**Smart Home project**

Summary report

Advisor: Mony Orbach

By: Daniel Levi and Lidor Elfasi

Contents

[Introduction 4](#_Toc160291112)

[Goals 5](#_Toc160291113)

[Developing tools 6](#_Toc160291114)

[Jetson nano 6](#_Toc160291115)

[Deep Learning 8](#_Toc160291116)

[Keras 9](#_Toc160291117)

[Convolutional Neural Network (CNN) 10](#_Toc160291118)

[MobileNet-V2 11](#_Toc160291119)

[GPIO (General Purpose Input/Output) 12](#_Toc160291120)

[OpenCV (Open Source Computer Vision Library) 13](#_Toc160291121)

[Data augmentation 14](#_Toc160291122)

[Hardware block diagram 15](#_Toc160291123)

[Project definition 16](#_Toc160291124)

[Define the problem 16](#_Toc160291125)

[Challenges 16](#_Toc160291126)

[Research 17](#_Toc160291127)

[Model 17](#_Toc160291128)

[Dataset 17](#_Toc160291129)

[Improvements 18](#_Toc160291130)

[- Hand Detector 18](#_Toc160291131)

[- Data Augmentation 18](#_Toc160291132)

[- Transfer Learning 18](#_Toc160291133)

[Training block diagram 19](#_Toc160291134)

[Training summary 20](#_Toc160291135)

[Inference block diagram 21](#_Toc160291136)

[Files hierarchy 22](#_Toc160291137)

[Summary 23](#_Toc160291138)

[References 24](#_Toc160291139)

# Introduction

# Goals

# Developing tools

## Jetson nano

The NVIDIA Jetson Nano is a small, powerful AI computer designed for makers, learners, and developers. It provides the capability to build practical AI applications, robots, and more. Key features include a microSD card slot, 40-pin expansion header, USB ports, HDMI output, and Gigabit Ethernet. It’s particularly useful for edge AI and IoT applications1.  
  
The NVIDIA Jetson Nano is a single-board computer specifically designed for AI and machine learning tasks. Here are some key points:  
Purpose: It’s intended for edge computing, meaning it performs AI computations directly on the device (rather than relying on cloud servers).  
Hardware: Jetson Nano features a quad-core ARM Cortex-A57 CPU, a Maxwell GPU with 128 CUDA cores, and 4GB of RAM.  
AI Acceleration: The GPU provides hardware acceleration for deep learning models, making it suitable for real-time inference.  
Use Cases: Jetson Nano is used in robotics, drones, smart cameras, and other embedded AI applications.  
  
The NVIDIA Jetson Nano is a small AI computer designed for makers, learners, and developers.  
Purpose: It allows you to build practical AI applications, robots, and more.  
Hardware: It has a quad-core CPU, a GPU, 4GB of RAM, and various ports (USB, HDMI, Ethernet).  
Use Cases: Jetson Nano is used in robotics, drones, smart cameras, and edge AI applications  
  
The NVIDIA Jetson Nano is a compact, energy-efficient AI computer designed for edge computing. Here’s what you need to know:  
Purpose: Jetson Nano is specifically built for AI and machine learning tasks. It enables developers, hobbyists, and students to create AI-powered projects without relying on cloud servers.  
Hardware: It features a quad-core ARM Cortex-A57 CPU, a Maxwell GPU with 128 CUDA cores, 4GB of RAM, and various I/O ports (USB, HDMI, Ethernet).  
AI Acceleration: The GPU provides hardware acceleration for deep learning models, making it suitable for real-time inference.  
Use Cases: Jetson Nano is commonly used in robotics, drones, smart cameras, and other embedded AI applications.

## Deep Learning

Deep learning is a subset of machine learning that involves training artificial neural networks with multiple layers to learn from data representations. Unlike traditional machine learning approaches that rely on handcrafted features, deep learning algorithms automatically learn hierarchical representations of data through successive layers of abstraction. This enables deep neural networks to effectively model complex relationships in data, leading to state-of-the-art performance in tasks such as image and speech recognition, natural language processing, and reinforcement learning. Deep learning architectures include convolutional neural networks (CNNs) for image processing, recurrent neural networks (RNNs) for sequential data, and deep belief networks (DBNs) for unsupervised learning.

Deep learning is a subset of machine learning. It involves neural networks with three or more layers. These networks simulate the behavior of the human brain, allowing them to learn from large amounts of data. Deep learning powers various AI applications, including image recognition, natural language processing, and self-driving cars2.

## Keras

Keras is an open-source deep learning framework written in Python that provides a high-level interface for building and training neural networks. Developed with a focus on user-friendliness, modularity, and extensibility, Keras allows developers to quickly prototype deep learning models with minimal boilerplate code. Keras supports both convolutional and recurrent neural networks, as well as their combinations in complex architectures such as siamese networks and autoencoders. It can run on top of backend engines such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK), providing flexibility and interoperability across different computational platforms.

Keras is an open-source Python library that provides an interface for building and training neural networks. It’s user-friendly, modular, and works seamlessly with popular deep learning frameworks like TensorFlow. Keras allows you to define and train neural network models with ease3.

## Convolutional Neural Network (CNN)

A screenshot of a video game

Description automatically generated

A Convolutional Neural Network (CNN) is a type of deep neural network specifically designed for processing structured grid-like data, such as images and videos. CNNs are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers use learnable filters to extract features from input images through convolution operations, preserving spatial relationships and reducing the dimensionality of the data. Pooling layers then downsample the feature maps, further increasing the network's translational invariance and computational efficiency. Finally, fully connected layers perform classification or regression tasks based on the extracted features, enabling CNNs to recognize objects, scenes, and patterns in visual data with remarkable accuracy.

CNNs are specialized neural networks for image analysis and computer vision tasks. They excel at recognizing patterns in images. CNNs consist of convolutional layers, pooling layers, and fully-connected layers. They automatically learn features from raw image data, making them ideal for tasks like image classification and object detection4.

## MobileNet-V2

MobileNet-V2 is a lightweight convolutional neural network architecture optimized for mobile and embedded devices with limited computational resources. It builds upon the original MobileNet architecture, incorporating improvements such as inverted residuals and linear bottlenecks to enhance model accuracy and efficiency. MobileNet-V2 employs depthwise separable convolutions, which factorize standard convolutions into separate depthwise and pointwise operations, significantly reducing the number of parameters and computations required for inference. This makes MobileNet-V2 well-suited for applications like image classification, object detection, and semantic segmentation on resource-constrained devices such as smartphones, drones, and IoT sensors.

MobileNet-V2 is a convolutional neural network architecture designed for mobile devices. It uses inverted residual structures and lightweight depthwise convolutions to achieve high accuracy while being computationally efficient. It’s commonly used for image classification and other mobile AI applications5.

## GPIO (General Purpose Input/Output)

GPIO refers to a set of pins on microcontrollers, single-board computers, and other embedded systems that can be configured to either input or output digital signals. These pins allow hardware peripherals and external sensors to communicate with the computing device, enabling a wide range of interactive and control applications. GPIO pins typically support basic operations such as reading the state of external switches, buttons, and sensors, as well as controlling the state of LEDs, motors, relays, and other output devices. GPIO programming involves configuring pin modes, reading and writing digital values, and handling interrupts or events triggered by external signals, providing a versatile interface for interfacing with the physical world.

GPIO (General Purpose Input/Output) refers to the pins on a microcontroller or single-board computer (like Raspberry Pi) that can be used for digital input or output. GPIO pins allow you to connect external devices (sensors, LEDs, motors) and interact with them programmatically.

## OpenCV (Open Source Computer Vision Library)

A logo with a red blue and green circle

Description automatically generated

OpenCV is an open-source library of programming functions primarily aimed at real-time computer vision tasks. It provides a comprehensive suite of algorithms and tools for image and video processing, feature detection, object recognition, camera calibration, and 3D reconstruction. OpenCV supports a wide range of programming languages, including C++, Python, Java, and MATLAB, making it accessible to developers across different platforms and domains. The library's modular architecture allows users to easily integrate and extend its functionality with custom algorithms and external libraries, enabling rapid prototyping and deployment of computer vision applications in fields such as robotics, augmented reality, medical imaging, and surveillance.

OpenCV (Open Source Computer Vision Library) is a powerful open-source library for computer vision and image processing. It provides tools and functions for tasks like image manipulation, object tracking, face detection, and more.

## Data augmentation

# Hardware block diagram

A close-up of a computer chip

Description automatically generatedA blue led with two small sticks

Description automatically generated with medium confidenceA white rectangular object with red and blue lines

Description automatically generatedתמונה שמכילה עיגול, עיצוב, רמקול

התיאור נוצר באופן אוטומטי

# Project definition

## Define the problem

A diagram of a person pointing

Description automatically generatedFirst, we had to name our problem so we could find the appropriate sources which would lead us to the solution.

We defined the task as a **Hand Pose Classification** task. The hand position is what we care about in the image, and it is what we must classify for understanding the action the user wants the system to do.

## Challenges

As part of the project definition, we also defined the main challenges we will probably meet while developing this project.

* We need to complete our task using Jetson Nano which is a not very strong edge device. We must think about our limited resources.
* We need to get a fast response from the system.
* We need to get high performance, even though the camera gets an image of the whole room and can get a lot of noise from the picture.
* We need to choose a dataset which will maximize the accuracy of the model, by being large enough and robust.
* We need to train a model without losing much time and effort, considering we will need to retrain the model several times.

We managed to solve some of the problems and reduce the effect of some others by pre-thinking.

# Research

## Model

The common solution for image classification tasks is CNNs. We decided to look for a CNN model which gives high performance, but still doesn’t use many resources of the system.

We had to dive in into the subject of CNN[[1]](#footnote-1) and found two networks which might fit the task-

* Resnet18
* MobileNet-V2

We decided to discover them both.

## Dataset

For simplicity, we chose to use a dataset of the numbers in sign language. These gestures are much simpler to remember for people who know sign language, and for people who don't.

![A collage of hands

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDsRXhpZgAATU0AKgAAAAgABQESAAMAAAABAAEAAAE7AAIAAAAHAAAIVodpAAQAAAABAAAIXpydAAEAAAAOAAAQ1uocAAcAAAgMAAAASgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEtoYWxpZAAAAAWQAwACAAAAFAAAEKyQBAACAAAAFAAAEMCSkQACAAAAAzU2AACSkgACAAAAAzU2AADqHAAHAAAIDAAACKAAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDE5OjExOjA0IDE4OjU4OjM3ADIwMTk6MTE6MDQgMTg6NTg6MzcAAABLAGgAYQBsAGkAZAAAAP/hCxlodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDE5LTExLTA0VDE4OjU4OjM3LjU1OTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5LaGFsaWQ8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwACAQECAQECAgICAgICAgMFAwMDAwMGBAQDBQcGBwcHBgcHCAkLCQgICggHBwoNCgoLDAwMDAcJDg8NDA4LDAwM/9sAQwECAgIDAwMGAwMGDAgHCAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM/8AAEQgAlQLTAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A/fyiiigAooooAKKKKACiiigAooozQAUUUUAFFFFABRRmigAooozQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUZozQAUUUUAFFFFABRRRQAUUUUAFFGc0UAFFFFABRRRnFABRRRQAUUUUAFFFFABRRRQAUUUUAFFGaM0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFIXCtjuaAFoppkUd6cGzQAUU0yKGxuGfSnZzQAUUZoouAUUA0buaLgFFG7FJvX1ouAtFIGBFJ5i+tFwHUUhcA0ucUAFFGaCcUAFFAbNFABRRRQAVy/xQ1u40DQY7i2k8qTztmSmf4HrqK5L4vRrL4ft93/Pyp/8AHHqqfxgefX/xZ8Qxf6vUv/IKf/EVnS/GvxNFJ/yEv/JZP/iKq+I9nl/LXK6nf+V89ewqNLsZ6nYX/wAb/EltZyTNqyQRxx+Y7vDD+7/8cry2/wD+Cjml6Xf+RP8AETSGf/pnDC//ALJXyR+31+2G+qeZ4P0m88uxj2Sai8L/ALyT+4lfNnhffqlxI837iD/Yr4XP+JqeEreypRPqMryH6xS55H6sxf8ABRLSZfLjXx9ohkk9TCn/ALJXSWP7Vmr+I7ffpfiCzvs/c8hIH/8AZK/I3VdZsIhs8mSeT+/XIxeIvK1DzrO8uLH5/vpPXm4fjDn+OmdVThj+SR+ums/tP/ETS7O7Zte2yW74KfYrX5P/ABysvWf2yfHUugeda69tn/68oP8A4ivif9nz456rdaHJZ6zr0mpQf8sPO/5Z1654X8bw38nk+Z/0zr3KOcUcT/DPJxGU1cN8Z9E/C/8Aao+IGveGbi4v9e866junj3/YrWP5AiH+5V/Xv2ofH0MO6317Z/25Qf8AxFeZ/Dm6SLS57bf/AKyfzP8AxyOtyLZf2+xq9yn8B4NS/OUdf/bY+KWlu+3xIP8AwXWv/wARVfT/ANu74mybPM8Qb/pp1r/8RXNfEHw55XmVwF1N9gjk/wCmdUSfRFt+2/44Mcfna9HH/wBuMH/xFO8aft7658MvAGo+KNe8R+RpWnoS7pZQSea5+5Gnyffr5Zude+ypvlf/AIHXHftGeN7b4l/sV/EaztXk8/Q0g1KP/gE8dctSpyQkzqwtPnnCEj6j+DX/AAVK8QfGHW/IstauIPOG+CK5srXMn/fCV6vP+1L44NxGkOtf6x8f8eUH/wARX5Jfsg+PP7C8WaHMs3l+XdJI+/8AuPX6TWEX7yN/+edceT4uWIXvndn2ChhakeQ9Mk/at8ZRXvk/2x/5Kwf/ABFeY+Ov25vih4fu9V+z+I41jt3cQ5061Oz/AMcqG/lc38j15z8RrXzLPXHl/wCWjv5devUilseLTqHrHwf/AG3viZ4r8NwTX3iTzppP4xp1rH/7JXKeO/27fjF4bv5I4vFreXvwmdMsf/jFcR8B7r7L4btE/wBipPiPFDrPnwr/AKyo5UbHbfD/AP4KLfFDVb/ydQ8SRyf9w61j/wDZK7y5/bQ+IkOr2v8AxPv9Fk+//oVr/wDEV8D/ABp+KFz8DfAeq+IbW3+1Xen7I4IX/wBXvd/L3vXof7KH7Rl58cvC9wmspbx6lpeyTzoU8uO4R/8AYrl+sUfa+w+0df1OtOj9Zh8J9lxftgeOML9o17b9ofYjpZQfu/8Axyqcv7X/AMRLPULi2k8QfvLc5/48bX7n/fFeRePNeh8OeF7S8urm3tbW3fzJ5ppPLjjSr0OtW/iP7Df2c1vdQXEHyTQv5kciV1e4cXv2uel3/wC1/wDESwTP/CRLs/v/ANn2v/xFUbr9s74lRQTuviFP3aOU/wCJfa9v+AVwt15NxZxov+rrDl/dWc8P+x5dUSeiax+2j8UofDV9NF4qKTxQeZG/9mWv/wAZo0z9tz4ma14SW4j8TbLop97+zrX/AOIrz+6tVuvD8if7Hl1wfhLxb/Y3h/yf9Z/yzrXlgd0aZ1kH/BRz41aV4uazvPGXmW6vgf8AEosU/wDaFeuaJ+2z8SNSghk/4ST/AK6f8S+1/wDiK/OH43ftJPa/tA3WiW9nHHBp8CSO7/6y43/vK+nvg34tTXvD1rMreZHIledh8RCtUlCJ3YjL50aEKsvtH0drP7YfxIsLfzv+En/d/wDYOtf/AIip9T/bB+IkMFrNF4h2xyffT+z7X/4ivG/Hl/5Xh/79Jo/iOHXdLjhjubeeSN9joj+Z5ddvuHH7I9oi/a5+I2xt3iDp/H/Z9r/8RVGL9sr4lfaGRvEHT/qH2v8A8RXn0t3+8kWqt3/rKfuGZ6foX7Y/xGv/ABM9rJ4h8yPyXk2DT7X/AOIqcftjfEKTX5rX/hIT5cf/AFD7X/4ivJfCUufHEn/XCrth/wAjxP8A7lTZF+zPT1/a0+I2k+K4YbrxJ59jqCfucafajy3/AO+KxPjT+178WPCVrHdaX4q8m3/jQ6Zayf8AtCuT+Jcf2AWNyv8AyznrN+I3i3Tf+Ff31zqVxbwWtnBvnmd/9XU1PZpXLp0bnqXwg/bV8deO9EZZ/EH+nBPvCztcflsq9e/tX/Eiynj8zXv3f8f+hWv/AMRXwbF+1JD8NPD+pa9pM0c8dnA8iI6f6x/4K7j9iP8Aab179oaTUtN8TXNvdTyWX2+B4YEj8v5/nT/x+vPjmFH2kIfzHo/2PUdOVb+U+0dI/ax8ba/oE80ereXcW7/On2WD/wCIp9r+1J44lf8A5DX7uRPk/wBCg/8AiK+UfG/7Tdn+zT8P9Y1vUoZL6T7VHaQWyP5f2id67j4BfG62+Mnw/j1WG2+yyee8E8O/zPLf/WV1fWKPtfY/aOb+z6nsvb/ZPcLv9q3x9bXGz+2v/JGD/wCIrk/Gv7YvxI0jxCsVv4i2wtapIVOn2p+fe4/uewrJ1T97XB/EaXHiC3f/AKYJ/wChyV0HPTpnoun/ALbHxNlT954mH/gutP8A4isq5/bm+Ky615aeKv3P93+zLXP/AKIrgYv3VnJXM2t2n9oSbn/jrOZ0ezpnu0/7b/xO8v5fEw/8F1r/APEVmxft0/FSW4Rf+Eq/8pdr/wDGK8T8Y/EvQ/h9p3na1qVvYwSP5ab3/jqx4S8R6f4tt47zTbmO6tZPuOlc/tIc/IafV/c5+Q9ru/26fidbf8zV/wCU61/+MVEf24PimZFVfFzf+C2y/wDjNeYy2CX/AO5as218OPFqG6KbzPLT7laGfsYHot/+338XLfzNvipv3b43/wBmWX/xmtn4e/t6/EzXYpI7rxMrzL9z/iXWnP8A45XidrF5unybqh8GxZuJ6BeygfRFt+2/8SjrkkLeIf3cf8A061/+Ip+u/tvfEbT9TtIYvEP+v+//AMS61/8AiK8Uiv8A7Nqm+X/lpsqbxHdJdeKNNdf3kdehGNM4vZnsA/bS+KGi+ILWS68SR3Wl6g/kJ/xL7WPyJP8AvisX44/ts/F7wVaw32k+KNlv/Gh0y1k/9oV85/tS/tVaV8G/FGl+FVsJNR1K4gTVp/n8uOzg3+XH/vyffqj+118fbnRv2T9V1XS3j/tKR7GCC52f6uCd/nf/AK6eXXDUxlCHPH+U7qeX1fc/vH138JP26/iB478ISTNr26+jTP8Ax5Wv/wARU+nftufECDULeO917fHImXxZWo/9kr4b/Yj+N1zFc+TdXPnx7/L3/wDPRHrsP24f2pbn4GeKPAeieH7Wzk1LxQk9/PczJ5n2eBH8v/yJJUxxtD6v9YKllNb639Xgfak37XXjv7ElxH4gjktz/GtjB/8AEVVP7WnxEO918QeZHvx/yD7X93/45XhP7PHjf/hYPwvkubhPIkvE+0bE/wCWb112ly+bbxzN/rJE8t/+mlehTjTnT54nl1Kc6dTkmehD9sL4iB/+Rg/8krX/AOIrF0f9tf4mX+oT7vEw8vfhE/s61/8AiK4m6i8rUN6/6uSuV8L3X/Ew/wCB1jUiuczmenax+2f8XP7amt7PxhGv7veiPpdr/wDGK0vDn7bPxK8R6RNC3iT7LqVudj50+1/+Irx+1l83xndf7lV/+EtS1+IEG1PL+0QeW/8A00rP3DPmZ1HhT/gob8XtH8fPpmteLDcQB8DOmWSH/wAchr1bxD+2L8RNO8lrfxD+7k/6h9r/APEV8AftffG6bwR+0RpulWFtb7P7OS/eb/npveSPZ/45X0v8L/FEPxL8D2Lo/wC/jrkw+IhOcqX8p3VMHWp0YVX9o92sP2uPH1/pkc0XiH95/H/oVr/8RWbqn7XHxQilgeLxJ8n8af2da/8AxFfLfw5/av0HWfixfeGNLmvJJNPvXsJJn/497h0++le1ap+61CRV/wCWn7yOt8PUhW+AxxGHrUZe8d7J+2L8Rx/zMP8A5T7X/wCIrO1f9sz4kWdl5i+KNreYFH/EvtMn1ziPjnPSuI1n91b71Suf8XTbLKzX+9MT+S1tG19TOjqz1HX/ANtf4m2Ph97iPxNtcAYP9nWp7j/pnWnp37YnxFm0K2lk8SDzpkHzmwtRg9enleleLeKHEng19rY464+laVqFn8FMrZ3Qw71I9kJ/pWnuPZHfGmtzS+Kv7c3xq8BatHInjEtZuwyBpNkQB9fKzXo/w+/bU8eeL/DK3DeId02zJb7DbD9PLrwfxJ4j0PxrpEeiyalpp1poty2n2hPOyB/dznPsea5/9n/xgfC+vSabdBtrO0eG7dR0rkpyjKTR01MNyrVWPqPRf2wvH1zeNFN4g3Nn5SLG2GP/ACHUmqftV/EaMkR+JNueB/oFsef+/VeRa1Z3GnXDS27KI1kDhiflx16/061q6Nrn/CRWV0GXyry1Iz/t9/5Vv7nNy9Tk9npc7bRv2wfiYPOS68SFnjbAP9n2vH/kKrVx+1z8R9p2+IiGxx/oFr/8brgYPLuj9z5pF3bvccmqk0vl3pDcLjP4VXLHsL2aOu1v9tj4mWGrw28fibaDHlx/Z1oTn/v1VLXv24fijY3tikPifaJnAcHTbTn/AMhV5b4quf8AitJB/CqgD61V8YyGPU9LYEBd4yaLQtcuVPS571f/ALY3xE+1/Zx4kKMyjB/s+06/jEa8r8W/t/8Axq8D+NVhufFzTWEkm1VGlWWPzEIP5Vk+JdfjOlR3UbeZNZzAtjuuM/yzXl/7cPxQ0/4YfD/TdUuLVrq51C/Wxt1LFVX5d5OfwIrCrVp04uUjfDYX2slBLc+wrT9sr4gal4YS8h8RESFAzE2Fr+P/ACzp3h/9sH4hanbyK/iItJgEYsLbpn/rnXy/+yV8WoPiD4HNozLCzAqATuwcZxmvQNRsrvw1febDJtVIs57VdOUJwU4k4jC+xqOEketat+1p8TljkSDxN5ciqSM6fa8Y/wC2dV9I/bF+JU1mvneIm8zHP/Evtf8A41XC22sf234ct9Q24bO2XHp0zUiQ+dHIeI3XHzf3h2rTliczp2Pef2aP2kvGnj/416LpOra39s0+68/zYfscEe/bBI45VAwwyg8HtX1fXwp+xtMsv7SXh/d95ftOPf8A0WavuuueaszKWjCiiipJCuR+MX/Iu2//AF9L/wCgPXXVxfxtvlsPDdoW/wCWl4kf5o9aUfjQHj/jKX7Jp/zV5D8X/iVD4I8B6rqVwnnwWdq8joj+X5ldV8VfFD6pqH2a1SSTy6+P/wDgoT8ZE8GeA4/Dazf8TLWP9eif6yOCuvMMR9Vws6sjrwOD9tW5T4w1Xxlc/EH4gaxeXH7v7ZdPPP8A9M6f4y8dzeB/C01/b2El9Bb/AH4U/wBZWdoGg3UXz/8ALP8A3P8AWVlfGPxa8Vv9n0uGTVdSt4HkS2T/ANnr8Nqf7RX9pM/U8LT9jQMPQf2jLz4v3F1YaDbeXJbw+ZdPN+7+zp/t1wel+F/EPh22uvs+vR6lPvedLbZ5f/AN9cz8EfFFzoNvdXNw8kdxrE7yajs/5aVqeLPG9toMv2lZv3dfaYPKcOqHvnztbHVvbWNrwv8AtQarpc/kyw3EF1bv5bw/6vy/9+vrz4LfHO51nwRYzMklr9nT77/8tN9fmn4cur/xv45utSuvtEkF4/mPsr6a+E3iO5lkg021e4kg/j315WHp8mK5KJ34z38N75+lnwq+IN5r/hdL9X8ue3n8v/rp/q69c8G+Mk1m3+b/AFlfPv7Mlg5+B8l43/LO9eN/++I66bQfiCul6p/y0r9HwvwH5jiPjPavEezVLORHryPxRYeVHOldrpfje21m3+V/Lk/uPXCfEvxHb6NL/pF1Z2slx9xJp0jkkrerJIwo3Z5X8WpX0bR9m/8A4+HrzXQdeS6+HXxN0pv9XeeHrr5P+uaVm/tGfF/xR4x8Z6l4Y8F2enyf8Iu6R6jeXn/LS6dPM8iD/rnHXlHg34g6raxaxDq0Plz3mnXVo/yeX5nyV83ic4pRqexPpsLk9b2ftjo/gPr3/EvtYVePzI0SSv0w+FeqPqnhjTbyX/V3lrG/z/7lfmX8OfBul2F5Y7bPy5/ITe/+rr7y/Zp+I80Wj+HNKZPtVjJAke93/eRpXFwvUvUmd3FFP9xCZ6bqkvm3Ejr/AKv+CuE+I0X/ABJ9V/3HrvtesP7LvJ1/5Z7/AJK4vxvF9q0fVdv7z78dfY1D4emcP4S1VvCWj6HM3+ouIPnrc8RXSRanI3/LCT94j/7FY+qWH2rwfBCv+st0SSOvP/GXjS5sNHvrCV/3f2V9j/8APOsanwHV8Zwn7cWlrL8F/FVrF+/kksvPg2f8tNjxyVwH7D3iO/0bWIPtDxwR6hZfJsf+4/mVe8R+Mk8b/B7xBpUsn7/T7K6/74dK8h/Za+Jj2ul2rrDcR+XImyZ0/v8A7uvmMyxHJjKUz7LK6PPgKlE+q/8Agq/4z+y/s7+A9N86SP8A4SDxDHvRP+WiQQeZ/wCzpXo37Cm+X4Bxyf8ALP8AtG62f+Q6+V/26/i0uvfET4SeG5LmPz9P0SfUp9//AD3nn8tP/IcFfXX7GSfYfgHBtSP/AF91I6J/10rrw9bnzOX+E48RT5MpjH+8ei2Evmx7Kzr+6/4mkaf89Hq3ay/Zb2T/AHKytU2f2wj/APPN6+gPl/tmray+S8kLV4z43mfS9L1zyf3clu7+XXsV1/x8b68r+I1t5WualG3+rvE8yrxHwHqUz8+fiXfXUXxYtbyL95PeJ5Du7/3K+xP2UPGT2vheOzun/f2c+z5P7n8FfH37Q8X9jeJJJP8Anzuq9U/Zt+I00n2HzYbiCO4Ty97p/rNn7xP/AGevjqeI+rY7/EfYVKf1nLf8B9C/tI/Gn/hF/C+q3m+Tz47V5INn+58lcB/wTdu7m68SSfvpJPMsnkn3/wDLR98deeftX/EFP7EukZ/+Pi6gsP8A2o9ezf8ABPu1SXVL65h8uST7Emx/+en7z/7CtsRiHPMKdMxwmFUMuqTZ9NS3/lahUl1dJLVO5/eyRu1Lf3X+j19QfI+zH+HJU/4TSTb/AM+T/wDocdaWlSeX4sumb/nhXM6DdNaaxPN/sVpapqiXV55y/wBzY9ZnRTpl34taok3guR/M8uOP95vr5J/aR/aH0fXfgh4gsLCaS6/tC12I+z938j+Z/wCyV7/8ULr+1PhfqVn/AM/FrPB/5Dkr837/AF3+1Pg/sV/Mks0eN/8AvivBzzGSouMP5j6DJcvhWTq/yml4o8RzWvw72N9ojgvLV/vp/HXvX/BM7WbbRviR4fdZpP8AiYQT2D73/vp5n/slfNPjjWf7U/Z3guY4bjy9PdPndP4Nnl16h+x5K/g3XPCupLfySSf2pY70f/f8v/2evlcHjP3lP/EfTYvD81Goeof8FItQmlvND0G38uSS41d7/Y7/AMCJ/wDZ173+wpYTWHwLkeRPL8zVH+T/AIBHXy1/wUJ1651P9ryxsLW2kuoNLsk8/Y/+red9/wD6L2V9ifsb2Dxfs72L/wDLO4urqRP/AEX/AOyV9Jh6nPmcj5/E0+TLInqkV+stvXD/ABB1BP7cgRv9X5H3/wDgdddajybfy689+KEvla5bv/yz8j/45X0h8rTpj7qXyoJPn+SvnD9qr4+3Pwq0ix/stI/t2oXXkI7p5nlp/HU3xp/bDm+HPiyDRLOxt7qSRPMd5nrwH9s34gw+N/D/AIPvI08v+0Hkn2b/APYrxcwzD93KEPiifQZfgJ+2hOcfdkcP8c/jxf8AxA1Tw5DeTefNG88jon/LSvpD9lX4tJ4T+EWsTXSeX/Z8/n7H/wByvh/xvL/Z/wAXNDhXzJPMtXkr1i1+I83h34T+I0k8yCS407+P/f8AL/8AZ68HD4x/WFP+6fQYjDr6o4f3j6S/Zp/bI1v4q+OI01JLf+zdQd/IRE8v7P8A3K+pfCWy6nndf7lfnd+xvaw6X4o0ORvM/dv5f3/9ivvz4Z3T/Z9Sf/Y+SvayfETrU+eZ4eeYeFGpCNMLWw+yyTw1B4ctfK1CdK6q1lttZt45v+Wn8dMGgwxXEjL/AKyvaPBOR1T/AJCEjf8APOoLDVP+KntYW/26n1mL/ieXyLXlfxg+INzo1vavZvJBPH+8d63lU5DH2Z5r/wAFLIktvip8OfEKpI/2yyn0mfZ/y0dH8xP/ACG715X8VfGd5r3wb1LTbya4kgt7JI/Jd/4EePZXqP7dmvJ4y+C+h6xbv+70/UbW/wB6f8s/M8yN68E+MniiHVPh/PeW83mR3mneQ7p/2zr5HMqnJjpf3on12X/vsDH+7I9i/ZfmhiuIHsPk+0WvmbP9tH/+NvTP2zfEc3jz9pnwjbNc+W/hvw9HH/108+eSSuY/Zptrbw54o8Mw2dzcSSXE72773/vwSf8AxFY/jK/vPF37Yni6/t5reSx0+f7B5L/3IE8v/wCLrnWM/wCEz2f9473h/wDhR9p/dPvX9kGb7L4HtbP/AKdUr1G6le1Nqn/POvGf2bpZtH/se1b93JcadBJs/wB9PMr2zxRF5tvHt/uV9xgf4ED4LGfx5kN/L/ocj/7FcP4cuvNkkeum1nzIvD90+/8A5YP/AOi6818JeI/K8xGpz+M4ahe1T4g6P4I1We81rUrPTY5HSBHmf/WPXF/FXx5DoP2XUlfz45PubH/1leSf8FGLV7Wz8K+IYHk8iz1R7R/+Bp/9hXK3PxG/4SP4X6UnneZJp++3/wDiP/IdePUxn750T0qeXw+rxxMCp+1fqi+LviZ4D1vzpIPtmlvA+z/Yn/8As6+if2S/G76Fp9ilw/mRx/u5H/3P3dfI3xku5vEfhvwP5U0cElvdXUDu/wD2zr2L4VeMn8L/AAn8TXktz5n9j2r3aP8A9sP/ALCvHwuI5MfM92ph+fAQPP8A9kG/ubr4xz6ls/d3mrz3aPv/ANZveSv0g+1/ao7Wb/phX5l/sR/bLXUNN+1W0lvJv+R/9Z5n7uv0p8GXX9qeE43/AOmFelw/79Kb/vHncTfxYf4Sf7V9qt64/wAe35s1s1H8TP8A+y/410VrK8UkiVyPxMv8TWKkZG98/wDjle1WlZXPn8LrURdvdUS88IvGzDKr83tXRaTIY9AaNwyMIw3zDGVC84rgNXWSKxZrZgsm0AE/dUngE/jXy7+yd+0xrWjfFK4tfE19dOLrUGstRSaQssB3smVz90BtmfY+lcNTHqk1GXXQ+lwmX+3jKa6K9jgNf8ZX3w+/bJ8U+dP5jWWrx3aSceasfyuFz6FfT6V9eeLJJLfxFHqsPy7mSQ7evzbWz+BIr44/bb02HQf2x/NuPMUatYqJPLIUB1d0yfXhV6eor6U8N/FKG++G2jTH/SpprNY8D5zlDtbJHccV4+CxEliZQbPVzKnz4WFaC1WhS/by+M+qWnhH4dix1C4sre6v7hL5YpNqyOIo2Xdjk/xHj0r3H9k3x1L4h8PxzTyGaWa12Ox53YJAPP0r5F/bQ163HhvwHJN5axf2nLIfMPABgUdK96/Yu8TL/Y9qvybcFMo2VHyhu3+9XRRxMnmXL0sTWw98tTSV76n0dbXZs4U3fxSFfwpup/vRuX72cD6dqbr8ZSGFo+md3FVxdCSJee4r6KWx8z7NHD63P9p8azLn+JR+lZfjvWXt9ZtV3bVjdV3HsT0qObW1j+Ic+5uPO28143/wUT1PUtC+Heg6xpkjx2EepIt8FfaWXlVyeo+crXFisR7ODktbHbhcP7aoqTdrtIyf22fiXq2j/DPRpNLu5bW3k1dIL/Y+1pAVfy8Ec43qOenHvXN/G7xVL8Zf+CfFxdPL9q1TwTq9tcsxfc0iCXaST6bJif8AgFebfE34lSa/8C9T0uY/aR5SXNuc/OFjxKh55zuVh649jWn8Dtdt/E3wZ8YeF2aP/ipPD1zApjPJmhhaRDj1JJ/74r5epjnUneT0krfefTYfDypU9VrGX4I9G/Yd8bXVruEy+Ws3+kIAeo4U/wDoQr7jhu18SeELhlZW/c9fcV+X/wCwl8SZ73xFpMMsM3k3mwK7DA/eIBj/AL6Vfzr9JfhBcfavD15C/wAuGYIW4DjBHBPXnjivXyPEKph3d7Ox52fYflxCa6q5H8NL9rrwXdQM2TG7Lj8a3lvvJkWAn+AMD6jFcf8AC6+W3vb61bj9+Rg/XFdNrinTtR4542/SvcPnalPsenfsgfP+0z4bYcD/AEr8f9Fmr7ur4N/Y1nL/ALSfhlWH3vtWP/AWavvLNYz3OWt8QUUUVJiFeb/tM2v2nwNa7TtaPUI3/wDHJK9Irgf2hYPtPgu1Ve96n/oD1pR+NAeB/b0ijk82GOOT+/sr8r/2vr+58SftQeKkv5P3lnevGn+5/BX6ffEb/iTaP5zyV+V/7Zskn/DXGpTM/wDx+WqSV5vF1Oc8BzH03D/+8cpY8EaCkun7tv7z+55HmVX1nS/sEck1wkcf999nl10Hw+sLm60+N7f/AFkifJ/zzrqpf2fby/j/ALS1a/8AtUn+sS2hTy46/HcGfpFT4LHwVdfBHWNL0eSaWw8v53+RH8ySNN9eXar4Tm1n4gWOjy+ZB8/mT708v/gFfp/f6XYWD/8AHnH5f8CU+X4Z+HviNYbNU0G3n/2/J/eR/wDA6+qp5xNQ9meL/ZsOf2h8T3/7MmlW2hxtpbXGjzxp5nyfvI5P+AV3H7JfhLyvFF890/nyW+yPfs8uvfviL8Kk8OeF/sdn+8tbz93A7onmR/7D1zfhz4c/8K+8QxwxJ+7k/eV2ZLTnOc60zhzipDk9mfTXwbuvsHw7khX/AFcl08mz/gEdYnjL/QNQ85au+BLryvBf/bd//QI682/a58ZX/hf4H+I7/S/+P63g8tHT/lnvr7qMuSh7Q/Oa1PnxPIHxV/ah0T4VeBPEc0fiHR4PEGj6dPPa2zz/ALzz9nyV+dXiP4g32p3H9vazf3mq32ofvPOmn8ySR66rwZoVn4ksLqG4fzJNQR45Hf8A2/79eO6RaTf2HdWF95kd94fnewm/6ZulfB4/NKmK0PusvyuFBanefszftSTeFvE+uQ3Ulwkn9ovIjvP5knzolexfEb4vp8S/DclnbvZwSSP573MKeXJ/uV8V+LYv7B8RSOj/ALvWIU2P/wA83So1+PeueE5razult7y1uHSPzt/lyJ/v1yywc6v8E9b65CEPZzPq7wtr2sXWuQOuvahH9n+4m+v0G/4J7eMpvHmiSJeeX/aWhukEj/8APwn8D1+Xvwv1S8tdQ+03Vz9qkk/gT/V29fX37FHxkm+Gn7RHh/zXkt9N1xP7Nut7/u/3n3P/ACJWmQ4r6tjI02ebnmH9tgz9JvFt15tvvWvPLDWP+JrqVtL5f+veu7m2X+n/APTSvD/G+syaNrF88X7uT7U/z1+oVD8zpnSazFbaDJH88cccn9+vEv2grBNL1iSa18uSC8g/grub/wARvf2/72T95s/jr59/ab/aM8N/BvSI5teubjzJH/0W2tk8yS4/v1y1KkIU/fOzD05zqe4eD2vxb03S/Hl9pX2+OT7RA9pdIn/LPenl1h/DTxQ+g6JY6bcTeZH8mzZ/sPXH/D74g+FZfg/Y3lvbefdb7qe6+RPtEk8jyffrG+Gni3VPFviC1hur/wAmOzTy4IURPLr8+xmO9rU55/ZP0bL8HClT5P5j1fVPHlv8VP2oPEF4qR31jpfkWCb0/wBZBap5f/xdfoB+xvavpen6Hbf6uD+zv36b/wC/+8r8+fAejW0vijVG8ny5LdEjgmR/Lkkd/vp/tx1+gH7KOspFqE8P+sjjg8tP+AV7PDf76rUrHk8UctKjTowPVtUlj0+4kSsu6i823hmX/gdWvHnnWt5vWsfS9ZWXT7qFv+WaP/6BX1h8XSNSXxRDa2+y4rzn4oapD9o3q/mRyJ8j1Y17WftVxI9eR/GnWXufB+q/Z5vLm+yz7HT/AJZ/JJWFfEfuz3KdP94fL/7S2sWOs+MNVewvLe+jt3+d4X8yON63NG8Zf2N4esbmV/3dvsn+T/lnXkfwCtba/wBPkt5Vjkjkg8zY/wDy0r1Tx5f2fg3wvBt8v93Akab/APlnX5njMw9pXjP+U/RMvyuEKEofzHM6h4zm8eeJLHd/fefY/wDy0319ffsUWvla5JcRp5Ef2pI/k/5abE/+zr5H8IWia9rED3ieZ/y0/wCmkdfYX7FEqHT7H/ppvk/77evX4fqTxOOlWmcWf4f6rgY0UfS2s7LWsqKX7fbyf9dKwP2hvjJo/wAINDgudYvPIkvP9RCn7y4uP9xK8u8Jftw6Df3EaSaVrFra3H/Lb5JPL/4BX2FbMKMKns5TPjsPl+IqQ54xPd/Dlr/pMj/89P3dUdU/dahOi/8ALOtjw5LDdaXBcwyRvBcfvEdP+WiVh39/Da+KJPNf93JWkzCmeL/EH4jXOn/EiOxlmk+wyP5aJ/yz+5Xy1aWD6pqHirwYsMn+jvPAj/6uP/Yr3f8Aa5tX0bUEvrf/AJZ7JE/4BXzfqnjx5fixfXlqn7vVESff/wAs4/kr5PiLWlz/AMp9Tw7pW5P5jH8EXV543/ZsvtNWHzLq4gktNj/u/nR67j4XxTeCH8MPfzRx2un3VrJevs/1aI8e96o61488N+CPB+xrzy/nff8A89Lh9/mPsT/rpXI6z48m8UeXcrDcWtjI/mO83/LOvjadTkn7Q+6+rwlD2cz0rWfiEnxa/ac8VeJ/9Igtby68yy85PL8yBE8tP/IaV+jH7Ktg1t8ANJhuP3kkaef/AN9/vK/M/wCFWvWGqaXd+Vc28klx+7jTf+8/36/Ub9n2VJfh3HDF/wAs08uvsOF/31apWmfG8UU4UYU6NMu3V/8A6RGn+3XHfEuJIdYj3J/BWr9qeXWJ0/551h/EG6+1axbv/wBMK+ymfGnwN+2lYJF8aIJmeSGP7L5e9P8Afrx340+I7aXw34Os/tMkn2PUZ4Pkf+B08xK+gv29dGSLVIL9k8zy5/LevBfHnw5/4Wl/Ybs/2G10+fz3TZ+8kr4bNKnscRP+8fbZXT9thof3WcL48g+w/FjwTNarcT/aLW6gf/lp5afJWh8afEbxaPpWlN5nmahOkD/7m/zHr0L7B/Ysmz/nmnl/cqO1uYdUuI/N8ufy/ufJ/q6+eWM5Zcx9N9U9pDlO8/ZQ0GGXxpY+VD5cnzyPs/3K+2fhff8Alpdf886+Qv2ZIrbRvHF1MyeXJ9l+RE/369+0b4s23hfS768lkjgtY0eSd5v+WaJX2mRuEMP7Rnw+eR58X7NHpujaz5VzOn+3W5Ff+bXx3rP7ZuseI9Y3+H7aztbGN/kmmg8yS4/+Ir6F+Evjy/8AGXge1v8AVLaO1urj+5/q5E/v16WHx1KtPkgcGMy2rQhzyNyWX/ipLr/frx39obRvKt7iZE/1dd/f3/8AxUF98/8Ay0rjvHcr69Z31tK/mfaIX2V6lT4DyqfxngOq6onjf4J+LvDFwnmTx2T3dr/002fvP/Rif+RK8ysJX+LXwnsbC1WPzLzZs/5529HjL/idfGfQ/D32+40q1vEee6eF/LkuE/54Vx3xG+KE3wR0+PTdBSzt/tE8++2/1nl/7dfB5xiITqxjH4on2WQ4fkpS5vhkeheF/Fr/AAl+IPh/Xrp/tVjpd1572yf6y42J5fyVl/BGLUNT1nWLzUof9K8QXUkiPD+8j3zz/c/8frh9BttY8b+H49Y1K5+1XVv+82J/6BXqf7I3iO2v/EnhWHybzzNU161nf5PLj/4+o9if+OV5tKpOcoUvsn0FWnDknV+0foBrWjP4N8cb4v3f9nwJAn/AE8uvW/NfWfC9jeMn7y4g+f8A368k+L+ofZfiRdbf9XcPXqlrL5Xw+t/+maeZX6xR+FH5LiNzE8Ry/wDEovoV+/8AZX/9Arwe/wDFEOg2++4fy5P4K9e1jVPtWl6rJ/zztZ//AECvm34o74/Lf/lnIlc+IOeZU/aSj/4WD8CvEGmyp58luiX8H+/B+8/9F76+O/Fv7Q6eHPD+m22kW0ck+ob40fZ+7j2V9O/8LBTRpI7a/wD9RJ+73vXxrf6Dptr8fodKv08zR9Hvp5E+T93J/cSvlc69zlrQPouH/fhLDVD1fwRqt54y8H2Ntcf6VPb3vno//PP5JN//ALJW5rPxBufDnwP8Y6DKskmq64kdpaon/PPf8/8A5Drxr4jfFS88G3kEPhm8+w+Zv3vDB5cdbGg2s0vg/wC3+dJfXUifPvfzJJK+V+uVoz5z66jh6PJ7I+k/2LdBfVJY02SRyWenTzujp/H+7j/9nr7g+C+s/atHk02X/XxweYlfKH7DN/DfweJvs/mSf2fpcEaO6eX/AB19NfAK++1axPJ/y02fPX33D1Plwlz4fiSXNi2jqrr/AEW4keuB+Lc++XT/APaLj8crXaeLb/y7ySH/AG64P41XrR6dYSx/6xRIR/46P616mI+E8TB/xonN+IfiD/wjGlSeZtk2fIoPO89MV8efHCWxsvjRJ4kmPk2PiSJZrmBDtUXKDZIB6bwoavZ9a18zSz218zeTcHJkHPl+/wCHWvn/APaU08XHhu5tWxMtq4u7d88vggde2Rng+tfN5pF1adlufZ5XU9jO72Zl/tqfESHx7ongXxet64urHzNJv54gQznaGjk5H8QT+dczd6t8QNQ8J6WVu5rfw/ZrJNai3uQk824g73HBI9u1cj4z1BfHfww1TS1KizWFHRuVIkjdWjI/3UDofXf6VV+Hf7Sdze+Ak0mO4aOWztjbESKGbB+U4PuM18vWxNSSU4fEtGfWUsJDl5Z/C9UfQGseO38dfDHwvZ3s8d5NodxcoHYA+dG4j2OSfcMPXj6V9QfsszfYNBt7iGNfJ84q+32jj444r86Pg2V1PVmZ5mLQ/KA8pKqo56dM1+hP7LOvQr8LytuyFVvSrlex8qOvQyWvKePvPseZm0YwwTUe59gfbIdb8OwXEP8AqXXaBj7pFZ4/dRqoGWyDj8ad8Px/xb1P4ud30qml8susRbW+XO0/Wv0B7XPhjxjxdepaeK9QmkkWNUmbknGDg1zvxS+z/Ff4Y654euJIXj1KxZEJ52uPmRx7hgp/CpvjhZyrr2peWu6Pz23egry6y8b/ANmoIbqTy1VggkxwBnvjmvLqa3i9nodtNcqU1utT5rt9Vt7HRbjRZkkbWLYvbyl/mZ3GAwGOgx3PB3Vg/Aj4sWvgwrasoS80272OxJ3bUPPT+8hb8sVuftATx+CvjvNqUabTrkMdwu3ld+drYxx6GuSHgez0GXUNVjXfd6g/nTBgcAdevSvhcVJ0qrj2dz7vBwVelz91qdT8A9P8V+AEudRtZre6m0G+ms4bMvyxgkKxtuHHUA56HFegfsnftoePPhl8XI77VfEl1qVnqEwt9U02/kVogjOd3lj+EpkEEdSor5p0f46X3gXxXcWr3TmG4wyy52+Zyc8Huc55q94T1zT/ABB4zVw3mysTK3PIJPWpo4urQ0js9S8bhqVWKct0rH7BaDqqeHviDeM2WtWkLxbjnCnHfozDI5GQRXpGuwfaBG6n5ZFEi8dQa+YP2bPH7eL/AIWaAtxObq4tEWwlkc/MTEdpP44H519P6xKYNCtmBwqxhfwr9EwtT2tKNR9j8+xFL2dZxO+/YylLftM+G12/d+05Pp/os1feY61+f37Dl79p/ac8P85ybnH/AICzV+gI61tLc8nE/F8haKKKk5wrhPj/AHkdh4Pt5JPufbEH/jj13deM/tt6l/Zfwos5F/6CaJ/5AnrSj8aA+W/jn8Rvt9zJDE/7uOvzR/br15/+F6WLq8nmfZfL+RP+mlfcHjfVHv8AzHr8/f24dZ+wfG/TXim8uSOH599a8SU/+E2R7mU+5ioHtn7Pt0kujx/aE8//AGHr2yw1T7T+52Rxx18r/CD4gQ2tvAnnfvI69u8OeN4bqTetzHHH/v1+AUanIz9MkTa9oMP/AAlEkLf8tPnSuu8JaOlhaeX/AKuSsmWw/t7ULWZX/eRp5iVY8R+LbXRtD+0+d5fyfcevSp1LmPM46kd/oH/CUeJJLWL95Hp8D38//AP/ALZsri/Ful+ZeRzV71+y18OXv/g/qvjO/hkj/wCEkdLSyR/+fVH/APakn/ouuL+NPhO20bVP9HfzPn+5X6ZlGVzo4ONSf2j4fMMZ7XESOY/4SNPC/wAL5LyX/Vxu/wD7Tr5Q/aC/aHvNe8P6rZy/8eMkEkb2yf8ALSvqH4g6N9q+Ac7/APLS31H/ANkjr4G+KsrWusTw/wC3XZiueGFseDRpwlijznSviNNoOoSJElxHHI//AC2Ty/LrB+LXiO3l+I9jeRv5baxZf6bs/wCWjwfx/wDfuup8W+Db+/8AIvLW2kvo9/79E/5aJWB8U9F87wJpv9l+G9UbWNHvfkSaDy/tCP8AfSvg6fJ7Wx+hL+Gec/tB3/8AxR1jfQ2ckMBk3wXOyuZk8C2/xI1LS4bP7smx7qaF/Mr0PxNPf/FLQbfwvJo95paW+z7VNc/8s0/2P9uvSfAPw7sfA2ix2ljZxrDG/wC8X/np/wDbK7KmMhRp2h8Rj9X55mb4I8JXHh3XI9LvP9Bn8hJLWbY8lveJ/fR//Rle4eHJbn+yo7eX/j4j+48M/wD4/VHwva20sa6bqX/Hr/x92t5s8ySzk/57p/0z/gkSprX/AIkOoXSMkcc//PGH/VyP/fT/AK6V4UffrXNp/wALU/T/APZp+I1z8Vf2e9D1i8/5DOx4Lp/+fh0fy99cl8QZf+JxqX/POSd6vfs0+HJvhz8G/D+lXX/H1Ha/v/3n+rd/3lYfxBuvK1zVU/6bv/6HX63hef2EOY/LK3J7efIcj4o1n+y9L86T93bxpvd/+edfnP8AtkeN/wDhY3xhkmmf/QY08iBH/wCWab/v19+fHwvL8N/3X/Lwib6/N39oPw59v+IF1u8z93D5ibHrzc8/g8h6eR/xuY8LubDUvCPiS+/su48uSOf5E2eZHcI/8FdJ8EPic099JczfJdRzbJk/5Zx/7lSWWo2/hLxJb69rTKukokFuGj52SJ/l62Ph34O03Wbie+sV+wyaxevd2v2lP3ckb/6uN/8ArpXy9bklQ98+0p+5M+gPh9f2f9sWM10kc8kc6SJN/wA+/wA9fdHw01R/BGoWN5GvnzSXvz22/wAuS4g31+dvhKX+wYLpPJkg+z/6I8L/APLN6+9vhV4jTVPiRpU1x/q7N0j+f/cr1uF/hkfP8SfYPprxj+9nRN/mR7PkevPNR32El9/1wf8A9ArrZrr7f97/AJd65XxF+9uJ/wDpoj19UfNUtzi7/XY/L218B/tj/EvWPEfjeTR11K4j02TfH5KP5cdfa3xk36Dp8nlfu5JE8tK/PX9pbzovEEjxfvJLd/Mrws05/Y8p9NlfJ7bnMn9njWb+61iD995ckaeRs/56V2Xjz4lvruuRvdRyR2Onv5cEL/6y8nT93/37rD/Ye1PSrDXLt9e8vyLxHgff/wAs/M8ys3xv5N/8QJ3W5+1Qef8AI/8AsV8JiKcOc+/w8vcPSvDlrNr2l/afOkgnj/eQQpP5cf8AuV9NfAz456J8G/hBpuvX9z+8s/3H2BH/ANMuHT+DZXy18Obryvn/AOWcf8dbPiTS01TWIJl/efaEeT7n+rqstzOeFc+QMywsMRCHOdF8QfiXrf7RnxAn8T6y/lz7/wDRYd/+j2cH/PBK+gv2c/gi/wAUdH/t6/S4sdD+5aon+svP9v8A6514f8L/AIaTeMtXj0e3/wBX/r7r/rglfcnw51+HS9Pgt1SOO1jTy0RE8uOOvfyHAvGT+s1j57NsyeGh9WonpPhewhsPB9jbW6+Xb2/7tE/55pXK+KLDzdc8xa62wuoYrePyv9XXO+Mv9GufOWvsqlM+Q9oeK/tNaWuvfD+62/6yz/8AQK+B/FHi258JawlzawyTyW7/AHEr9A/jJdpF4f1KH/n8g8tEr4G1mw+0+LLq2b+/XyWe/wC7yPosg/3mJwvjL426X4luGuL+4t4ZLf8A57P5fl1vfD/4jL4n0CBZfL8u4T79eS/tUfDf+w76D5f9Y/3667wF8JfsOj6c+l3my+2J/o037yO4/wBj/Ykr5Grh6X1dTW59l9Yn7Q+iPCN//Y1nBtht/Lj/AHkDoiV+m/7OmspL/ocTeZBcWqTo6SV+Vngi6TS7d3ZJPL2fJvf/AFdfo5+xlf8A2bwHpLyv+8uNLg8t/wDgFfT8I1Je8fN8UfZO6i1BLXxxfQy/8tKx/iNF9g1SPb+8j8itX4oaZJYaxHqUf+ruE+euV8R699qj/e/8s08uvtT5E+V/20tUhufCk7t9/wC1fJXiWg3VzLpUflQ+X/tu9eqftr69D/wg3y/6zz68h+FXiNNQ0va3l18FxH8Z9pw3U9yQeKIvNs/9Im8+T+4lc/FfyS/efyI//IldJrNr9lv5P9XIkn365+L97ceSifvJP3lfL6n1B1XgjWX8Ea4mq2aSSQR/u56PiX8ZJviD5mj2CSR6bv8AMnR/+Xh//jdSeF4vstv5LN5nmffR6o3WhQ6fqF1cN/q4/wB4lelRx1ZUvYI4amDour7dnY/s0/DlPEfjyxs5fMj0ezdJ73/pp/cT/tpX3xLpdtLZweSkccezy02V8/fBb4cp4I+H+m+ZD5d9eJ9run/23r3Pwlf/AG/R40/5aR197lOD9hRufCZxmHt61kcPrMP/ABNL7/frivFuqJo3mXMsnkRx/wAb16B4j/dXF1J/03eN68B+NOs/b7i+VW/d2/7tK7sR7kDyqfvzPlf9snVPtXjCDVdBupLW6jfzLWb/AFflvXi3iXx1qnjLxF9j1yzjg1K4Tz4Ln/V295sr1v4+2qyyaa8qeZ5e+SP/AKZvXkv7V/xe07Wvhnoem2mj3Fpr2j3Xnw3/AP0zf+Cvi61OFbE8kz7TL6nJhjtvhV43ml0eOz3/ALzZ/BXsXhH4gzaDeabeM/7jR54J02f7D+ZXy/8ADDwdeaNZ2s3nXD/J5k8yP/x7v/t/9M69s0vVM+F75bj955ieWj7K8lLkxFoHpyqc9I/Tv4qf8TTxRprp/rLz9/H/ALj/ALxK9Q8J6ymp/DeT/lo9v+4f/pm9edeN4obDR/DN5/y0/su12P8A9sI66z4Qf8iPPu/1moO++v1rD/Cj8yxG5iRf6V4T1J/+Wmyf/wBArxrxloX9q+H9n/LSP94lemRao+jR6rbt/rNk8f8A45XH6p/qf+2dY4g56h80fFWw8qy/55yR18cfGS/uYtZukt7a4nm/v/6uOOvtj48S+bqEiLXyT8UbDzfEt8n+3Xy+bfwT3sh/inj994x1KLw7Jc3XlyeXs3vv/eR1678JfG639nHarN+72eYnz/6xK8I1PQbm68bf2ar/AOi3kn75K9e8JeF0sPMubfy5ILf76f8ALS3/APtdfOYinD2dz6j2nvn3J+wf4teXxxrFnceZ/wATDSH/APHJI6+qP2eL/wCy6xsb/l43xpXxT/wT7v3uvjhY2ez/AFml3X/oFfYPw5/5Gixs4v8AV28/mV9pw/8A7ofF5/8A7wd/43l83xhGn/PSuV8Ywrd31nC67g3mKR7HbWx8Vrp7TxJa3i/6uSszUtTi1O+0+Vfv4kD/AF+X/wCvXr1FfQ8vD/xEfP3xH8MHRdfuo2BMW87D/eFfPXxkiksrG4jJ3RZbaD2B4xX2R8R9Mh1K4k3Iu3dxz92vlD45aOqjUINyyiPLKa8PGx5WfRYeo7Hyb4q165gsJrOzt7i4jjU542ov1Pf8K8iWS4GptHujs5HGAUJ3qc19IxeHWvvBl5cqn7tSQwQ5K84rwW28ISap8SmhkVliWQbnHOB1P6V8fRqQUp3PuPbN04Hf/BLxAumXcIjkMfmNseSQ7g5/vZ6c9Pxr7w/Yr8ZJHoOr6BJK32yWaK9iB4Eo+ZXx9Pk/P0Br418MfD+3jSSS0Xb5CEPCq5DL2Zf5nvX0f+xdEdW+LFmq7Y1ht5bqRkJbCgBcNnkcstPLa3+3Lk6nHmUVLCSv0P0w+E2orqPgLYzYMKlWHoaxbCQvPM33gknHPvUPwJ1A3KapN92OXEOztxyD+lU7LUm0fxTdW8nTzc4Poa/S5awSPg7nH+INOXWdU1iFvmEmW5HQ182/EDSDbw3kLL+8Un5T9a+mL++U6hq0nC7pdoPtXkPxb0KO70+4kZf3kaFlYd64MVT906KVTWx8Y/GzxG0Vtp9tdRmZtLuC1vIi5dVI5Hvzx+Nea+OviZrgtZGa0t7O3dflDtlwPXjj869Q+MNmBeWbnK7p9jN6Vwf7SHhFtI0BmTc8LRDL+hPavhM0s8Uj7bJqjjh2keF3GrnV7yPz2+1NLIqhnP3Rnsa7/wCDWpJYatF5cbReTN5bFv8AWHB6NXNfDXwbHq04+1Rt9nhUDZ/tH3r1fSfBkazyalCFWRQEuFx99h92UD6fL+OelY4qtFQ5UehCPO9T7L/Yb8WNam80jd5jRzpdxnPZhh/yKj8/rX3pe3633gKO4D5VkypHfGM1+d37CpYeJ9Q1B1XbZ2USswHG58kD6lVb6d8ZFfePgkyXPwpELZZpQ8iL/cyen5V9xkrlLCpSPh84sq7seofsESeZ+0r4TdWLLI13/wCkk9fopivzZ/4J4atn9p/wtat1U3YH4Wk5r9Js816lT4vuPncR8QUUUVJziJ92vmz/AIKha5caB8ANLmt28uQ69AmfrBPX0mn3a+Zv+CqWkzar+zfa/Z13yW+tQz7fpDPVU9zSn8R+bnij4taraRv++j/74r4t/aq1S4+IXiCC8v8A93JG/lp5P/LSvprxldf6yvnD4tWH2/y0/wBZ+/8AMffXLmmInPCzR7GD5PbwKnw+8RvpWnwJK/7z/bevVPBHiKa/uPlm8v8A3K8g0vwl/wAS+Pc/kfPXofwzv7DwvcR/armOT+5vevw2XL7Q/SqJ9N/DnXvstvGm+SST/brJ+LcX/CW+KNK0SB5I/tk6Rzp/z0+euc8N/EOHzI/sfmXX+4lT+Db/AM39pjwdbSv5k95dfP8AP+7r6zhWhCrjoRqHi59iOTDy5D9FPiXDD4c+G+m6bbpHBHGiRoif8s0RK+X/AIyaW8Wsb2/1clfR/wAY7r95HD/z7wR18/fF+b7VcR7a/bMZSvT5Yn53h6hyM1ims/C+ewb/AJeJ3/7Z/JHX58/tGaDNo3iu+jlTy5LN/nr9ELWwe18Nx7v+e7/+06+Lf2vtGW1+J+pbk8yO4RJP/HK+axVPmo+zM6NTkr3PK/BGqP8AY96/+P121/o3leG/t/8Ar5Nn3681sLr7D5cO/wDdx/u69M8L+LbWXQ9lx+8jr80xFNwqH6XhanPQPBbA+Vbz3N1+8nknfz9/+/Wxo2vJLJsifzvn8vf/AOyf7ElXPiBpdmdU1GazeOP7Q/meS9cpoNrNo1xHJLDJ5n+r3/8APRP7kn9+pFUPW9Lv0h0uPd+7g3/Jv/1lm9bHwlsP7e+JFi8v7yPT38/ZXl9/43+zW+9f7nl/f/1le/fCr4V3/wAPo9KfUE8vUtYtUnff/wAs9/7zZXs5Hl/tq3tJfZPFzjGclHkPsz4VePPt+lRws/8ABWL48+0/2pqs3/LDz331wnhLXn0by5l/5d3+dK9JupVutDnmb/l8TzK/Qz4Socf4y0v/AISP4X7F/wBZs+Svz1+PulpF8VIdySR/aIPLr9FPKaX4fx7f9Zbv5lfCP7ZEXlfET7Yv7vy714/++68vOKfPRO/J6nJWPnv4z/B2zW30e8k8xv7QvfL2b/3cdbfhG/hk1jyW8uRf9Xs/5Z11XiO6s/E/wvns5ZP9O09/MtX/AOeb15z4SlmtdYkmWHzI5H+evh5XnDlZ919s9bhtf7U8UaVDv8z7RdWtp8/+sk+eOPY9fZmtWz/Dn4oXWmr/AKu3nk8ivjv9mnT/APhPf2mPAem3X/HrJq8Ej7/+Wmz95/7JX3x+0P4Ne/8AFkGsQp/HJ59fUcN0+SlI+Z4gqe/E9G8B+PU1TT/mf955Hl07R9Uh13R5EuP9fGleIeHPEd1o2obFf93/AH663/hKHsPEmlbH/wCPyfyHr6Q+fJP2jNGSLw/BJF5n/PSvz/8A2kdA+y+JPtDJ/rH8uv0m+KGjf2p4fjRv7nl18I/tN+EnsN6XX7uS3evFzin7h7mW1LTPL/hLoM2jR30K6bbzx3ifJcu/l/Z/9v8A66Vjf2WkOsf9MLf/AMiV3/w5v0utLkT/AJaVxfiO6TS9QkT/AFkm/elfA1OebP0aPLCmdPo1/wAp5Vd/oMsNhZ/abj/WW/7xN9eXeEpXluY5pP3f+xX0T+y18HE+JfiCHVdch/4kdu/lwQv/AMvkn/xutMHl860+WJy4zMIUafNI9m/YP8EJFo+q6xfw/vtYTyIN/wDyzgr1CTRk0vzEZ/4/kosNMfwvqEc0X+o+T5Ep/je6T+19Nf8A5YSP+8r9IweHhhqPsoHwOIxE61b2kjo/C9/c6XqH2a4/5Zx1uap5OqW8aS/8tKZYaWlzqEc3+xT9etfst5BtrsOP2h478b/C72v3X8yONPnr4N+Ldg/hz4mSSf6v7Q9fpV8ZPDj3Wjzuv/LSCvz5/abtktNUgmavl84w/uSge9leI5KkJnkP7TVt/bHhzR5v+Wcl6kdXfCV1DdXHk/8ALOtHxlpaeLfhBvV447rS5/P+euH8B6pDdapJ5r+XJ/Bvr4anT9z2Z9tWqe9zns0t0915ab/Mkk/d7/8AnpX6I+EtLm8B6PY28X7uOzgSP/xyvzk+EB/4Tf4yeGdDZ4/+JhqlrG7/APA6/Tew16GX4gX2m3X+o/gr7PhfD8kJTPk+IMRzzjA7jRtZXxv4HkSX/j4jrznxHKsGqSQ/8s9ldj4c0uTwlqny/vLG4ri/iDF9g8ST7f8Alp+8Svqj5v2h8i/tpaN9q8Hybf8AVxz+XXzD8L9Ue1kkRn8v7PJ5dfZn7Uul/wBqeF75P9Z8nn18K3Wsp4X8SyJL5n7x/M+Svlc/w/PA+gyPGckz3LWf9K8NyTL5cj+XXF2t+lhcb2/5af36qxfFrOn/AGZV/d/7dcrr/iNrq4kmX95HH/BXxvs2fafWD1Cw8UQ2vzf7HmV6b+zn8Krn4v8Aii11LVP3eh28/mIn/LS8/wDtdeV/sofD7/hfviidrrzINH0d089P+Wlw/wDcr7k+Hvgj+xreP7Kkccdv/An9yvqsnyf/AJf1j5XNM4n/AAIG34s0x7Dy3V6t+EvED6XJIrVB4oun/s+NG/v1V0aXzfL/AO/dfXHy5N4o1R4k1Kb/AJ6fvEr5z8eWry2dx/y0/v19RS2CX1tsZf3exI3/AO+K8B+I3hiS1vL6GNJJPLd99ViKfuGOHqe+fJ/x9sPO0fTZl/5Z3Xl/9/K8J+MXh1NY0+xfZ5cf2pI0/wC+6+oPi1oyXWh3ULf9dErx3VNGh8ZfD+dG/d3Vm/n/APxFfC5jT5K3tEfZZXU56XIVfAd19l8Qb7d/Ikj+5/8Asf8APOuq1S1TS4JEt4/Itf8AWbN/7uP/AHK8n8Oap9g8QedL5n/PRHT/AJZ13l1r39qXkFs3+okdI53/AOB1wUMP+9TOypiP3R+u3xLsP7U8B+Ebm3/1EenWv/oiOpPhz4y/sYx203+r3/JXQayYYvh5a7P3kFnaps/3ESvPJbD+2fD/APaVn/yzTzNlfqlP4T4GW50/xa0DytQkvLf/AFdxA8n/AI5XkfxB1R7DwvGi/wCskfy99ezS3/8Abvw/3f8ALSzgeN6898ReHIde0OeFv+AVGIOWR8u/EaL+Ovmn4q2D/wDCcT7f+WiJJX1Z8X/Dk2jJIkv+r/gevmX4yfutQgm/7Z18tmlPnonsZXU5Kx4/rOg/2N8RN7J/q7V5P+B123w4v0l8za/lyR/+gf8AxuqPxQ+x3+n2N/F/r9nlvWT4Dv5tLt/uSf7D18p9g+tl8Z9if8EyYk/4ac2fu/8AkEXWxP8Ann/q6+sNBupvC/iSd2/1kc718p/8EgtmqfHHxHeXD/6Vb6X5af8AA3/+wr7E8WxW0viySz/1ckieZX3mR0+TDHxmcVOfEnVa9LD4y8FyPF/rLf8AeVwtpI1rKu7+Ef0rX+H11No2qX2m3H/LT7lQ29us+v6lbt96AR4/HdmvVZ5tJ2lc4rx3I15pN4YQTIOevTmvmX4hWzJHOrNuZw24nrX1p4w8N/2Zh4zuWQjP0zzXy98a7f7H4k1DA2ndhfQCvHzCDsezh6x458BtLXUdE1i1Zc4uypU9SM1494y8OW/gz4katbw7lhtW+RmHUsefy6V6j8OfGlr4P8f31tNNFGszGRcsBk1yv7SsVnqHieO7sWw12v7wqdyk/hX55UhONdo+4wlRSopk3gHVA9szbijj5Qy9WB4/XNfTP/BPbw5/at1451JI1MltZQRKVGNm+Uk/n5Yr5E8PXF5pel7Wj2BUwJa/Qr/glF4PjT9n3xHrU+GuNa1ERE9dqRJx+ZLV7uQ4VSxXOuh52a13HD8vc90+EvigaAPIZtqTFWJP+fWum+JWkrJPb6hB1kIDEdzXBppaa1DcNZNtktiQVHGdvP8ASu58Kao+veCWjk+a4tY9/Pav0A+O5keZahctbG4En/LSY5964f4rFh4YvJoW/wCWfT68GvUr7RI9Whyy/M2Tn0NeU/F2yksdAvoecqnI/EVxYrYqL1PjX45qx063bduEN2hOO+SB/WtT4laFb6z4FmmkjWaKKDeFI+8QuR+uKh+Kqq1jcBhlVYSD8Of51F4d8a2et+ApLeSSOZWiKuquNwNfn+bUmqikj7fJ6q9meFaAq2lvDtb5inKDjIJyf8K9E0a6ZLBXHy7lwcj+H0/CvOba1kHieTyEaSCIkAdGH4Gu00m4m8Qa1p+kwwzJLqE8Vtg8bS7hev41x0abqSUTulinBXR9y/sqfDhvCf7Ofh/UDGyz+IbiW/lYjloyQkY+gVScf7VfTvwu8WRm1js5GVduUUeoIqjqHhKy8J/C3SNPiVVtdJtUhU46BVA/Uk1yawTaalrqFuzGNZBuIPrwP51+nYSl7KnGK7HweKq+1nJvufRv7E+hHR/23/Cm3/Vt9sb87K4r9JD94V+en7FFwurftQ+CbxRn5bsMw9fsU9foX1IrSe55db4haKKKkxCvEf27l3/CCx3f9BdP/RE9e3Vzfjz4d6P8RtJSw1mz+22scwnWPzXj+f1+Qg/xGqiB+Ifxu0L/AIRzxpqtv/yzjfzI/wDcr5l+KEv2XQ55l/d+W/8A38r+gLxb/wAE2vgv48vPtWqeC/tVxs2bv7Xvo+P+ATiuX1L/AII4fs46vA0Vx8O/MjbqDr+qf/JNcuIp88LHbh8V7N3P5/8Awv4seXVE/wB/5/8ApnXfRfvLmPbcyeX/ALcCSV+1Nt/wQ6/Zfsp90Pw0aN/UeJNX/wDkqty2/wCCQn7Pdl/qfh+0e3013UuP/Jmvz6vwdi51OeE4n1NPiigvsH45eCJby61iP7RNJJHGnyb6x/i/4y1D4f63o/iTRpPLvvD+qQX8e9P9Ym/5/wDyHX7Yw/8ABKf4C2k/mDwN+9/vf21qH/x+qmvf8Ehf2efEsEiah4BFxHJ1zrmo8/8AkevRyvh3F4SvCtzR905MZn9CtCULM+X9L+PGlfGTQ59VsLmO4j2J9z/cry/xHr39qeII031+gngH/gll8C/hlaSRaH4KmsY5E2Mn9u6jJkf8DnNbMX/BOr4Ox3XnL4PPmf3v7Xvv/j1fp39pR5LWPlD8/tZiSLT4E/1cckNfFv7Zulf8VJ53/TBK/eK4/YZ+Fd3Htm8KtIMbedTu/wD49XI+M/8AglL8A/Hax/2t4B+2bE2D/idajHx/wCcV5dZ84fbufzI+KPEb6XrGxZpI45P7ieZWzoPiK8+x70m/d/xvN+78yv6HL7/ggh+ybqV59om+E4kn27N//CS6xnH/AIFUr/8ABBf9lGSDyf8AhVn7v+7/AMJJrH/yVXyuYZLUrT54H0WDzqNGHJJH89uoW018Y5mubfy7hPnTyK2NL0tbDT/3tz/B5m/yK/f5P+CD/wCytGvy/C//AMuTV/8A5Kqf/hxn+y7s2f8ACslx/wBjDqv/AMk15/8Aqzif5o/idn+sFPsz+cm6tbmbxRYzfu/sv2pN+/8A36/TG/8ABsPjzwnpd/cf8f3kJJv/AOelfoEP+CF/7LYnWT/hWHzI+8f8VFqvX/wKr1Ky/wCCf3wjsNPjt4fCp8mMbUB1W9fH/kavo8nwc8LCUJni5jjI4n4T8gdZ0aa1g8mVNleqfJJ4Hj3f8s7VP/QK/Si6/wCCcvwavv8AXeDVk+uqXv8A8ep3/DvX4P8Al7P+ER+Xbsx/al70/wC/1ez7Q8epT5z8rorpNK8Pyf8ALSONHr4R/brtfKs9Ydv3f/LRK/o3m/4Jx/BeW28tvBn7v0/tW+/+PVxfxE/4Iu/s0/FnTfs/iD4arqEO3Zt/t7VI+P8AgFyK58R75ph/3c7n8sWl6y9ro8k0ryf7iV0fg2WztbeSbZJJ/fd3/wBZX9J3/EPd+yH5Hl/8Kij2f3f+Em1jH/pVVqP/AIIF/slxWnkL8J0EP93/AISPWP8A5Kr5mpkdWXU+m/tykujP52/2br+20b9pDwXfwzfvJNUg/jr9TfFNr/bNvJt/efP89fa+lf8ABAv9k3Rr21u7X4UrDcWcnnwP/wAJHq/7t/X/AI+q9etf+CfPwhsU2xeEfLGNvGq3v/x6vfyuhLDw5Jni4/ERxEueB+POveEptP1SN1T+OpL+XzfGmlf88/tSbP8Avuv2Buf+Cd3wdu/9b4PWT/uJ3v8A8eqo/wDwTX+Cc00MjeCY99u/mRn+1L75H9f9dXf7Q4OVn51+I4kl8PyP/sV8I/t4fubj5f8AlpBX9CN1/wAE/wD4R30HlS+FPMj9Dql7/wDHq4Pxv/wRn/Zt+Jkedc+HDX318Q6rH/6BdCssR+8hY66dXkP5htB+IVzFHIlv/wADdKmi+061qHzJJv8A4Hr+kAf8G8X7H8S/u/hHIv08Wa3/APJlWI/+Dfr9kuL7vwtmH/c163/8mV8lVyGpKd4s+ip8QR5LSTP5wrXxRf2uqWtg0Mf+vRHf/Yr9MfBtrDYeH9KezhjgtY4E2Qp/yzr9ELb/AIIEfsnWs6yR/CvDo+8H/hJtY6/+BVeqWP8AwTW+Cul2MdrD4L2Qx/dQavfcf+Rq9zK8H9Xj755uYZksR8B+cOl6yl15cMtWvEfhL7VZyIv/ACzk8xK/R7/h3b8HI/L/AOKQP7vp/wATa+4/8jVoD9h34Wp/zK//AJUbv/49Xre0PL9ofn34DuvtVhBu/wBZHVrxlYeX5bV98WP7Dnwt0xt1v4X8v6ajd/8Ax6rV1+xf8Nb+LZL4b3p/2ELr/wCOUe0D2h+e/i2WGXwvP5v9yvzL/bmlSwvLqFf9Xbu9f0X6j+wd8KdXsvJuPC7SQ/3P7UvV/wDa1ebeOv8Agih+zL8TGkfXPhn9uM33/wDiodVjz/3xdCuPEU/bG1HEch/MTo3je8/s+RJbyOCCT926P+88ytjwlo1na3G9rmOSP+4iV/Rof+Der9kN49o+E7Knp/wlWt//ACZVz/hwT+yf5Pl/8Kr+T0/4SbWP/kqvl6nD9WR9FT4gpqFrM/n/APgZL/Y3x08M3i3Plxx6pBIn/fdfpT4ysJotYj1KL/l4TzK+2tJ/4IKfso6FeQ3Nr8LfJnt38yNx4n1j5G/8Cq9hH7APwk+yrF/wih8uPoG1S9/+PV72V4OeGhyTPHxuYKtPnifn58KvG/8Ab0n2G6/1/wDt1h/HP/iV+JIF/wCnJP8A0OSv0Zsf+Cevwg03UPtlv4TaO4/56DV77/49Unif9gb4T+MpUfUvCjTSKnljGq3seE9PkmFescPtD8U/i1rP9s3EkK/vPMTy6/P/AOMml/YPiDfWzTeZ9jfy6/qJl/4JR/AGZ9zeAef+w1qP/wAfrzzW/wDggJ+yX4j1aa/u/hW811cPvkkPibWVLn/wKrzcww8q1PkgdeBxkaM+eR/OBbeLfK0+BLeaOOP/AFfyJWVrPjGSW38mOaSeT+5v8uv6Tf8AiHt/ZF/6JU//AIVetf8AyZUcn/Bu/wDsgyvub4TMz+v/AAlet/8AyZXztPIKq3Z7v+sFPsz8Sf8AgmTvl1DxBC/7vy3SRK+6/BsvleY7f6vzPLr9APht/wAEbP2b/hEs3/CN/Dk6b9o/1mNf1STf/wB93JrtrL/gnp8HdLL+R4Q8rzDufGq3vP8A5Gr6jD0+SjGnM+dxGIjOftD80/Ftgl1p8+2uO0u6ksJNjf36/WOT/gn38JJR83hPP/cVvf8A49VOT/gm78GZT83g3/yr33/x6ugx9ofnXo8P2qP/AIAlc3qlqlr4lkdkj8yRH3/J/rK/Ui1/YJ+E9jHsh8JGNf8AsJ3f/wAeqG8/4J//AAjv5N03g9ZJPfUrv/49XV9YhyWMT+fP9oywh8OapqSf8s43fZXxuPG9/wD2jfeTN5cEbvG+/wD5aJX9T3in/gjt+zj4/kkfV/hut1JJ9/8A4nupR5/74uRXGJ/wb3fsgx+Zs+EMa+Z9/wD4qTWOf/JqvAzDL/bfAepg8w9jI/mm8GxW0tv5zXO+ST+BErYu7WH7RJcxTeXHb/cT/npX9JFl/wAEAP2SdLt/Kh+EsMcfoPEWr/8AyVSf8OA/2R8N/wAWljPmdf8AipNY/wDkqvMo5JVhO/MehUzum1ax8c/CrxH/AMJT+zvoe1/M+2aXBsf/AIBXMfC/xl/wjl5JYXH+o3/cev1G8Gf8E6vg38PfDFppOk+Dfsum2MflwQHVb1xGv/A5qLj/AIJtfBe5m8x/BcbN6nVL7/49X2EcRofN9T4DurBLDw/fTWr/ALm4gfen/AK47yvN0f5a/UC3/YT+FdtpT2cfhUi1kXY8f9p3eCP+/wBVSP8A4J7fCGJdq+E//Krff/Hqn2hM9T8a/jTYfavB90jf8u7+ZXwd+0rqn2TT5IV/1nn/ACV/TbrX/BLr4F+I7SSG88D+dHN99TrWorn8p6858Uf8ECv2TfGR3aj8KftDb9/PifWF5/C7rzcRh5TOnD1fZn8wMWqf6PAks1x5dw/8Fdday2dhp+zZ/B/HX9H8f/Bu3+x4jx/8Wjb9z9z/AIqrW+P/ACcq9d/8G/v7JF/Dsl+FAdffxNrH/wAl18/UyCrP7R739tU+zPxE/wCCWGsw6D+0RPDE/wDx8ac/8dfZHxk1R9L8aWt/FX6EfD7/AIId/sv/AAl8TLrWgfDAafqUabI5R4j1WT/xx7op+leia3/wTg+C+u+X9s8G/aPL+7nV77j/AMjV9Ll9P2NH2czwMZU9tV9pE/Nnw5rNr4tt4LlX8u6jrSsbFm1y4uG5a4UBv+Ak/wCNfofo3/BNf4K+HJvMs/Bvkyev9rXzfznrUg/YN+FLybh4WJZRtJOqXhI9v9cfb2rp9oc8dz80viBa+XZoSOmM18n/ALTlmtnr11IrbN2HB9ulfu/qH7A3wo1OEpN4V3r6f2neD+UorjPGH/BIn9nzx3L5mqfD83TsuCw13Uoz/wCO3A/nWeIj7RHZTrxify2/Fi/a/wDiBM0G393hSQPw60Wl8k5W1lSZ2iwTlvlNf0qXX/BvT+yJe3j3EnwlZpZDlifFOs8n/wAC6kt/+DfL9kezuvPj+E7LN3YeKNZ/+S6+YxWS1qk7xsj3KOd0qceWzP5055Ld9EjtPljLH+9ya+9/+CTmrRxfAXWLBZHfyL+QgHtkZHFfp7ef8EC/2UbzaZPhbI7L0/4qjWBj/wAm67b4N/8ABJH9n/4B293D4V+H/wDZUd62+Yf27qNxvP8A20nau3KctqYWV5NGOPzalXikkz81dE8RSeDfHF4siM1vO3zZHAzXq2hWcEg+1WLKsdwu1oweoNfeGp/8E2Pgvq9000/gtXkbqf7VvVP6TVqaL+wV8K/DsHl2fhhoY/7p1W9YfrNX0HtDyXWjbRH5uizVNQuE/iicqD6V5f8AG/TGa2uhhj5iHPHXvX65f8ME/CkTtJ/wivzSHcx/tK75P/f2svXf+Cbfwb8Rk/avBqyEgjP9r3ynB6/dmH86zraxsjONXW5/N78crlNPsrjccLtYcd6+e/C2rtY3bSMXWNWyBHnBzxzX9Qvij/ghn+y54xiZdR+F/wBoVuufEmrIT/3zdCucj/4N5/2QVj2x/CTarcYHinWue/8Az914OOymVfWLPZwecKirNH832izW/wDwk8lwzM7MMkdAa6PR5PL8e6XfxyeWLe8hdQH6kOtf0VWv/Bv5+yXaIqp8KWUYx/yM+sfz+106H/g3/wD2TIH3L8KpNysGGfE+sHkHIP8Ax9+tcGFyOvTnztnVUzyjKLikz5D1G/8A+Ej+FqtGzOZrRXBPfKjP61yfwl8TwyWcml321k34Un2FfqRpn/BPL4R6NokOn2/hNks7dBGiHVr5iq+m4zFv1rPT/gmT8E0n81fBJSTOc/2xf8n/AL/kV9fF2seDLEJu6R81fsF6cdG/aT8LwrJ5kLtdsp/7dJq/RGvO/h9+yp4D+F3iCz1TQ9B+w31iGEEv224l2bkZG4eQg5ViOR3r0SiW5zSld3CiiipJCiiigAooooAK8Z/aH/a5s/gX448K+E9M8H+NPiR428YRXd9ZeHfC8dl9sjsrTyxc3s817dWtrDBHLPaw/vJw7yXCIiP85T2avl39o3wl42+HX7angb4weFfAusfE3SbTwjrHg3WdF0TUNNtdVsXubqwvba9j/tC5tYJIAbKaORPPVwZYHRHAfYAewfs2fH7w/wDtQ/BbQ/HnhqS7Oj68km2K7tzb3dnPDM8E9rPH/wAs5oJ45IXTPDxuOa9Crwb/AIJ3/AjxB+zn+y7p+i+Ko7O08TavrGteJ9VtLWf7RBYXOq6rd6lJapJ/y0EH2ry9/wDH5ea95oAKKKKACiiigA6UUUUAeT/Az9rX4d/tM+KPE+l+A/E9p4luPCMlsupS2UMxsz54k8t4bkoILpCYZU3wSSIjxuhIdMV6xXxf+yZ8abTxZ/wUm+N11D4X+Kmm6b4007w/b6PqWs/DnX9I0+8ksYL77UDdXVlHDHs8yPZ5jp5m/wCTfX2hQAUUUUAFFFFABRRRQB5N8ff2sfBv7MMmnr4wu9f+2axFdXdtY6F4d1TxDePa23l/abr7Np0E80cERmgDzOgjR7iNC+ZEz2/w/wDH+ifFbwRo/ibw3qdnrmg+ILGHUtNv7R/MgvLeZBJHKj/3HXB/GvAv+CiP7UXij4DeFtD8M+DPDfji4174gS3Nj/wlWkeCNV8UWHgiFFQzX1xBp9tcSST4cfZ4NgSafh3jjSRx6P8AsgeFPCnw8/ZZ8B+HfAtn4ksfBvhrRrfSNItte0u703VEt7VBCnnwXkMM6SfJ/wAtI0z9/oaAPWKKKKACiiigAooooAK8N0D9vb4X+J/jja/Dy21zUofEl9qF/ounTTeHdUg0fU72yMn2q1tdTeBLG5ngMEySQwzvIjwToRmOTHuVfAnxH1zxD8X/ANuv4a6lotj8c/EF34R8dTrd+FPGfgZ9N8IeDrIWl3Yz65Y6nDa2sF1P5Z/ceZfahlNRk2W6P+8tQD77ooooAKKKKACiiigAooooA8L8a/t0eD/A/wC2P4G+Cclnr2oeLvHEM863Njaq9hooS1uLqNLyZ3R0eeOyuvLREf8A1Hz+WCm/3SvzxtP+Cd3x6+HH7WHwo8TWHjj4deLdLtviNrfjPxPrUvgSa01SP7bpd3aj7U/9tYn/AHDwWMH2eBPISOB3R0jkR/0OoAKKKKACiiigAooooAK8Buf22Li//aO8S/Dnw78J/id4wm8G3mn2eu65ps2h2+m6Yb2BJ0ci61OC6eNI33v5cDng7A/Fe/V8HfttfstX37SHxY1Cz8P/ALN9noXxI/tvSJtI+OkNxokMmn21q9rM919pSb+2BMkcc9qLb7O8ch2I8nkSO6AH3jRRRQAUUUUAFFFFABXN/Ev4m6D8HPAOseKfFGr2Og+G9BtnvNR1G+m8m3soE+/I79ABXSV8v/8ABWn4Ya18Wv2INesdB07xJrU2lazoGv3umeH5p49X1Cx0/WbK+vYrXyXSQ3H2SCfy0jPmPJ5aJ85SgD1r4J/tB6F8edNvbnQbHxxYxae6xzJ4j8F6z4bkbcP4E1G2geYcffj3jNejV8jf8E79N1a3+Lfxam0eX4xN8Gbs6OPC0fxJk1uTVo9TSGf+1HgXW/8AiZLZlP7OCef8nnpdGPg5b65oAKKKKACiiigAooooA8U+OX7b/gn9mzUp7fxVa/ElobGx/tG6v9F+HHiLXdOt4Rv3s95ZWU8EewI5dHcOg644z6H8OPH2m/FvwDoPirQ5rm40PxFp8OqadLPaT2sssE6LJG7wzKkkeUdTskRHXkEA9PCf+ChvhHX/AI+WXgT4K6Tpeuy+H/ilrHleNNXispPsGn+HLL9/fW006LsSS+/c2KRvjfHdTumfJevpSzs4dNto4YY44YYU2oirsRFoAtUUUUAFFFFABRRRQBm+M/Gek/DnwhqniDX9SstG0PQ7SW/1HULyZYbaxt4kLyzSyMQqRoiszMxAVVJJABNec/s9/tufDv8Aaf8AEV9o/hW/8RRaxYadBq7af4g8Kat4bvJ7GdnSK7gh1G2t3ngLRspliDorbQSCygn7d1lrWp/sW/Fa18OeENP+IGuXXhXUYLPw3f5+y65I9u6i2lAZCyPkqUV0ZgdodCQ4+V/+CZvhjVNG/a+v7jTV+LvjDwn/AMK5sbG88U/E7wfqvhzU9BvILkrbaNpov44Wns3iNzNLiKaYSQQvcX120kQiAP0AooooAKKKKACiiigCO6bbbs3y/KM5Y4Ax3P0r5s8I/wDBS7wz4k+Iul6evg34hWvgjW/FVz4H0rx9c21l/YGp6xA88LW6qtyb1Fa4tri3WeW0SBpYlUSHzYy30rMrPEyrwW4z6V+fnir9gub4rftbeE/+EZ+EXxR+FfhTw58Q/wDhP/EOs6t8Qg3hrWJbS4a5iGm6DZ6tc2/n3l75U7SXFnbbIhcMSJ3VKAP0DWQMf/rU6o4oypHG3GTjOf8AP/16koAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAGzKzxMF4bHGc9fwqva2phmDbex54z2znHcnHPovbvaooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooopIAooopgFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//2Q==)

Although we trained the model over all the numbers, we used only the 0-5 numbers. In that way, we reduced the noise in our model. For example, the numbers 7 and 8 are similar, 6 and 9 are also not far from each other. These are examples of cases where the model can be confused and more likely fail.

## Improvements

In the previous section we mentioned the challenges of the project development. As part of our research, we found solutions to some of these problems.

### Hand Detector

At the end, the camera gets a picture of the room. For using the model, which we trained on pictures of hands, we had two ways to deal with the input picture-

* + Separate the picture into tiles and activate the model on each tile of the picture.
  + Activate a hand detector model on the picture and cut the hand part of the picture. This part will be the input to our model.

We chose to use a hand detector which was developed by OpenCV library. This hand detector has good performance and accuracy.

### Data Augmentation

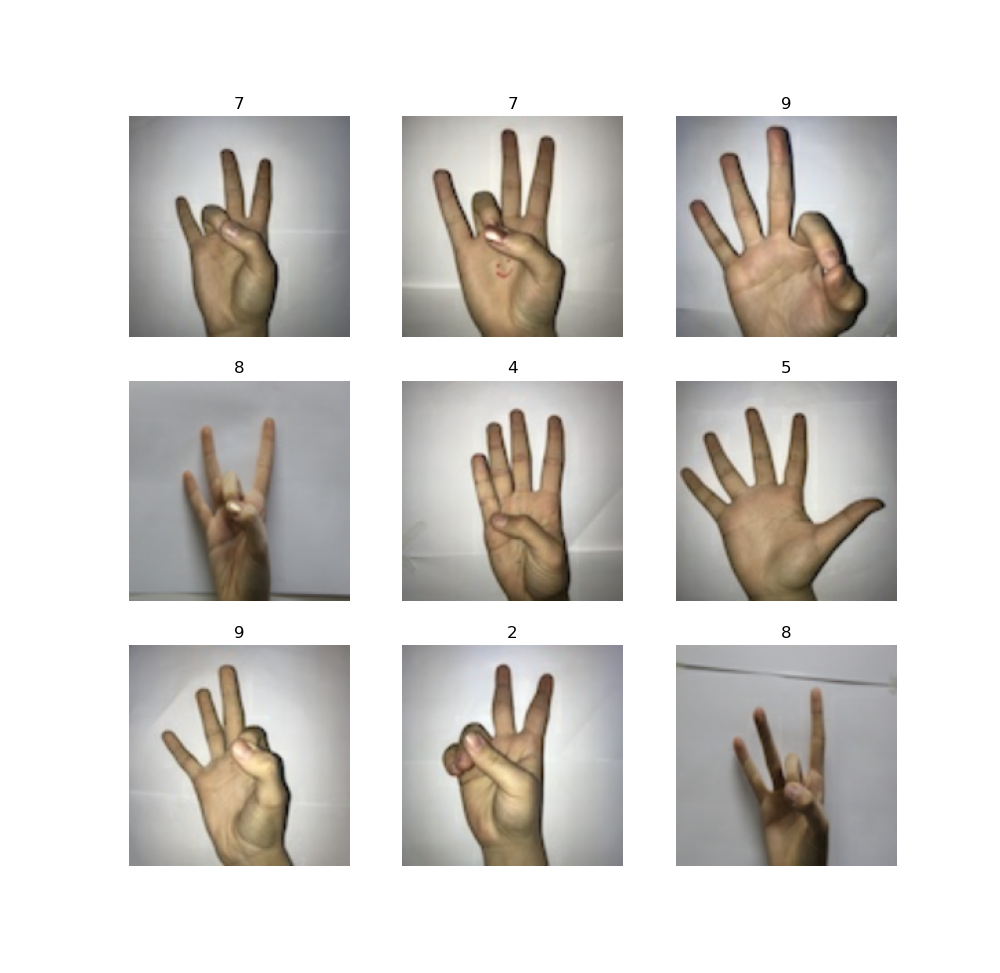
For training the model to be less sensitive to variations in the angle and the direction of the hand, we generated new pictures out of the exist dataset. This operation can be done relatively simply by using common libraries like Keras.

### Transfer Learning

Instead of training the whole model over and over with any little change, we used the Transfer Learning method. Transfer Learning is the method to reuse a pre-trained model on a new problem, with just a little change and a training of a few layers.

We took a pre-trained MobileNet-V2 from Keras, changed the last few layers to fit to a 10-label classification task, and trained just these few layers. The rest of the layers have been already trained over the ImageNet task, and that’s good enough for us.

# Training phase

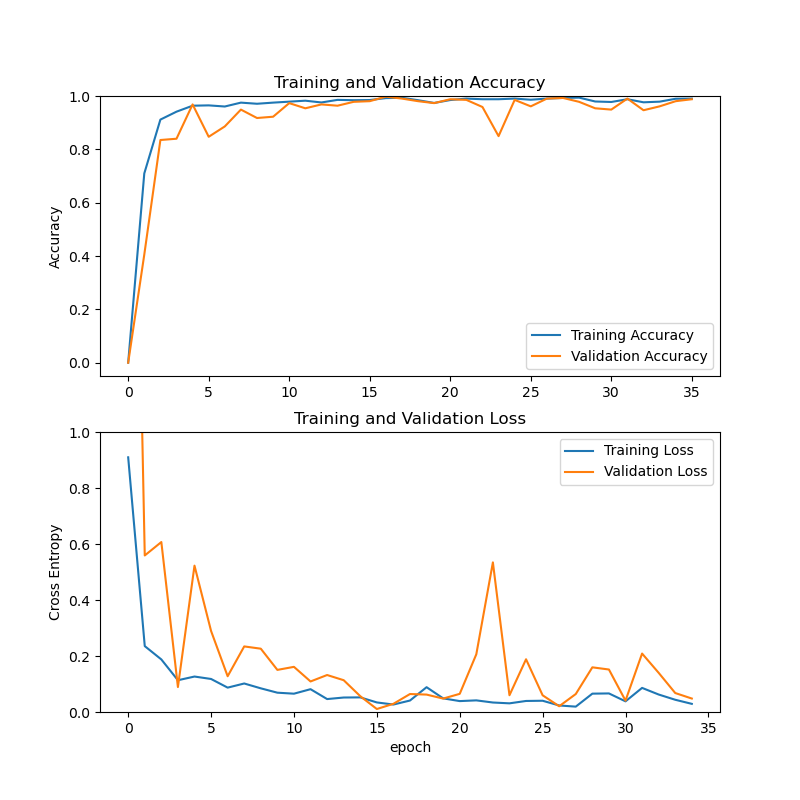
1. A screen shot of a computer code

   Description automatically generatedData loader  
   Keras library provides us easy API for loading a dataset which is organized in directories.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   Now, we have a dataset which is divided into a training set and a validation set. Each sample is labeled with its actual label.
2. A screen shot of a computer code

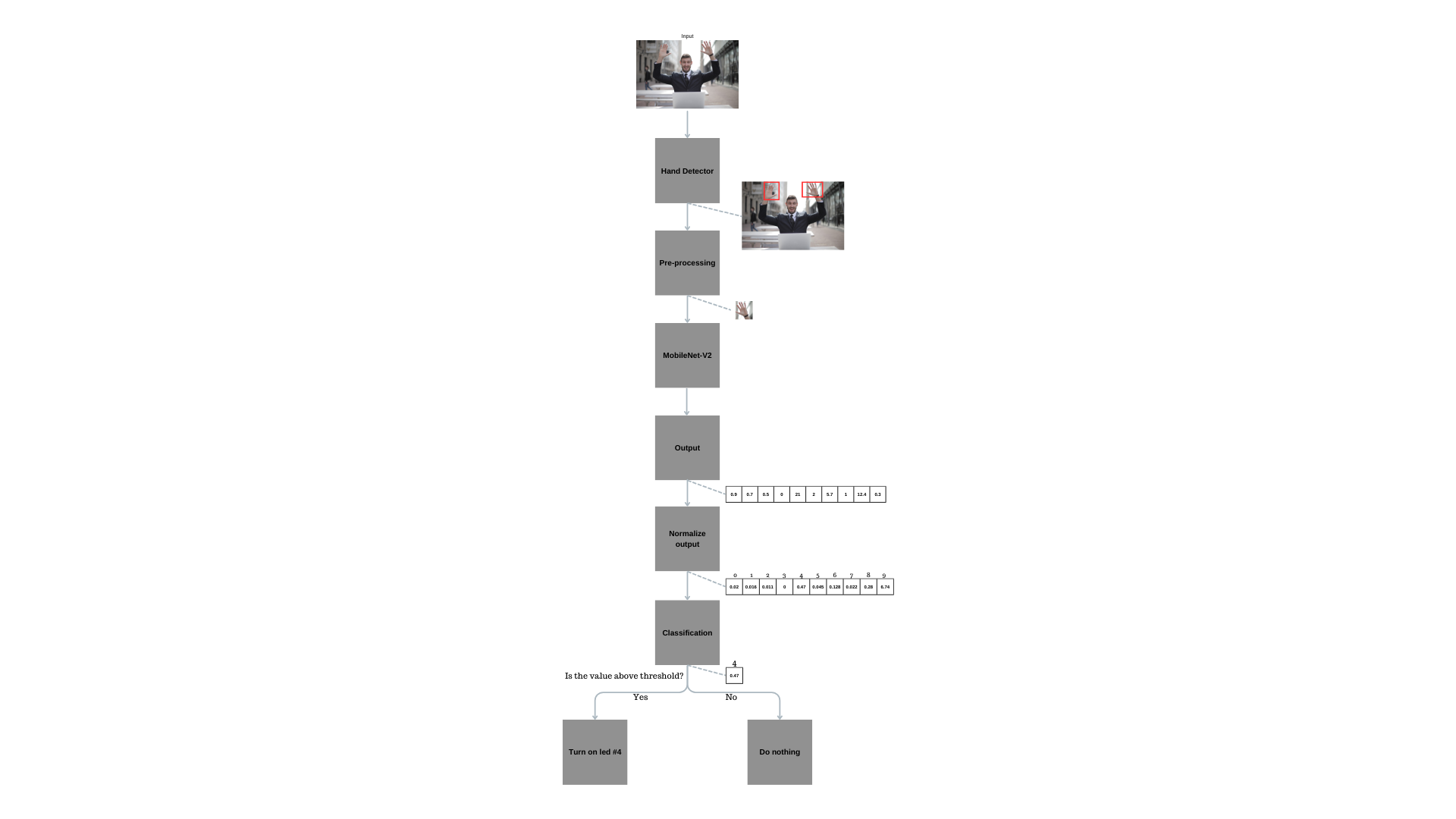
   Description automatically generatedData augmentation  
   We create a data augmenter, which is responsible to generate new images with variations in the direction and the angle of the hand
3. A black and white text

   Description automatically generatedModel create  
   First, we load the MobileNet-V2 with ImageNet pre-trained weights  
     
     
   As can be seen in the picture, we chose to load the model without it’s final part, which we will need to change. This model was trained over ImageNet, a classification task with 1000 different labels, while what we need is a model with only 10 classification options.  
     
   Just after that we mark the “base model” as not trainable  
     
     
     
     
     
   Now, we can put the pieces together and create the whole pipeline we A screenshot of a computer program

   Description automatically generatedneed  
     
   The input will pass through the data augmenter, after that it will pass through the preprocess algorithm, then through the basic pre-trained model and finally it will pass through the last few layers we added(the only trainable layers in the model).
4. A white background with black text

   Description automatically generatedTraining  
   In that step we train the last few layers we added  
     
   We chose to run 35 epochs, with Adam optimizer and learning rate=0.001. Those parameters were chosen based on the need to shorten the training duration time and on the other hand, the need to prevent overfitting.   
     
   As can be seen from the next graph, we got high accuracy and low loss rate on the validation set after those 35 epochs.  
   

# Inference block diagram



# Files hierarchy

The project files are located and maintained in a GitHub repository:  
<https://github.com/DanielLevi6/44167-ProjectA-SmartHome>

* Classification
  + Data
    - SignLanguageNumbers
  + Model
    - MobileNet-V2
  + trainingScript
    - mobilenet\_training.py
* HandDetector
  + hand\_detector.py
* SmartHome.py

For running the project, we should just activate the SmartHome script on the main directory.

# Summary

# References

#### Online courses and tutorials:

* [Convolutional Neural Networks (Coursera course by Andrew Ng)](https://www.coursera.org/learn/convolutional-neural-networks)
* [TensorFlow official tutorials](https://www.tensorflow.org/tutorials)

#### Academic papers:

* [Hand Pose Classification Based on Neural Networks (Rashmi Bakshi)](https://arxiv.org/ftp/arxiv/papers/2108/2108.04529.pdf)

#### Dataset:

* [Sign Language Digits Dataset (Turkey Ankara Ayrancı Anadolu High School)](https://github.com/ardamavi/Sign-Language-Digits-Dataset)

1. [Convolutional Neural Networks (Coursera course by Andrew Ng)](https://www.coursera.org/learn/convolutional-neural-networks) [↑](#footnote-ref-1)