

# Robô seguidor de linha utilizando visão computacional e redes neurais embarcadas em Raspberry Pi

Igor Gallon

# Agenda

- Objetivos
- Caso de Estudo
- Metodologia
- Implementação
- Resultados e Conclusão
- Propostas de melhoria



# Objetivos

Desenvolver um robô capaz de seguir trajetórias predeterminadas de maneira automática

- Explorar o uso de processamento de imagens
- Explorar o uso de redes neurais
- Desenvolver um sistema embarcado em Raspberry Pi

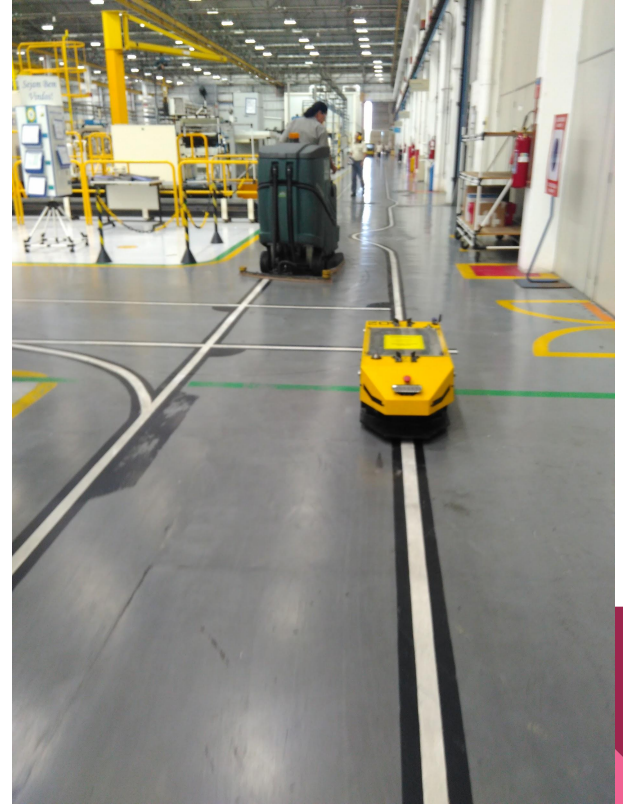


# Caso de Estudo - AGVs Embraer

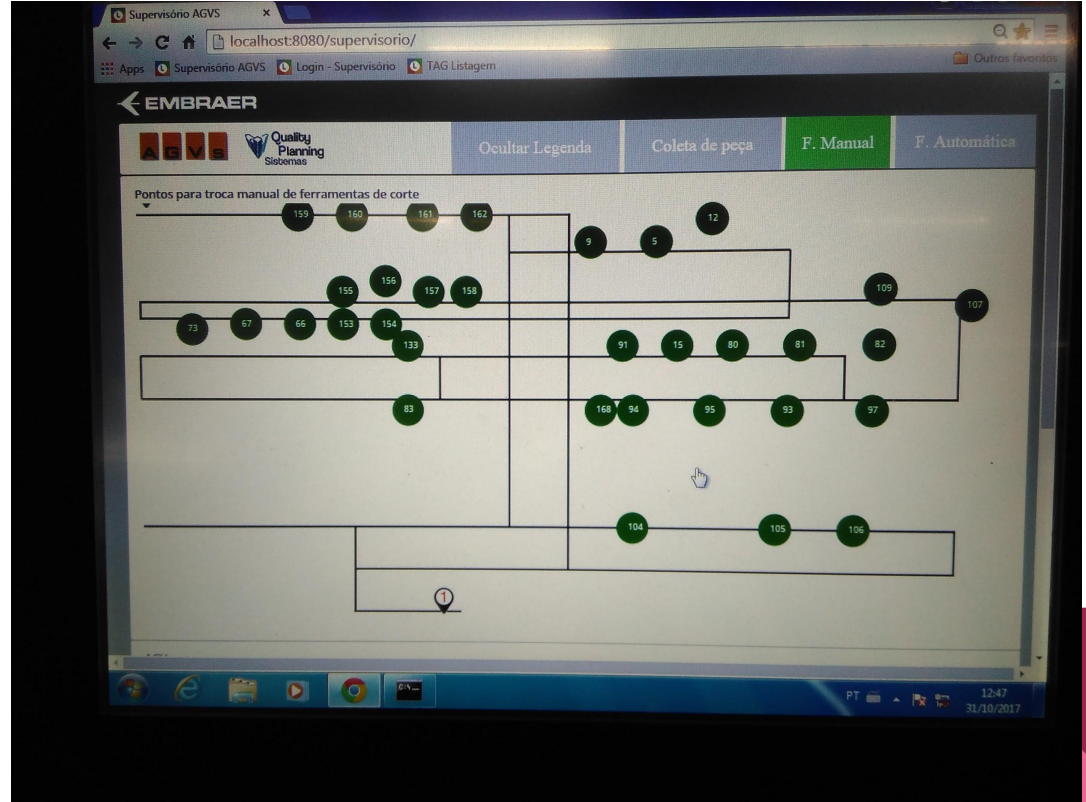
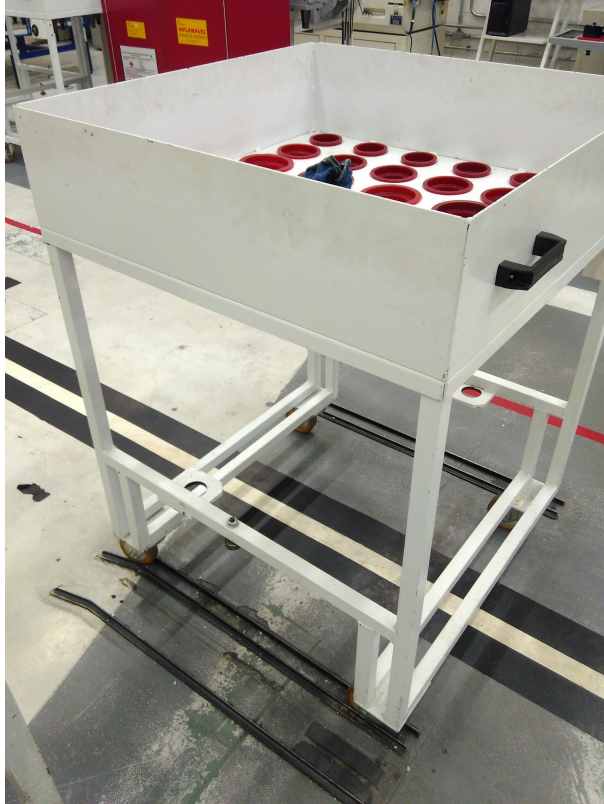
- Centro de Usinagem de peças metálicas - Embraer (Unidade Faria Lima), São José dos Campos
- AGVs utilizado para transportar carrinhos de ferramentas
- Faixa preta e branca pintada no chão de fábrica
- Sistema de comando central dos AGVs na Ferramentaria



# Caso de Estudo - AGVs Embraer



# Caso de Estudo AGVs Embraer

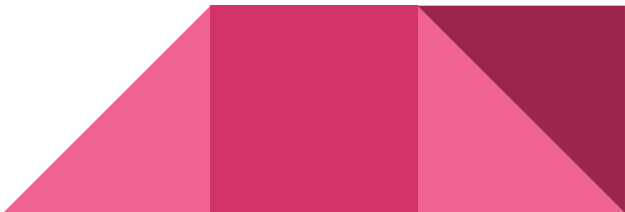


# Caso de Estudo - AGVs Embraer

Implementação:

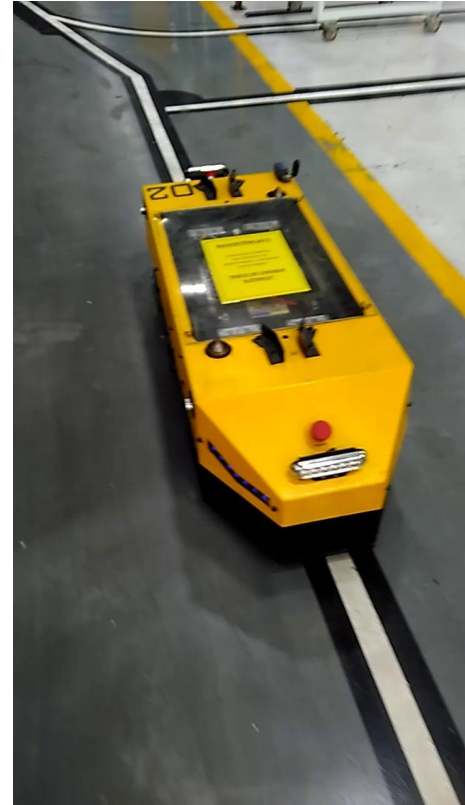
- Barra de sensores ópticos

Problemas:

- Desgaste do trajeto com o passar do tempo
  - Desnível do piso
  - Luminosidade, poeira
- 



# Caso de Estudo AGVs Embraer





# Metodologia

## Estudo das tecnologias usadas

- Bibliotecas para tratamento de imagens, Redes Neurais
- Placas Raspberry Pi (com Raspbian OS) e Arduino
- Comunicação por Python Sockets
- Controle dos movimentos dos motores



# Metodologia

## Implementação iterativa e incremental

- Preparação do SO na Raspberry
- Instalação das bibliotecas
- Implementação do sistema
- Montagem do robô

Testes quantitativos da Rede Neural gerada



# Raspberry Pi

- Categorizada como *single-board computer*
- Baixo custo (entre U\$5 e U\$35)
- Alto poder de processamento e funcionalidades (Raspberry Pi 3 Model B)
- Desenvolvimento em Python
- Grande número de bibliotecas, implementações desenvolvidas pela comunidade open source
- Sistema Operacional baseado em Linux



# Raspberry Pi 3 Model B

- Processador quad-core ARM Cortex A53 1.2GHz
- GPU Broadcom VideoCore IV
- 1GB memória RAM DDR2
- LAN Ethernet, WiFi, bluetooth 4.1 embutidos
- Saída HDMI
- 4 portas USB
- Barramento CSI (Picamera)
- Consumo médio 125mA (1.2W) - 5V



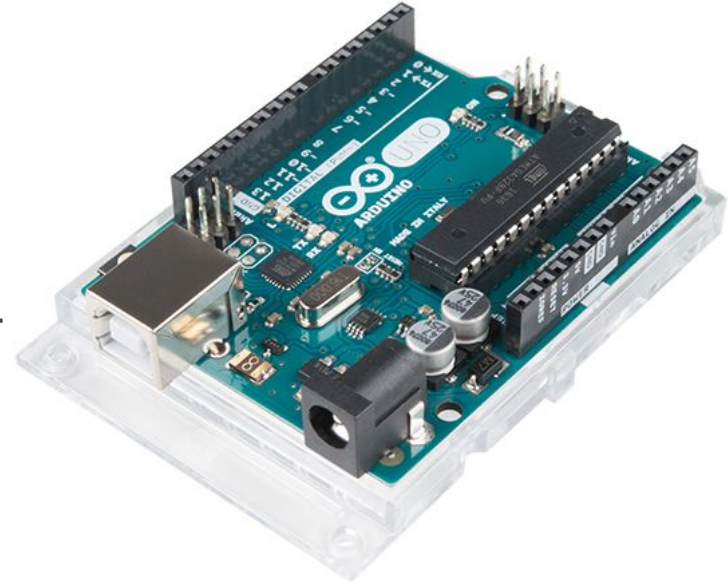
# Picamera

- Módulo de câmera para Raspberry Pi (barramento CSI)
- Resolução de 5MP
- Grava vídeos 1080p a 30FPS



# Arduino

- Plataforma de prototipagem eletrônica open source
- Placa única contendo microcontrolador Atmel AVR (Atmega 328 16Mhz - Arduino UNO)
- Linguagem de programação própria, semelhante à C/C++



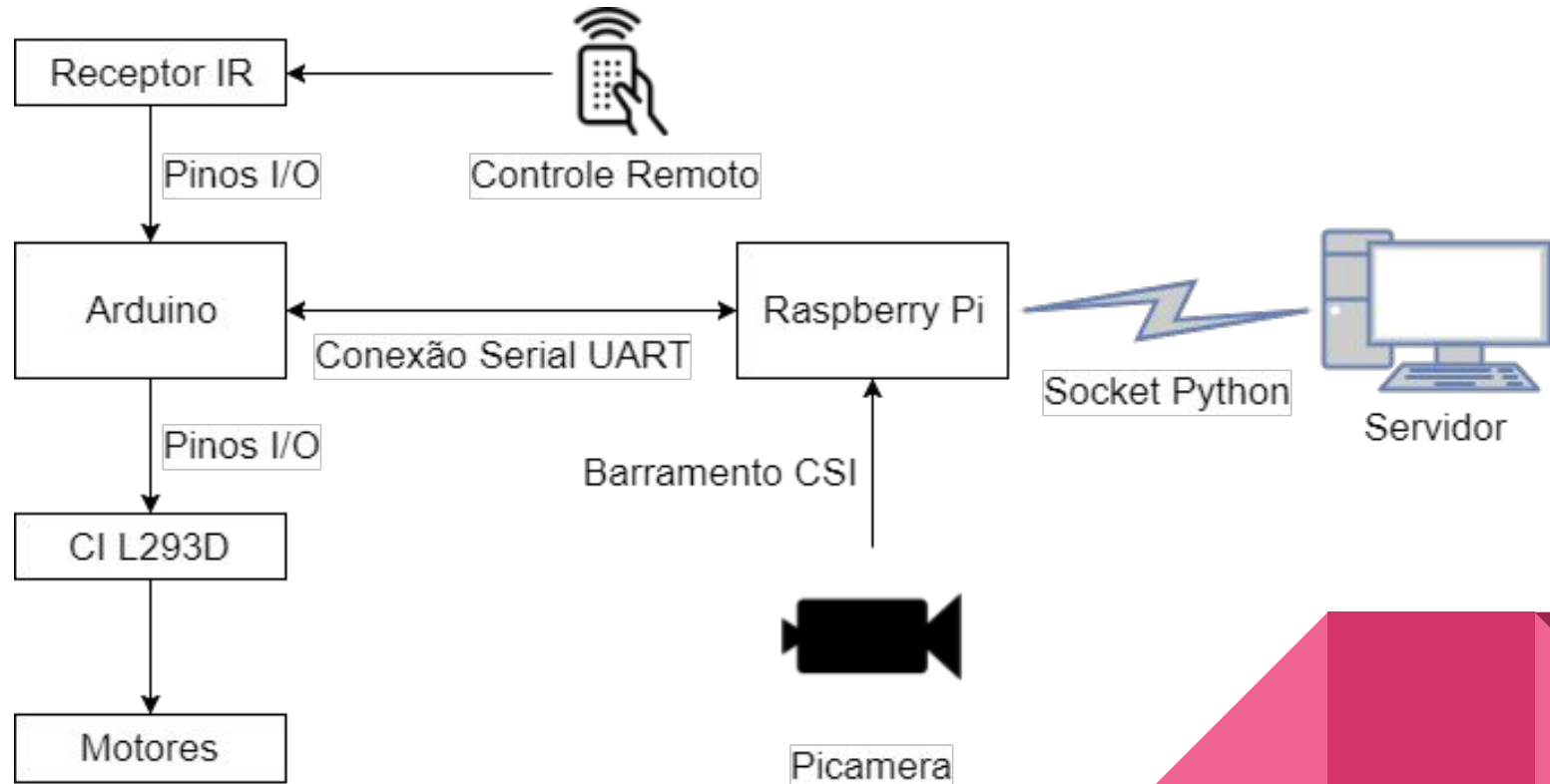


# Implementação - Modos de operação

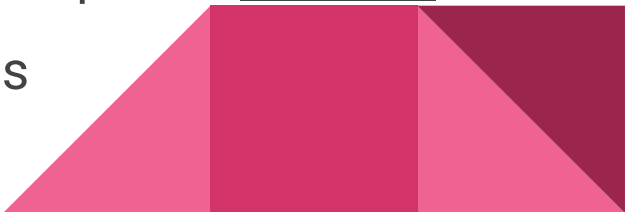
- Treinamento
- Predição



# Implementação - Arquitetura Geral



# Implementação - Dados de treinamento

- Controle remoto envia sinais para Arduino
  - Arduino decodifica os sinais e transforma em comando de movimento para os motores (Frente, Trás, Esquerda, Direita)
  - Arduino envia o comando para a Raspberry Pi
  - Raspberry Pi captura o frame da Picamera, monta mensagem contendo o frame e o comando e envia via Socket para o Servidor
  - Servidor aplica filtros nos frames e armazena-os
- 

# Implementação - Mensagem enviada

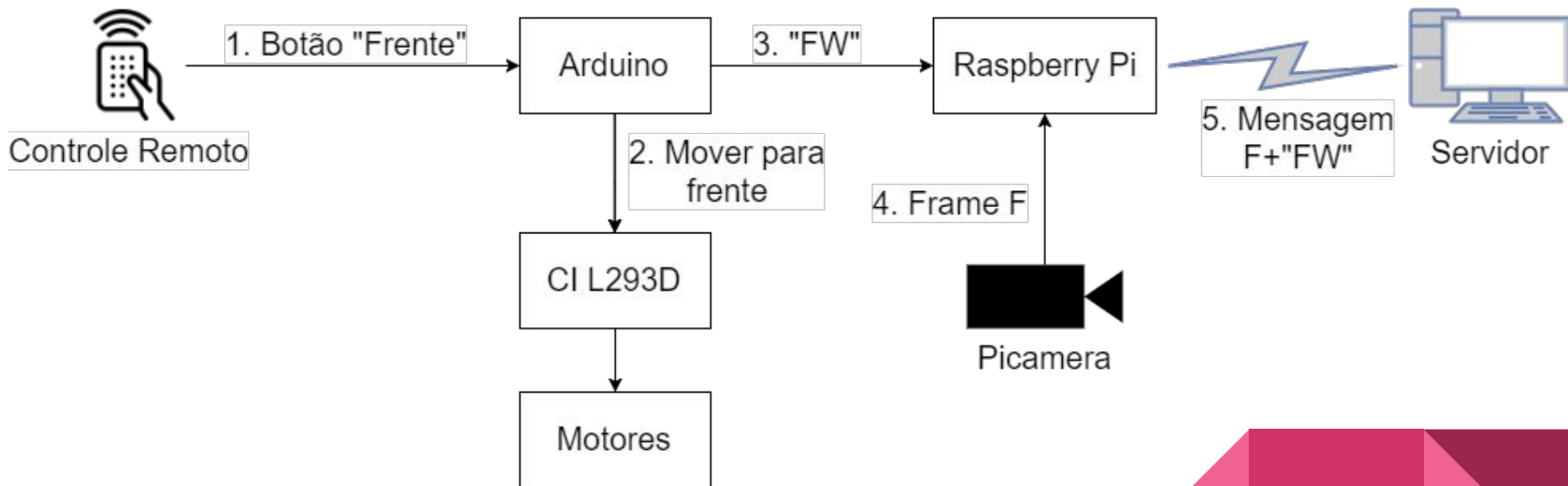
Controle Remoto	Mensagem Serial	Classificação
Backward	BW	0
Forward	FW	1
Left	LF	2
Right	RH	3

int	int	long	bytes
BV	CL	FRAME_LEN	FRAME_BYTES

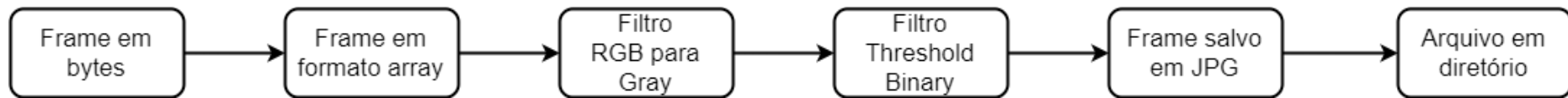
**BV** - bit de validação

**CL** - classificação do frame

# Implementação - Coleta de dados




# Implementação - Filtros (OpenCV)

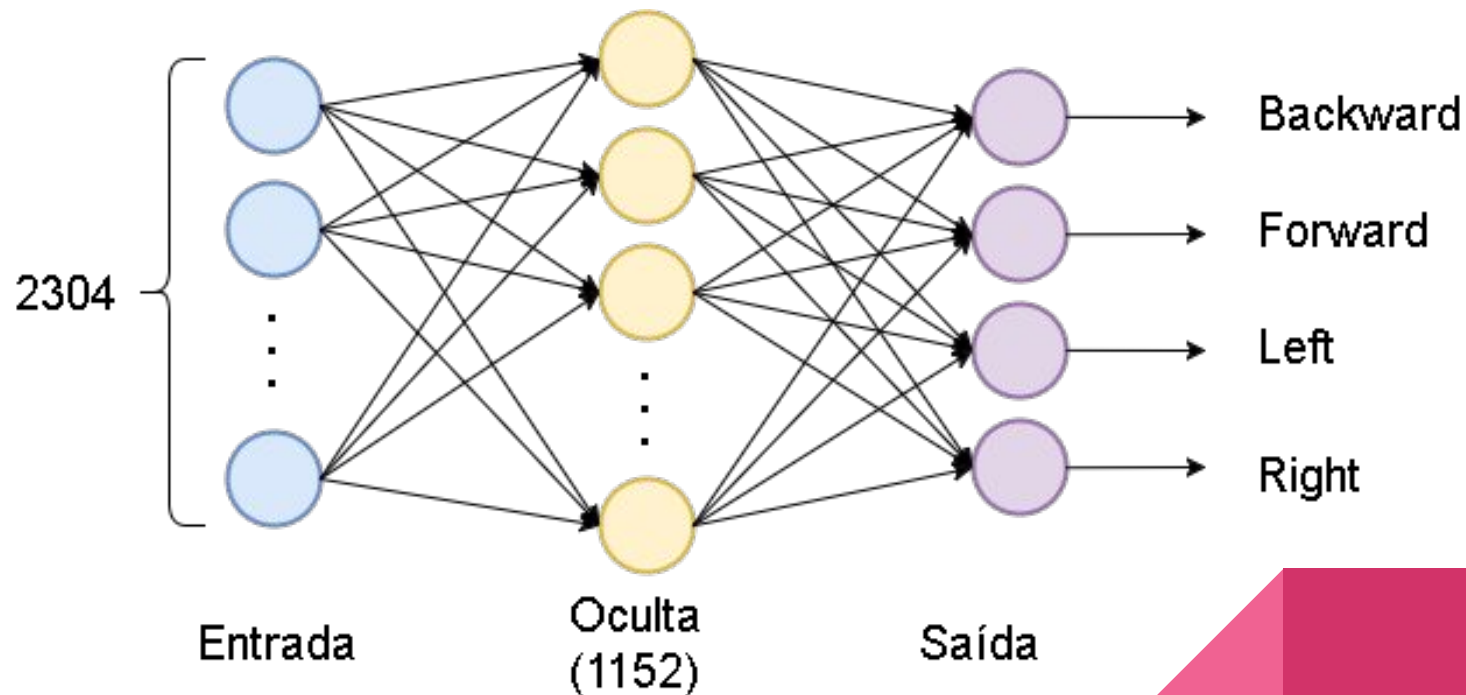




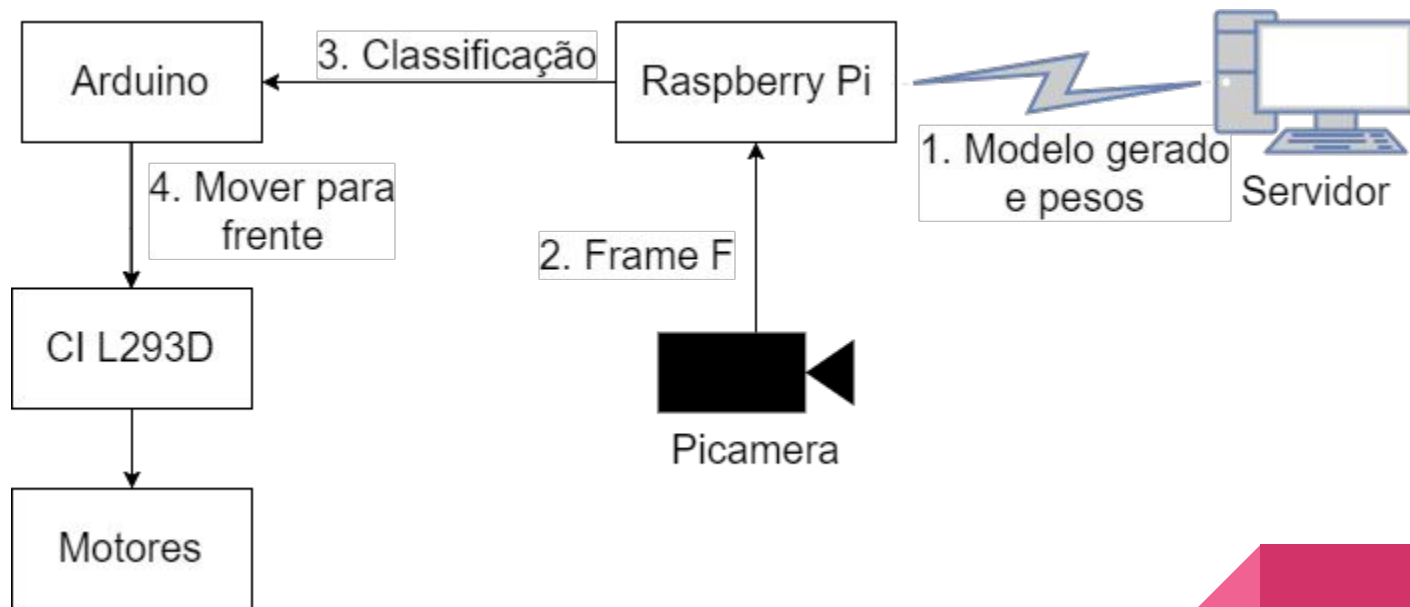
# Implementação - Treinamento

- Camada de entrada: 2304 neurônios (32 x 24 x 3)
  - Camada de saída: 4 neurônios (Frente, Trás, Esquerda, Direita)
  - Camada oculta: 1152 neurônios (ajuste de hiperparâmetros)
  - Função de ativação: ReLU
  - Dados de treinamento: 75%
  - Dados de teste: 25%
  - Dados de validação: 20% (dos dados de treinamento)
- 

# Implementação - Rede Neural



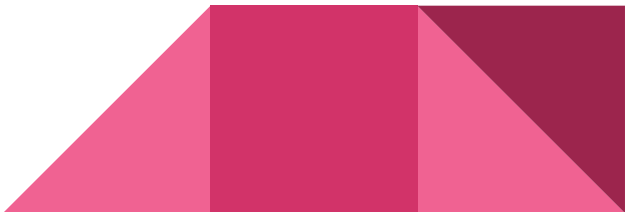
# Implementação - Predição



# Resultados

GridSearch: testa todas as combinações de parâmetros

Parâmetros testados:

- Neurônios (camada oculta): 72, 144, 288, 576, 1152
  - Função de ativação: ReLU, Sigmoid, Softmax
  - Batch\_size: 10, 40, 80
  - Epochs: 10, 50, 100
- 

# Resultados

Melhor resultado:

- Neurônios (camada oculta): 1152
- Função de ativação: ReLU
- Batch\_size: 80
- Epochs: 50

Taxa de acerto na classificação: **74.46%**

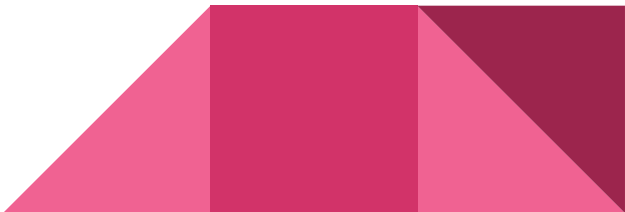


# Custo do projeto

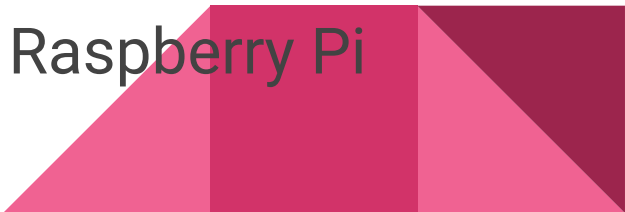
Componente	Custo estimado
Raspberry Pi 3 Model B	R\$ 160,00
Cartão microSD 16GB	R\$ 30,00
Câmera Picamera	R\$ 40,00
Arduino UNO Rev. 3	R\$ 50,00
CI L293D	R\$ 12,00
Chassis	R\$ 50,00
Fontes de Energia	R\$ 60,00



# Conclusão

- Possível contorno ao problema de má leitura de sensores ópticos por meio de tratamento de imagens
  - Rede Neural possível de ser embarcada em Raspberry Pi
  - Solução relativamente barata
- 

## Próximos passos

- Ajustar a Rede Neural de maneira a melhorar a acurácia de classificação
  - Sistema de controle de velocidade dos motores
  - Embutir o processo de treinamento na Raspberry Pi
- 



Obrigado!