

**DIEGO DUARTE MENESCAL**

**ANÁLISE DOS DESAFIOS DE UMA SOLUÇÃO IoT PARA DETECÇÃO  
DE INFRAÇÕES DE TRÂNSITO**

Monografia apresentada ao PECE –  
Programa de Educação Continuada em  
Engenharia da Escola Politécnica da  
Universidade de São Paulo como parte dos  
requisitos para conclusão do curso de MBA  
em Internet of Things.

São Paulo  
2018

**DIEGO DUARTE MENESCAL**

**ANÁLISE DOS DESAFIOS DE UMA SOLUÇÃO IoT PARA DETECÇÃO  
DE INFRAÇÕES DE TRÂNSITO**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Internet of Things.

Área de Concentração: Internet of Things

Orientador: Prof. Dr. Jaime Simão Sichman

São Paulo  
2018

## FICHA CATALOGRÁFICA

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catalogação na publicação  
Serviço de Biblioteca e Documentação  
Escola Politécnica da Universidade de São Paulo

Duarte Menescal, Diego  
ANÁLISE DOS DESAFIOS DE UMA SOLUÇÃO IoT PARA  
DETECÇÃO DE INFRAÇÕES DE TRÂNSITO / D. Menescal -- São  
Paulo, 2018. 50 p.

Monografia (MBA em Internet Of Things) - Escola Politécnica da  
Universidade de São Paulo. PECE – Programa de Educação Continuada  
em Engenharia.

Universidade de São Paulo. Escola Politécnica. PECE – Programa de  
Educação Continuada em Engenharia II.t.

## **RESUMO**

Dado o grande volume de informações geradas na era digital, novas tecnologias aparecem para lidar com a demanda de processamento associada, sendo uma delas a chamada Internet das Coisas (IoT). Existem já propostos na literatura modelos de referência que servem de guia para o desenvolvimento uma solução de IoT, entretanto, existem desafios a ser solucionados em diversas camadas desses modelos. Nesse trabalho, será feita uma análise de tais desafios em um contexto de deteção de infrações de trânsito, que utiliza a tecnologia de reconhecimento de caracteres ópticos (OCR) para extrair texto a partir de imagens de placas de veículos obtidas por uma rede de câmeras.

.

## **ABSTRACT**

Given the huge information volume generated in the digital era, new technologies emerge to deal with this associated processing demand, one of those being the so-called Internet of Things (IoT). In the literature, one can find reference models that guide the development of IoT solutions; however, there are challenges in several layers of these models. In this work, we analyze these challenges in the context of application of detection of traffic infractions, using optical character recognition (OCR) technology for text extraction from vehicle plates images captured by a camera network.

## SUMÁRIO

|  |           |
|--|-----------|
|  | Pág.      |
| <b>1. INTRODUÇÃO .....</b>   | <b>7</b>  |
| 1.1 Motivação .....  | 7         |
| 1.2 Objetivo .....   | 8         |
| 1.3 Justificativas .....   | 8         |
| 1.4 Estrutura do Trabalho .....                                    | 9         |
| <br>   |           |
| <b>2. DETECÇÃO DE VEÍCULOS EM VIAS DE TRÂNSITO .....</b>           | <b>10</b> |
| 2.1 Detecção usando Radar .....                                    | 10        |
| 2.2 Obtenção da Imagem .....                                       | 12        |
| <br>   |           |
| <b>3. CONCEITOS BÁSICOS .....</b>                                  | <b>13</b> |
| 3.1 Internet das Coisas (IoT) .....                                | 13        |
| 3.2 Reconhecimento de Caracteres Ópticos (OCR) .....               | 22        |
| 3.3 Inteligência Artificial (IA) .....                             | 29        |
| 3.4 Computação de Alto Desempenho (HPC) .....                      | 37        |
| <br>   |           |
| <b>4. SOLUÇÃO IoT PARA DETECÇÃO DE INFRAÇÕES DE TRÂNSITO .....</b> | <b>42</b> |
| 4.1 Proposta de solução IoT .....                                  | 42        |
| 4.2 Análise dos Desafios .....                                     | 46        |
| <br>   |           |
| <b>5. CONSIDERAÇÕES FINAIS.....</b>                                | <b>49</b> |
| <br>   |           |
| <b>REFERÊNCIAS .....</b>   | <b>50</b> |

## 1. INTRODUÇÃO

### 1.1 Motivação

Pode-se perceber claramente nos dias de hoje o avanço tecnológico presente no cotidiano das pessoas. Tal avanço levou à miniaturização dos dispositivos computacionais, agora com mais capacidade, mobilidade, acessibilidade, portabilidade e conectividade. O resultado é uma concentração elevada de tecnologias embarcadas em um simples dispositivo como um celular.

Por outro lado, o volume de informação disponível para consumo aumentou exponencialmente. Os dispositivos necessitam de maior capacidade para tratar essa informação, recorrendo ao uso de mais memória, processadores mais rápidos e com capacidade de paralelizar tarefas, discos de armazenamento maiores e mais rápidos, como os discos de estado sólido (em inglês, Solid State Drive - SSD). Além da alta demanda para mudanças em hardware, o software passou também por grandes mudanças. Novos algoritmos foram desenvolvidos para lidar com esses dados, atuando como agentes transformadores de dados em novos dados mais agregados. A combinação de hardware e software que atendam essa nova demanda por informação tem impactos no ramo dos negócios, permitindo novos empreendimentos com foco em tratamento de dados.

A área de Internet das Coisas (em inglês, Internet of Things – IoT) é exemplo de tecnologia que agregou um conjunto de tecnologias já pré-existentes para definir novos conceitos. O elemento chave dessa agregação é a Internet, permitindo que haja comunicação, e portanto, possibilitando que dados sejam trocados entre cada elemento.

Dado que a tecnologia de IoT permitiu um novo fluxo de dados para ser potencialmente analisado, é importante que essa análise seja executada no menor tempo possível e com eficiência, garantindo uma tomada de decisão mais ágil e correta. Para tal, uma sequência de elementos contidos nesse arcabouço de IoT precisam ser avaliados, desde a aquisição dos dados, passando pela comunicação destes dados, construção de uma base para análise desses dados, e o processamento desta base, fornecendo uma resposta esperada.

Um exemplo de aplicação que interliga as tecnologias mencionadas acima é o sistema de detecção de veículos em vias de trânsito. Neste domínio, um veículo ultrapassa um ponto onde se situa um sensor ligado a uma câmera fotográfica que pode tirar

uma foto quando este veículo cometer alguma infração das leis de trânsito. A imagem é a forma de registro da infração, e deve ser armazenada para processamento e extração da placa do veículo. Sob o ponto de vista de uma cidade grande como São Paulo, por exemplo, a quantidade de imagens obtidas por um sistema deste tipo gera uma enorme demanda em volume de dados. Tais dados precisam ser pré-processados e analisados com alta precisão e alto desempenho.

Dentro do cenário de IoT, essa aplicação oferece uma diversidade de tecnologias, tais como processamento de imagens, reconhecimento de caracteres ópticos, inteligência artificial e computação de alto desempenho.

Esta cadeia de elementos e tecnologias representa a motivação desse trabalho, mostrando os desafios de manter eficiente toda a infraestrutura de IoT.

## **1.2 Objetivo**

O objetivo deste trabalho é realizar um survey analítico, apresentando quais são os desafios que surgem para o desenvolvimento de uma aplicação de IoT para detecção de infrações de trânsito, que utiliza a tecnologia de reconhecimento de caracteres ópticos (OCR) para extrair texto a partir de imagens de placas de veículos obtidas por uma rede de câmeras.. Para tal, utiliza-se um modelo de referência proposto em [19]. Este modelo de referência é dividido em camadas, e cada uma possui uma funcionalidade e dispositivos envolvidos. Dessa forma, os desafios apresentados são estruturados segundo essas camadas, dando uma visão modular de uma aplicação complexa de IoT.

## **1.3 Justificativas**

O desenvolvimento de uma aplicação de IoT envolve uma série de problemas como performance de hardware, de software e de comunicação de redes. Além dos problemas que já ocorrem em soluções convencionais, no contexto de IoT aparecem novos problemas, como por exemplo sensores não confiáveis, data analytics, dados não estruturados, novos protocolos de rede, Tais questões podem apresentar oportunidades para uma nova demanda de pesquisa e desenvolvimento.

## 1.4 Estrutura do Trabalho

Este Capítulo apresentou o cenário que motivou este trabalho, além de explicar o seu objetivo e justificativa para a sua realização.

O Capítulo 2 introduz uma visão geral de um sistema de detecção de veículos em vias públicas, descrevendo as etapas para detectar o veículo e como obter a sua imagem. Esta aplicação serve como base para a aplicação de IoT definida pelo capítulo 4.

O Capítulo 3 apresenta, de forma resumida, as tecnologias utilizadas numa aplicação deste tipo: Internet das Coisas (IoT), Reconhecimento de Caracteres Ópticos (OCR), Inteligência Artificial (IA) e Computação de Alto Desempenho (HPC). É apresentada uma visão geral de cada uma delas, identificando seus conceitos básicos e exemplos de aplicações.

O Capítulo 4 descreve uma possível aplicação de IoT, que utiliza as tecnologias descritas no capítulo 3, visando resolver o cenário de deteção de veículos introduzido no capítulo 2. Neste capítulo, apresenta-se o modelo de referência de IoT para analisar os desafios que surgem nesse contexto.

O Capítulo 5 apresenta as conclusões deste trabalho, evidenciando como os desafios encontrados no capítulo 4 podem se tornar oportunidades de pesquisa e desenvolvimento para trabalhos futuros.

## 2. DETECÇÃO DE VEÍCULOS EM VIAS DE TRÂNSITO

Para evitar que os veículos circulem em vias de trânsito em velocidades elevadas ou que cruzem semáforos na luz vermelha, são utilizados dispositivos que possibilitam detectar a presença de um veículo em trechos de rua e obter uma imagem desse veículo durante o momento da infração.

### 2.1 Detecção usando Radar

Existem diferentes dispositivos que se destinam a detetar a presença de um veículo em trechos de rua. Estes possuem diversos tipos e nomes, tais como o Radar, o LiDAR, e os detectores fixos [25].

O dispositivo chamado Radar (figura 2), do acrônimo Radio Detection and Ranging, pode ser um dispositivo móvel (tipo pistola) ou montado sobre uma base fixa (um tripé). O Radar tipo pistola utiliza como princípio o efeito Doppler para calcular a velocidade do veículo: a frequência da onda emitida é comparada com a freqüencia de onda refletida pelo veículo que chega ao Radar. Já o tipo tripé emite constantemente um feixe de onda diretamente no solo, formando uma espécie de cortina. Quando um veículo passa por essa cortina, ele interrompe a emissão do feixe por um instante de tempo. Esse intervalo é usado então para calcular a velocidade que foi atravessada na cortina.

Ao contrário do Radar, que usa ondas de rádio, o dispositivo LiDAR (figura 1) usa um sinal luminoso para calcular a velocidade do veículo. Também pode se apresentar como um dispositivo móvel ou fixo. Em ambos os casos, a forma de cálculo é a mesma. Entretanto, o intervalo de emissões das ondas do LiDAR é muito curto, entre 100 e 600 milissegundos [25], tornando difícil um veículo conseguir burlar a medição da velocidade.

Figura 1: Exemplo de um dispositivo LiDAR [25].



Figura 2: Exemplo de um dispositivo Radar, ou “pardal”.



Fonte: <http://pioneiro.clicrbs.com.br/rs/geral/cidades/noticia/2014/09/tres-pardais-sao-instalados-na-rota-do-sol-entre-vila-seca-e-tainhas-4594185.html>

Por fim, os detectores fixos, popularmente chamados de “pardais”, medem a velocidade de forma diferente das anteriores. No próprio asfalto, são instalados 3 sensores que emitem um campo eletromagnético constante. A velocidade é calculada no momento que o veículo atravessa esse campo, causando uma perturbação. Cada um dos sensores cumpre um papel importante para a detecção e medição. A detecção ocorre quando o veículo passa pelo primeiro sensor. Nele, um contador de tempo é iniciado para calcular a velocidade. Ao passar pelo segundo sensor, já é possível calcular a velocidade entre os dois sensores. O terceiro sensor serve para confirmar o cálculo anterior, e se caso a velocidade aumente, esta é registrada ao final.

Para detectar que um veículo ultrapassou o semáforo no sinal vermelho, um campo eletromagnético é emitido dos sensores do asfalto assim que a cor do semáforo ficar vermelha. Quem ultrapassar o cruzamento nessa situação irá causar o distúrbio no campo e este será detectado.

## 2.2 Obtenção da imagem

Uma vez que o veículo tenha sido detectado pelos sensores e cometido alguma infração, uma câmera fotográfica é usada para registrar a imagem do infrator. Nessa imagem, o objetivo é observar a placa do veículo e extrair suas informações, para que sejam aplicadas as devidas medidas legais . A Figura 1 ilustra essa situação:

Figura 3: Imagem obtida por câmera ligada a radar.



Fonte:

[https://edisciplinas.usp.br/pluginfile.php/2971935/mod\\_folder/content/0/DM%20-%20Reconhecimento%20Autom%C3%A1tico%20de%20Placas.pdf?forcedownload=1](https://edisciplinas.usp.br/pluginfile.php/2971935/mod_folder/content/0/DM%20-%20Reconhecimento%20Autom%C3%A1tico%20de%20Placas.pdf?forcedownload=1)

Dependendo da forma que a imagem for obtida, a sua qualidade pode ser alterada, tornando difícil a identificação das placas. Os fatores podem estar relacionados com o estado de conservação da placa, com as condições climáticas do momento da imagem e do estado de conservação e limpeza da câmera [25]. Como tais fatores são externos ao sistema de detecção, esse último não tem como se ajustar para os problemas listados acima.

### 3. CONCEITOS BÁSICOS

Neste capítulo são abordadas, de forma resumida, as tecnologias relevantes para o tratamento do problema de que trata este trabalho: Internet das Coisas (IoT), Reconhecimento de Caracteres Ópticos (OCR), Inteligência Artificial (IA) e Computação de Alto Desempenho (HPC). O objetivo do capítulo é prover o leitor com informações suficientes para a compreensão da análise dos desafios para o problema tratado neste trabalho, i.e., o desenvolvimento de uma aplicação de IoT para detecção de infrações de trânsito, que utiliza a tecnologia de reconhecimento de caracteres ópticos (OCR) para extrair texto a partir de imagens de placas de veículos obtidas por uma rede de câmeras. Tais desafios são assunto do capítulo 4.

#### 3.1 Internet das Coisas (IoT)

A Internet acaba de dar um avançar um passo, onde não apenas estão interconectados os computadores, mas também outros dispositivos, como telefones, relógios, eletrodomésticos, e automóveis entre outros. Cada um desses equipamentos é uma “coisa” conectada à Internet.

A aderência a essa nova tendência tem crescido, permitindo novas oportunidades de projetos inovadores. Segundo [13], podem-se citar algumas áreas de interesse para esta nova abordagem, como agricultura, saúde, automação residencial e industrial, energia, telecomunicações, forças armadas, varejo, setores automotivo e aeronáutico, meio ambiente e cidades inteligentes. Baseado nessas áreas, pode-se perceber a grande variedade de projetos que podem surgir, além de desafios para serem superados, contribuindo de forma incremental para o desenvolvimento completo do IoT.

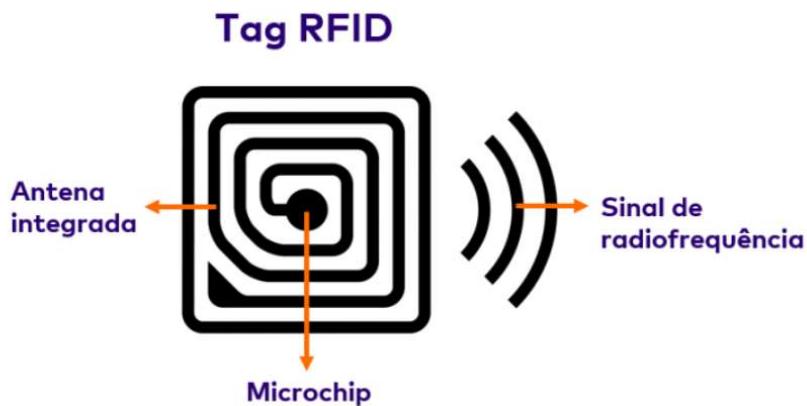
Para garantir a capacidade de conectividade, comunicação e mobilidade dos dispositivos, algumas tecnologias existentes contribuem para o aprimoramento de aplicações de IoT. Tais tecnologias, bem como um modelo de referência para as aplicações de IoT, são apresentados a seguir.

### 3.1.1 Identificação por Rádio Frequência - RFID

RFID é uma tecnologia já consolidada, e tem como objetivo conectar dois dispositivos através de ondas eletromagnéticas. A forma de conexão proposta pelo RFID é chamada peer-to-peer, ou P2P, assim não existe um dispositivo intermediário.,.

Um sistema de comunicação RFID é composto por 3 elementos principais [13]: um transponder, um leitor e um middleware. O transponder é o elemento que vai armazenar os dados. Um exemplo simples é o uso de tags, conforme mostra a Figura 4.

Figura 4: Tag RFID para armazenar dados.



Fonte: <https://www.mandae.com.br/blog/etiquetas-rfid-como-funcionam-e-quais-sao-as-suas-vantagens/>

Pela Figura 4, é possível notar que existe uma antena integrada que emite o sinal de rádio frequência com os dados contidos no microchip interno.

O leitor é responsável por receber esse sinal enviado pela tag, por exemplo para converter o sinal eletromagnético em um sinal digital que representará o dado enviado pelo transponder. A Figura 5 ilustra um exemplo de leitor RFID.

Figura 5: Leitor RFID.



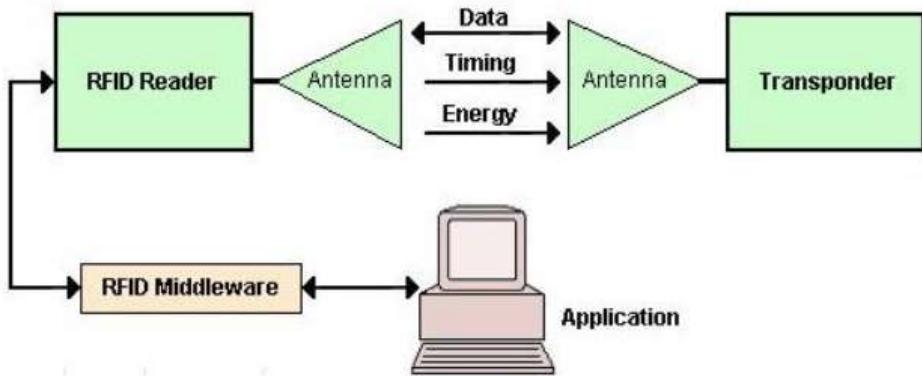
Fonte: <https://researchdesignlab.com/iot-enabled-rfid-reader.html>

O leitor geralmente pode estar em uma base estática, e quando uma tag RFID se aproxima, ele interpreta corretamente o sinal emitido.

Por fim, o middleware atua como um agente que coleta a informação obtida pelo leitor e envia para algum sistema de armazenamento de dados, tipicamente um banco de dados. Um middleware é um software que faz a intermediação entre o dado obtido no ambiente e o dado que chega ao banco de dados. Através do acesso ao banco de dados, uma aplicação específica de análise pode fazer um bom uso desses dados.,.

A Figura 6 ilustra um sistema simples de IoT, usando RFID como meio de comunicação.

Figura 6: Sistema de IoT com RFID.



Fonte:

<http://www.devx.com/supportitems/showSupportItem.php?co=31108&supportitem=figure1>

### 3.1.2 Comunicação de Campo Eletromagnético Próximo

Nesse padrão de comunicação chamado de Near Field Communication (NFC), os dois dispositivos estão localizados muito próximos um do outro, na ordem de polegadas (1 polegada é igual a 2,54 centímetros) e realizam transferência de dados entre si.

### 3.1.3 WiFi

Mais conhecido nos ambientes domésticos, o WiFi é um protocolo de rede de acesso sem fio, utilizando rádio frequência para propagação do seu sinal. Ao contrário do RFID, o modo de conexão WiFi se utiliza de um dispositivo intermediário para estabelecer a conexão entre dois dispositivos quaisquer. Esse dispositivo intermediário é chamado de ponto de acesso, representado na Figura 7.

Figura 7: Ponto de Acesso.



Fonte: <https://www.eletronicasantana.com.br/acess-point-wireless-n-300mbps-tl-wa901nd-tp-link/p>

Outra diferença técnica entre o WiFi e o RFID diz respeito aos espectros de frequência utilizados para propagação do sinal. Para o RFID, as frequências usadas são 125 KHz, 13.56 MHz e entre 800 a 900 MHz [13]. Para o WiFi, existem diversas opções de espectros de frequência, indicadas na Figura 8.

| Aspect                 | Standard IEEE | Frequency   |
|------------------------|---------------|---|
| WiFi Wireless Fidelity | 802.11        | Channel Number 1 - 14<br>2401- 2473 MHz – Lower Frequency<br>2412- 2484 MHz –Middle Frequency<br>2423- 2495 MHz – Upper Frequency |
| White-Fi               | 802.11af      | 470 - 710MHz  |
| Microwave Wi-Fi        | 802.11ad      | 57.0 - 64.0 GHz ISM band (Regional variations apply)<br>Channels: 58,32, 60.48, 62.64, and 64.80 GHz                              |

Figura 8: Frequências usadas pelo padrão WiFi [13].

Um exemplo simples de como se aplica o uso do WiFi dentro do contexto do IoT pode ser representado pelo uso de celulares com aplicações de sensores (temperatura ou pressão atmosférica) conectados a um ponto de acesso em um ambiente doméstico.

O ponto de acesso tem um papel importante, possibilitando a conexão dos dados com outros dispositivos, através de uma rede cabeada.

### **3.1.4 ZigBee**

O ZigBee é um padrão de comunicação por rádio frequência de baixa potência, direcionado para dispositivos que estejam dentro de um raio de distância pequeno. O uso desse padrão pode ser usado para alcançar longas distâncias, usando dispositivos intermediários para repassar o sinal.

Para o cenário de IoT, este é o padrão de comunicação mais compatível. Embora o uso de comunicação por rádio frequência seja possível num ambiente com vários sensores, a emissão de um sinal consome energia do dispositivo. Considerando um sensor como um equipamento que pode ser posicionado em qualquer lugar, mesmo distante de uma fonte de energia, ele necessitaria portanto de uma bateria interna para permitir seu funcionamento. Dessa forma, é indicado que qualquer tarefa deve ter um baixo consumo de energia, inclusive a emissão de sinal de rádio para comunicação. O Bluetooth é outro padrão, que atua de maneira similar ao ZigBee, mas seu alcance é menor, com limite de até 30 metros, enquanto que o Zigbee pode atingir mais de 100 metros [13].

Pelas descrições acima, pode-se perceber que tipicamente uma aplicação de IoT comporta a aquisição de muitos dados através do uso dos sensores. Esses sensores, por sua vez, se conectam através de uma rede específica, seja direta ou através de um ponto intermediário (um ponto de acesso, por exemplo); esta rede se conecta a uma outra rede, e assim sucessivamente, até que todos os dados coletados sejam enviados para um ponto central, onde é feita a sua análise. Intuitivamente, pode-se ver a formação de uma infraestrutura complexa, onde existem redes distintas, com protocolos distintos e diversos dispositivos distintos garantindo o fluxo contínuo de dados. Tal complexidade motiva a busca de um modelo de referência, como mostrado a seguir.

### **3.1.5 Modelo de referência para aplicação de IoT**

O grau de complexidade de uma aplicação em IoT se mostra elevado, visto que novos dispositivos podem ser inseridos e conectados na infraestrutura. A forma como cada

dispositivo é conectado, os tipos de conexão, e os tipos de dados trafegados são elementos para projetar uma arquitetura de IoT.

Segundo [18], quando se define uma arquitetura de IoT é possível notar alguns requisitos, tais como:

- *Clareza para o gerenciamento de redes*, pois cada funcionalidade deve ser avaliada independente do fabricante;
- *Facilidade de entendimento* aos desenvolvedores, para que saibam quais funcionalidades são necessárias e como estas devem trabalhar em conjunto;
- *Facilidade do processo de padronização*, permitindo interoperabilidade e redução de custos na implementação.

A Figura 9 ilustra a forma como os dispositivos se comunicam dentro de um ambiente de IoT:

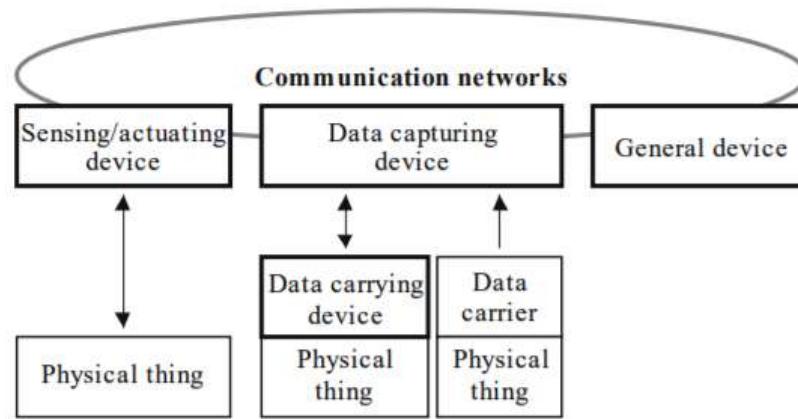


Figura 9: Dispositivos em um ambiente IoT [18].

Segundo a Figura 9, pode-se perceber como se dá a mudança do ambiente real (mundo físico) para o ambiente virtual: as “coisas” precisam ser anexadas a um dispositivo capaz de obter os dados, seja uma tag de RFID, um código de barras ou até mesmo um sensor. Os dados obtidos são então enviados a outros dispositivos capazes de realizar um processamento inicial sobre os dados, como por exemplo a conversão de um sinal do tipo analógico para digital. Nesse caso, o dado em formato digital permite aos programas de computador interpretarem os dados em outra etapa de processamento.

Outra análise baseada na Figura 9 é o surgimento de novas redes para interconectar as “coisas” com os dispositivos de coleta de dados, uma rede entre os dispositivos em

si, e uma rede que irá enviar os dados para um processamento central, vista no topo da Figura 9. Cada rede possui seu conjunto de regras, e para que cada uma possa se comunicar sem conflito, é necessário o uso de dispositivos chamados de *gateways*. Os *gateways* atuam como conversores de protocolos, garantindo a interoperabilidade entre as redes. A Figura 10 ilustra a interconexão desses dispositivos.

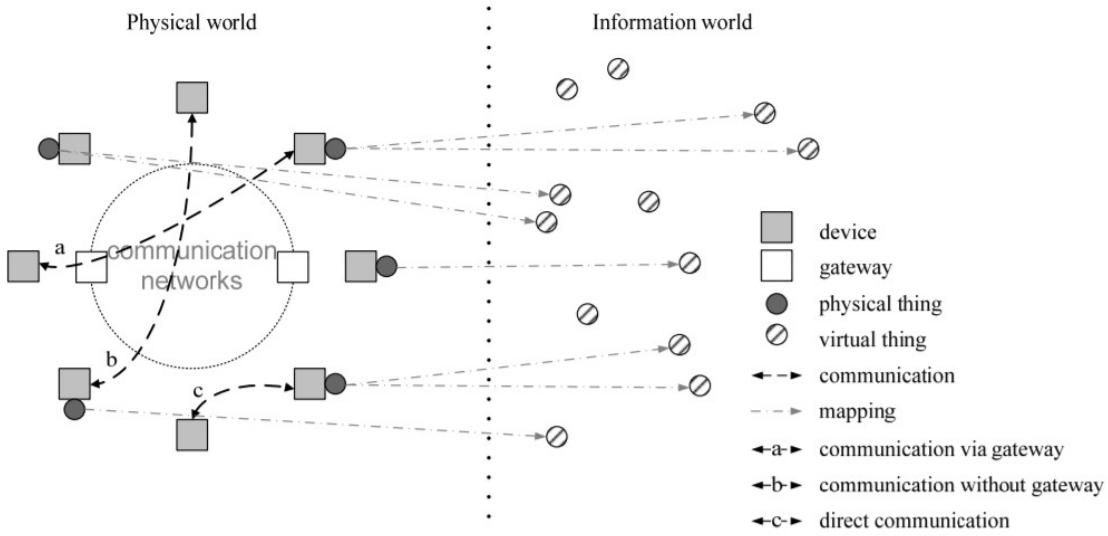


Figura 10: Junção do mundo físico com o virtual [18].

Além disso, como indicado na Figura 10, algumas comunicações podem não passar diretamente pelo *gateway*; entretanto, quando se necessita atingir outras redes que possuem protocolos diferentes, o *gateway* se mostra necessário.

Segundo [18], é possível identificar um dos desafios que surgem em uma arquitetura de IoT, ainda mais quando se deseja expandir sua infraestrutura. Garantir a conectividade entre os dispositivos conectados ao *gateway* é uma tarefa desafiadora, uma vez que uma falha nesses dispositivos pode causar problemas para a transmissão correta dos dados. Por conta dessa alta carga de responsabilidade para um dispositivo, existem três categorias de *gateways* [18]:

- Aqueles que podem conectar vários tipos de dispositivos e suas tecnologias;
- Aqueles que permitem o uso de tecnologias para redes locais de alto alcance;
- Aqueles que permitem diversas aplicações, como gerenciamento de rede e segurança.

Para uma melhor compreensão de como se define uma arquitetura de IoT, o modelo exibido na Figura 10 complementa o modelo da Figura 9, onde apenas se informa

como os dispositivos se conectam na rede. A Figura 11 descreve as funcionalidades, bem como as diferentes camadas, que envolvem todo o ambiente de IoT, permitindo entender a funcionalidade de cada elemento que irá compor uma infraestrutura de IoT.

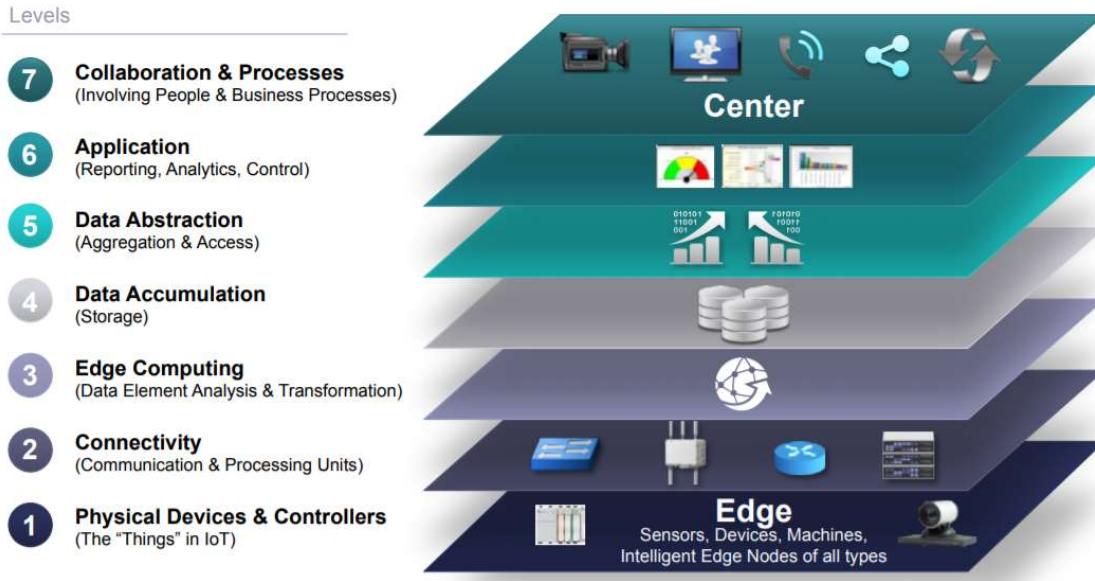


Figura 11: Modelo de referência para aplicação de IoT [19].

Na camada 1, se situam as “coisas” e os dispositivos que carregam os dados. Como exemplo, têm-se os sensores, leitores de RFID, tags de RFID, código de barras, câmeras, etc.

Já a camada 2 envolve a conectividade dos elementos da camada 1, recebendo os dados em um formato digital. Por exemplo, pode-se utilizar comutadores, roteadores e pontos de acesso. Essa camada irá realizar a conversão de protocolos, garantindo que as redes das camadas superiores possam se comunicar com os dispositivos da camada 1.

A camada 3 consiste na fronteira para chegar a um *data center*. O volume de dados de entrada é muito grande, e por isso eles precisam ser reajustados para que possam ser armazenados de uma maneira mais uniforme. Esta camada também é chamada de *fog computing* [18], remetendo a uma névoa que paira bem próximo aos dispositivos representados nas camadas 1 e 2.

A camada 4 desempenha o papel de armazenamento dos dados transformados na camada 3, servindo de base para as camadas 5 e 6, que efetivamente são as responsáveis pelo acesso à base de dados e pela implementação dos algoritmos de

análise de dados. As técnicas de inteligência artificial e de computação de alto desempenho, descritas mais adiante neste capítulo, atuam nesta camada 6.

Finalmente, a última camada representa a interface com o usuário final, ou seja, aquele que irá consumir todos os serviços descritos nas camadas inferiores.

Sendo o objetivo deste trabalho analisar os desafios de IoT em um contexto de detecção de infrações de trânsito, apresenta-se a seguir outra tecnologia essencial neste contexto.

### **3.2 Reconhecimento de Caracteres Ópticos (OCR)**

O termo OCR, que vem do inglês *Optical Character Recognition*, corresponde a uma técnica que converte uma imagem contendo texto em um formato digital que permite a um computador processar este texto de modo conveniente.

Por se tratar de reconhecimento de texto, o grau de complexidade para realizar tal tarefa é elevado, visto que existem diversos tipos de linguagem, fontes e estilos presentes em textos e imagens do cotidiano. Cada linguagem possui um conjunto de regras para construção de palavras e acentuação, e em alguns casos essas regras são exclusivas de certas linguagens, como por exemplo o português do Brasil, que possui diversos acentos (circunflexo, til, trema, agudo, crase) e letras diferentes, como o cedilha.

Historicamente o OCR teve seu início na década de 1950, quando uma máquina chamada GISMO era capaz de reconhecer uma quantidade limitada de caracteres (23 no total), sendo que um por vez [24]. As décadas seguintes não renderam muitos avanços em termos de pesquisa acadêmica, se destacando os trabalhos realizados pelas agências governamentais e grandes empresas como bancos, redatoras de jornal e companhias aéreas [24].

Ainda que exista uma grande distância entre a capacidade de reconhecimento de texto por uma máquina e por um ser humano, pesquisas recentes mostram um desenvolvimento de novas técnicas para melhoria na acurácia e tempo de execução de sistemas de OCR para uma grande variedade de documentos. Um exemplo que mostra esse avanço é a ferramenta *open source* Google Tesseract, que oferece um *software* capaz de reconhecer texto em imagens em mais de 100 idiomas distintos. Esta ferramenta ainda permite que sejam feitos ajustes por qualquer usuário, e tem

se tornado objeto de pesquisas acadêmicas ou até trabalho de empresas, garantindo uma melhoria gradual e com grande abrangência.

Um sistema de OCR pode ser classificado de acordo com o tipo de texto presente na imagem: texto impresso por máquina ou escrito manualmente. Para se avaliar o grau de complexidade de um OCR frente a esses dois tipos de texto, observam-se alguns aspectos do texto, tais como tamanho das letras, espaçamento entre letras e palavras, distância entre linhas e sobreposição de palavras. Sendo assim, um OCR com texto impresso por máquina é mais simples, visto que os aspectos de tamanho, espaçamento e distância entre letras e linhas é uniforme, permitindo reconhecer o texto com menos trabalho. A sobreposição de palavras não ocorre, evitando que os caracteres fiquem irreconhecíveis. O caso de escrita manual é mais complexo: como cada ser humano possui sua forma de escrita, o trabalho de buscar padrões no texto torna-se mais difícil. Além disso, existe uma grande probabilidade de que ocorra sobreposição de texto, aumentando mais o grau de dificuldade do problema de OCR.

As Figuras 12 e 13 ilustram as diferenças destes dois tipos de texto.

Figura 12: Exemplo de escrita manual



Fonte: <https://threefeelings.com/curso-caligrafia/>

Figura 13: Exemplo de escrita por máquina.

Estimada señorita:

Son de tal magnitud mis deseos de formalizar mis relaciones con Ud. que goso en comunicaria a todas horas del dia, que daria mi pobre corazon, perturbado ante una joven tan bella, por dar gusto a mis grandes y poderosas conocimientos que se ven atravesados por agujones. He sido informado de que Ud. es tan pura asi como amable, modesta, simpatica y bonita, que espero que no oponga resistencia a mi natural carisma, mi gallarda presencia y mi garbo, que es capaz de destrozar el mas fuerte corazon, que sienta tan solo un leve y minimo cariño. Esperando a unirnos sentimentalmente y preferentemente sin mas demora, permiteme acompañarla a la hora y sitio que Ud. tenga por gusto.

Un admirador

Fonte: <https://www.pinterest.co.uk/pin/520095456946612602/>

O desafio do OCR para texto manual é ainda maior quando se consideram duas possíveis técnicas de reconhecimento: online e offline. A técnica online reconhece o texto ao mesmo tempo em que este é escrito. Já na técnica off-line, o reconhecimento é realizado após o texto ser escrito. Claramente, a técnica offline, é mais complicada, visto que os traços de escrita não possuem um padrão uniforme e ainda com o alto risco de sobreposição de texto. O processamento online é menos complexo por que durante a escrita em “tempo real” as características do caractere que está sendo escrito ainda não foram distorcidas pelo restante do texto.

Mesmo diante de um desafio grande como é o OCR, já existem diversas aplicações que já fazem uso das metodologias pesquisadas. As principais aplicações que utilizam OCR são:

- Leitura de cheque bancário
- Verificação de assinatura
- Processamento automático de formulários
- Validação de passaportes
- Reconhecimento automático de placas
- Auxílio a pessoas com deficiência visual

Um sistema de OCR consiste em um conjunto de processos que realizam diferentes etapas do problema de reconhecimento de texto em imagem. A Figura 14 ilustra os processos envolvidos.



Figura 14: Processos de um sistema de OCR. Fonte: Autor.

### 3.2.1 Aquisição da Imagem

Esta é a fase inicial, onde a imagem é adquirida através de alguma fonte externa, como por exemplo um scanner ou câmera fotográfica digital. Durante essa etapa, a imagem passa por vários métodos de tratamento, tais como compressão, digitalização ou até binarização. Originalmente, a imagem é composta de três intensidades de cores, o chamado padrão RGB (*Red, Green, Blue*). Isso resulta em uma imagem com vários tons de cores. Já na binarização, último método citado, a imagem é convertida inicialmente para apenas tons de cinza e depois é aplicado um limiar (*threshold*) para definir apenas duas cores na imagem: preto e branco.

### 3.2.2 Pré-processamento

A aquisição da imagem não é suficiente para se tentar extrair o seu conteúdo, visto que algumas imagens podem conter interferências que prejudiquem a sua extração. Essas interferências são chamadas de ruídos, e dependendo do tipo de informação que se deseja extrair, tudo o que atrapalha pode ser considerado um ruído. Por exemplo, na Figura 15 pode-se ver que a assinatura é considerada ruído se a intenção é extrair o texto de máquina ou vice-versa.

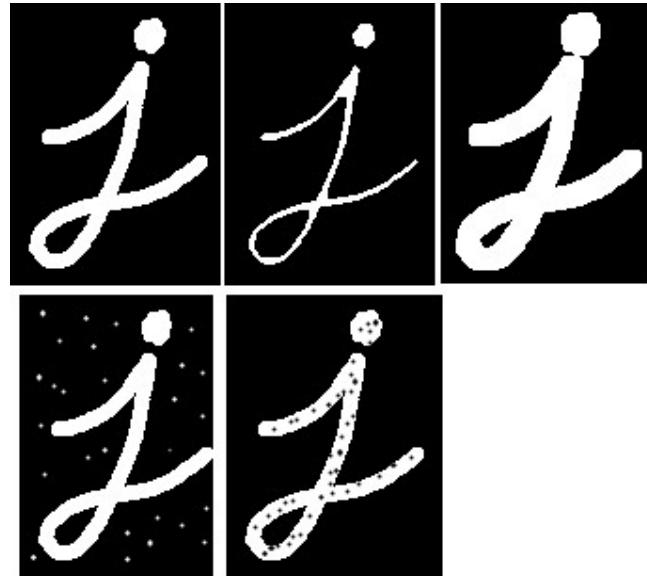
Figura 15: Imagem com ruído.



Fonte: <http://poncheverde.blogspot.com/2015/12/lula-assinou-tambem-mp-512.html>

O objetivo dessa etapa é reduzir ou eliminar, se possível, o ruído que venha a interferir no processamento do texto. Dentre algumas técnicas para remoção de ruídos, pode-se citar as operações morfológicas: erosão, dilatação, abertura e fechamento. Essas técnicas visam alterar as intensidades de cores da imagem, permitindo atenuar os ruídos e manter o texto intacto para extração. Na Figura 16, pode-se visualizar como cada operação morfológica atua sobre a imagem.

Figura 16: Operações morfológicas. No canto superior direito, a imagem original. Da direita para a esquerda e cima para baixo as operações de erosão, dilatação, abertura e fechamento.



Fonte: [https://docs.opencv.org/trunk/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html)

### 3.2.3 Segmentação de caracteres

Nessa etapa, a imagem é varrida de forma a separar todos os caracteres presentes, além de respeitar o posicionamento nas linhas e o espaçamento para diferenciar o intervalo entre duas palavras, por exemplo. A Figura 17 ilustra o resultado da segmentação de caracteres em uma imagem de placa de veículo.

Figura 17: Segmentação de caracteres em uma imagem.



Fonte:

[https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/a/Automatic\\_number\\_plate\\_recognition.htm](https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/a/Automatic_number_plate_recognition.htm)

### 3.2.4 Extração de Características

Após separar todos os possíveis caracteres da imagem original, conforme visto na Figura 17, devem-se extrair as características de cada caractere obtido na etapa anterior. Essas características representam os traços e curvas que identificam o caractere analisado. Por exemplo, na figura 17, o caractere 4 poderia resultar em 3 segmentos de reta, sendo dois deles com um vértice comum e o terceiro cruzando verticalmente o segmento horizontal. Nessa etapa, cada caractere começa a ser quantificado para auxiliar a próxima fase, a classificação de caracteres.

### 3.2.5 Classificação de Caracteres

Uma vez quantificadas as características distintas de cada caractere, resta atribuir uma classe a cada caractere. Uma classe corresponde a um caractere esperado, sendo assim ela possui um conjunto de valores associados para facilitar a identificação. Por exemplo, uma imagem com 3 segmentos de reta poderia representar o número 4 ou a letra H, mas nunca a letra C. Assim, é possível mapear as características da imagem obtida, verificando qual representação de caractere melhor se enquadra para o tipo específico de imagem. Pode-se perceber o quanto complicado é reconhecer escritas feitas à mão olhando essa etapa. Cada maneira de escrever gera um conjunto de características distintas, o que dificulta o processo de

classificação desse caractere. A única forma de se obter sucesso é através de padrões previamente estudados que ajudem a classificar um caractere com mais acurácia.

### 3.2.6 Pós-processamento

Mesmo depois que os caracteres foram classificados, não se pode garantir que o sistema de OCR possua uma acurácia alta o suficiente para fornecer resultados consistentes. Portanto, para assegurar essa eficiência aplicam-se mais filtros de classificação e outras técnicas relacionadas a linguagem em questão, tais como análise de contexto, processamento léxico e uso de dicionários.

### 3.2.7 Utilização de OCR no modelo de referência de IoT

Para ilustrar a utilização da tecnologia de OCR no modelo de referência para aplicação de IoT, descrito na Seção 3.1.5, consideremos o uso de uma câmera fotográfica. Esta estaria representada na primeira camada do modelo de referência. Para que o processo do OCR seja aplicado, a imagem captada pela câmera teria de ser transmitida (camada 2), armazenada e por fim tratada (camada 3), de acordo com as etapas do processo de OCR. A Figura 18 ilustra estas etapas.

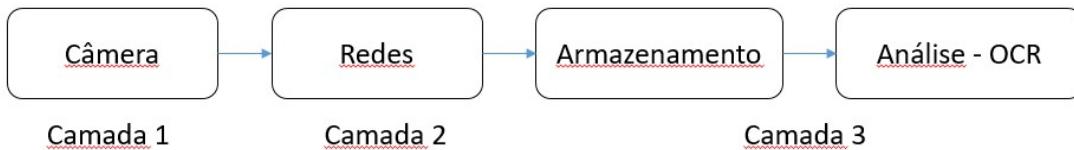


Figura 18: Utilização de OCR no modelo de referência de IoT. Fonte: Autor.

## 3.3 Inteligência Artificial (IA)

Existem várias definições para o que seja Inteligência Artificial ou IA. De acordo com [23], o campo de IA lida com métodos que realizam tarefas que o ser humano costuma realizar e que envolvam o uso de cognição. Dessa forma, quando se desenvolve um programa de IA, pode-se afirmar que ele precisa armazenar conhecimento, para aplicá-lo em uma resolução de problemas e possivelmente alterá-lo, através da adquisição de novo conhecimento fornecido pela experiência.

Ainda segundo [23], existem 3 componentes fundamentais em um programa de IA: (i) Representação de conhecimento, (ii) Raciocínio e (iii) Aprendizado.

A representação de conhecimento utiliza uma linguagem para armezá-lo. O uso de uma representação símbólica, por exemplo através de linguagens lógicas, facilita a sua compreensão para a resolução de um problema determinado.

O raciocínio consiste em o conjunto de métodos para resolver problemas. Segundo [23], um programa de IA que possui raciocínio precisa ser capaz de monitorar e controlar quais ações devem ser executadas para resolver o problema em questão.

A partir dessa visão, pode-se perceber que a resolução de um problema em IA se resume a um problema de busca, onde o programa de IA precisa representar as diferentes possibilidades e escolher convenientemente uma delas, para atingir resultado desejado. Um exemplo que ilustra o problema de busca é o problema do caixeiro viajante, onde deseja-se achar o caminho de menor custo que passa por um conjunto de cidades as cidades.

Por fim, a aprendizado, ilustrado na Figura 19, envolve uma série de etapas que possibilitam que um programa aperfeiçoe seu desempenho com a experiência.

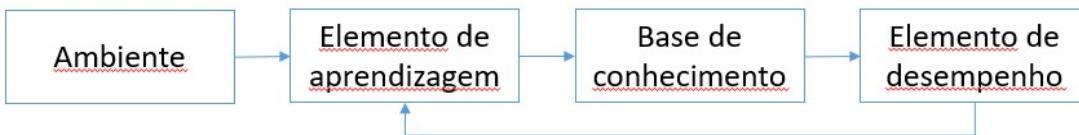


Figura 19 : Modelo de aprendizado de máquina [23]

Após realizar uma leitura do ambiente e guardar as informações relevantes em sua base de conhecimento, o programa define quais ações ele deve seguir através de seu raciocínio, e com base nos resultados obtidos, um elemento de desempenho atua como um mediador, ou supervisor, que indica se as decisões tomadas com base no raciocínio resolvem o problema em questão, ou se foram mais eficientes. Através de um laço de realimentação, esta avaliação altera o elemento de aprendizagem nas próximas resoluções de problemas.

### 3.3.1 Aprendizado de Máquina

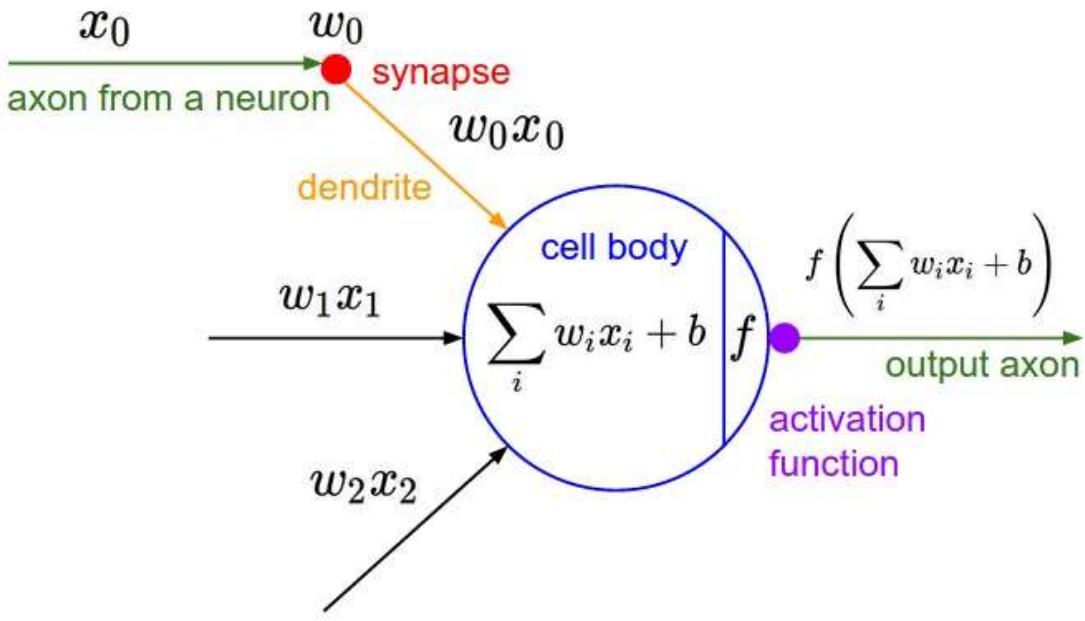
A Figura 19 ilustra como se dá o processo de aprendizado de forma simples. Contudo, existem formas variadas de aprendizado para um programa de IA: (i) Aprendizado supervisionado, (ii) Aprendizado não supervisionado e (iii) Aprendizado por reforço. No aprendizado supervisionado, sabe-se de antemão quais comportamentos de saída são esperados, sendo então possível compará-los com as saídas do programa. Dessa forma, pode-se estabelecer uma medida de erro, representado pela diferença entre a saída esperada e a efetivamente obtida. Este erro serve para ajustar o modelo na forma de retroalimentação, resultando no reajuste do modelo para minimizar o erro existente..o ajuste do modelo é feito numa primeira fase, denominada fase de treinamento. Uma vez treinado, o programa é utilizado numa fase de resolução (recall em inglês) quando efetivamente pode ser utilizado para solucionar problemas. Já no aprendizado não-supervisionado, a saída esperada não é conhecida a priori. Um exemplo é a tarefa de agrupamento de dados, por exemplo num levantamento para oferecimento de crédito bancário.. Não existe uma definição inicial de quantos e quais as características dos grupos desejados. Os algoritmos utilizam técnicas matemáticas que permitem separar os dados segundo um padrão definido durante a execução do algoritmo. Ao final, se espera ter grupos distintos de dados, onde dados de um grupo possuem alguma afinidade com outros integrante do grupo; por exemplo, contas bancárias de funcionários públicos que obtiveram crédito nos últimos 5 anos. Finalmente, o aprendizado por reforço é semelhante ao aprendizado supervisionado, com a diferença que não se sabe exatamente o comportamento esperado da saída, mas apenas uma avaliação se a saída está convergindo para uma direção correta ou não. Os algoritmos então aplicam uma bonificação ou uma penalidade, conforme o caso, para cada saída gerada. Quando mais próximo do objetivo esperado, mais bonificações são recebidas e vice-versa. O algoritmo procura uma solução que maximize as bonificações. Entretanto, pode-se levar em consideração também a quantidade de passos até atingir tal objetivo. Portanto, o problema de otimização consiste em maximizar as bonificações e minimizar o número de passos envolvidos.

### 3.3.2 Redes Neurais

Dentro do campo de IA, as redes neurais representam uma forma de aprendizado supervisionado,.

A inspiração para o termo redes neurais veio da biologia, onde os neurônios são conectados entre si formando uma densa rede. A capacidade do cérebro de adquirir conhecimento e aprendizado levou a uma modelagem matemática de um neurônio artificial, conforme a Figura 20.

Figura 17: Modelo matemático de um neurônio.



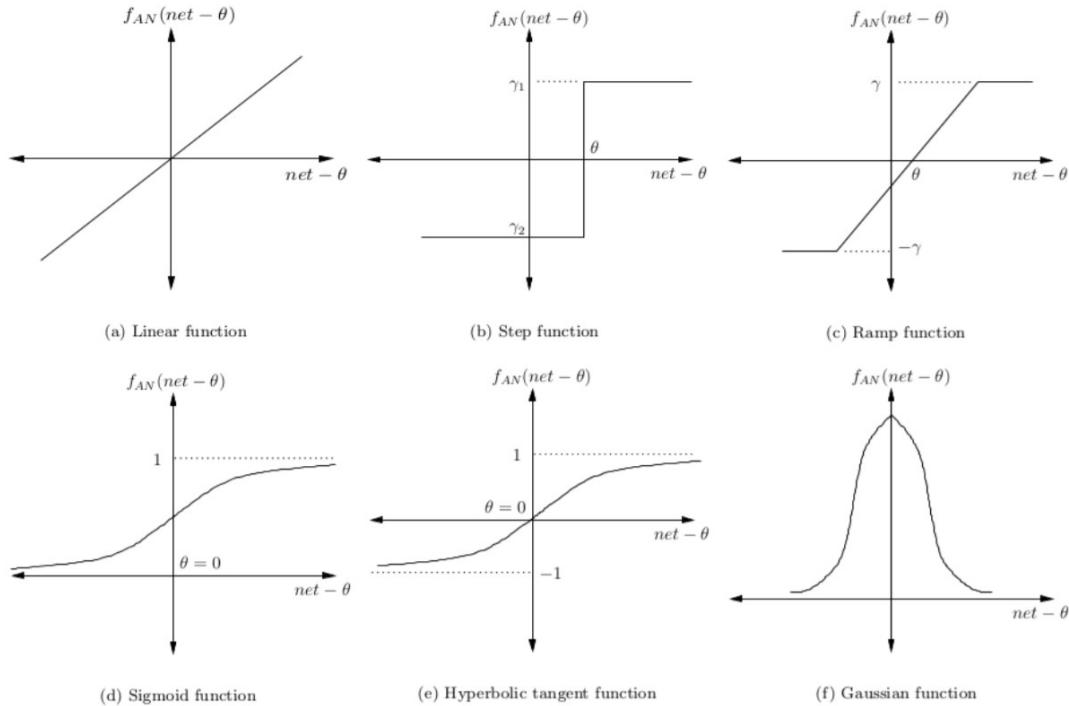
Fonte: <http://cs231n.github.io/neural-networks-1/#intro>

Observando a Figura 20, é possível destacar componentes importantes do neurônio, como as entradas (vetor  $x$ ), os pesos sinápticos ( $w$ ), a função de ativação ( $f$ ) e a saída ( $y$ ). A soma ponderada das entradas com os pesos sinápticos excitam o neurônio, através de uma função de ativação não linear. Isso significa que, dependendo do peso estabelecido, as entradas podem ser completamente suprimidas (multiplicação por zero). A saída do neurônio tipicamente tem dois valores: excitado (ativo) ou não excitado (inativo).

Assim como na biologia, o neurônio possui um limiar de excitação, ou seja, dependendo do estímulo aplicado a saída tem um comportamento diferente. A função

de ativação possui comportamento não linear e similar; após computar todas as entradas ponderadas a função de ativação usa esse valor para gerar uma saída. A Figura 21 ilustra alguns exemplos de funções de ativação.

Figura 21: Funções de ativação.



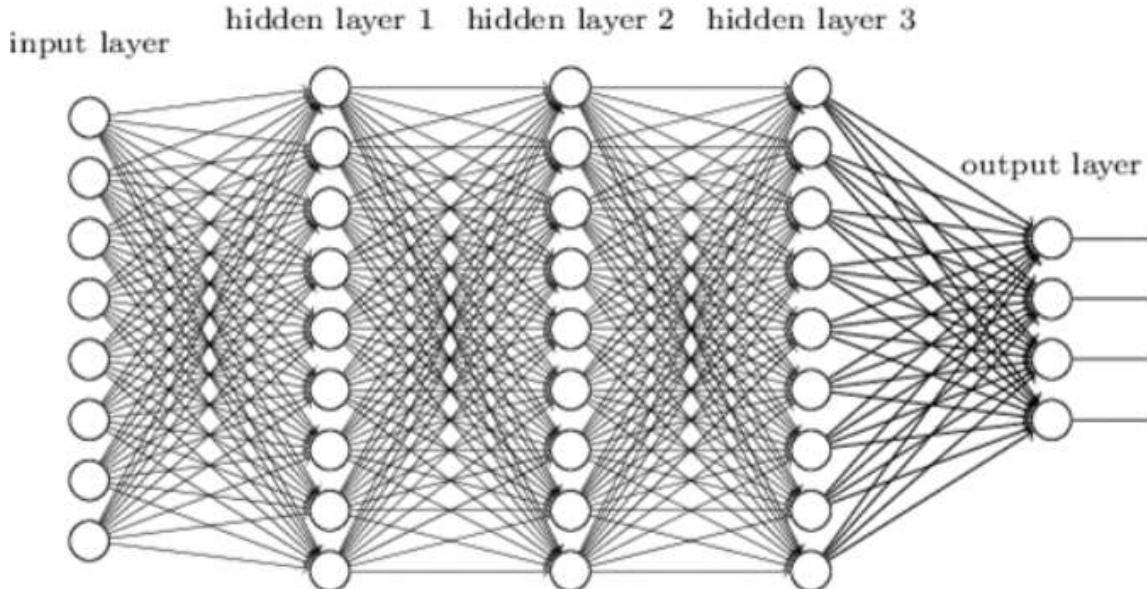
Fonte: <http://www.turingfinance.com/misconceptions-about-neural-networks/>

Dependendo do tipo de função escolhida para o neurônio, o comportamento da rede neural é afetado, visto que em alguns casos os valores de entradas podem ser suprimidos ou não pela função de ativação escolhida.

### 3.3.2.1 Rede neural com múltiplas camadas

Uma forma simples de como se modela uma rede neural pode ser observada na Figura 22.

Figura 22: Rede Neural com várias camadas.



Fonte: <http://houseofbots.com/news-detail/1442-1-what-is-deep-learning-and-neural-network>

A disposição dos neurônios é feita em linha para cada camada. Quando uma rede neural possui camadas entre a entrada (input) e a saída (output), essas camadas são chamadas de camadas ocultas (hidden). Caso uma rede neural possua mais de uma camada oculta, ela pode ser chamada de rede neural profunda.

Do ponto de vista matemático, cada saída de uma camada oculta representa o valor da função de ativação aplicado às entradas correspondentes. Essa saída, por sua vez, é entrada para outra camada oculta, que irá ter outra função de ativação. Na saída final da rede neural, o que se pode observar é uma composição de funções de ativação representando toda a rede neural matematicamente. A fórmula abaixo mostra como a rede neural da Figura 22 pode ser representada matematicamente:

$$y = f(\sum w * f^3(\sum w^3 * f^2(\sum w^2 * f^1(\sum w^1 * x + b^1) + b^2) + b^3) + b)$$

### 3.3.2.2 Aprendizado de uma Rede Neural

Por se tratar de um modelo de aprendizado supervisionado, a saída produzida por uma rede neural é comparada com uma saída previamente definida. A diferença entre

essas duas saídas será o complemento para refinar o raciocínio da rede neural, melhorando os resultados do aprendizado.

Para garantir que o aprendizado esteja convergindo para a solução desejada, é necessário aplicar uma forma de quantificar a proximidade da solução desejada. Essa forma pode se chamar de função erro, mas existem outros nomes também como custo, perda e distância. Em todos os casos o objetivo é oferecer uma medida para qualificar o aprendizado para a resolução de um problema.

Segundo [21], alguns exemplos de funções erro são:

- (i) Erro quadrático médio
- (ii) Erro quadrático médio logarítmico
- (iii) Norma L2
- (iv) Erro absoluto médio
- (v) Entropia cruzada
- (vi) Proximidade de cossenos

Dessa forma, o modelo de aprendizado de uma rede neural pode ser representado como mostra a Figura 23.

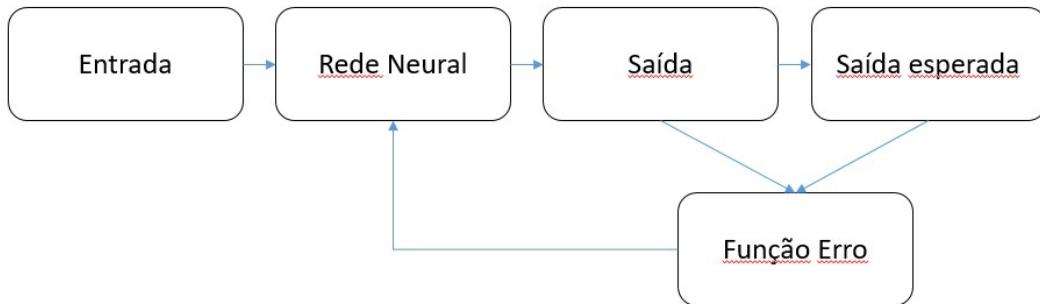


Figura 23: Rede Neural e aprendizado. Fonte Autor.

Conforme visto anteriormente, a saída produzida pela rede neural é uma composição de funções. Essa, junto com a saída esperada, que é um valor fixo da mesma dimensão da saída da rede neural, servem de entrada para a função erro, que quantifica o progresso do aprendizado.

Na Figura 23, é possível notar uma ligação da função erro com a rede neural, o que sugere que a saída dessa função atua como entrada na rede neural. Entretanto essa ligação apenas representa que a função erro vai atuar sobre os pesos sinápticos.

Percebendo que a função erro contém o modelo da rede neural com seus parâmetros, o processo de aprendizado pode ser definido como a redução do erro entre as saídas da rede neural e a esperada. Quando o erro se reduz a zero, pode-se afirmar que a saída da rede neural é igual a esperada.

Do ponto de vista matemático, o processo de aprendizado em uma rede neural pode ser considerado um problema de otimização de função, buscando minimizar o erro.

Durante o processo de otimização da função erro, os pesos sinápticos da rede neural são alterados para que ela produza saídas diferentes, justamente buscando a caminho da otimização. A fórmula a seguir ilustra como os pesos são atualizados:

$$\mathbf{w} = \mathbf{w}_i - \eta * \frac{dL}{dW}$$

Onde o termo  $\mathbf{w}_i$  corresponde ao valor do peso no estado anterior e o termo  $\eta$  se chama taxa de aprendizado.

Ao final da otimização, a função erro (L na figura) converge a zero e as atualizações se encerram, resultando em um aprendizado correto.

### 3.3.3 Utilização de Redes Neurais em OCR

Dentro da estrutura de processos de um sistema de OCR, a etapa de classificação de caracteres consiste em dizer qual caractere corresponde àquele que foi segmentado na etapa anterior. Esse problema é chamado de classificação, ou também de reconhecimento de padrões.

Esses tipos de problemas se enquadram em um conjunto de problemas que as redes neurais têm capacidade para resolver.

Um exemplo de caso de uso é a ferramenta Google Tesseract (<https://github.com/tesseract-ocr>), que tem como foco realizar OCR para imagens em diversas línguas. Todas as etapas, exceto a de processamento de imagem, são aplicadas pelo Tesseract e durante a classificação dos caracteres, a ferramenta faz uso de um modelo específico de rede neural, chamado de LSTM (*Long Short Term Memory*).

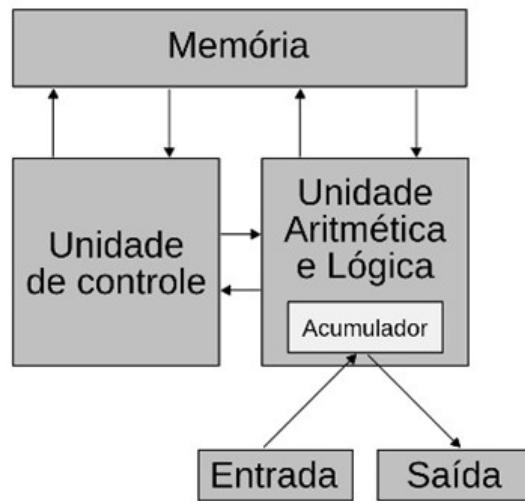
### 3.4 Computação de Alto Desempenho (HPC)

Este termo se refere às grandes infraestruturas de computação voltadas ao processamento de grandes quantidades de dados. Essas infraestruturas são dotadas de equipamentos de última geração, contendo grandes quantidades de processadores, memória e espaço de armazenamento em disco. Essas infraestruturas podem ser chamadas de *Data Centers*, e seu grande objetivo é processar grandes volumes de dados da forma mais rápida possível.

#### 3.4.1 Arquitetura de Von Neumann

A execução de um programa de computador pode ser dividida em operações mínimas, chamadas de micro-operações. Esta é a unidade fundamental para processar um dado. Todo e qualquer computador, seja pessoal ou até mesmo em um *Data Center*, segue a mesma arquitetura fundamental, a chamada arquitetura de Von Neumann, conforme mostra a Figura 24.

Figura 24: Arquitetura de Von Neumann.



Fonte: <https://www.slideshare.net/flaviogomesdasilvalisboa/o-que-e-programao-de-computadores>

Um programa fica armazenado na memória do computador. Durante a sua execução, a Unidade de Processamento Central, irá acessar a unidade de memória para buscar, decodificar e executar a próxima instrução. Tal processo é realizado iterativamente

até finalizar o programa. Tipicamente, as primeiras instruções envolvem ler entradas (por exemplo, operações de teclado ou mouse) e as últimas escrever em saídas (por exemplo, no *display* ou em uma porta de comunicação).

Entretanto, tal padrão de funcionamento pode ser um gargalo quando se deseja solucionar um problema complexo, onde diversas operações são demandadas quase que ao mesmo tempo. Segundo [1] pode-se citar como exemplos de alta complexidade os sistemas de previsão (clima, financeiro) e simulação (processos, fenômenos).

### 3.4.2 Formas de Execução de um programa

De maneira básica, o fluxo de execução de uma micro operação descrito acima se divide em 5 etapas: Buscar (*Fetch*), Decodificar (*Decode*), Executar (*Execute*), Acessar a memória (*Access*) e Escrever (*Write back*).

A etapa de Buscar corresponde ao acesso à memória pela CPU, em busca da localização da instrução do programa que deve ser executado. A memória possui posições numeradas, analogamente a uma rua com cada casa numerada.

Na etapa de Decodificar, a CPU encontrou a instrução do programa e precisa interpretar as informações que estão ali presentes.

A etapa de Execução do programa representa a parte importante, quando todas as instruções para analisar e processar os dados são realizadas. Durante essa etapa, a CPU irá realizar vários acessos na memória, todos de acordo com a localização feita na primeira etapa.

Após finalizar a execução do programa, a CPU faz o acesso na memória para escrever os resultados da execução do programa, servindo como localização para futuras execuções. Pode-se perceber que essas etapas formam um ciclo de instruções, chave fundamental para qualquer programa que venha a ser executado. A Figura 25 mostra como são executados 3 programas.

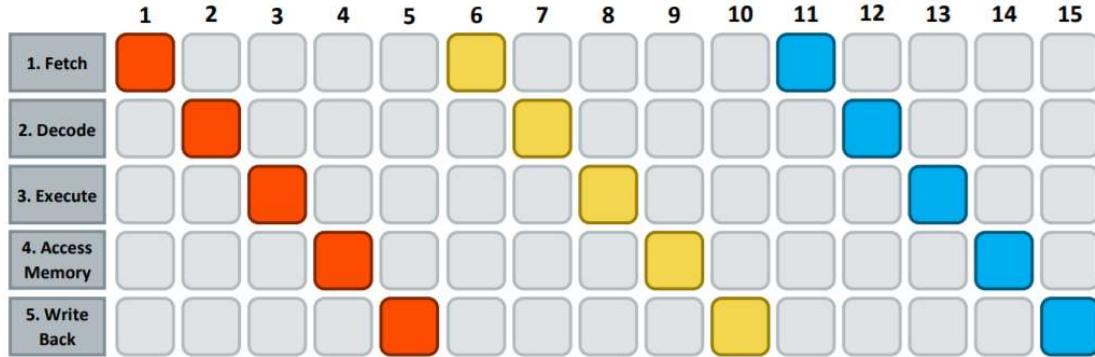


Figura 25: Ciclo de instruções para 3 programas em série [12].

Pode-se perceber que existe uma dependência sequencial entre as instruções, de forma que não podemos realizar a etapa de *Decode* antes da etapa de *Fetch*. Caso esta forma de execução sequencial fosse aplicada para o processamento de uma rede de sensores sem fio, provavelmente o tempo de resposta não seria adequado. Em ambientes críticos, onde o tempo de resposta é o requisito principal para segurança, essa forma estaria fadada ao fracasso.

Para melhorar o tempo de execução deve-se buscar uma forma de evitar o tempo ocioso de um programa. A Figura 26 mostra uma outra sugestão, adicionando paralelismo na execução.

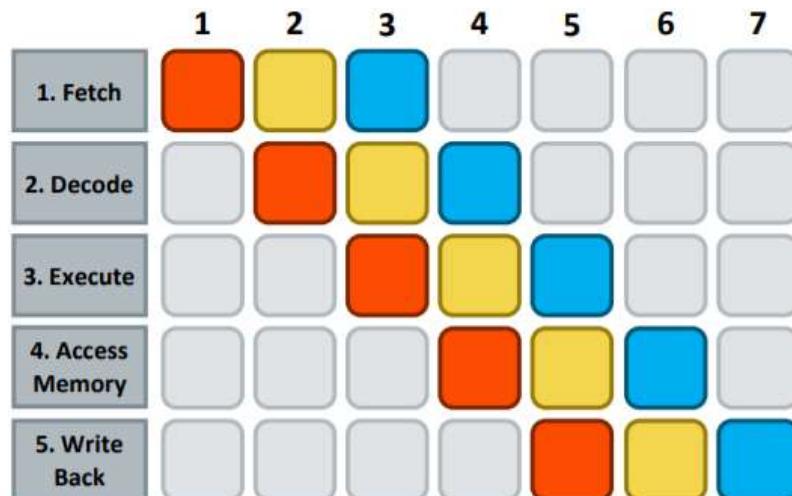


Figura 26: Ciclo de instruções para 3 programas em paralelo [12].

Neste caso, a dependência entre as etapas para um mesmo programa é mantida. Entretanto, vários programas podem ser executados simultaneamente. Por exemplo, enquanto um programa A (vermelho) está executando, o programa B (amarelo) está decodificando e o programa C (azul) está buscando a próxima instrução. O paralelismo aplicado reduziu o tempo de execução final, permitindo novas execuções dentro do tempo ocioso ilustrado na Figura 25.

Além dessa técnica, existe um vasto conjunto de opções para garantir que as execuções de programas sejam cada vez mais paralelizadas e em menor tempo. De acordo com [9], o consumo energético dentro de um data center é um dos grandes desafios para quem possui tal infraestrutura. Quanto mais tempo computando instruções, mais energia é gasta, e por isso a utilização de HPC contribui para que as execuções sejam realizadas consumindo a menor quantidade de memória e no menor tempo possível.

Outra preocupação é o uso de memória para programas de análise de dados. Devido ao grande volume de dados a serem processados, é comum que muitas variáveis precisem ser armazenadas para auxiliar outras execuções. Por conta disso, pode-se perceber que a necessidade de usar mais a memória também está aliado com o tempo gasto para acesso a esta memória.

Por conta disso, outro fator a ser considerado, junto com o consumo energético e de memória, é o tamanho da banda de acesso à memória. Este fator vai informar as limitações de tempo para acesso em memória, mostrando que uma banda elevada garante acesso mais rápido, por mais que seja necessário um volume de acessos cada vez maior.

Portanto, pode-se perceber que a utilização de técnicas de HPC busca otimizar o consumo de recursos e tempo em um *Data Center*, garantindo assim que uma aplicação do tipo IoT, por exemplo, possa adquirir mais dados via uma rede de sensores, e realizar a análise de dados obtidos para determinação da tomada de decisão.

### 3.4.3 Utilização de HPC em OCR

Retomando o exemplo citado na Seção 3.2.7, durante a fase de análise da imagem obtida por câmeras na camada 1 do modelo de IoT, cada etapa do OCR aplica algoritmos para processar a imagem. A imagem é tratada como uma matriz de duas

dimensões, onde cada posição possui um valor de intensidade da cor da imagem. Essa intensidade segue o padrão RGB (*Red, Green, Blue*), com valores entre 0 até 255.

Cada etapa do OCR aplica uma série de operações matemáticas sobre a matriz de entrada, que é a imagem. Do ponto de vista computacional, uma operação sobre uma matriz bidimensional de tamanho N por M, leva no pior dos casos,  $N \times M$  passos para se completar o processamento. Por isso, dependendo da quantidade de imagens, do tamanho da imagem e sua qualidade, o custo computacional de análise se eleva, podendo comprometer a cadeia de execução inteira da solução de IoT.

Dessa forma, as técnicas de HPC permitem tornar a execução destas tarefas mais rápida, otimizada o uso da arquitetura do computador. Como foi descrito anteriormente, é possível realizar a execução de um programa de duas formas: serial e paralela. A segunda é a mais apropriada para o caso de OCR, visto que o processamento é feito sobre conjunto de dados de grande volume, e que se a matriz for dividida em pequenas partes, cada parte não é dependente da outra, o que permite a aplicação paralelizada do processamento.

Suponha uma matriz NxN; nessa matriz será aplicado um processamento em cada célula (i,j) da matriz. Se a execução é serial, cada célula (i,j) será processada uma a uma, de cada vez, totalizando  $N \times N$  passos. Caso a execução seja paralela, é possível executar um número K de células por vez, com  $K > 1$ , reduzindo o total de passos para  $(N \times N)/K$ .

Por fim, o uso de HPC garante menor tempo de execução do processamento de OCR, além de utilizar a arquitetura interna de forma mais otimizada, contribuindo para melhor consumo de memória e menos energia.

## 4. SOLUÇÃO IoT PARA DETECCÃO DE INFRAÇÕES DE TRÂNSITO

Uma vez introduzida a problemática do domínio de aplicação no Capítulo 2 e os conceitos básicos das principais tecnologias associadas no Capítulo 3, apresenta-se agora uma proposta de solução IoT em um contexto de deteção de infrações de trânsito, que utiliza a tecnologia de reconhecimento de caracteres ópticos (OCR) para extrair texto a partir de imagens de placas de veículos obtidas por uma rede de câmeras. Discute-se também os principais desafios que seriam encontrados para a implementação da proposta.

### 4.1 Proposta de solução IoT

Uma solução IoT que utiliza a tecnologia OCR deve necessariamente realizar análise e processamento de imagens. O processamento de imagens, como detalhado na Seção 3.2, envolve diversas etapas, desde o tratamento para extração do máximo de características da imagem, até a extração de informações que possibilitem identificar um caractere na imagem. A análise consiste em utilizar um algoritmo de inteligência artificial, como os discutidos na Seção 3.3.1, para identificar e classificar corretamente os caracteres contidos na imagem; em nosso caso particular, para detetar a identificação do veículo infrator.

A seguir, será exemplificada uma solução para este problema, fazendo uma correspondência com o modelo de referência para aplicação de IoT apresentado na Seção 3.1.5.

Os sensores que servirão de base para esta solução serão radares utilizando detectores fixos, conforme descrito no capítulo 2. Serão distribuídos em diversos pontos da via pública, formando uma rede de radares com detectores fixos. Junto aos radares, existirão câmeras para capturar imagens da identificação dos infratores. A Figura 24 ilustra uma forma de representar essa topologia de rede.

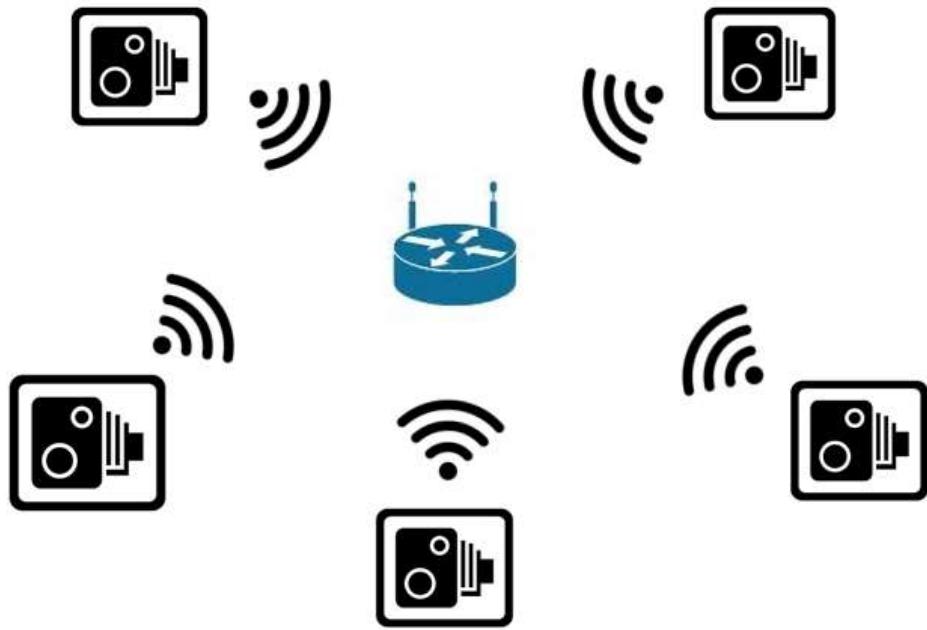


Figura 27: Rede de câmeras sem fio, conectadas a um ponto de acesso central.

Fonte: Autor

Considerando o modelo de referência, a camada 1 é formada pelos sensores que irão detectar que um veículo passou, extraíndo informações como velocidade por exemplo. As câmeras, que estão conectadas aos sensores e ao ponto de acesso, vão obter imagens segundo os dados informados pelos sensores.

Os pontos de acesso formam a camada 2, estabelecendo uma rede que irá interligar as “coisas” com a rede das camadas superiores. Considerando que a aplicação analisa veículos na área urbana em geral, o volume de imagens coletadas é proporcional à quantidade de veículos que passam em um determinado ponto.

As imagens obtidas possuem um grau de qualidade associado. Dependendo desse grau, as informações que serão extraídas podem ser comprometidas ou não, mesmo que se apliquem técnicas sofisticadas de processamento de imagens. Para garantir que os dados sejam obtidos, deve-se estabelecer um grau mínimo de qualidade, sem comprometer o tamanho da imagem que será armazenada para análise. Uma imagem de alta qualidade consome mais espaço de armazenamento, além de maior uso das redes de comunicações. Por isso, a escolha de um grau mínimo de qualidade evita

que os pontos citados anteriormente gerem maiores problemas. A Figura 28 mostra a diferença de graus de qualidade em uma imagem genérica.

Figura 28: Imagens com graus de qualidade diferentes.



Fonte: <http://www.4designweb.com.br/blog/imagem-em-alta-vetores-ai-psd-hein/>

A imagem da esquerda permite que mais informação seja coletada, enquanto que a da direita pode não conter informação suficiente para análise. Vale ressaltar que a aplicação do OCR para extrair as informações depende muito desse grau de qualidade, além dos métodos de processamento. Uma imagem com baixo grau de qualidade pode resultar em uma extração errada de informação, gerando resultados negativos no processo de classificação.

Para transformar as imagens em informações mais padronizadas e simplificadas, é necessário passar pela camada 3. Nela, todo o processamento de OCR pode ser realizado, resultando em um formato de dados mais simples: texto. Esse novo formato pode ser a base para as etapas subsequentes. Nesta primeira abordagem, a maior parte do processo computacional se situa na camada 3.

Entretanto, pode-se utilizar outra abordagem quanto às camadas onde será realizado o processo de OCR. Ao invés de transformar completamente imagens não estruturadas em dados estruturados textuais, pode-se também apenas aplicar uma camada de pré-processamento da imagem, aplicando filtros e padronizando seu tamanho. Dessa forma, as camada 4 e 5 irão armazenar imagens pré-processadas enquanto que o processo de OCR será realizado na camada 6 onde haveria mais esforço computacional envolvido. Analisando-se o modelo de referência, pode-se perceber que o grau de complexidade para a análise de dados se situa na camada 6, enquanto que a camada 3 realiza operações mais simples para formatação dos dados para melhor armazenamento. Mesmo que se use uma base de dados de imagens, já existem formas recentes para armazenar tais tipos de dados, permitindo que o acesso

aos dados não seja feito diretamente ao disco (hard drive ou solid state drive), e sim através da memória, o que permite mais agilidade no acesso aos dados.

Ainda na camada 6, utilizam-se tecnologias de Inteligência Artificial para classificação dos caracteres obtidos das placas na última fase do processo de OCR. Tipicamente, se aplicam técnicas como o aprendizado supervisionado mencionado na Seção 3.3.1. Essas técnicas utilizam essencialmente operações matriciais (multiplicação, soma, transposição), que exigem alto poder computacional devido à presença de grandes loops para realizar esses cálculos. Para isso, utilizam-se métodos que aceleram o processamento, como por exemplo o uso de matrizes esparsas quando se tem muitos zeros numa matriz completa, ou a transposição de matrizes para produto matricial para acelerar a quantidade de operações por ciclo de máquina. Além disso, usam-se também bibliotecas com foco em otimização de hardware, permitindo que o data center (camada 6) seja melhor aproveitado. O uso da computação de alto desempenho é importante nesta fase para que se otimizem os recursos computacionais.

A Tabela 1 resume a correspondência entre o modelo de referência para aplicação de IoT e a solução IoT proposta nesse trabalho:

| Camadas | Modelo de referência para aplicação de IoT | Solução IOT proposta                         |
|---------|--|--|
| 1       | Dispositivos físicos, "coisas"             | Detektore fixos e câmeras                    |
| 2       | Dispositivos de conectividade              | Pontos de acesso                             |
| 3       | Transformação dos dados                    | Pré-processamento de imagem                  |
| 4       | Armazenamento de dados                     | Armazenamento da imagem pré-processada       |
| 5       | Acesso aos dados                           | Acesso as imagens                            |
| 6       | Análise de dados e relatórios              | Aplicação de OCR, utilizando técnicas de HPC |
| 7       | Interação com usuário final                | Interface com o Usuário Final                |

Tabela 1: Correspondência entre modelo de referência de IoT e solução proposta.

Fonte: Autor.

## 4.2 Análise dos Desafios

Tomando como base o modelo de referência modelo de referência para aplicação de IoT apresentado na Seção 3.1.5, pode-se perceber que existe um grau de complexidade atribuído a cada uma das camadas definidas. Essa complexidade, por sua vez, pode se traduzir em um desafio para quem deseja aplicar as recomendações de cada camada.

Na camada 1, o foco está em estabelecer quais dispositivos serão usados para aquisição e comunicação de dados. Os desafios envolvidos são:

- Estabelecer uma rede com dimensões capazes de atender a demanda de informações coletadas pelos sensores,
- Garantir que exista a tradução de protocolos de comunicação entre os dispositivos da rede,
- Possuir meios para gerenciar cada nó integrante da rede, permitindo um controle e monitoramento onipresente.

No cenário para deteção de infrações de trânsito, deve-se identificar qual a quantidade de radares (e câmeras associadas) a ser utilizada, onde serão fisicamente dispostos na via, e como controlar e monitorar se cada um dos dispositivos (radares e câmeras) estão funcionando do modo previsto.

Na camada 2, os equipamentos tradicionais de rede de comunicação são utilizados para preparar a interconexão com o ambiente mais complexo. Seus desafios são os mesmos da camada 1, pelo fato de que o objetivo é garantir a conectividade para transmissão de dados. No cenário para deteção de infrações de trânsito, a instalação de uma rede de comunicação corretamente dimensionada pode ser um grande desafio, especialmente em grandes cidades como o caso de São Paulo.

Os desafios que surgem na camada 3, onde ocorrem os procedimentos de transformação de dados, estão relacionados aos procedimentos que utilizam as tecnologias de OCR, IA e HPC, que como descrito anteriormente necessitam de grande poder computacional. O algoritmo de OCR utiliza técnicas de IA para extração de características da imagem processada, e por conta disso, o uso de processadores, memória e armazenamento são usados com frequência. Para garantir que todo o processamento computacional seja mais eficiente, é importante o uso de técnicas de HPC em cada equipamento presente na camada 3. Sendo assim, pode-se listar os desafios nesta camada:

- Utilizar técnicas de processamento de imagens mais eficientes, tanto em espaço como em tempo de processamento
- Utilização de algoritmos de classificação de caracteres mais rápidos e eficazes
- Utilizar técnicas de HPC para otimizar o uso do poder computacional, partilhando o processamento de vários dispositivos
- Utilizar servidores com alto poder de computação para IA (processadores, memória e armazenamento)

No cenário para deteção de infrações de trânsito, pode-se citar algoritmos especializados em filtrar ruídos e detetar carcteres em placas de veículos. Na camada 4, toda a informação transformada é armazenada, servindo como base de dados para as camadas superiores. Como nenhum procedimento é utilizado nessa camada, existe apenas um desafio a ser tratado:

- Utilizar formas de armazenamento mais eficazes para redução na latência para acesso

No que diz respeito ao hardware, trabalhar com dados armazenados em um formato do tipo HDD (Hard Disk Drive) é mais lento do que utilizar um formato do tipo SSD (Solid State Drive). Especificamente no cenário para deteção de infrações de trânsito, caso se opte por uma alternativa onde na camada 1 seja feita apenas o pré-processamento, deve-se utilizar também gerenciadores e bancos de dados especializados no armazenamento de imagens.

Nas camadas 5 e 6, são realizados procedimentos semelhantes aos da camada 3, visto que são executados mecanismos de análise de dados e geração de relatórios. Apenas tais procedimentos são realizados num nível de abstração distinto. Além disso, necessita-se de novos mecanismos para acessar e controlar as informações, Por fim, a camada 7 serve como uma interface para acessar todos os serviços prestados nas camadas inferiores, e por conta disso seu principal desafio consiste em desenvolver interfaces de acesso ao usuário final. No cenário para deteção de infrações de trânsito, poderia-se pensar em painéis onde o usuário pudesse acessar serviços que identificassem regiões da cidade onde ocorrem mais infrações.

A Tabela 2 apresenta de forma resumida a associação destes desafios a cada uma das camadas do modelo de referência para aplicação de IoT.

| Camadas | Modelo de Referência para aplicação de IoT | Solução IOT proposta                         | Desafios encontrados   |
|---------|--|--|--|
| 1       | Dispositivos físicos, "coisas"             | Detectores fixos e câmeras                   | Dimensionamento de rede para atender a demanda, tradução de protocolos entre dispositivos, gerenciamento de rede                       |
| 2       | Dispositivos de conectividade              | Pontos de acesso                             |  |
| 3       | Transformação dos dados                    | Pré-processamento de imagem                  | Busca por técnicas de processamento de imagens, algoritmos de reconhecimento de caracteres, HPC, hardware com alto poder computacional |
| 4       | Armazenamento de dados                     | Armazenamento da imagem pré-processada       | Novas formas de armazenamento de dados para redução de latência de acesso  |
| 5       | Acesso aos dados                           | Acesso as imagens                            |  |
| 6       | Análise de dados e relatórios              | Aplicação de OCR, utilizando técnicas de HPC | Busca por técnicas de processamento de imagens, algoritmos de reconhecimento de caracteres, HPC, hardware com alto poder computacional |
| 7       | Interação com usuário final                | Interface com o Usuário Final                | Desenvolvimento de Interfaces para o Usuário Final   |

Tabela 2: Correspondência entre o modelo de referência para aplicação de IoT e os desafios encontrados. Fonte: Autor.

## 5. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo mostrar quais desafios que surgem para o desenvolvimento de uma aplicação de IoT para detecção de infrações de trânsito, que utiliza a tecnologia de reconhecimento de caracteres ópticos (OCR) para extrair texto a partir de imagens de placas de veículos obtidas por uma rede de câmeras. O campo de estudo sobre IoT é multidisciplinar, e por isso mostrar todos os seus principais aspectos é uma tarefa complexa. Por esta razão, nesse trabalho adotou-se como base ilustrar um projeto de IoT através de recomendações sugeridas por um modelo de referência.

Além disso, indicou-se como outras tecnologias são intimamente associadas a um projeto de aplicação de IoT, mais especificamente OCR, IA e HPC. O papel de cada um delas foi indicado, relacionando-as com as camadas do modelo de referência.

Através deste estudo, percebe-se que uma aplicação de IoT oferece novas oportunidades de trabalho e pesquisa. Como se tratam de aplicações tipicamente multidisciplinares, existem oportunidades futuras nas áreas de redes de sensores sem fio, protocolos de rede, redes de comunicação de dados, armazenamento de dados, análise de dados, processamento de imagens, inteligência artificial, e análise de desempenho.

## REFERÊNCIAS

- [1] COUTINHO, L. C. M.; COELHO, M. H.; POZZEBON, E. **Agentes Inteligentes Conversacionais**: Conceitos Básicos e Desenvolvimento de uma Aplicação. *Anais SULCOMP*, Araranguá - SC - Brazil, 16 Fevereiro 2017. 2.
- [2] BATISTA, G. E. D. A. P. A. **Pré-processamento de Dados em Aprendizado de Máquina Supervisionado**. *Instituto de Ciências Matemáticas e de Computação - USP*. São Carlos, SP - Brazil, p. 232. 2003.
- [3] BERKHIN, P. **A Survey of Clustering Data Mining Techniques**. 2nd *International Conference on Computing for Sustainable Global Development*. New Delhi: [s.n.]. 4 Maio 2015. p. 47.
- [4] DA SILVA, G. C. **Detecção e Contagem de Plantas utilizando técnicas de Inteligência Artificial**. *Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina*. Florianópolis, p. 94. 2017.
- [5] SHANCANG, L.; XU, L. D.; SHANSHAN, Z. **5G Internet of Things**: A survey. *Journal of Industrial Information Integration*, p. 3, Março 2018.
- [6] BORGES, A. **Uma Arquitetura para Internet das Coisas para Análise da Concentração de Monóxido de Carbono na Grande São Paulo por meio de técnicas de Big Data**. *Engenharia Elétrica - Universidade Presbiteriana Mackenzie*. São Paulo, p. 105. 2017.
- [7] AZEVEDO, M. M. G. O. D. **Análise do Desempenho de Métodos de Inteligência Artificial Baseados no Comportamento das Plantas**. Coordenação do Curso de Pós-graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão. São Luís - Maranhão, p. 80. 2017.
- [8] MESQUITA, V. H. L.; HEO, M. M. D.; SILVA, B. S. **Visualização de Progressos de Otimização em Implementações para Alto Desempenho**. *IBM Research Brazil*. São Paulo, p. 4. 2016.
- [9] BOYLE, P. et al. **Accelerating HPC Codes on Intel Omni-Path Architecture networks**: From particle physics to Machine Learning. *Cornell University Library*. [S.I.], p. 17. 2017.
- [10] CALORE, E. et al. **Energy-efficiency evaluation of Intel KNL for HPC Workloads**. *University of Ferrara*, Itália. Ferrara, p. 11. 2018.
- [11] BAO, Y.-G.; WANG, S. **Labeled von Neumann Architecture for Software-Defined Cloud**. *University of Chinese Academy of Sciences*. Beijing - China, p. 15. 2017.
- [12] INTEL CORPORATION. **Intel Vtune Amplifier Tuning Guide for the Intel Xeon Processor Scalable Family, 1st Gen**. *Intel Corporation - United States*. [S.I.], p. 15. 2017.

- [13] BHUVANESWARI, V.; PORKODI, R. **The Internet of Things (IoT) Applications and Communication Enabling Technology Standards**: an Overview. *International Conference on Intelligent Computing Applications*. Cluj-Napoca: [s.n.]. 2014. p. 6.
- [14] IHS MARKIT. IoT Trend Watch 2018. **IHS Markit, 2018**. Disponível em: <<https://ihsmarkit.com/Info/0118/iot-trends-ebook.html>>. Acesso em: 20 Agosto 2018.
- [15] PREETHI, N. **Performance Evaluation of IoT Result for Machine Learning**. *Transactions on Engineering and Sciences, Department of Computer Applications, Jain University*, Bangalore, v. 2, n. 11, p. 4, Novembro 2014.
- RUILONG, D. et al. Toward Power Consumption-Delay Tradeoff by Workload Allocation in Cloud-Fog Computing. *IEEE International Conference on Communications - Mobile and Wireless Networking Symposium*. Londres: [s.n.]. Junho 2015. p. 5.
- [17] JAYAVARDHANA, G. et al. **Internet of Things (IoT)**: A Vision, Architectural Elements, and Future Directions. *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*. Palladam, India: [s.n.]. 5 Outubro 2017. p. 19.
- [18] STALLINGS, W. **Foundaments of Modern Networking**. [S.I.]: Addison Wesley, 2015.
- [19] CISCO. **Building the Internet of Things**. Cisco. [S.I.], p. 26. 2014.
- [20] SANCHES, C. H. et al. **Técnicas de Suavização de Imagens e Eliminação de Ruídos**. *Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação*, Frederico Westphalen - RS, p. 21-30, Novembro 2015.
- [21] VON ZUBEN, F.. **Introdução às Redes Neurais** (Parte 1). *Faculdade de Engenharia Elétrica e Computação*, Unicamp. Campinas - SP, p. 14. 2008.
- [22] CHANGHUAU, I. **Loss Functions in Artificial Neural Networks**. Agency of Science, Technology and Research. Singapura, p. 9. 2018.
- [23] HAYKIN, S. **Redes Neurais**: Princípios e Prática. 2a. ed. [S.I.]: Bookman.
- [24] ISLAM, N.; ISLAM, Z.; NOOR, N. **A Survey on Optical Character Recognition System**. *Journal of Information & Communication Technology-JICT*, Malásia, v. 10, n. 2, p. 4, Dezembro 2016.
- [25] Como funcionam os radares e outros aparelhos de fiscalização eletrônica? **Flatout**. Disponível em: <<https://www.flatout.com.br/como-funcionam-os-radares-e-outros-aparelhos-de-fiscalizacao-eletronica/>>. Acesso em: 26 Agosto 2018.
- [26] BERNARDI, E. **Os sistemas de identificação veicular, em especial o reconhecimento automático de placas**. Escola Politécnica da Universidade de São Paulo. São Paulo, p. 136, Março 2016.