

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
ENGENHARIA ELÉTRICA

RAVI HELON DE MELO SERAFIM FERREIRA

Relatório de Estágio Supervisionado
LIEC - Laboratório de Instrumentação Eletrônica e Controle

CAMPINA GRANDE
12 de dezembro de 2019

RAVI HELON DE MELO SERAFIM FERREIRA

Relatório de Estágio Supervisionado
LIEC - Laboratório de Instrumentação Eletrônica e Controle

Relatório de Estágio Supervisionado submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Orientador: Rafael Bezerra Correia Lima

CAMPINA GRANDE
12 de dezembro de 2019

RAVI HELON DE MELO SERAFIM FERREIRA

Relatório de Estágio Supervisionado
LIEC - Laboratório de Instrumentação Eletrônica e Controle

Relatório de Estágio Supervisionado submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

APROVADA EM: 10/12/2019

Rafael Bezerra Correia Lima

Orientador
UFCG

George Acioli Júnior

UFCG

CAMPINA GRANDE
12 de dezembro de 2019

Agradecimentos

Entendo que este se trata do espaço reservado a agradecer àqueles que foram de alguma forma essenciais na conclusão do curso ao dedicar o trabalho que marca seu fim a estas pessoas.

A dificuldade é determinar onde se traça a linha entre aqueles que foram essenciais e devem ser mencionados e aqueles que não contribuíram de forma significativa. Aquele desconhecido que eu vi falando sozinho, mas foi algo que mudou minha vida, deve ser mencionado? Talvez seja demais agradecer a essas pessoas, mas eu acredito que tudo o que faz de mim a pessoa que sou e que me ajudou a chegar vivo, saudável e são onde estou agora merecem meu agradecimento, contudo tenho apenas uma página.

Não pode faltar inicialmente o agradecimento à família, por terem contribuído com minha formação enquanto indivíduo e pelo apoio das mais diversas maneiras, com menção especial aos meus irmãos, Radharani, Raphael, Erick e Ananda, aos meus pais, Angela e Eliabe, e meus avós Eunice, José Ferreira e Ana. Também a meus primos Lucas, Tomas e Danielle. Não posso esquecer também dos familiares sem laços sanguíneos, João Pedro França, Gabriel, Moisés e José Adeilmo Junior.

Gostaria também de agradecer a todos os meus amigos e em especial alguns dos que estiveram comigo quando mais precisei, sendo eles Allender Villar, **Amanda Villarim**, Breno, Camila Caroline, Dário Macêdo, **Darlanny Diniz**, Egydio, Elias Gabriel, Erika Rizzo, Giuseppe, João Pedro Souza, **José Adeilmo Nunes Barbosa Junior** (Denovo), Kaio Nikelisson, Lucas Oliveira, Lucas Porto, Márcio Almeida, Matheus Ferreira, Osmar Lucas, Paulo Roberto, **Robson Donato**, Safire Torres, Ulisses Gomes, Victor Hugo, Victor Andrade, Walter Guedes, Weverton e Wislayne.

Impossível esquecer também dos devidos agradecimentos aos professores. Péricles, George Acioli e **Rafael** pela oportunidade de trabalhar no LIEC, Alex, Adenes, Anderson, **Hamurabi**, **Adeilson** e Ribamar pela contribuição para a pessoa que me tornei e Antônio Marcus, Carlos Wilson, Edmar, Epaminondas, **Eustáquio**, Horácio, **Iraponil**, Maurício e **Olindina** por terem contribuído para minha formação, mas também como amigos em diversos momentos. Importante expressar minha gratidão também a Adail e **Tchai** pelo suporte no decorrer do curso.

Por fim acho importante mencionar algumas das figuras históricas que permitiram que o conhecimento chegasse até mim, das quais vou expressar explicitamente apenas algumas das que mais me inspiraram. Sendo elas Fatima al-Fihri, René Descartes, Jostein Gaarder, **Richard Feynman**.

“Let no man pull you so low as to hate him.”
(Martin Luther King Jr., Paul’s Letter to American Christians)

Resumo

Este relatório descreve as atividades e resultados obtidos pelo aluno Ravi Helon de Melo enquanto estagiário no Laboratório de Instrumentação Eletrônica e Controle, no qual o mesmo ficou incumbido de desenvolver um programa capaz de, por meio de técnicas de processamento de imagens e aprendizado de máquina, identificar placas de carros de forma a contribuir para melhoria na fiscalização em vias públicas.

Palavras-chaves: Aprendizado de Máquina, OCR, RNN, Processamento de Imagens

Abstract

This report describes the activities and results achieved by student Ravi Helon de Melo as a trainee at Laboratório de Instrumentação Eletrônica e Controle, in charge of the development of a software capable of using image processing and machine testing techniques to identify license plates and help improve law enforcement on public roads.

Key-words: Machine Learning, OCR, RNN, Image Processing

Lista de abreviaturas e siglas

ONU	Organização das Nações Unidas
RNN	Rede Neural Recorrente
ANN	Rede Neural Artificial
LSTM	Long Short-Term Memory
ReLU	Unidade Linear Retificada
OCR	Reconhecimento Ótico de Caracteres
ALPR	Reconhecimento Automático de Placas de Carro
API	Interface de Programação de Aplicativos

Lista de símbolos

Σ	Somatório
∇	Gradiente
$F(x) _{x=a}$	Função de x com x avaliado no ponto a
\vec{X}	Vetor X
\hat{y}	Valor estimado de y
\tanh	Tangente hiperbólica

Sumário

1	Introdução	10
2	Laboratório de Instrumentação Eletrônica e Controle	12
3	Conceitos de Processamento de Imagens	13
4	Aprendizado de Máquina	15
4.1	Visão Computacional	15
4.1.1	Redes Neurais Artificiais	16
4.1.2	Redes Neurais Recorrentes e LSTM	18
4.2	Tesseract OCR	20
5	Atividades Realizadas	21
5.1	Utilizando Imagem Estática	21
5.2	Utilizando Vídeo	21
5.3	OCR	24
6	Conclusões	26
	Referências	27

1 Introdução

Em Maio de 2011 a ONU lançou a Década de Ação pela Segurança no Trânsito 2011 - 2020 objetivando redução na quantidade de acidentes e mortos no trânsito, que já chega a nona causa de morte no mundo chegando à cifra de 1,25 milhão de pessoas mortas por ano [1].

No Brasil, comparativamente, apesar da quantidade de acidentes ter diminuído bastante nesses últimos 10 anos, morreram mais pessoas em acidentes de trânsito de que na guerra na Síria [2] e uma das maneiras de mitigar essa problemática é melhorar a fiscalização.

Neste contexto o presente documento trata-se do relato das atividades desenvolvidas pelo seu autor enquanto estagiário no laboratório de instrumentação eletrônica e controle, o qual, no exercer tal função, ficou incumbido do desenvolvimento de um software capaz de reconhecer veículos automotivos por meio de vídeo e, a partir desta informação em conjunto com marcações de tempo e localização vindos do próprio dispositivo de gravação, determinar a velocidade média com a qual tal automóvel transitava na via rodoviária em que o equipamento como um todo estiver instalado.

Cumprir esta tarefa é equivalente a, em termos mais direcionados à aplicação prática, reconhecer os automóveis em imagens adquiridas por cameras instaladas nas rodovias, fazer os cálculos de velocidade baseando-se no tempo que determinado veículo é identificado por diferentes equipamentos cuja distância relativa é conhecida e então relacionar com os dados relativos à legislação e limites de velocidade na via correspondente.

Sendo assim as seguintes etapas intermediárias foram traçadas:

1. Localização de placas de identificação veicular;
2. Segmentação da área identificada nas imagens;
3. Aplicação de método de identificação ótica de caracteres;
4. Implementação dos cálculos de velocidade e verificação contra dados legais;
5. Desenvolvimento de interface Humano-Máquina;
6. Integração das funcionalidades anteriormente citadas à plataforma em nuvem;
7. Realização de testes em campo de forma a validar o produto final.

No que diz respeito ao texto em si no capítulo 2 será apresentado o agente de integração. Em seguida, nos capítulos 3 e 4, são apresentadas as principais soluções e dificuldades associadas ao problema a ser resolvido, mais especificamente no que se refere ao reconhecimento de

placas de veículos em sí, Aprendizado de Máquina e Programação em nuvem, respectivamente. Por fim os resultados, assim como as dificuldades encontradas, são apresentados no capítulo 5 e as conclusões decorrentes no capítulo 6.

2 Laboratório de Instrumentação Eletrônica e Controle

O Laboratório de Instrumentação Eletrônica e Controle, cujas instalações podem ser vistas na Figura 2.1, é parte do Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande e é composto por professores altamente qualificados, assim como técnicos e alunos, de graduação, mestrado e doutorado.



Figura 2.1 – Laboratório de Instrumentação Eletrônica e Controle. Fonte: Capturada por um colega.

O referido laboratório dispõe de computadores em ótimas condições para desenvolvimento de software, assim como diversas plantas para estudos em controle e os mais diversos equipamentos eletrônicos, tais como microcontroladores, sensores e atuadores para trabalhos em instrumentação eletrônica e sistemas embarcados.

Os projetos realizados são principalmente nas áreas de instrumentação eletrônica, controle e automação, incluindo trabalhos que envolvem Controle Avançado, Aprendizado de Máquina, Interface Humano-Máquina, Sistemas Embarcados, Modelagem, Identificação e programação em nuvem. O estágio ao qual este trabalho se refere é focado em aprendizado de máquina.

3 Conceitos de Processamento de Imagens

Para o desenvolver do trabalho em questão muitos conceitos teóricos de processamento de imagens são necessários e este capítulo visa abordar algumas definições fundamentais.

Pixels são definidos em [3] como o menor elemento endereçável independentemente cujas cor própria requer uma combinação das três cores primárias. Quanto maior a quantidade destes em uma imagem de tamanho fixo mais difícil de percebê-los visualmente de forma individual, o que leva a uma percepção de uma melhora qualitativa da imagem, representada pela grandeza denominada Resolução, também definido em [3].

Cada pixel tem um valor de vermelho, verde e azul associado ao mesmo de forma a determinar a sua cor. Sendo assim uma forma de armazenar uma imagem digital é na forma de uma matriz tridimensional onde uma dimensão indica a cor enquanto as outras duas determinam a localização do ponto. O valor em determinado ponto significa a intensidade daquela cor naquele ponto.

Vídeos digitais, por sua vez, são feitos pela composição de várias dessas imagens em rápida sucessão de forma a levar a uma percepção de movimento. Cada imagem dessas é um elemento que na computação é intitulado Frame, que é definido em [3] como "uma imagem de muitas conexas que formam um filme". Se cada imagem é uma matriz tridimensional então pode-se representar o frame pela quarta dimensão e assim tratar todos os elementos a serem processados enquanto matrizes.

Em algumas situações a informação que se deseja resgatar na imagem não tem relação alguma com as cores presentes na mesma, como é exemplo os próprios ALPR e OCR, já que o interesse está nos caracteres. Nessas situações pode-se tornar mais prático lidar com imagens em escala de cinza ao invés da imagem colorida, já que aquela tem apenas duas dimensões ao passo que esta é codificada em três, como explicado anteriormente.

Existem diversas técnicas para retirar as cores de uma imagem, tais como as que podem ser vistas em [4] e [5]. O estudo em [6] inclusive faz uma apanhado quanto às diferentes técnicas e o impacto que podem ter no processamento. Neste trabalho, contudo, a conversão para escala de cinza foi utilizada apenas como uma etapa preliminar para a posterior binarização, mencionado por [7] como uma das etapas mais importantes para ALPR.

Binarização torna uma imagem em escala de cinza com múltiplos tons em uma imagem com apenas as cores preta e branca. Diversas técnicas são avaliadas por [8]. As técnicas utilizadas neste trabalho e pelo MATLAB são as descritas em [9], no caso com limiar global, e em

[10] para o caso adaptativo.

Binarização com limiar global é quando é escolhido um limiar, um valor que represente uma intensidade entre 255(Branco) e 0(preto), que será comparado com o restante da imagem. Se o valor de determinado *pixel* for menor que o limiar ele será preto na imagem binarizada, em contrapartida se estiver acima do limiar o *pixel* será configurado como branco. Quando o limiar é adaptativo significa que haverá uma ponderação pelo valor dos seus vizinhos além do próprio limiar global.

Com a imagem binarizada é possível a análise de bordas, caracterizado por quando em dada sequencia um ou mais pixels de um cor é seguido por um pixel da outra cor.

Outro conceito importante é a segmentação é um assunto vastamente estudado na análise de imagens quando aplicado à identificação de objetos, uma vez que nesse gênero de problema existem áreas da imagem que são úteis e que são destacadas ou utilizadas.

Utilizam-se muitos métodos diferentes para cumprir esta tarefa, mas em geral são muito específicas às aplicações em questão e podem demandar muito poder de processamento ou condições especiais para poderem ser utilizados, desta forma é interessante evitar esse tipo de abordagem nos casos onde é possível.

Um último conceito que foi utilizado no trabalho foi a varredura entrelaçada consiste em realizar uma análise em uma imagem, mas, quando uma abordagem ordinária, progressiva, analisaria uma linha seguida de sua imediatamente superior esta abordagem muda esta ordem. Supondo 10 linhas numeradas, ao invés de analisar na ordem 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 seria feita na ordem 1, 5, 9, 2, 4, 6, 8, 10, 3, 7 indo inicialmente de 4 em 4, voltando ao início e indo de 2 em 2, e em seguida de 1 em 1, sem repetir.

Esta técnica foi utilizada em ALPR anteriormente por [11] e pode servir para reduzir o tempo de execução da análise.

4 Aprendizado de Máquina

Aprendizado de máquina é um assunto que vem sendo amplamente discutido em várias mídias, oferecendo por vezes perspectivas que podem parecer tão irreais que escapam à capacidade de serem concebidas. Desde coisas que são apenas impressionantes, como prever, a partir de sinais de vibração, a condição de um motor elétrico até coisas que pensávamos serem exclusividades da espécie humana, como argumentar [12], mentir[13],[14] e contar histórias originais[15][16].

Uma das primeiras definições para aprendizado de máquina foi dada por Arthur Samuel na qual é dito que é o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"[17], contudo essa definição é muito generalista e ajuda pouco no que diz respeito ao entendimento e das técnicas que se utilizam e portanto sejamos mais precisos. Outra definição foi cunhada por Tom M. Michell na qual "diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e performance P, se sua performance P nas tarefas em T, na forma medida por P, melhoram com a experiência E"[18].

Pode-se dizer que se tratam de métodos capazes de transcender aquilo que foi explicitamente programado e exercer funções mais complexas e aqui objetiva-se estudar as capacidades reais de tal ferramenta e, mais especificamente, como pode ser utilizada na solução do problema em mãos.

Tais problemas podem ser categorizados de duas formas diferentes, quanto à forma de aprendizado, podendo ser supervisionado ou não-supervisionado, e quanto ao resultado que se espera, sendo um classificador, um regressor, um detector de anomalias, reconhecimento ótico de caracteres, entre outros. Pertencentes a essas classificações destacam-se também vários métodos, tais quais Redes Neurais, Máquinas de Vetores de Suporte, Árvores de Decisão, etc. No presente trabalho, cujo interesse era o reconhecimento de caracteres, foi utilizada uma rede neural recorrente previamente treinada de código aberto conhecida como Tesseract OCR associada à API da OpenALPR.

4.1 Visão Computacional

A área de visão computacional apesar de não ter surgido como subtópico [19] [20] de aprendizado de máquina, teve avanços tão significativos[21] desde a introdução deste que hoje em dia é quase impensável a dissociação.

Uma das tecnologias associadas a essa área é o reconhecimento ótico de caracteres a qual torna computadores, por sua vez, capazes de identificar texto em imagens e traduzir para

formato de texto.

Existem duas abordagens para lidar com essa tarefa, uma que envolve segmentação[22] dos caracteres para análise de símbolo em símbolo e outra que não precisa dessa etapa[23].

Como o reconhecimento de caracteres em separado é reconhecido como uma tarefa cujos resultados obtidos já são altamente satisfatórios então é na fase de segmentação que ocorrem a grande maioria dos erros encontrados[24], contudo segmentação é uma tarefa relativamente custosa e sendo assim vem sendo favorecidos métodos sem segmentação, já que uma das vantagens de utilizar aprendizado de máquina é poder pular esta fase.

As redes neurais convolucionais são uma opção popular em visão computacional e já foi aplicado com certo sucesso ao problema em mãos[25], porém, assim como as redes neurais recorrentes tradicionais [26], não oferecem resultados comparativamente melhores de que aqueles obtidos pelos métodos com segmentação [24]. Arquiteturas LSTM para as RNN por sua vez vem encontrando relativo sucesso [27] e é o método utilizado na API utilizada na obtenção dos resultados explicitados no Capítulo 5.

4.1.1 Redes Neurais Artificiais

Contudo, antes de tratar das RNN com arquitetura LSTM faz-se necessário um entendimento, mesmo que superficial, das redes neurais artificiais e de aprendizado profundo.

Essas foram introduzidas em 1943 por Warren McCulloch e Walter Pitts[28], um neurofisiologista e um matemático, respectivamente. Obteve relativo sucesso até os anos 60 o que levou à crença de que estaríamos lidando com máquinas realmente inteligentes em um futuro próximo[29].

Porém, devido à capacidade computacional da época, isso se mostrou impraticável naquele primeiro momento e os investimentos não se mantiveram o que as levou a cerca de duas décadas no esquecimento. Nos anos 80 o interesse foi renovado, devido a novas técnicas para se realizar o treino, mas caiu novamente devido ao advento das Máquinas de Vetor de Suporte, que pareciam mais precisas e ter bases teóricas mais sólidas[29].

Recentemente voltaram a crescer em popularidade devido a alguns fatores tais quais a disponibilidade de grande quantidade de dados, computadores com capacidade de processamento muito maior, outras melhorias nos algoritmos de treino e o fato de que algumas das limitações teóricas se mostraram inofensivas na prática[29].

As Redes Neurais são muito versáteis e podem ser utilizadas nas mais variadas aplicações, mas pode-se descrever o problema a ser resolvido de forma a, dadas variáveis de saída, discretas ou contínuas y , categóricas ou não, obter parâmetros x que sejam capazes de minimizar, por meio de uma relação $F(\mathbf{x})$, uma função custo $L(y, F(x))$.

A inspiração do método é no modelo de transmissão de informação no cérebro humano, daí a denominação de Redes **Neurais**. Onde as entradas podem ser visto como os estímulos externos, as quais seriam repassadas a neurônios que enviam informação de como o corpo deve reagir.

Para facilitar o entendimento é comum o emprego de diagramas tais quais o presente na Figura 4.1. A primeira camada, com circunferências amarelas, são as entradas, X_1, X_2, X_m , conectados à seguinte por meio das ponderações, W_1, W_2, W_m . Em verde tem-se a denominada camada oculta e pode ser composta por mais de um nós, ou neurônio, os quais atuam na combinação de suas entradas de acordo com a função de ativação ϕ . Por fim \hat{y} é a saída estimada e y a saída real, se disponível.

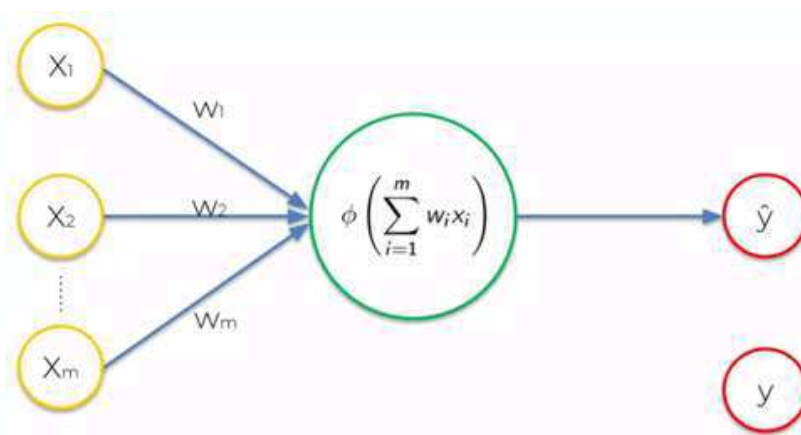


Figura 4.1 – Diagrama Representativo de uma ANN Adaptado de [30]

O funcionamento do método é dividido em três etapas, Treino, Validação e Implantação.

Na etapa de treino objetiva-se determinar primeiramente hiperparâmetros, como quantidade de camadas ocultas, nós por camada oculta, funções de ativação, dentre outros e, posteriormente, os valores dos pesos. O resultado final deve ser um modelo capaz de aproximar a saída por meio das entradas.

Essa etapa se baseia em uma publicação de Donald Hebb na qual é proposto um modelo para o neurônio biológico em que a conexão entre neurônios fica mais forte quando é ativada repetidas vezes. Sendo assim esse treinamento se trata de uma mímica desse processo ao ajustar os pesos iterativamente de forma às interações entre nós que levaram a respostas corretas serem reforçadas em contraponto às que levaram a respostas errôneas.

Este foi por anos um gargalo para o campo até que foi desenvolvido o método do gradiente descendente [31], relevante até hoje, no qual a atualização a cada iteração é baseada no gradiente, ponderado pela taxa de aprendizado η , da função custo C .

Sabe-se que, para uma função $C(\vec{x})$ diferenciável ao redor de um ponto A , a direção para sua mais rápida diminuição é a do negativo do gradiente naquele ponto, ou $-\nabla C(\vec{x})|_{\vec{x}=A}$,

Portanto pode-se escrever a seguinte expressão para atualização dos valores de peso a cada iteração t :

$$w_i^{(t)} = w_i^{(t-1)} - \eta^{(t)} \nabla C(\hat{y} = F(\vec{x}), y)$$

Se uma rede neural é composta por grandes quantidade de camadas ocultas ela é dita de aprendizado profundo. Isso possibilita que diferentes níveis de abstração possam ser aferidos às respostas obtidas após cada camada oculta, contudo existem alguns problemas associados a esses tipos de arquiteturas.

Alguns dos problemas são o tempo extenso necessário para treino e quando o gradiente é propagado para as camadas mais baixas o seu valor vai ficando cada vez mais baixo ou, em alguns casos, cada vez mais alto em efeitos conhecidos como desaparecimento e explosão do gradiente, respectivamente.

Cabe aqui explicitar-se a importância das funções de ativação, as quais representam o processamento da informação pelo cérebro e determinação de o que será repassado às camadas seguintes. São por meio destas que há adição das não linearidades que diferenciam as redes neurais de regressões lineares, mas também é por meio de sua escolha que os problemas de explosão e desaparecimento do gradiente são mitigados, já que estas limitam as saídas de cada nó, possibilitando melhor controle sobre as variáveis que percorrem a rede neural. Normalmente são escolhidas funções sigmóides ou ReLu cujas saídas estão sempre entre 0 e 1.

Após o treino o modelo obtido deve passar por uma avaliação em se utilizando dados de entrada diferentes daqueles usados para o treino, com saídas conhecidas, para verificar os resultados são satisfatórios para a aplicação em questão. Se não forem deve-se repetir a etapa de treino, se forem pode-se partir para a implantação.

4.1.2 Redes Neurais Recorrentes e LSTM

Redes neurais recorrentes são semelhantes às redes neurais comuns, com o adendo de que as saídas de um nó serão realimentados de forma influenciar na próxima etapa e fazendo com que esta configuração leve em consideração o passado do sistema ao fazer suas estimativas. Uma ilustração dessa configuração pode ser visto no diagrama na Figura 4.2

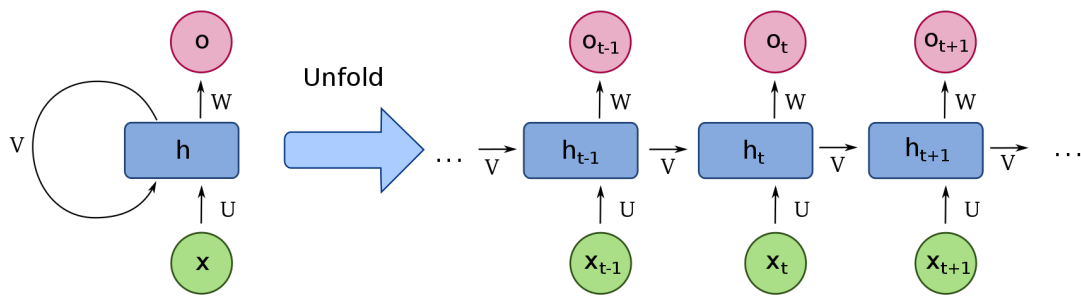


Figura 4.2 – Diagrama de uma RNN. Fonte: Por François Deloche - trabalho próprio, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=60109157>

Contudo é fácil perceber que, pela sua própria estrutura, que os problemas associados às redes de aprendizado profundo são agravados, já que há ainda mais parâmetros a serem determinados, associados àquelas conexões com os estados passados.

É nesse contexto que foram introduzidas as Redes Neurais Recorrentes com memória de longo e curto prazo, cuja contribuição principal é a adição de portões os quais são encarregados de filtrar quais informações serão realmente repassadas com o objetivo de mitigar o problema de desaparecimento de gradiente[32].

Há diferentes configurações de LSTM, cada qual com seus respectivos portões. Os portões mais utilizados por sua são os de entrada, responsável por filtrar apenas aquelas entradas que são relevantes, de saída, encarregado de selecionar aquilo que será passado como resultado e conseqüentemente às próximas iterações e o de esquecimento o qual leva em consideração quais daquelas informações das iterações passadas serão relevantes e quais não serão.

Uma representação da unidade de LSTM, ou *LSTM unit*, para uma determinada iteração t está representada no diagrama da Figura 4.3

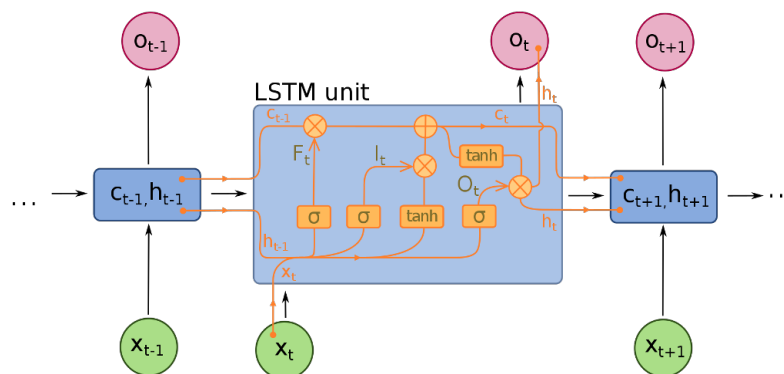


Figura 4.3 – Diagrama de uma LSTM. Fonte: Por François Deloche - trabalho próprio, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=60149410>

4.2 Tesseract OCR

Para a realização do reconhecimento dos caracteres para este trabalho foi utilizada API disponibilizada por OpenALPR que se utiliza do mecanismo do Tesseract OCR.

O Tesseract OCR começou a ser desenvolvido enquanto código aberto pelo grupo de pesquisa Hewlett-Packard Laboratories Bristol and pela companhia Hewlett-Packard Co, entre 1985 e 1994, com algumas mudanças adicionadas em 1996, para portabilidade com Windows, e 1998 para melhor adaptação ao C++. Em 2006 passou a ser desenvolvido principalmente pelo Google [33].

Historicamente era um dos três softwares de OCR de melhor precisão, em 1995, e se mantém como uma opção competitiva até hoje [33].

Apesar de versões mais antigas utilizarem métodos por segmentação, a versão 3.5 sendo a mais recente destas e ainda sendo utilizável, as versões mais atuais, cujas atualizações ainda são desenvolvidas, são baseadas em LSTM.

5 Atividades Realizadas

Em se tratando de uma área na qual o estagiário em questão tem pouco domínio os primeiros dois meses foram dedicados aos estudos referentes ao que tem sido realizado neste campo e quais são as mais robustas tecnologias que podem ser utilizadas para auxiliar a resolução destes problemas.

Realizados os estudos adequados a primeira etapa a ser superada foi o desenvolvimento de um filtro para separar o veículo em um cenário genérico, em se utilizando tanto de arquivos em imagens estáticas quanto em vídeo. Tanto as imagens quanto os vídeos foram capturados com um *smartphone* Samsung J3. As imagens com resolução de 2576x1932 e os vídeos com resolução de 1280x720.

O tratamento das imagens foi feita por meio do ferramental disponível no MATLAB.

5.1 Utilizando Imagem Estática

A primeira tentativa foi feita inicialmente reduzindo a resolução da imagem, realizando uma diminuição e posterior aumento no seu tamanho, como pode ser visto na comparação das Figuras 5.1a e 5.1b. Isto acontece porque quando o tamanho de uma imagem é reduzido ela deve ser representada em menos *pixels* e portanto a informação contida em muitos desses que são abandonados é perdida e não será recuperada na operação de aumento da imagem.

O próximo passo foi a binarização e a partir daí determinados os contornos presentes, visto na Figura 5.1c. Em seguida foram separados apenas os contornos mais longos, de forma a apenas os contornos de estruturas maiores, tais quais os carros, sobrassem na imagem, como na Figura 5.1d. Por fim utilizou-se um retângulo circunscrito ao contorno como uma máscara para segmentar na imagem original, como pode ser visto na Figura 5.1e.

Este método não obteve resultados consistentemente, já que dependendo das condições da imagem ocorrem situações tais quais a representada na Figura 5.1f, onde o contorno de vários corpos acabam se conectando e tornando o filtro final inútil.

5.2 Utilizando Vídeo

A próxima tentativa foi por meio de vídeo. Aqui o primeiro passo é, a partir de diferentes momentos gravados, *frames*, sendo o primeiro de referência, em algum momento em que a via estivesse livre esta imagem poderia ser a adquirida, e o segundo, no sistema já implementado seria quando o veículo acionasse o sensor.



(a) Imagem Original

(b) Imagem com resolução reduzida

(c) Imagem dos contornos



(d) Apenas Contornos da Imagem

(e) Resultado final desejado

(f) Possível contorno em uma má situação

Figura 5.1 – Primeira tentativa de segmentação. Fonte: Autoria Própria

Nas Figuras em 5.2 tem-se o resultado da tentativa de segmentar a placa do veículo na Figura 5.2a.

O primeiro passo, na ausência de imagem sem um veículo, foi utilizada de referência uma imagem de cerca de 20 frames antes em uma câmera de 30 fps. Supondo uma velocidade de 15 m/s, por se tratar de um estacionamento, nesse tempo ele deve percorrer pelo menos 5m, que já seria suficiente para que não houvesse sobreposição entre os carros nas duas imagens, já que a grande maioria dos veículos não tem mais de 5m. O problema em si, contudo, não é a ocorrência sobreposição o importante é que a placa e seus arredores, tais como a sombra e a parte da estrutura do carro próxima à placa, não sejam muito sobrepostos com elementos muito parecidos.

No segundo passo foi realizada a subtração das duas imagens, efetivamente retirando aquilo que é comum às duas, mais especificamente a estrada, resultando, após binarização, no filtro presente na Figura 5.2b.

Para obter o segundo filtro, da Figura 5.2c, foi utilizado o primeiro filtro em operação xor e posterior binarização em relação a cada uma das cores da imagem da qual se espera

segmentar a placa, já que no Brasil as placas são, normalmente, pretas no branco, ou prata, de forma a filtrar ainda mais elementos de cores parecidas que não estivessem sendo eliminados pelo primeiro filtro.

É importante salientar que a imagem da qual se deseja retirar a placa é que foi negativada na operação de subtração, de forma a seus *píxeis* brancos se tornarem pretos e vice-versa. Isso faz com que no segundo filtro a operação de xor não elimine as semelhanças entre o primeiro filtro e a imagem de interesse.

As máscaras foram então utilizadas para cortar retângulos de interesse na imagem binarizada os quais foram analisados linha a linha, exemplificada pela linha na Figura 5.2d, de forma a procurar um padrão de maior densidade de presença de bordas. Para esta análise foi utilizado algoritmo de emaranhamento para melhorar a eficiência, a busca sendo realizada inicialmente pulando de 1/4 da quantidade vertical de *píxeis* e dividindo-se por dois a cada iteração.



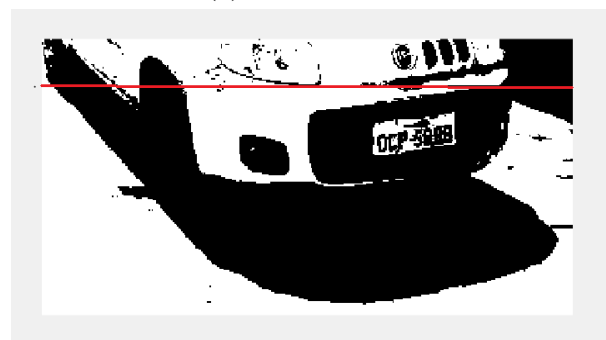
(a) Imagem da qual se deseja reconhecer a placa.



(b) Primeiro Filtro



(c) Segundo Filtro



(d) Resultado final desejado

Figura 5.2 – Esquema final para localização da placa. Fonte: Autoria Própria

Candidatos a placa devem ter de 14, para a placa com menos bordas (I I I 1 1 1 1) a cerca de 42, para as placas com maior número de bordas (W W W - 0 0 0 0), em um espaço de cerca de um quinto da imagem original, sendo analisadas em histogramas, tal qual o presente Figura 5.3.

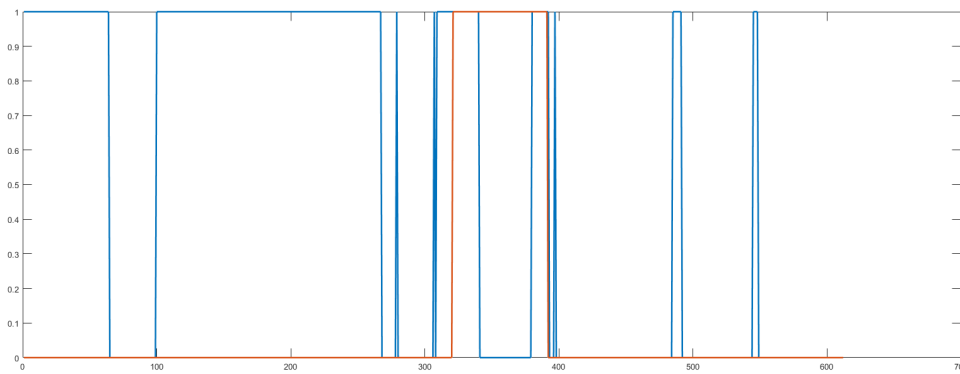


Figura 5.3 – Histograma das Linhas Analisadas na Imagem. Fonte: Autoria Própria

Algumas das imagens segmentadas podem ser vistas na Figura 5.4.

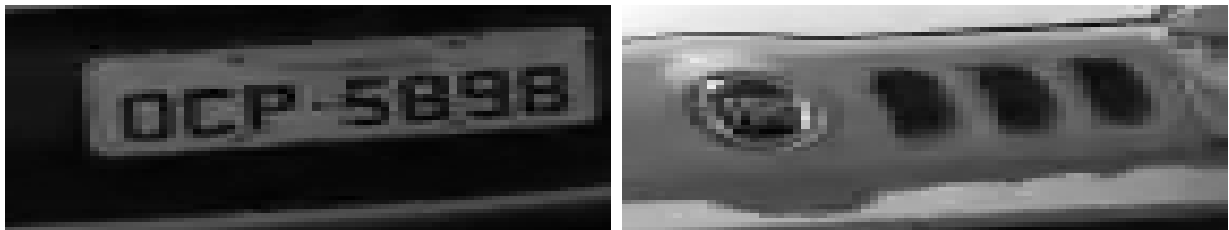


Figura 5.4 – Candidatos de Placa de carro

5.3 OCR

O plano de ação era em seguida realizar o reconhecimento de caracteres e, caso seja encontrada alguma placa, terminar a execução do programa, contudo as etapas até aqui descritas foram realizadas em MATLAB com intenções de serem traduzidas para uma linguagem mais favorável à programação remota em seguida, portanto essa interrupção da continuidade do código seria implementada uma vez na nova plataforma, com a ferramenta de reconhecimento de caracteres já escolhida.

Sendo as principais linguagens de programação, no que concerne aplicações em aprendizado de máquina e nuvem, Python ou C# optou-se por utilizar a última em combinação com a API do *Azure Computer Vision*. Devido a dificuldades técnicas relacionada à API, mesmo utilizando-se diversas imagens com diferentes configurações e qualidades, ficou decidido pela utilização da API da OpenALPR que se utiliza do mecanismo Tesseract OCR, o qual na sua versão mais recente utiliza RNN em arquitetura LSTM, fazendo com de todo o procedimento realizado anteriormente desnecessário, já que se trata de um método que não tem necessidade de segmentação e poderia atuar na imagem completa.

Os resultados obtidos são dados em conjunto com a confiança para as possibilidades que o modelo identificou, como pode ser visto na Tabela 1, que mostra as três melhores estimativas,

assim como a confiança de que cada uma está correta. Os resultados obtidos em geral levam a uma exatidão muito alta com confiança normalmente acima de 70%.

Placa	Estimativa	Confiança(%)	Estimativa	Confiança(%)	Estimativa	Confiança(%)
NPV - 8031	NPV - 8031	70,97	NPV - 9031	70,48	NPV - 031	65,65
OCP - 5898	OCP - 5898	87,87	CP - 5898	76,79	QCP - 5898	76,74
OFA - 2351	OFA - 2351	85,67	DFA - 2351	76,68	FA - 2351	75,86
OFX - 4405	DFX - 4405	73,08	OFX - 4405	70,54	FX - 4405	67,72
QFB - 3968	QFB - 3968	94,10	QF3 - 3968	81,13	FB - 3968	80,99
QFH - 4385	QFH - 4385	94,28	QH - 4385	81,02	FH - 4385	80,87

Tabela 1 – Algumas placas testadas e avaliadas pela API de OCR. Fonte: Autoria Própria

Notasse que a grande maioria dos erros são referentes às letras e que muitas vezes é substituída por alguma placa que não seria válida, faltando uma letra ou número. Com a mudança da placa para o novo padrão sul americano a eficiência pode ser afetada por apresentar letras e números juntos e na semelhança entre o zero e a letra O, que serão diferenciadas por apenas um traço naquele.

Alguns resultados desejáveis não foram alcançados devido a limitações ligadas ao próprio estagiário, suas competências e o tempo disponível. O cálculo da velocidade, apesar do armazenamento do azure fornecer o tempo com precisão de segundos, não foi feito por não haver alvo no que diz respeito à posição dos sensores, sendo assim a distância se mantém uma incógnita. A aplicação é local, sendo a única funcionalidade em nuvem a chamada da API utilizada e, por fim, não há interface humano-máquina.

6 Conclusões

Na lei número 11.788, de 25 de setembro 2008, está escrito, no seu artigo primeiro, no que se refere ao estágio, "visa à preparação para o trabalho produtivo de educandos que estejam freqüentando o ensino regular em instituições de educação superior, de educação profissional, de ensino médio, da educação especial e dos anos finais do ensino fundamental, na modalidade profissional da educação de jovens e adultos" e em seu respectivo paragrafo primeiro "o estágio visa ao aprendizado de competências próprias da atividade profissional e à contextualização curricular, objetivando o desenvolvimento do educando para a vida cidadã e para o trabalho".

No laboratório de Instrumentação Eletrônica e Controle o estagiário redator do presente texto foi incubido de uma tarefa ligada à área de formação além de ter lidado com metas realistas e prazos na entrega dos resultados apresentados no Capítulo 5, tal como se espera no mercado de trabalho, o que favorece o desenvolvimento das habilidades necessárias para a vida profissional. Também tratando-se de um trabalho significativo para a sociedade, como explicitado ainda no Capítulo 1 e compartilhando o espaço com multiplas outras pessoas, como mencionado no Capítulo 2, pode-se dizer que o objetivo legal e curricular do estágio em questão foi alcançado.

Quanto aos resultados, apesar de ainda haver necessidade de trabalho no futuro para que se possa dizer completo, os principais objetivos, de segmentar e identificar a placa de carros foram cumpridas. Fica proposto para trabalhos futuros a integração do projeto à nuvem e o desenvolvimento de uma interface humano-máquina capaz de identificar a infração de trânsito e ser utilizada pelos agentes da lei.

Referências

- [1] Nações Unidas. Década de ação pela segurança no trânsito (2011-2020), nov 2019.
- [2] Portal do Trânsito. Estudo mostra que em 10 anos o trânsito brasileiro matou mais que a guerra da síria, nov 2019.
- [3] Rudolf F Graf. *Modern dictionary of electronics*. Elsevier, 1999.
- [4] Karl Rasche, Robert Geist, and James Westall. Re-coloring images for gamuts of lower dimension. In *Computer Graphics Forum*, volume 24, pages 423–432. Wiley Online Library, 2005.
- [5] Amy A Gooch, Sven C Olsen, Jack Tumblin, and Bruce Gooch. Color2gray: salience-preserving color removal. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 634–639. ACM, 2005.
- [6] Christopher Kanan and Garrison W Cottrell. Color-to-grayscale: does the method matter in image recognition? *PloS one*, 7(1):e29740, 2012.
- [7] Satadal Saha. A review on automatic license plate recognition system. *arXiv preprint arXiv:1902.09385*, 2019.
- [8] Mahua Nandy and Satadal Saha. An analytical study of different document image binarization methods. *arXiv preprint arXiv:1501.07862*, 2015.
- [9] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [10] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *Journal of graphics tools*, 12(2):13–21, 2007.
- [11] A Broumandnia and M Fathy. Application of pattern recognition for farsi license plate recognition. *ICGST International Journal on Graphics, Vision and Image Processing*, 5(2):25–31, 2005.
- [12] Curiosity. Listen to the princess and the fox, an oddball fairy tale written by an ai, aug 2019.
- [13] Curiosity. Robots of the future will lie to us, and that’s a good thing, aug 2019.
- [14] Curiosity. Facebook a.i. programmed to negotiate teaches itself to lie, aug 2019.
- [15] Curiosity. Read these (legit terrifying) horror stories by an artificial intelligence, aug 2019.

- [16] Curiosity. Meet ibm watson, the computer that can argue better than anyone, aug 2019.
- [17] P. Simon. *Too Big to Ignore: The Business Case for Big Data*. Wiley and SAS Business Series. Wiley, 2013.
- [18] T.M. Mitchell. *Machine Learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education, 1997.
- [19] A Jay Goldstein, Leon D Harmon, and Ann B Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.
- [20] Michael Kirby and Lawrence Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*, 12(1):103–108, 1990.
- [21] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.
- [22] Thomas M Breuel. The ocropus open source ocr system. In *Document Recognition and Retrieval XV*, volume 6815, page 68150F. International Society for Optics and Photonics, 2008.
- [23] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [24] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait. High-performance ocr for printed english and fraktur using lstm networks. In *2013 12th International Conference on Document Analysis and Recognition*, pages 683–687, Aug 2013.
- [25] Yoshua Bengio, Yann LeCun, and Donnie Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks, and hidden markov models. In *Advances in neural information processing systems*, pages 937–944, 1994.
- [26] Andrew William Senior. Off-line cursive handwriting recognition using recurrent neural networks. 1994.
- [27] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008.
- [28] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

-
- [29] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.
- [30] Ludolouis. Comment fonctionne un réseau de neurones artificiels, Mar 2019.
- [31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [33] Ray W Smith. History of the tesseract ocr engine: what worked and what didn't. In *Document Recognition and Retrieval XX*, volume 8658, page 865802. International Society for Optics and Photonics, 2013.