d Thermopil array sensor is $87. There is currently no separate cost in the software part. Because using opensource software.

## 4.2 Environmental,Political and Social Costs

Infrared sensor arrays have emerged in recent years and are products that allow the application of thermal imaging technologies to consumer electronics. With the introduction of Covid-19 into our lives, it is frequently encountered in the field of thermal imaging. Since a consumer electronics product using these products has not been produced in Turkey yet, it is considered to contribute to the national knowledge. In the project where privacy is prioritized, the privacy of the person will be ensured by using a thermopile array sensor. Also, the project is executed by Vestel A.Ş. so the project is expected to inspire different projects within Vestel. The fact that these sensors are newly introduced to the market will pave the way for innovative applications in many fields and thus method changes that can turn into patents. For example, although gesture recognition is performed with cameras conventionally, it offers a significant innovation in methodology, using passive infrared arrays. Smart mirrors that can be controlled with gesture recognition can have a positive effect on the preference of the products in question by the consumers. In doing so, using a technology that does not impair personal privacy will ensure that the users put the product into their lives safely and the applications related to the product become widespread.

# 5. Conclusion

The reason for the project is that the system made with the existing infrared sensors does not work as desired. The main purpose of the system is to renew the smart mirror again to appeal to the upper segments. 8 features have been defined for the new gesture detection system that we will add to the smart mirror. However, if the device is desired to be released to the market, the number of transactions can be increased without touching the smart mirror by increasing the gesture suggestions. The Thermocouple system, which dates back to a very old past, is becoming increasingly popular today. Based on this formation, it was decided to use the Thermopile array sensor, aiming both to prevent reflections around and aim at the confidentiality of privacy. The most important points in determining the sensor are the frame rate and the pixel value to make the image clearer. The conclusion reached in the research based on these two properties was the use of the HEIMANN HTPA32x32d sensor. Also the project needs a processor. While the tablet processor in the Smart Mirror was assumed to be sufficient at the first stage, it was later concluded that the performance of the tablet processor was insufficient for this project. While moving on to the Raspberry Pi, the Raspberry family, which is one step ahead in price-performance comparison in the market, is considered sufficient for the project in terms of speed and operating system. Image acquisition was the next step. At this stage, the periphery-python library in the python language to get the image comes to the fore with its ability to read large pixels and give fast results. If we assume that the image taken is an offset image, the offset values are calculated and stored separately for each pixel via EEPROM, which is one of the advantages of the selected sensor. With the data received from the EEPROM address, the image is finally in the clearest form. Thermopil array sensors detect all objects that emit infrared rays, so it is necessary to isolate the hand from the background image. The Otsu method, which is the most advanced of adaptive thresholding methods, is one of the techniques that will yield the best results for this project. Now there is only a hand shape in the image. In the system with 8 gestures, it is possible to separate two as Static and Dynamic gestures. The not too complicated MHI (Motion History Image) method will

be used to distinguish between the two types of gestures. By comparing the images taken as two frame sets, it will be understood whether there is a difference or not, and whether the image will be dynamic or static will be distinguished. Two separate CNN (convenantial neurol network) clasification algorithms will be implemented for dynamic and static images. Among the image classification methods, it is the CNN method that stands out for gesture detection. Finally, the Raspbery Pi and Smart mirror will be integrated. Looking at the project, the most innovative methods will be used on the basis of the method to be used.

# Bibliography

[1] Otsu, N. "A threshold selection method from gray-level histograms". IEEE Transactions on Systems, Man, and Cybernetics, volume 9, no. 1, pages 62–66, 1979.

[2] Schilz, J. "thermophysica minima thermoelectric infrared sensors ( thermopiles ) for remote temperature measurements ; pyrometry", 2001.

[3] Gonzalez, R. Digital image processing. Prentice Hall, Upper Saddle River, N.J, 2008.

[4] Ahad, M. A. R. Motion History Images for Action Recognition and Understanding. Springer London, 2013.

[5] Ghafouri, S. and Seyedarabi, H. "Hybrid method for hand gesture recognition based on combination of haar-like and HOG features". In 2013 21st Iranian Conference on Electrical Engineering (ICEE). IEEE, 2013.

[6] Prasuhn, L., Oyamada, Y., et al. "A HOG-based hand gesture recognition system on a mobile device". In 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014.

[7] Chen, W.-H. and Ma, H.-P. "A fall detection system based on infrared array sensors with tracking capability for the elderly at home". In 17th. IEEE, 2015.

[8] LeCun, Y., Bengio, Y., and Hinton, G. "Deep learning". Nature, volume 521, no. 7553, pages 436–444, 2015.

[9] OpenCV. "Open source computer vision library". 2015opencv, 2015.

[10] Yuan, X., Martínez, J.-F., et al. "An improved otsu threshold segmentation method for underwater simultaneous localization and mapping-based navigation". Sensors, volume 16, no. 7, page 1148, 2016.

[11] Kawashima, T., Kawanishi, Y., et al. "Action recognition from extremely low-resolution thermal image sequence". pages 1–6, 2017.

[12] Li, G., Tang, H., et al. "Hand gesture recognition based on convolution neural network". Cluster Computing, volume 22, no. S2, pages 2719–2729, 2017.

[13] Lupp, S. "Htpa32x32dr2l5.0/0.85f7.7ehic thermopile array with lens optics rev3.0". Technical report, Heimann, 2018.03.28.

[14] Ba, N. L., Oh, S., et al. "A 256 pixel, 21.6 uw infrared gesture recognition processor for smart devices". Microelectronics Journal, volume 86, pages 49–56, 2019.

[15] Pinto, R. F., Borges, C. D. B., et al. "Static hand gesture recognition based on convolutional neural networks". Journal of Electrical and Computer Engineering, volume 2019, pages 1–12, 2019.

[16] Tateno, S., Zhu, Y., and Meng, F. "Hand gesture recognition system for in-car device control based on infrared array sensor". pages 701–706, 2019.

[17] V., A. and R., R. "A deep convolutional neural network approach for static hand gesture recognition". Procedia Computer Science, volume 171, pages 2353–2361, 2020.