



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RECONHECIMENTO DE PLACAS VEICULARES EM CENÁRIOS
COMPLEXOS UTILIZANDO O MÉTODO DO SUBESPAÇO

Anderson Sousa de Jesus

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Engenharia
Elétrica, PPGEE, da Universidade Federal
do Amazonas, como parte dos requisitos
necessários à obtenção do título de Mestre
em Engenharia Elétrica.

Orientadores: Waldir Sabino da Silva Júnior
Eulanda Miranda dos Santos

Manaus
Novembro de 2021

ANDERSON SOUSA DE JESUS

**RECONHECIMENTO DE PLACAS VEICULARES EM
CENÁRIOS COMPLEXOS UTILIZANDO O MÉTODO DO
SUBESPAÇO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 07 de dezembro de 2021.

BANCA EXAMINADORA



Prof. Dr. Waldir Sabino da Silva Júnior, Presidente
Universidade Federal do Amazonas



Prof. Dr. Carlos Augusto de Moraes Cruz, Membro
Universidade Federal do Amazonas



Prof. Dr. Gabriel Matos Araújo, Membro
Centro Federal de Educação Tecnológica

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

J58r Jesus, Anderson Sousa de
Reconhecimento de placas veiculares em cenários complexos
utilizando o método do subespaço / Anderson Sousa de Jesus .
2021
70 f.: il.; 31 cm.

Orientador: Waldir Sabino da Silva Júnior
Coorientadora: Eulanda Miranda dos Santos
Dissertação (Mestrado em Engenharia Elétrica) - Universidade
Federal do Amazonas.

1. Placas veiculares. 2. Método do subespaço. 3. Redes neurais
convolucionais. 4. Visão computacional. I. Silva Júnior, Waldir
Sabino da. II. Universidade Federal do Amazonas III. Título

Agradecimentos

- Primeiramente à minha mãe (Rosa), minhas irmãs (Gleice e Keityane) e minha namorada (Ana) pelo apoio e compreensão em todos os momentos.
- A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.
- Ao meu orientador Waldir Sabino, pela amizade, compreensão, paciência, orientações e por sempre me ajudar nos momentos difíceis da execução desta dissertação.
- Ao Osmar Silva e Ruan Belém, por possibilitarem a conciliação entre os horários de trabalho e estudo, além de compreenderem a importância desta formação para o desenvolvimento profissional.
- A professora Eulanda Santos e Bernardo Gatto, por todo o aconselhamento e inspiração ao longo dessa jornada.
- A todos os meus amigos pela amizade de longa data, pela compreensão, apoio e respeito.
- Apenas gratidão a todos que me ajudaram, muito obrigado.

Resumo da Dissertação apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

RECONHECIMENTO DE PLACAS VEICULARES EM CENÁRIOS COMPLEXOS UTILIZANDO O MÉTODO DO SUBESPAÇO

Anderson Sousa de Jesus

Orientadores: Waldir Sabino da Silva Júnior

Eulanda Miranda dos Santos

Programa: Pós-Graduação em Engenharia Elétrica

Nesta dissertação, é proposto um sistema completo para a realização do processo de detecção e reconhecimento de placa de licenciamento veicular em imagens onde o processo de aquisição foi realizado com a câmera e o veículo em movimento, e que possui variações na iluminação e na resolução, além de cenários complexos. Como etapas de pré-processamento foram utilizados a conversão para a escala de cinza e o detector de objetos YOLO para realizar a detecção do carro. A primeira etapa da metodologia proposta foi a criação de bases de imagens, na qual foi utilizado o detector de objetos YOLO para detecção da placa e seis tipos de degradação para aumento de dados: ruído gaussiano, ruído de Poisson, ruído laplaciano, mudança de escala, rotação e mudança de contraste. Em seguida a metodologia proposta consistiu na aplicação de duas técnicas para o reconhecimento de placas veiculares, na primeira técnica utiliza-se o método do subespaço mútuo, e no segundo método utiliza-se como *baseline* redes neurais convolucionais. Os resultados obtidos na abordagem utilizando o método do subespaço mútuo teve como melhor resultado a acurácia de 57% e tempo médio de predição de 0,33ms, enquanto que o resultado da abordagem baseada em redes neurais convolucionais obteve como melhor resultado a acurácia de 94% e tempo médio de predição de 200ms.

Palavras-chave: placas veiculares, método do subespaço, redes neurais convolucionais.

Abstract of Dissertation presented to UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

LICENSE PLATE RECOGNITION IN BACKGROUND COMPLEX USING THE SUBSPACE METHOD

Anderson Sousa de Jesus

Advisors: Waldir Sabino da Silva Júnior

Eulanda Miranda dos Santos

Department: Postgraduate in Electrical Engineering

In this dissertation, a complete system is proposed for carrying out the process of detection and recognition of vehicle license plate in images where the acquisition process was carried out with the camera and vehicle in motion, and which has variations in lighting and resolution, as well as complex scenarios. As pre-processing steps, conversion to gray scale and YOLO object detector were used to perform car detection. The first stage of the proposed methodology was the creation of image bases, in which the YOLO object detector was used to detect the license plate and six types of degradation for data augmentation: Gaussian noise, Poisson noise, laplacian noise, scale change, rotation and contrast change. Then the proposed methodology consisted of the application of two techniques for vehicle license plate recognition, in the first technique the mutual subspace method is used, and in the second method convolutional neural networks are used as baseline. The results obtained in the approach using the mutual subspace method had as the best result the accuracy of 57% and average prediction time of 0.33ms, while the result of the approach based on convolutional neural networks obtained as the best result the accuracy of 94% and average prediction time of 200ms.

Keywords: licence plate, subspace method, convolutional neural network.

Sumário

1	Introdução	1
1.1	Objetivos da Dissertação	4
1.2	Organização da Dissertação	5
2	Fundamentos Teóricos	6
2.1	Processamento digital de imagens	6
2.2	Aprendizagem de máquina	9
2.3	Redes neurais	9
2.4	Métodos de visão computacional utilizando inteligência artificial . . .	14
2.5	Método do subespaço	15
2.6	Método do subespaço mútuo	16
2.7	Fluxo de processo do método do subespaço mútuo	17
2.8	Trabalhos relacionados	18
3	Metodologia	24
3.1	Introdução	24
3.2	Reconhecimento de placas veiculares utilizando redes neurais convo- lucionais	25
3.3	Reconhecimento de placas veiculares utilizando o método do subes- paço mútuo	30
4	Base de dados	37
4.1	Introdução	37
4.2	Detecção do veículo	39
4.3	Detecção da placa do veículo	40
4.4	Rotulação dos caracteres das placas veiculares	40

4.5	Aumento de dados utilizando degradações	40
5	Experimentos e Resultados	45
5.1	Setup	45
5.2	Questionamentos principais	46
5.3	Procedimentos Experimentais	46
5.4	Métricas de desempenho	48
5.5	Análise dos resultados	48
5.5.1	Resultados da técnica baseada em subespaços	48
5.5.2	Resultados da técnica baseada em redes neurais	51
5.5.3	Resultados por classe com variações nos tipos de degradações	52
5.5.4	Resultados por classe com variações nos níveis de degradações	54
6	Conclusão	57
	Referências Bibliográficas	60

Lista de Figuras

3.1	Diagrama em blocos do processo de reconhecimento de placas veiculares utilizando CNN.	26
3.2	Conversão da imagem do espaço de cor RGB para escala de cinza. . .	28
3.3	Diagrama em blocos das etapas de treinamento do método de classificação por subespaço.	32
3.4	Diagrama em blocos das etapas de predição do método de classificação por subespaço.	33
4.1	Diagrama em blocos do processo de criação da base de dados. . . .	38
4.2	Exemplos da base de dados original.	39
4.3	Processo de detecção da placa do veículo.	40
4.4	Exemplo da aplicação de rotação em uma imagem.	41
4.5	Exemplo da aplicação de ruído gaussiano em uma imagem.	42
4.6	Exemplo da aplicação de mudança de gamma em uma imagem. . . .	43
4.7	Exemplo da aplicação de mudança de escala em uma imagem. . . .	43
4.8	Exemplo da aplicação do ruído Poisson em uma imagem.	44
4.9	Exemplo da aplicação do ruído laplaciano em uma imagem.	44

Lista de Tabelas

2.1	Comparação entre a metodologia proposta e trabalhos relacionados. .	23
3.1	Variáveis do classificador baseado em redes neurais.	27
3.2	Variáveis do classificador do método do subespaço mútuo.	34
5.1	Degradações aplicadas nas partições de cada base de imagens.	47
5.2	Resultados da técnica baseada em subespaços.	49
5.3	Resultados da técnica baseada em redes neurais.	51
5.4	Análise dos resultados por classe de variados tipos de degradações. . .	53
5.5	Análise dos resultados por classe de variados níveis de degradação do ruído Gaussiano.	55

Capítulo 1

Introdução

A visão computacional é uma área de pesquisa que busca fornecer a percepção visual humana (detecção de objetos, contagem, noção de diferentes dimensões) para que computadores possam interpretar imagens e vídeos [1]. A interpretação de imagens e vídeos é realizada por meio da utilização de diferentes métodos, sendo que esses métodos podem ser divididos em duas grandes subáreas: processamento digital de imagens (PDI) e aprendizado de máquina. Processamento digital de imagens remete ao uso de complexas etapas computacionais para que se permita analisar, recolher informações e aplicar melhorias em imagens [2]; e aprendizado de máquina é a utilização de algoritmos para extrair informações de dados brutos e representá-los através de algum tipo de modelo matemático [3]. Dentre diversas aplicações que empregam técnicas de visão computacional, destaca-se as seguintes como exemplos: realização de detecção de doenças [4], desenvolvimento de veículos autônomos [5] e classificação de grãos de café [6].

Outra aplicação que tem atraído considerável atenção dos pesquisadores é o reconhecimento automático de placas de veículos [7]. O reconhecimento automático de placas veiculares é a capacidade de extração e reconhecimento automático dos caracteres da placa de um veículo a partir de uma imagem [8]. Com o rápido aumento do número de veículos, há necessidade de melhorar os sistemas existentes de identificação de veículos, já que este aumento causa uma série de problemas, tais como acidentes de trânsito ou vulnerabilidade do espaço público ao crime. A gestão deste novo cenário é bastante desafiante, e este problema tem encorajado o desenvolvimento de sistemas automáticos para auxiliar neste controle [7]. Em particular,

a detecção automática de placas veiculares pode contribuir a monitorar adequadamente os veículos e aliviar grandemente a carga de gestão de tráfego, além disso, a detecção automática de placas veiculares também pode ser aplicada em muitos outros cenários, por exemplo, para a cobrança de pedágio, violações de excesso de velocidade, vigilância e gestão de estacionamento e condomínios residenciais [8].

Embora o reconhecimento automático de placas veiculares tenha sido estudado durante muitos anos, ainda é uma tarefa desafiante detectar e reconhecer seus caracteres com alta taxa de acurácia e baixo custo computacional. Na literatura, foram propostos muitos algoritmos para o reconhecimento automático de placas veiculares [7]. Os métodos tradicionais envolvem características tais como cor [9], borda [10] e morfologia [11], que estão principalmente confinadas por condições rigorosas. Para amenizar este problema, métodos baseados em inteligência artificial foram concebidos, que aprendem automaticamente as características a partir dos dados adquiridos. Recentemente, este tipo de métodos de detecção tem rendido resultados muito impressionantes em acurácia [12], [13], [14]; contudo, o consumo de tempo e hardware é significativamente mais elevado do que o de técnicas tradicionais. Todos os métodos citados enfrentam vários desafios no desenvolvimento de um sistema escalável e eficaz de reconhecimento da placa de veículos [7]. Estes desafios incluem: necessidade de imagens de alta resolução como entrada, dificuldade no reconhecimento de diferentes formas e cores de placas, diferentes tipos de veículos e estradas, condições meteorológicas variáveis, resultado mais fraco sob iluminação e fundo complexo, dificuldade em prever quando vários carros estão presentes em uma única imagem.

A classificação de subespaços tem sido um dos problemas pesquisados de forma recorrente em visão computacional, onde um conjunto de imagens de um objeto a ser classificado é compactamente representado por um subespaço em espaço vetorial de alta dimensão [15]. A análise do subespaço em visão computacional é um nome genérico para descrever uma estrutura geral de comparação e classificação de subespaços. Uma abordagem típica na análise de subespaço é o método do subespaço (SM) que classifica um vetor padrão de entrada em várias classes com base na distância ou ângulo mínimo entre o vetor padrão de entrada e cada subespaço de classe, onde um subespaço de classe corresponde à distribuição dos vetores padrão da

classe no espaço vetorial de alta dimensão [15]. O SM tem sido conhecido como um dos métodos mais úteis no campo do reconhecimento de padrões, uma vez que seu algoritmo é muito simples e pode lidar com a classificação de múltiplas classes [16].

O método de subespaço mútuo (MSM) é uma extensão natural do SM [17]. O SM usa um único vetor como entrada para o cálculo da similaridade, enquanto MSM usa o subespaço de múltiplos vetores, calculando a similaridade através da distância entre subespaços. Portanto o MSM classifica um conjunto de vetores de padrões de entrada em várias classes baseadas em múltiplos ângulos canônicos (principais) entre o subespaço de entrada e subespaços de classe [15]. Métodos baseados no subespaço têm sido empregados em várias aplicações de visão computacional, incluindo reconhecimento facial, reconhecimento de objetos e reconhecimento de gestos [18]. Em geral, os métodos baseados no subespaço ignoram a dimensão ideal de cada distribuição de classe, tratando todas elas com a mesma dimensão [16].

Nossa proposta, consiste na detecção e reconhecimento de placas veiculares em ambientes complexos. O sistema proposto é composto em quatro etapas. A primeira etapa consiste na criação de uma base de imagens com auxílio do detector de objetos YOLO, sendo que esta base de imagens será usada para o treinamento das técnicas de reconhecimento e para execução de testes para validação do método proposto. Na segunda etapa utiliza-se o método de subespaço para realizar o reconhecimento de placas de licenciamento veiculares em ambientes complexos através dos ângulos canônicos entre os subespaços não lineares, que são gerados a partir dos padrões de imagem de cada classe. Na terceira etapa aplica-se, seguindo uma abordagem de aprendizado supervisionado, o treinamento de redes neurais convolucionais para o reconhecimento de placas de licenciamento veiculares em ambientes complexos, sendo que o objetivo dessa etapa é criar um *baseline* para comparar os resultados obtidos do método baseado em subespaço. Na última etapa realiza-se uma análise dos resultados obtidos dos métodos propostos desenvolvidos nas etapas anteriores.

Esta dissertação apresenta como destacável contribuição a base de dados concebida. As imagens das bases de dados existentes e disponíveis, normalmente utilizam imagens com ambiente controlado e com a qualidade da imagem bem superior, resultando em uma excelente visibilidade da placa do veículo. A base de

dados concebida neste trabalho de pesquisa apresenta alguns fatores que aumentam a complexidade da detecção do objeto de interesse, como: o processo de aquisição foi realizado com a câmera e o veículo em movimento; variações na iluminação; variações na resolução das imagens; *background* complexo.

Com a metodologia proposta desenvolvida então a predição para detecção e reconhecimento de placas veiculares será aplicada através das seguintes etapas: inicialmente será realizada a conversão da imagem de entrada para a escala de cinza; então será utilizado o detector de objetos YOLO para realização da detecção do carro na imagem e, em seguida, realizar a detecção da placa veicular; por fim será utilizado o método proposto para reconhecimento das placas.

1.1 Objetivos da Dissertação

O objetivo principal desta dissertação é estabelecer um método completo para realização do processo de detecção e reconhecimento de placas veiculares em imagens com o objeto de interesse em movimento em um *background* complexo.

Objetivos Específicos da Dissertação

Os objetivos específicos desta dissertação são os seguintes:

- (1) Conceber uma base de dados com imagens reais de veículos e câmera em movimento com resoluções diferentes e ambiente complexo;
- (2) Investigar a técnica para reconhecimento de placas veiculares baseado no método do subespaço;
- (3) Investigar a técnica para reconhecimento de placas veiculares baseado em técnicas de aprendizado de máquina;
- (4) Examinar o comportamento da metodologia proposta com diferentes tipos de degradações nas imagens;
- (5) Comparar os resultados obtidos com as duas metodologias propostas para o reconhecimento de placas veiculares;
- (6) Definir um modelo de pesquisa que sirva como alicerce para trabalhos futuros.

1.2 Organização da Dissertação

Esta dissertação está organizada conforme abaixo.

- No Capítulo 2 são apresentados os fundamentos teóricos utilizados nesta pesquisa. São explicados conceitos de processamento digital de imagens, conceitos de aprendizagem de máquina e uma fundamentação sobre o método do subespaço. Realiza-se uma descrição detalhada do fluxo de processo do método do subespaço mútuo. Ainda temos a revisão bibliográfica apresentando as técnicas atualmente mais abordadas nas literaturas para detecção e reconhecimento de placas veiculares. E por fim, é realizada uma breve comparação entre as literaturas e o trabalho proposto nesta dissertação.
- No Capítulo 3, é descrita a metodologia proposta, que pode ser dividida em dois processos, sendo o primeiro baseado em redes neurais convolucionais e o segundo baseado no método de subespaço. Foi exposto um diagrama de blocos para cada processo da metodologia proposta para dar uma visão macro ao leitor sobre cada proposta e nas seções seguintes deste mesmo capítulo foi detalhado cada item do bloco correspondente a esta metodologia.
- No Capítulo 4, apresenta-se a base de dados utilizada nesta pesquisa, especificando cada etapa do processo, indo desde a detecção do veículo até o tipo de degradação aplicada. Com isso, descreve-se de forma detalhada as informações da base de dados utilizada nesta dissertação, contemplando o cenário de aquisição, características da base, método de detecção da placa e a forma de rotulação dos caracteres.
- O Capítulo 5 foi dedicado aos experimentos e resultados obtidos. Inicialmente é descrito o *setup* dos experimentos, detalhando as ferramentas utilizadas para desenvolver o método proposto. Então apresenta-se os questionamentos principais que este trabalho pretende responder. Na sequência descreve-se os procedimentos experimentais aplicados. Por fim os resultados são exibidos, mostrando as diferenças entre as duas metodologias propostas.
- No Capítulo 6 são apresentadas as considerações finais da pesquisa e as propostas de trabalhos futuros.

Capítulo 2

Fundamentos Teóricos

2.1 Processamento digital de imagens

Os sistemas de processamento digital de imagens (PDI) consistem em um conjunto de técnicas que permitem a aquisição de imagens por equipamentos diversos, como tomógrafos médicos, câmeras fotográficas, satélites e outros, bem como a representação e transformação destas imagens [19]. A finalidade é extrair e identificar informações contidas nas imagens de forma automática por meio de máquinas [20]. Processos baseados em PDI podem ser utilizados em diversas aplicações, como por exemplo, na área médica identifica-se lesões ou regiões do corpo atingidas por câncer [21], [4], ou em automação industrial, onde um sistema de PDI auxilia a inspeção de placas de circuito impresso ou a inspeção de soldas [22].

Uma imagem pode ser definida como uma função bidimensional, $I(x, y)$, em que x e y são coordenadas espaciais (plano), e a amplitude de I em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando x , y e os valores de intensidade de I são quantidades finitas e discretas, chamamos de imagem digital [20]. Para que as técnicas de PDI sejam empregadas nas imagens adquiridas, torna-se necessária a definição de etapas para sua aplicação [23]. Nesse contexto, as principais etapas para análise e quantificação das informações são: aquisição, pré-processamento, segmentação, extração de características, reconhecimento e interpretação. Essas etapas serão melhor definidas a seguir.

Considerando-se a aquisição de uma imagem, pode-se comentar que é o pro-

cesso de conversão de uma cena real tridimensional em uma representação bidimensional adequada [20]. Os dispositivos de aquisição de imagens mais utilizados atualmente são as câmeras *charge coupled device* (CCD) e as câmeras *complementary metal oxide semiconductor* (CMOS) [2] que são amplamente utilizados em câmeras digitais e outros instrumentos que utilizam sensores de luz.

Durante o processo de aquisição das imagens é frequente a ocorrência de alguns tipos de degradações que são comumente denominadas ruídos [19]. A função da etapa de pré-processamento em um sistema de processamento de imagens, é aprimorar a imagem que será utilizada nas etapas posteriores, buscando a eliminação da ocorrência destes ruídos. As operações efetuadas para aprimorar uma imagem envolvem técnicas que visam melhorar a aparência ou converter a imagem para um formato mais adequado, permitindo a análise da mesma por uma máquina [24]. Um dos processos mais utilizados no pré-processamento de imagens é a transformação para tons de cinza [25]. Imagens em tons de cinza com profundidade de 8 bits fornecem 256 níveis de cinza tendo a luminosidade representada por um único valor que varia do preto ao branco. Para o caso de imagens coloridas, ao invés da intensidade de luz ser representada por um único valor, cada pixel é representado pela combinação de três valores no sistema RGB, ou seja, três planos de 8 bits (24 bits), totalizando mais de 16 milhões de cores diferentes [20]. Outros processos (ou métodos) de pré-processamentos comumente utilizados são a equalização de histograma [25] e suavização [20].

Considerando-se a segmentação, o objetivo é encontrar uma imagem subdividida em partes como forma de buscar o isolamento de conjuntos de pixels que representem as regiões de interesse procuradas na imagem [25]. O nível de detalhe em que a subdivisão é realizada depende do problema a ser resolvido. A maioria dos algoritmos de segmentação baseia-se em uma das seguintes propriedades básicas de valores de intensidade: descontinuidade e similaridade. Na primeira categoria, a abordagem é dividir uma imagem com base nas mudanças bruscas de intensidade, como as bordas (por exemplo detector de bordas Canny [26]). Na segunda categoria estão baseadas na divisão de uma imagem em regiões que sejam semelhantes de acordo com um conjunto de critérios predefinidos [20], como as cores (por exemplo limiarização [25]). Há outros métodos de segmentação que utilizam aprendizado

de máquina, sobressaindo-se entre elas as aplicações de segmentação semântica [27] onde o objetivo é compreender melhor a cena ao atribuir a cada pixel no interior da imagem, uma categoria de objeto. Destacam-se entre as aplicações de segmentação semântica de imagens: nos veículos autônomos, onde é necessário um entendimento preciso do ambiente; no diagnóstico médico [28], onde células, tecidos e órgão de interesse são segmentados; no desenvolvimento de robôs que navegam e manipulam objetos no ambiente.

A extração de características desempenha um papel crucial na detecção de objetos porque ajuda a encontrar pistas que podem isolar objetos significativos do fundo do ambiente. A extração de características visa reduzir uma grande imagem a um pequeno número de elementos visuais [29]. Métodos poderosos de extração de elementos são normalmente utilizados para construir modelos de classificação de imagens, especialmente para extrair elementos visuais das propostas de regiões para classificação posterior, tais como arestas, cantos e contornos. Existem vários métodos para a extração das características, sejam eles tradicionais como o *histogram of oriented gradients* (HOG) ou métodos mais sofisticados como *convolution neural network* (CNN) que extraem várias características diferentes ao longo das suas camadas. Outros processos que podem ser citados também são: *speeded up robust features* (SURF) [30], *features from accelerated segment test* (FAST) [31], *binary robust independent elementary features* (BRIEF) e *oriented FAST and rotated BRIEF* (ORB) [32].

A etapa de reconhecimento de padrões busca identificar objetos presentes na imagem que foram isolados pelas etapas anteriores. Cada objeto apresenta um padrão e as medições realizadas são baseadas nas características desse padrão [25]. São considerados pertencentes à mesma classe de padrões, objetos que possuam uma ou mais características semelhantes. Exemplo de técnicas utilizadas nessa etapa são: *k-nearest neighbours* (kNN) [33], *support vector machine* (SVM) [34] e rede neural artificial (RNA) [19].

2.2 Aprendizagem de máquina

O sucesso recente das aplicações de inteligência artificial depende do desempenho e das capacidades dos modelos de aprendizagem de máquinas [35]. Nos últimos anos, foi proposta uma variedade de algoritmos e técnicas para tornar a aprendizagem de máquinas mais efetiva e confiável [36]. A aprendizagem de máquinas nos permite enfrentar tarefas que são demasiado difíceis de resolver com programas fixos escritos e concebidos por seres humanos [3]. A partir de um programa científico e do ponto de vista filosófico, a aprendizagem de máquinas é interessante porque o desenvolvimento da nossa compreensão da aprendizagem de máquina implica desenvolver a nossa compreensão dos princípios que fundamentam a inteligência [37].

As tarefas de aprendizagem de máquinas são geralmente descritas em termos de como o sistema de aprendizagem de máquinas deve processar um exemplo. Um exemplo é uma coleção de características que foram quantitativamente medidas a partir de algum objeto ou evento que queremos que o sistema de aprendizagem de máquina processe [3]. Ou seja, os algoritmos de aprendizagem de máquina são algoritmos capazes de aprender com os dados. Muitos tipos de tarefas podem ser resolvidos com a aprendizagem de máquina. Algumas das tarefas mais comuns de aprendizagem de máquinas incluem: classificação [38], regressão [39], detecção de anomalias [40], entre outras.

2.3 Redes neurais

As redes neurais artificiais (RNA) foram introduzidas por McCulloch e Pitts [41] e foram inspiradas em redes neurais biológicas, como as dos cérebros humanos. As RNA são geralmente organizadas em várias camadas e cada camada possui vários neurônios artificiais ligados à camada anterior e a seguinte, sendo estas ligações similares a sinapses. Este tipo de rede é composto por 3 tipos principais de camadas [29]: camada de entrada, composta pelos neurônios que recebem e processam os dados de entrada; camadas ocultas que são camadas intermediárias utilizadas para calcular e agregar as ativações que ligam a camada de entrada à camada de saída, sendo que, o número destas camadas e destes neurônios utilizados afeta o desempenho do algoritmo; e camada de saída, onde a previsão final da rede é calculada, sendo esta

previsão podendo ser comparada aos dados reais de saída, de forma a atualizar os parâmetros da rede.

Uma rede neural artificial é composta por várias unidades de processamento. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma RNA vem das interações entre as unidades de processamento da rede [41]. A operação de uma unidade de processamento pode ser resumida da seguinte maneira: sinais são apresentados à entrada; cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade; é feita a soma ponderada dos sinais que produz um nível de atividade; e se este nível de atividade exceder um certo limite a unidade produz uma determinada resposta de saída [41].

As abordagens de redes neurais podem ser classificadas em 3 tipos principais: a abordagem supervisionado que é quando tentamos prever uma variável dependente a partir de uma lista de variáveis independentes, a abordagem não-supervisionado que é utilizado quando não temos os dados de entradas rotulados e precisamos achar uma representação mais informativa dos dados que temos. A terceira abordagem é a chamada aprendizagem por reforço, em que a máquina tenta aprender qual é a melhor ação a ser tomada, dependendo das circunstâncias na qual essa ação será executada [42]. Algumas das principais arquiteturas de redes neurais utilizadas são [3]: redes *multilayer perceptron*, redes neurais convolucionais, redes neurais recorrentes, *long short-term memory* (LSTM), *deep auto-encoders* e *generative adversarial network* (GAN).

A rede neural convolucional (CNN, *convolution neural network* em inglês) é constituída por duas fases primárias de extração e classificação das características [4], sendo que na extração das características realiza-se uma convolução contínua e operações de agrupamento a fim de detectar as características na imagem e utiliza-se a camada totalmente ligada para classificar as características extraídas. As camadas na CNN estão dispostas em três dimensões separando os neurônios de uma camada para a outra. A saída é normalmente reduzida a vetores únicos com base nas pontuações de probabilidade de cada classe [4]. Os blocos principais das CNN são as camadas convolucionais [29]. As camadas convolucionais geram novas

imagens, designadas como mapas de características (*feature map* em inglês). Cada mapa de características é a saída de um filtro aplicado a uma imagem da camada anterior [29].

Normalmente, o filtro é de tamanho reduzido, 3×3 ou 2×2 pixel, por exemplo. O filtro desloca-se através da matriz da imagem, sobrepondo-se ao longo da imagem [20]. Se o filtro exceder a matriz da imagem, é necessário aplicar *padding*. Isto é feito ou adicionando zeros à imagem ou um preenchimento válido, como espelhamento ou replicação. A CNN é o tipo de rede neural profunda (DNN, *deep neural network* em inglês) mais popular [43]. Um dos seus pontos fortes é o reconhecimento de padrões numa imagem, sendo atualmente o estado da arte na classificação e detecção de objetos [29]. Exemplos de CNN são a LeNet-5 [44], a AlexNet [45], GooLeNet [46], VGG-16 [47] e ResNet [48]. A seguir, apresenta-se algumas peculiaridades das redes neurais.

- Retropropagação: As topologias das camadas de redes neurais criam um desafio para o seu correto treinamento [49] devido a complexidade de análise do erro no caso de aplicação de uma função de ativação tipo limiar, por exemplo. Desta forma, a fim de sobrepujar este desafio, é usado uma técnica de aprendizagem supervisionada chamada retropropagação (*backpropagation*, em inglês) [3]. A retropropagação é um método de aprendizado supervisionado e uma generalização do método dos mínimos quadrados médios. O aprendizado ocorre no perceptron, alterando os pesos de conexão após o processamento de cada parte dos dados, com base na quantidade de erro no resultado comparado ao resultado esperado [49].
- Funções de ativação: A função de ativação define a saída de um neurônio, decidindo se este deve ser ativado ou não [29]. Mapeia os valores resultantes dos cálculos dos neurônios para um intervalo predefinido desejado. Esta função de ativação deve informar os erros cometidos pela rede para as camadas anteriores com a maior precisão possível [49]. Existem várias hipóteses na escolha da função de ativação. A função de ativação sigmoide normaliza a saída entre 0 e 1, enquanto que a função tangente hiperbólica está centrada em zero e normaliza entre -1 e 1. Estas funções estão sujeitas ao problema do gradiente de fuga que acontece quando a rede se recusa a aprender mais, tornando-

se demasiado lentas para fazer uma previsão precisa [29]. A unidade linear retificada (RELU) [50] pode ser utilizada para resolver este problema. Existe também a função de ativação softmax, tipicamente utilizada na camada de saída em problemas de classificação. Esta função normaliza as saídas para cada classe entre 0 e 1 e as divide pela sua soma, dando as probabilidades de cada valor de entrada pertencer a uma classe específica [29].

- **Erro e Função de Perda:** Uma função de custo ou perda é uma medida que avalia o quanto eficiente um modelo de previsão é em relação ao resultado esperado [29]. A otimização dos pesos e outros parâmetros está diretamente ligada à escolha de uma função de custo adequada [49]. Não existe uma única função de perda que funcione para todo o tipo de dados, dependendo de vários fatores, como a presença de *outliers* e escolha do algoritmo. O erro quadrático médio (MSE) é a função de custo mais utilizada [29]. MSE é a soma das distâncias quadráticas entre o valor obtido e valores previstos. Estas funções não são utilizadas apenas para medir a performance do modelo de previsão mas também na fase de treino na escolha dos pesos e bias ideal que minimiza o erro [29]. O bias é uma constante que ajuda o modelo adaptar-se melhor aos dados fornecidos, dando liberdade para ter um melhor desempenho.
- **Dropout:** Na técnica de *dropout*, em cada etapa de treino, cada neurônio tem uma probabilidade, chamada de taxa de abandono, que é a probabilidade de ser temporariamente ignorado durante essa etapa de treino [29]. Ao contrário da Regularização L1 e L2, o *dropout* não depende da modificação da função de custo, em vez disso, no *dropout* é realizado a modificação da própria rede. Regularização pode ser definida como qualquer modificação feita em um algoritmo de aprendizagem com o objetivo de reduzir o erro de generalização, mas não necessariamente o erro de treinamento [3]. Logo esta técnica reduz co-adaptações complexas de neurônios, já que um neurônio não pode confiar na presença de outros neurônios em particular. É, portanto, forçado a aprender recursos mais robustos que são úteis em conjunto com muitos subconjuntos aleatórios diferentes dos outros neurônios. Logo esta técnica é uma forma de garantir que o modelo seja robusto para a perda de qualquer evidência individual.

- *Pooling*: Entre camadas convolucionais sucessivas existem camadas de redução que são utilizadas para diminuir a variância e a complexidade computacional, mantendo as características importantes ao mesmo tempo. A partir destas camadas é possível reduzir a dimensão dos mapas de características, reduzindo por sua vez o número de pesos da rede neural e a utilização da memória da rede, permitindo acrescentar mais características. Esta operação reduz a vizinhança de píxeis a um único píxel. Esta redução pode ser feita utilizando técnicas como o valor máximo da vizinhança ou a média dos valores da vizinhança [29]. O *pooling* ajuda a fazer com que a representação se torne aproximadamente invariantes a pequenas mudanças na entrada. Essa invariância significa que se traduzirmos a entrada por uma pequena quantidade de valores, mesmo assim, os resultados não se alteram. O *pooling* pode complicar alguns tipos de arquiteturas de redes neurais que utilizam informação de valores altos e/ou baixos, tais como máquinas Boltzmann e autoencoders [3]. Mais detalhes sobre esta técnica pode ser encontrado na Seção 3.2.
- Aumento de dados: A técnica de aumento de dados é utilizada para enriquecer o banco de imagens através da adição de dados modificados a partir do banco de imagens original. Através desta adição de dados é possível aumentar o banco de imagens e melhorar a generalização [29]. Dados limitados são um grande obstáculo na aplicação de modelos de aprendizagem profunda, como redes neurais convolucionais. Frequentemente, classes desequilibradas podem ser um obstáculo adicional, embora possa haver dados suficientes para algumas classes, igualmente importantes, mas classes abaixo da amostra sofrerão com baixa precisão específica da classe [51]. As principais técnicas utilizadas no processo de aumento de dados são [52]: transformações geométricas, transformações de espaço de cores, filtros de núcleo, imagens misturadas, *random erasing*, aumento de espaço de recursos, treinamento adversarial, aprimoramento baseado em *generative adversarial networks* (GAN) e transferência de estilo neural. Os detalhes dos tipos de aumento de dados utilizados nesta dissertação está acessível na Seção 4.5.

2.4 Métodos de visão computacional utilizando inteligência artificial

Visão computacional é uma das tarefas mais estudadas no campo da aprendizagem profunda [53]. A evolução da inteligência artificial e da aprendizagem de máquina está a causar mais melhorias e evoluções na tecnologia de visão computacional, proporcionando uma maior confiabilidade nas suas aplicações [54]. A utilização de visão computacional com técnicas de aprendizagem de máquina fornece uma análise profunda da informação de uma imagem, e permite reconhecer os padrões com maior robustez em cenários complexos como variações de angulações, escala e condições de iluminação, partes ocultas, pouco contraste em relação ao fundo da imagem, entre outros [53].

Até 2012, os trabalhos de visão computacional se baseavam, predominantemente, em Máquina de Vetores de Suporte, Florestas Aleatórias e PDI [53]. A partir do trabalho de Krizhevsky, Sutskever and Hinton [45], que venceram a competição *ImageNet Challenge* ao classificar mais de 1000 diferentes classes de imagens com alta performance e precisão, as técnicas de inteligência artificial ganharam mais destaque neste campo [55]. Atualmente, visão computacional é um campo muito amplo, abrangendo variadas formas de processamento de imagens e vídeos, visando a identificação de feições, alocação em classes e até mesmo o desenvolvimento de novas habilidades visuais [3]. Dentre as tarefas mais comuns envolvidas em visão computacional estão a detecção, reconhecimento, segmentação, classificação e localização.

Uma nova abordagem à detecção de objetos chama-se *You Only Look Once* (YOLO), o que significa que uma imagem pode ser predita sobre o que são os objetos e onde estão num relance [56]. O algoritmo é baseado em redes neurais convolucionais e foi o primeiro método de aprendizado profundo a enquadrar a detecção de objetos como um problema de regressão para caixas de delimitação espacialmente separadas e probabilidades de classe associadas, que são previstas com uma única rede neural a partir de imagens completas numa única avaliação [57]. Além disso, a YOLO selecciona GoogLeNet, e não VGG-16 como rede de base, uma vez que as suas taxas de cálculo em relação à precisão é muito mais rápida do que a VGG [57].

O YOLO rompe o limite de velocidade máxima da CNN e realiza um excelente equilíbrio de velocidade e precisão. É extremamente rápido e em tempo real superando os métodos mais avançados como o Faster R-CNN com ResNet. Como limitações a YOLO impõe fortes restrições espaciais às previsões de caixa de delimitação tais como o reconhecimento de pequenos objetos em grupo e ainda se esforça para generalizar objetos novos e em incomuns relações de aspecto ou configurações. Também não é perfeito que a função de perda do YOLO trate os erros da mesma forma em pequenas e grandes caixas de delimitação [53].

A YOLOv3 possui 53 camadas convolucionais e utiliza um extrator de características chamado Darknet-53 do framework Darknet [58]. O Darknet é um framework de código aberto no qual permite customizar parâmetros de acordo com sua capacidade de processamento. É possível visualizar o treinamento por iterações: a cada conjunto de imagens que entra na RNA, é exibido um score de informações da RNA em relação à aquele conjunto [59].

2.5 Método do subespaço

A análise do subespaço em visão computacional é um nome genérico para descrever um quadro geral de comparação e classificação de subespaços [15]. Uma abordagem típica na análise de subespaço é o método de subespaço (SM) que representa cada conjunto de imagens de referência como um subespaço e depois mede cada vetor do conjunto de teste para o subespaço de referência com base nos ângulos entre eles [60].

O método do subespaço é um dos métodos de classificação mais eficazes na análise do subespaço, e foi desenvolvido por dois investigadores japoneses, Watanabe e Iijima, de forma independente [61]. O conceito do método do subespaço deriva da observação de que um conjunto de imagens de um objeto pode ser efetivamente modelado por um subespaço de baixa dimensão de um espaço vetorial de alta dimensão. A tarefa de reconhecimento nesta estrutura é comparar uma imagem de entrada ou um conjunto de imagens de entrada com os conjuntos de imagens de cada classe de referência, e classificar o conjunto de imagens de entrada na classe correta [60].

O SM tem sido conhecido como um dos métodos mais úteis no campo do reconhecimento de padrões, uma vez que o seu algoritmo é muito simples e pode lidar com a classificação de múltiplas classes [17], entretanto, o seu desempenho de classificação não foi suficiente para muitas aplicações na prática, porque os subespaços de classe são gerados independentemente uns dos outros e, com isso, não há razão para assumir a priori que cada subespaço de classe é o subespaço de classe linear ideal em termos de desempenho de classificação [15]. Para lidar com este problema, o SM foi alargado com duas extensões típicas que são o método do subespaço ortogonal e o método do subespaço de aprendizagem. O método do subespaço ortogonal executa o SM para um conjunto de subespaços de classe que são ortogonalizados com base no quadro proposto por Fukunaga e Koontz [62]. Os métodos de subespaço de aprendizagem executam o SM a um conjunto de classes de subespaços, cujos limites são ajustados para suprimir a classificação de erros para os vetores padrão de aprendizagem [15].

2.6 Método do subespaço mútuo

Um dos métodos de correspondência mais desenvolvidos para lidar com um conjunto de vetores de múltiplos padrões como entrada é o método do subespaço mútuo (MSM). O MSM é uma extensão natural do SM, o SM usa um único vetor como entrada, enquanto o MSM usa o subespaço de múltiplos vetores, com isso, o MSM se classifica com base nos ângulos canônicos entre um subespaço de entrada e os subespaços de cada classe. Estes subespaços são gerados a partir dos conjuntos de imagens do objeto desconhecido e os conjuntos de imagens de referência de cada classe, respectivamente.

As características dos objetos de classe SM e os MSM tornam ambos excelentes para lidar com múltiplas classes por sua natureza, ao contrário dos classificadores binários, incluindo o SVM. Esta capacidade de lidar com múltiplas classes é extremamente importante para aplicações de várias classes, tais como reconhecimento facial.

O conceito do MSM está intimamente relacionada com a análise de correlação canônica (CCA). O MSM alcançou alto desempenho em reconhecimento de comple-

xos objetos 3D usando um conjunto de imagens de sequência de imagens ou imagens de vários ângulos do objeto. Isto se deve ao fato de que a semelhança entre duas distribuições de imagens de vários pontos de vista de objetos reflete a semelhança da forma 3D entre os dois objetos. Em particular, o MSM tem sido conhecido por ser adequado para o reconhecimento facial, porque o subespaço que inclui qualquer padrão de imagem facial com todas as possibilidades de condições de iluminação, pode ser gerada a partir de imagens de rosto a partir de três condições de iluminação diferentes.

O método do subespaço também pode ser utilizado em conjunto com técnicas de inteligência artificial baseada em redes neurais convolucionais. O método do subespaço é aplicado antes das camadas convolucionais, contribuindo assim para redução da dimensão da imagem e realizando a extração de características para cada classe [63]. Este procedimento pode ser utilizado em técnicas de reconhecimento em geral, trabalhar com imagens hiper espectrais, entre outras possibilidades.

2.7 Fluxo de processo do método do subespaço mútuo

Todo o processo do MSM consiste em uma fase de aprendizagem e uma fase de reconhecimento [15]. Na fase de aprendizagem todos os subespaços das K classes são gerados a partir de um conjunto de imagens de referências de cada classe e da aplicação da análise dos componentes principais (PCA) [15]. Para isso o primeiro passo é calcular a matriz de autocorrelação dos conjuntos de imagens de cada classe da base de imagens, por fim os vetores de base ortonormal de cada classe são obtidos com os N autovetores correspondentes dos maiores autovalores calculados [60].

Na fase de reconhecimento as similaridades de uma imagem de entrada com todos os K subespaços das classe são calculados, em seguida, a imagem de entrada é classificada na classe do subespaço com maior semelhança, sendo que, se a maior semelhança é inferior a um valor limiar fixado antecipadamente, o vetor de entrada é classificado em uma classe de rejeição [60].

Logo o cálculo da similaridade pode ser analisado geometricamente através da operação da diferença entre os subespaços de entrada e de referência (de cada classe),

sendo assim uma extensão natural do vetor de diferença entre dois vetores [18]. Esta definição é baseada nos vetores canônicos que formam os ângulos canônicos entre os subespaços [60].

Os ângulos canônicos (ou ângulos principais) são então cruciais para os métodos baseados em subespaço porque medem a similaridade/distância entre os conjuntos analisados [17]. Os ângulos canônicos são ordenados de forma que o primeiro par de vetores canônicos calculados formam o menor ângulo canônico, o segundo par de vetores canônicos calculados formam o segundo menor ângulo canônico, e assim por diante [60]. Sejam $\mathbb{V}_1$ e $\mathbb{V}_2$ dois subespaços p -dimensionais de $\mathbb{R}^d$ e V_1, V_2 duas matrizes tais que suas colunas formam sistemas ortonormais para $\mathbb{V}_1, \mathbb{V}_2$, respectivamente. Então o cosseno, $\cos(\theta_i)$, dos ângulos canônicos θ_i entre $\mathbb{V}_1$ e $\mathbb{V}_2$ são os valores singulares de $V_1^T V_2$ [64]. Portanto o cálculo dos ângulos canônicos pode ser resolvido de forma simplificada através da utilização da decomposição do valor singular (SVD).

O SVD pode ser visto como uma etapa de redução da dimensão dos dados, objetivando uma redução significativa no custo computacional. Seja $X_{m \times n}$, com $m \geq n$. Então existem matrizes ortogonais $U_{m \times n}$ e $V_{m \times n}$ tais que $X = UDV^T$, onde as matrizes U e V são matrizes ortogonais e D é uma matriz diagonal que contém os autovalores de X . As colunas das matrizes U e V , são chamados autovetores à esquerda e à direita de X , respectivamente. Aqui U e V podem ser vistos como os vetores de base que abrangem os padrões especiais e sequências temporais, respectivamente [65].

2.8 Trabalhos relacionados

No trabalho proposto por Edson Neto e Samuel Gomes [10] tem-se como objetivo desenvolver um sistema capaz de reconhecer *license plate* de veículos brasileiros para controle de acesso em estacionamentos privados, onde apenas os usuários cadastrados têm permissão para acessar o local. Assim, este trabalho busca uma maneira de tornar o controle de acesso de carros mais preciso e seguro. Neste trabalho foram utilizadas técnicas de processamento digital de imagens, como *Hough transform*, morfologia, limiarização e detector de bordas Canny para extrair carac-

teres, assim como mínimos quadrados, *extreme learning machine* e *neural network multilayer perceptron* para identificar os números e letras. O sistema proposto foi testado com 700 vídeos com resolução de 640×480 pixels e formato AVI, garantindo acesso somente quando a placa foi registrada. Como resultado o sistema alcançou uma taxa de precisão de 97% na extração e reconhecimento de caracteres de números e uma taxa de 96,7% na extração e reconhecimento de caracteres de letras. Assim, o sistema final tem uma taxa de precisão de cerca de 94%.

O trabalho proposto por Rayson Laroca, Evair Severo [12] apresenta um sistema ALPR em tempo real baseado em redes neurais convolucionais (CNNs) e no detector de objetos YOLO. Como neste trabalho é processado quadros de vídeo, foi empregado redundância temporal, de modo que se processa cada quadro de forma independente e, em seguida, combina-se os resultados para criar uma previsão mais robusta para cada veículo. A primeira etapa consistiu em treinar duas redes CNN, sendo uma para realizar a detecção do veículo e a outra para realizar a detecção do *license plate*. Uma vez que o *license plate* foi detectado, foi utilizada uma CNN para segmentação e reconhecimento de caracteres. Para validação do método proposto foram utilizados dois conjuntos de dados. O primeiro conjunto de dados é composto de 2000 quadros de 101 vídeos de veículos, o sistema alcançou uma taxa de reconhecimento de 93,53% e 47 quadros por segundo (FPS). O segundo conjunto de dados utilizado foi o UFPRALPR, projetado para o ALPR. Esse conjunto de dados contém 150 vídeos e 4500 quadros capturados quando a câmera e os veículos estão em movimento e também contém diferentes tipos de veículos (por exemplo, carros, motocicletas, ônibus e caminhões). No conjunto de dados UFPRALPR foi atingida uma taxa de reconhecimento de 78,33% e 35 FPS.

Neste trabalho, proposto por Hui Li e Peng Wang [14], é abordado o problema da detecção e reconhecimento de *license plate* com base em redes neurais profundas. Primeiramente uma rede neural convolucional de 4 camadas e 37 classes (10 dígitos, 26 letras maiúsculas e 1 classe de não-caracteres) é treinada para detectar caracteres em uma imagem, sendo empregado em forma de janela deslizante em toda a imagem para detectar a presença de texto e gerar um mapa de saliência de texto. Em seguida os falsos positivos são eliminados com o auxílio de um classificador binário de *license plate*, sendo capaz de ser usado para detectar *license plate* de vários estilos

(por exemplo, de diferentes países), independentemente das diversas cores, fontes ou tamanhos de placas. Para o reconhecimento de placas é utilizado redes neurais recorrentes com memória de longo curto prazo (LSTM, do inglês *long short term memory*) tendo como grande atrativo a capacidade de realizar o reconhecimento de *license plate* sem a necessidade de segmentação dos caracteres, com isso é possível explorar informações contextuais mais complexas e evitar erros causados pela segmentação, contribuindo para um melhor desempenho e maior precisão na etapa de reconhecimento. Como desvantagem deste trabalho o método de janela deslizante utilizado para detecção de *license plate* é muito lento para aplicação em tempo real. Este método proposto obteve precisão de 92,47% no reconhecimento de caracteres em *license plate*.

A pesquisa apresentada por Rayson Laroca, Luiz Zanlorensi [13] tem como objetivo desenvolver um sistema ALPR baseado totalmente no detector de objetos em tempo real chamado YOLO. O sistema foi desenvolvido de ponta a ponta, passando primeiramente pela detecção de *license plate*, em seguida pela segmentação dos caracteres, e por fim o reconhecimento dos caracteres. Os modelos de CNN empregados foram treinados através da utilização de imagens de vários conjuntos de dados, com a adição de várias técnicas de aumento de dados, para que fosse atingido um alto nível de robustez em diversas condições. O sistema proposto atingiu uma taxa média de reconhecimento de 96,9% em oito conjuntos de dados públicos (de cinco regiões diferentes do mundo) utilizados nos experimentos, superando os trabalhos anteriores em 4 dos conjuntos de imagens utilizados: ChineseLP, OpenALPR-EU, SSIG-SegPlate e UFPR-ALPR. Este trabalho proposto alcançou também taxas expressivas de quadros por segundo (70 FPS) em uma GPU de alta performance (NVIDIA Titan XP), podendo funcionar em tempo real mesmo quando há quatro veículos na cena. Uma contribuição adicional neste trabalho foi que os autores rotularam manualmente 38.351 caixas de delimitação em 6.239 imagens de conjuntos de dados públicos.

Neste trabalho, proposto por Sergio Silva e Claudio Jung, [66], apresenta-se um método ALPR de ponta a ponta baseado em *deep learning*. O sistema apresenta duas redes consideravelmente rápidas e pequenas baseadas em YOLO, operando em modo cascata, com pesos transferidos de redes extensivamente treinadas. A primeira

rede detecta vistas frontais/traseiras de carro e detecta *license plate*, e a segunda detecta e reconhece caracteres dentro da região de *license plate* extraída no primeiro processo. O tempo médio de execução ficou em 13ms utilizando GPU de alta performance. O processo de reconhecimento desenvolvido explora o uso de aumento de dados através da mistura de placas reais com caracteres sintéticos (variando o número de caracteres e tipos/tamanhos de fonte) para lidar com conjuntos de dados de treinamento limitados (todo o sistema foi treinado usando apenas 196 imagens reais), e os resultados mostram que esta metodologia contribuiu de forma significativa para um aumento na precisão nos resultados. Além disso, foi apresentada uma técnica de coerência temporal para estabilizar melhor o reconhecimento de caracteres em vídeos. O método foi testado com conjuntos de dados disponíveis publicamente (SSIG, UFPR-ALPR e OpenALPR) contendo *license plate* brasileiros e europeus, alcançando taxas de precisão de 89,15% no conjunto de dados SSIG, 85,42% no conjunto de dados UFPR-ALPR e 85,19% no conjunto de dados OpenALPR.

Fei Xie e Ming Zhang [11] propuseram um novo algoritmo de detecção de *license plate* e reconhecimento de caracteres baseado em um modelo combinando extração de características e BPNN (*backpropagation neural network*) sendo adaptável em iluminação fraca e cenários complexos. Primeiramente foi aplicado pré-processamento para reforçar a relação de contraste da imagem original do carro. Em seguida as regiões candidatas de *license plate* são encontradas e a região real da *license plate* é localizada com precisão pelo método de projeção integral. Então um novo modelo de extração de características é projetado usando três conjuntos de características (dois conjuntos de características de densidade transversal e um conjunto de características de distância de borda), treinando estes vetores de características através do BPNN para completar o reconhecimento dos caracteres da *license plate*. Para validar os resultados obtidos foram realizados a coleta de 100 imagens de carros, sendo que em 95 as *license plates* foram corretamente detectadas. De um total de 700 caracteres válidos foi verificado que 684 deles são identificados corretamente, o que mostra que a taxa de reconhecimento de caracteres é de 97,7%. O tempo médio de predição foi de 46,1ms.

Na Tabela 2.1 é mostrado um resumo das principais informações dos trabalhos relacionados em comparação com a metodologia proposta da presente pesquisa.

Todos os trabalhos relacionados utilizaram GPUs de alta performance para treinamento e predição, contribuindo para um valor maior de FPS em seus resultados, porém a utilização em aplicações práticas cotidianas e em larga escala sugere a utilização de hardware mais modesto para tornar a aplicação mais viável economicamente. A metodologia proposta utilizou apenas CPU e apresenta tempo de predição consideravelmente menor que os trabalhos relacionados. Logo a grande contribuição desta dissertação é propor um método que consiga realizar o processo completo de reconhecimento de placa veicular em tempo real sem a exigência de grande poder computacional para tal. Outro ponto a considerar é o fato que as bases de dados utilizadas nos trabalhos relacionados apresentam cenários menos complexos que as utilizadas nesta dissertação, sendo que a maioria apresenta as seguintes características: boa qualidade de imagem, vista frontal/dianteira do veículo, veículo e câmera paradas. Enquanto que a base de dados utilizada nesta pesquisa apresentava cenários complexos, veículo e câmera em movimento, posições alternadas do veículo, condições climáticas e de iluminação variadas e alternância na qualidade das imagens.

Tabela 2.1: Comparação entre a metodologia proposta e trabalhos relacionados.

Autor	Base de dados	método de pré processamento	método para detecção de <i>licence plate</i>	método para reconhecimento de <i>licence plate</i>	resultado / acurácia	tempo de predição
Edson Neto e Samuel Gomes [10]	Base de dados de autoria de Neto	conversão para a escala de cinza + detecção de bordas	<i>Hough transform</i> + filtragem por <i>aspect ratio</i>	<i>extreme learning machine</i>	99,00%	50ms
Rayson Laroca, Evair Severo [12]	UFPR-ALPR	detecção do carro com o detector de objetos YOLO	detector de objetos YOLO	CNN	78,33%	28,5ms
Hui Li e Peng Wang [14]	AOLP	Conversão para a escala de cinza	CNN para detecção de texto + classificador binário	redes neurais recorrentes LSTM	92,47%	—
Rayson Laroca, Luiz Zanlorensi [13]	ChineseLP, OpenALPR-EU, SSIG-SegPlate e UFPR-ALPR	detecção do carro com o detector de objetos YOLO	detector de objetos YOLO	segmentação + CNN com coerência temporal	96,90%	14,3ms
Sergio Silva e Claudio Jung [66]	SSIG-SegPlate, UFPR-ALPR e OpenALPR	detector de vistas frontais/traseiras de veículo	detector de objetos YOLO	segmentação + CNN com coerência temporal	92,41%	13ms
Fei Xie e Ming Zhang [11]	base de dados de autoria de Xie	Conversão à escala de cinza + ajuste de contraste + detecção de bordas	operações de erosão e fechamento morfológico + filtragem por <i>aspect ratio</i>	extração de características + redes neurais baseadas em retropropagação	97,70%	46,1ms
Anderson Jesus (método proposto nesta pesquisa)	base de dados de autoria de Garcia	Conversão à escala de cinza + detecção do carro com o detector de objetos YOLO	detector de objetos YOLO	método do subespaço mutuo	57,00%	0,33ms*

Capítulo 3

Metodologia

Neste capítulo, descreve-se o método proposto nesta dissertação. Em síntese, o capítulo está descrito como segue: Na Seção 3.1 é realizado uma introdução dos tópicos que vão ser apresentados no restante do capítulo. Na Seção 3.2 apresenta-se o método para realização do treinamento do reconhecimento de placas veiculares utilizando redes neurais. Na Seção 3.3 é descrito o procedimento para realização do treinamento do reconhecimento de placas veiculares utilizando subespaço.

3.1 Introdução

Na literatura, é possível encontrar diversos métodos de classificação de imagens. Técnicas implementadas utilizando inteligência artificial [67], [28], [68] destacam-se devido a sua grande capacidade de generalização em relação a outros métodos. Para obter grande acurácia as técnicas que utilizam inteligência artificial exigem alto poder computacional e extensas bases de dados, tornando os métodos impraticáveis em várias situações. Uma maneira de contornar isto consiste na aplicação do método de aumento de dados [51] com objetivo de estender as imagens da base de dados e por consequência realizar a replicação de versões diferentes dos objetos de interesse. Métodos por subespaços se destacam justamente por não necessitar de grande poder computacional e conseguir se sobressair mesmo sem grandes bases de dados [69], [70] e [71].

Especificamente neste trabalho, propõe-se uma investigação da utilização de inteligência artificial e do método de subespaço para realização de classificação de

placas veiculares em ambientes complexos. Em uma breve descrição geral: o sistema proposto é composto em quatro etapas. A primeira etapa consiste na criação de um banco de imagens com auxílio de um detector de objetos desenvolvido utilizando *deep learning* [72], [73], [74]. Este banco de imagens, abordado no Capítulo 4, é usado para o treinamento das técnicas de classificação e para execução de testes para validação dos métodos propostos. Na segunda etapa aplica-se, seguindo técnicas de Deep Learning, uma abordagem de aprendizado supervisionado com a arquitetura de redes neurais convolucionais [75], [76], [77] para a classificação de placas de licenciamento veiculares em ambientes complexos. Na terceira etapa utiliza-se o método de subespaço para realizar a classificação de placas de licenciamento veiculares em ambientes complexos através dos ângulos canônicos entre os subespaços não lineares, que são gerados a partir dos padrões de imagem de cada classe. Na quarta etapa, apresentada no Capítulo 5, realiza-se uma análise dos resultados obtidos dos métodos propostos de classificadores desenvolvidos nas etapas anteriores.

3.2 Reconhecimento de placas veiculares utilizando redes neurais convolucionais

Nesta Seção, descreve-se em linhas gerais o classificador para detecção de placas veiculares utilizando redes neurais convolucionais. Inicialmente, é realizada a conversão das imagens de RGB para *grayscale*. A seguir, são executadas, em sequência, as camadas de convoluções e as camadas de *pooling*. A camada de convolução é composta por filtros que possuem como objetivo a extração de características. Aplica-se o processo de *pooling* para reduzir o tamanho das matrizes das imagens. Então as matrizes são transformadas em apenas um vetor e por fim são utilizadas as camadas densas que possuem todos os neurônios, de uma camada, conectados com todas as entradas da camada anterior. Na saída das camadas densas apresenta-se o resultado de classificação do método proposto. Na Figura 3.1 é mostrado o diagrama das etapas do classificador e na Tabela 3.1 apresenta-se as variáveis utilizadas na Seção 3.2. Nas próximas seções detalha-se cada uma das etapas deste classificador.

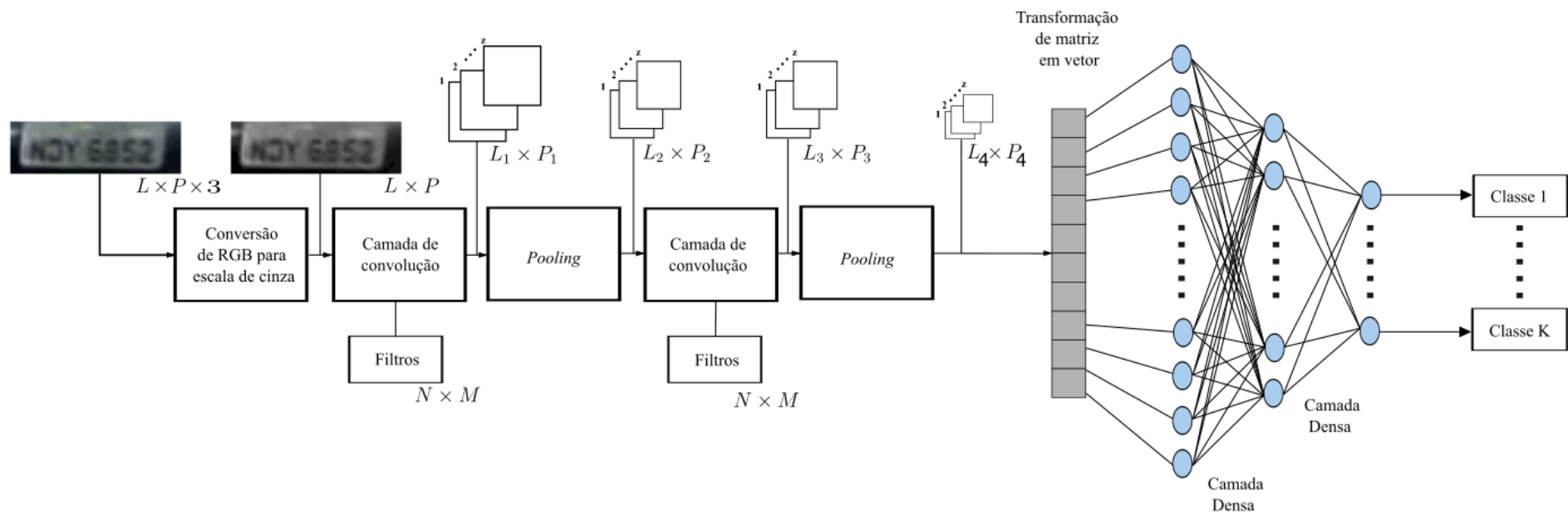


Figura 3.1: Diagrama em blocos do processo de reconhecimento de placas veiculares utilizando CNN.

Tabela 3.1: Variáveis do classificador baseado em redes neurais.

Descrição	Variável
Dimensões dos filtros das camadas de convolução	$N \times M$
Dimensões das imagens	$L \times P$
Dimensões das imagens após a 1° camada de convolução	$L_1 \times P_1$
Dimensões das imagens após a 1° camada de <i>pooling</i>	$L_2 \times P_2$
Dimensões das imagens após a 2° camada de convolução	$L_3 \times P_3$
Dimensões das imagens após a 2° camada de <i>pooling</i>	$L_4 \times P_4$
Quantidade de imagens na partição de treinamento	Z
Quantidade de classes da base de imagens	K
Imagem	I
Dados de entrada da camada densa	D
Filtros da camada convolucional	F

Foi utilizada neste projeto de pesquisa a função de custo chamada entropia cruzada binária. Esta função de custo compara cada probabilidade de classe prevista à saída desejada da classe real 0 ou 1, gerando uma pontuação de penalização com base em quão longe ela está do valor real esperado. Foi utilizado no processo de treinamento das redes neurais o algoritmo de otimização baseado no gradiente descendente conhecido como Adam. O otimizador Adam é um algoritmo baseado em gradiente de primeira ordem de funções objetivas estocásticas, baseado em estimativas adaptativas de momentos de ordem inferior.

Conversão RGB para escala de cinza

Nesta etapa, é realizada a conversão da imagem de RGB para escala de cinza pois esta conversão pode melhorar o contraste da placa veicular e diminuir o tempo de processamento dos algoritmos através da redução da dimensão da imagem. A descrição matemática é feita como segue: primeiramente, supõe-se, como entrada entrada do bloco, uma imagem $I(L_x, P_y) = \{I_R; I_G; I_B\}$ colorida no modelo de cores RGB onde I_i corresponde a matriz da componente de cores i e $L_q \times P_q$ as dimensões das imagens durante os processos das camadas das redes neurais. A imagem em escala de cinza é obtida conforme a Equação (3.1) [20]:

$$I(L_x, P_y) = 0,114I_R(L_x, P_y) + 0,587I_G(L_x, P_y) + 0,299I_B(L_x, P_y) \quad (3.1)$$

onde I é a imagem em escala de cinza, I_R a componente vermelha da imagem RGB, I_G a componente verde da imagem e I_B a componente azul da imagem RGB. A Figura 3.2 ilustra a conversão do espaço de cor RGB para a escala de cinza.

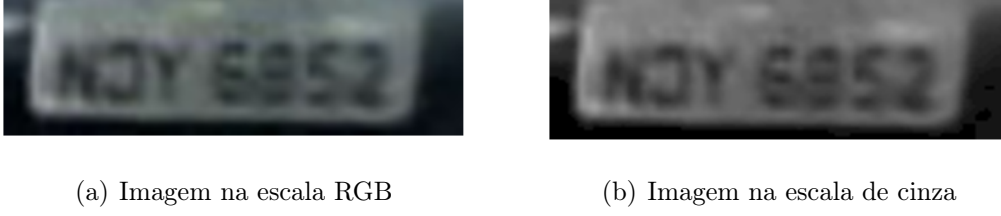


Figura 3.2: Conversão da imagem do espaço de cor RGB para escala de cinza.

Camada de convolução

Para a camada de convolução, considera-se duas entradas, a primeira trata-se da imagem em escala de cinza I e a segunda são os filtros F (conhecido também como *kernel* ou máscara) [78]. As dimensões do filtro definem o tamanho da vizinhança que cada neurônio da camada irá processar. A saída da camada de convolução é referida como mapa de ativação [79], sendo que um mapa de ativação é gerado para cada filtro usado na camada de convolução. Esse processo é realizado conforme a Equação (3.2),

$$I'(x, y) = \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} \sum_{j=-\frac{M}{2}}^{\frac{M}{2}} F(i, j) I(x + i, y + j) \quad (3.2)$$

onde N , M são as dimensões do filtro e x , y a posição dos pixels na imagem.

Os resultados obtidos consistem nas informações que melhor abstraem a entrada, desta maneira somente as características relevantes são propagadas, devido a isso este bloco é um dos mais importante de uma CNN [80]. Os neurônios na primeira camada convolucional não estão conectados a cada pixel da imagem de entrada, mas apenas aos pixels em seus campos receptivos (pequena região delimitada dos dados de entrada). Cada neurônio na segunda camada convolucional por sua vez, está conectado apenas aos neurônios localizados em seus campos receptivos e assim sucessivamente. A rede neural utilizada no sistema proposto possui apenas duas camadas convolucionais.

Camada de *pooling*

Ao se estabelecer a arquitetura da CNN é adicionado uma camada de *pooling* entre camadas de convolução, sendo que a função dessa camada é reduzir a dimensionalidade dos dados na rede [81]. A camada de *pooling* tem a função de agrupar um conjunto de dados (neste trabalho a entrada é dividida em partições de 2×2) e de cada agrupamento é selecionado um valor para os representar. Essa escolha pode ser feita por diversas funções. Neste trabalho utiliza-se a função de máximo.

A função de máximo parte da análise do maior valor de uma região, pois nesse método é considerado que os valores máximos no mapa de características representam aonde estão localizadas as características relevantes [82], e ao aplicar a camada de *pooling* uma porcentagem de informação é descartada. A equação (3.3) descreve o processo de *pooling*, onde $1 \leq i \leq H$, $1 \leq j \leq W$, $I_{x,y}$ é o valor máximo da região particionada, J são os valores dentro das partições. É importante mencionar que a camada de *pooling* não reduz a profundidade da entrada, ela apenas reduz a altura e a largura do conjunto de dados.

$$I_{x,y} = \max(J_{xH+1, yW+1}, \dots, J_{xH+1, yW+W}, \dots, J_{xH+H, yW+W}) \quad (3.3)$$

Transformação de matriz em vetor

Nos processos seguintes é necessário que os dados de entrada estejam em forma de vetor, devido a isso é necessário que os dados que estão em forma de matriz sejam transformados em vetores. Matematicamente, os dados das matrizes seguem as posições conforme a equação (3.4). Com a transformação das matrizes em vetores os dados vão ser reposicionados conforme a equação (3.5).

$$D_{L_{4x} \times P_{4y}} = \{D_{(L_{41}, P_{41})}, \dots, D_{(L_{41}, P_{4y})}, \dots, D_{(L_{42}, P_{4y})}, D_{(L_{4x}, P_{41})}, \dots, D_{(L_{4x}, P_{4y})}\} \quad (3.4)$$

$$D_{1 \times L_{4x} P_{4y}} = \{D_{(1, L_{41}(P_{41}-1)+P_{41})}, D_{(1, L_{42}(P_{41}-1)+P_{41})}, \dots, D_{(1, L_{4x}(P_{4y}-1)+P_{4y})}\} \quad (3.5)$$

Camada densa

As camadas densas se situam ao final da rede, após as principais características dos dados de entrada terem sido extraídas nas camadas convolucionais colocadas

no início da rede [83]. Em resumo, esta camada conecta todos os neurônios da camada anterior a cada um dos T neurônios da camada posterior, sendo que T é igual a K (onde K é a quantidade de classes do conjunto de classificação) na camada de saída.

Algoritmo 1: Camada densa

```

1 para todo neurônio  $j$  da camada totalmente conectada faça
2   para toda entrada  $i$  da camada anterior faça
3      $u_j = \sum_{i=1}^n x_i w_{i,j} + b_j$ 
4      $y_j = f(u_j)$ 
5   fim
6 fim

```

Durante cada época do treinamento, o erro é calculado e propagado de volta, com isso alguns parâmetros como: os pesos da rede neural (sinapses) e os filtros (detectores de características) são ajustados para ajudar a otimizar o desempenho [84]. Inicialmente, os filtros procuram por determinadas características, mas tais características podem não estar corretas, então os filtros são ajustados para procurarem por novas características [85].

3.3 Reconhecimento de placas veiculares utilizando o método do subespaço mútuo

Nesta Seção, descreve-se em linhas gerais o classificador para detecção de placas veiculares utilizando o método de subespaço. O primeiro passo é idêntico ao método de redes neurais convolucionais, ou seja, é realizado a conversão das imagens de RGB para *grayscale* pois informações de cor não contribuem de forma significativa para classificação das placas veiculares. Em seguida, aplica-se um redimensionamento da imagem. Então, cada matriz de imagem é transformada em vetor para, por fim, realizar-se a concatenação dos vetores imagens de cada classe de placa de veículos. Cada conjunto de vetores de classe é decomposto em autovalores e armazenado sua resultante. Na etapa de predição, realiza-se, em uma imagem, todo o processo descrito acima, e após aplica-se o método de subespaço para determinar qual classe possui um ângulo mais próximo da imagem de entrada. Na Figura 3.3 mostra-se o diagrama das etapas do classificador e na Figura 3.4 apresenta-se

o processo de predição utilizando a técnica proposta. Na Tabela 3.2 descreve-se as variáveis utilizadas na Seção 3.3. Nas próximas seções detalha-se as etapas deste método.

Redimensionamento das imagens

Redimensionar uma imagem é o processo de alterar suas dimensões, seja apenas largura, altura ou ambos [86]. Esta alteração pode tanto aumentar as dimensões da imagem quanto diminuir as dimensões da imagem. Além disso, a proporção da imagem original pode ou não ser preservada na imagem redimensionada. Este processo é realizado para que obtenha-se padronização das imagens da base de dados. O tipo de redimensionado utilizado será definido pela técnica de interpolação aplicada [87]. Alguns exemplos de técnica de interpolação são: bicúbico, bilinear, Lanczos e outros. Neste trabalho, utiliza-se a interpolação bilinear. O método bilinear trata-se de uma extensão do método linear, operando no espaço bidimensional. A interpolação bilinear utiliza a média ponderada dos pixels originais que se encontram ao redor do novo pixel [88]. Este método pode ser calculado através da equação (3.6), da seguinte maneira:

$$A_{x,y} = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} H_{11} + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} H_{21} + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} H_{12} + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} H_{22} \quad (3.6)$$

onde o x e y são as posições do pixel a ser interpolado, x_i e y_j são as posições dos pixels conhecidos, e $H_{i,j}$ são os valores dos pixels adjacentes. Apesar da média ser ponderada, o fato de utilizar apenas os quatro vizinhos mais próximo ao seu redor, proporciona à imagem transições mais suaves em locais de altas frequências.

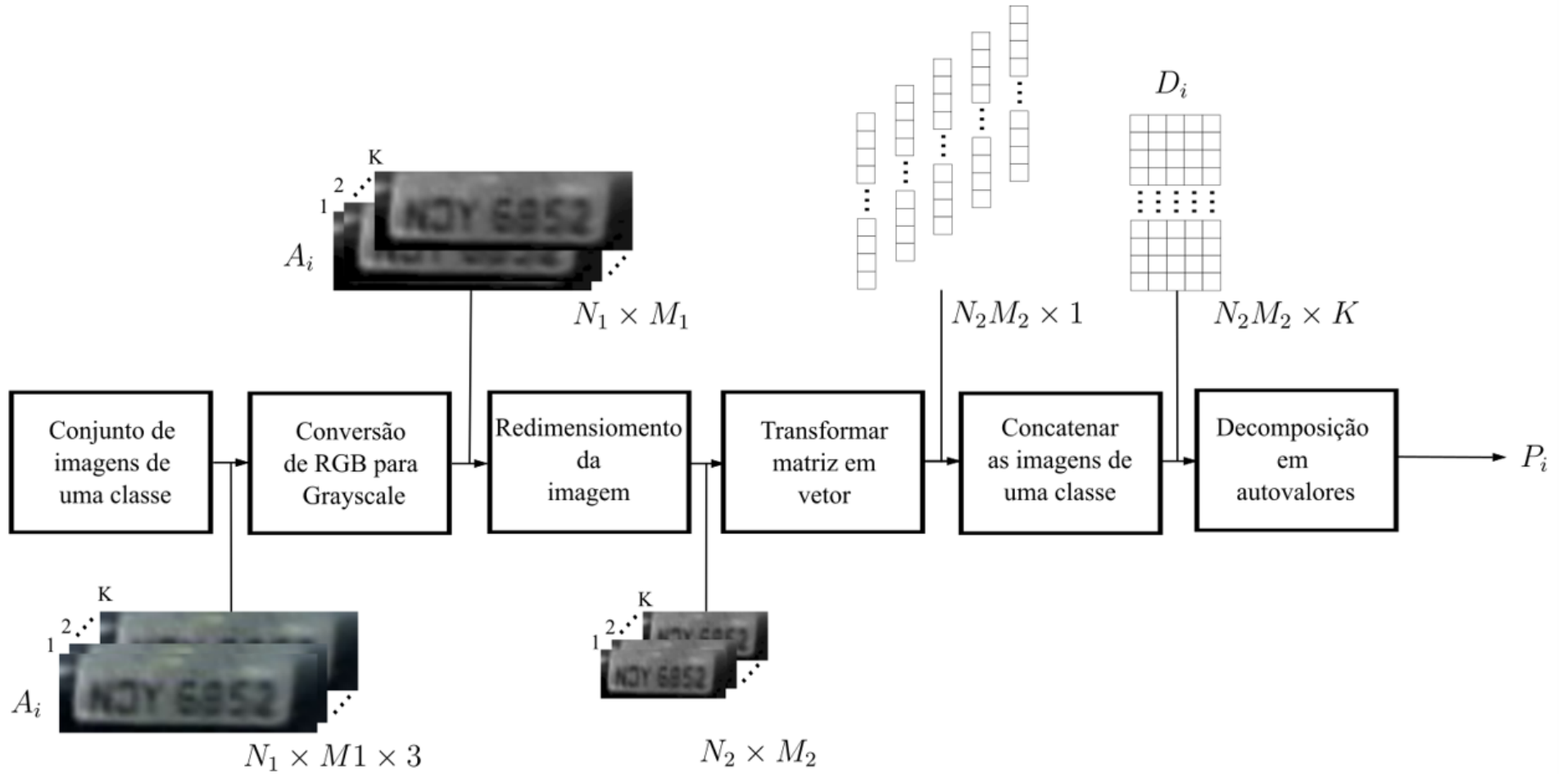


Figura 3.3: Diagrama em blocos das etapas de treinamento do método de classificação por subespaço.

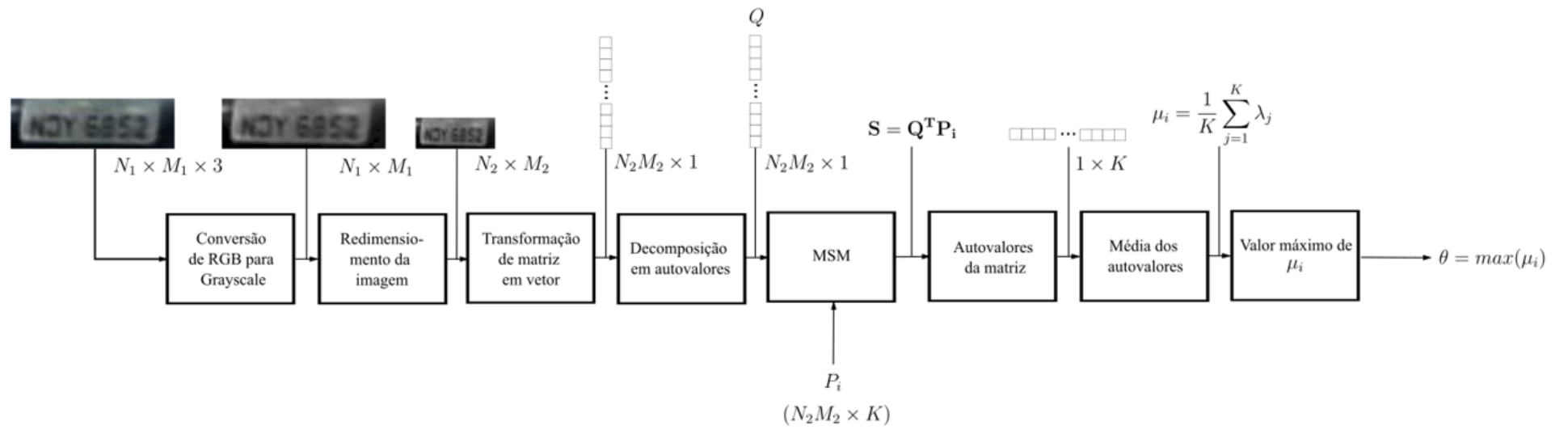


Figura 3.4: Diagrama em blocos das etapas de predição do método de classificação por subespaço.

Tabela 3.2: Variáveis do classificador do método do subespaço mútuo.

Descrição	Variável
Imagens	A
Quantidade de imagens por classe	K
Dimensões das imagens	$N_1 \times M_1$
Dimensões das imagens após o redimensionamento	$N_2 \times M_2$
Matriz com as imagens concatenadas de cada classe	D
Subespaço de cada classe	P
Subespaço da imagem de predição	Q
Matriz de similaridade da imagem de predição	S
Média dos autovalores	μ_i

Concatenação das imagens de uma classe

Após a transformação da matriz de cada imagem em um vetor é realizado o agrupamento de todos os vetores de imagens de uma mesma classe para formar a matriz de dados de cada classe da base de dados. Seja v_j um vetor imagem de determinada classe (D_i) da base de dados e K a quantidade de imagens de cada classe da base de dados. A concatenação dos vetores imagens de uma classe ocorre como mostrado na equação (3.7).

$$D_i = [v_1, v_2, \dots, v_{K-1}, v_K] \quad (3.7)$$

Decomposição em valores singulares

A decomposição em valores singulares, conhecida como SVD (do inglês, *Singular Value Decomposition*), é uma ferramenta importante da análise matricial [89]. O SVD auxilia no processo de ordenação das informações contidas na matriz, com isso parte do seu conteúdo é destacável [90]. O primeiro passo para o cálculo da SVD de uma matriz é a determinação dos seus autovalores através da equação (3.8), conforme a seguir:

$$\det(Q - \lambda I) = 0 \quad (3.8)$$

onde Q é a matriz na qual está sendo calculado o SVD, I é a matriz identidade de mesma dimensão que a matriz Q e λ são os autovalores da matriz Q . Em seguida, é encontrado o autovetor de cada autovalor através da equação (3.9), conforme a seguir:

$$(Q - \lambda_i I)v_i = 0 \quad (3.9)$$

onde λ_i é o autovalor encontrado no primeiro passo e v o autovetor correspondente. Os valores singulares da matriz Q são determinados utilizando a equação (3.10). Logo, com a equação (3.11), é realizado a decomposição em valores singulares da matriz Q .

$$\sigma_i = \sqrt{\lambda_i} \quad (3.10)$$

$$Q = U \Sigma V^T \quad (3.11)$$

Para tal, considera-se U uma matriz ortogonal $M \times M$, cujos elementos são determinados pela equação (3.12). O termo Σ é uma matriz $M \times N$ cujas entradas diagonais são os σ_i valores singulares de Q e as demais entradas são nulas e V uma matriz ortogonal $N \times N$, construída a partir de um conjunto ortonormal de autovetores da matriz Q , $\{v_1, v_2, \dots, v_N\}$.

$$u_i = \frac{1}{\sigma_i} Q v_i \quad (3.12)$$

Método de subespaço mútuo

O método de subespaço mútuo executa o processamento de reconhecimento nas imagens inseridas através do cálculo da similaridade dos valores singulares que foram registrados como o dicionário de cada classe e os valores singulares que foi obtido a partir dos dados de entrada [91]. Sendo d_p a dimensionalidade do subespaço de entrada P e d_q a dimensionalidade do subespaço de referencia Q_i (por conveniência, $d_p \leq d_q$), os ângulos canônicos entre P e Q são definidos através da equação (3.13).

$$\cos^2(\theta_i) = \max \frac{(u_i \cdot v_i)^2}{\|u_i\|^2 \|v_i\|^2} \quad (3.13)$$

onde $u_i \in P$, $v_i \in Q$, $u_i \perp u_j (i \neq j, i, j = 1 \sim d_p)$, $v_i \perp v_j (i \neq j, i, j = 1 \sim d_q)$, $\|u_i\| \neq 0$, $\|v_i\| \neq 0$, com $(\cdot)$ representando o produto interno e $(\| \|)$ representando

a norma dos vetores.

Sendo Φ_i e Ψ_i bases ortonormais dos vetores dos subespaços P e Q respectivamente. Um método prático de encontrar os ângulos canônicos é computando $X = Q^T P$, onde $Q = [\Phi_1, \dots, \Phi_{d_p}]$ e $P = [\Psi_1, \dots, \Psi_{d_q}]$. Com $[k_1, \dots, k_{d_q}]$ sendo os valores singulares da matriz X . Os cossenos dos ângulos canônicos $[\theta_1, \dots, \theta_{d_q}]$ podem ser obtidos como $[k_1, \dots, k_{d_q}]$ [92].

O MSM (*mutual subspace method*) original usa apenas um ângulo canônico mínimo θ_1 para definir a similaridade. No entanto, como os ângulos canônicos restantes também possuem informações para classificação é utilizado a média dos t menores ângulos canônicos como a métrica de semelhança [15]. O valor S da equação (3.14) então reflete a similaridade estrutural entre dois subespaços.

$$S = \frac{1}{t} \sum_{i=1}^t \cos^2(\theta_i) \quad (3.14)$$

Todo o processo do MSM é igual ao do SM (*subspace method*), exceto que um vetor de entrada é substituído por um subespaço de entrada.

Capítulo 4

Base de dados

4.1 Introdução

Os experimentos dessa dissertação foram desenvolvidos utilizando a base de dados criado pelo autor Garcia [22] em sua pesquisa sobre detecção de placas veiculares. No diagrama em blocos da Figura 4.1 se encontra as etapas para a criação da base de dados desta dissertação.

A base de dados original foi criada entre os meses de março e setembro do ano de 2016 na cidade de Manaus. Segundo o autor Garcia [22] a aquisição da base de dados foi realizada portando um aparelho celular com câmera posicionada na parte inferior do banco de uma bicicleta. Além disso, a aquisição dos frames foi concebida com três resoluções diferentes. Desta forma, as aquisições de imagens foram efetuadas percorrendo a ciclofaixa em toda extensão da via nos dois sentidos e em horários distintos com condições ambientais diversas. Na Figura 4.2 é mostrado alguns exemplos de frames da base de imagens original.

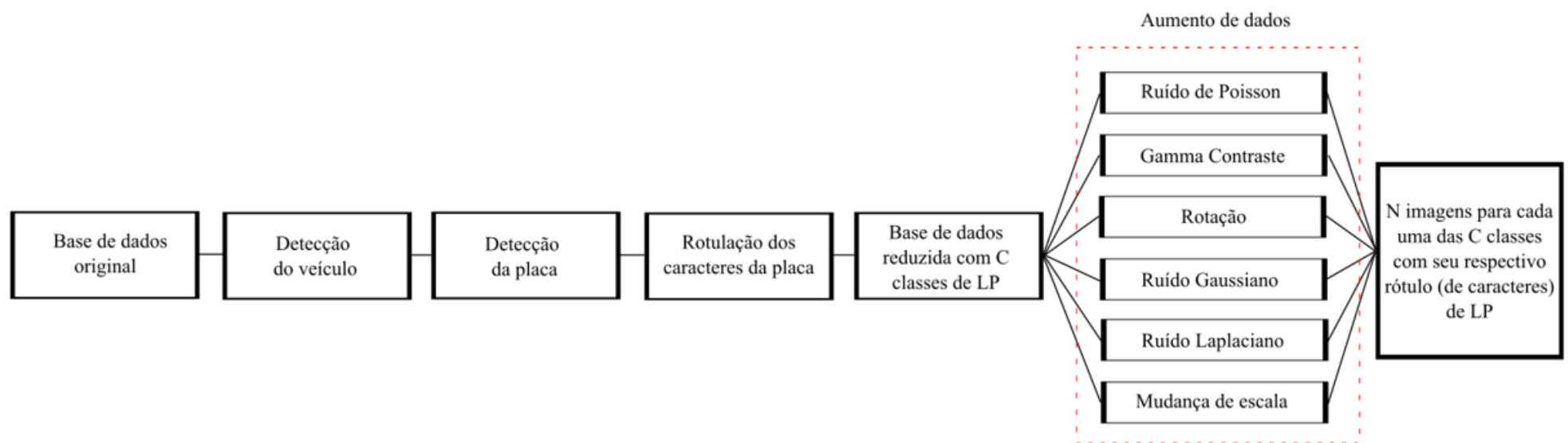


Figura 4.1: Diagrama em blocos do processo de criação da base de dados.



Figura 4.2: Exemplos da base de dados original.

4.2 Detecção do veículo

Como observado os frames dos vídeos dos dados de entrada possuem cenário complexo, o que torna o processo de localização da placa de forma direta bem desafiante, com o intuito de simplificar o processo foi colocado uma etapa intermediária de detecção do veículo antes de ser aplicado a detecção de placa. Para esta finalidade foi utilizada a arquitetura de *deep learning* para detecção de objetos chamada Yolo (You Only Look One) desenvolvida pelos autores Redmon *et al.* [56]. A YOLO utilizada neste trabalho inclui 19 camadas convolucionais e 5 camadas max-pooling, seguidas por funções de ativação softmax para classificar o objeto. A imagem de entrada é dividida em células de grade. A célula da grade, que contém o centro do objeto, é responsável por prever as coordenadas da caixa delimitadora. A YOLO detecta uma série de objetos nas imagens ao mesmo tempo e como em muitos cenários há possibilidade de ter mais do que um veículo na imagem foi definido que a função de detectar o veículo retorna apenas a localização do veículo que se encontra mais próximo da bicicleta.

4.3 Detecção da placa do veículo

Para detecção da placa do veículo foi utilizado um sistema robusto e eficiente baseado no detector de objetos YOLO de última geração que foi desenvolvido pelos autores Laroca *et al.* [12], sendo desenvolvido Redes Neurais Convolucionais (CNNs) treinadas e ajustadas para cada estágio ALPR para que sejam robustas sob diferentes condições (por exemplo, variações na câmera, iluminação e fundo). Este trabalho desenvolveu um sistema que realiza o processo completo de detecção e reconhecimento de placa, porém pra este trabalho só foi utilizado a parte do sistema onde é realizado a detecção da placa. Uma das razões que tornam esse sistema tão robusto foi a utilização da base de dados de várias regiões do mundo (inclusive do Brasil) tornando uma escolha adequada para utilização no sistema proposto.



Figura 4.3: Processo de detecção da placa do veículo.

4.4 Rotulação dos caracteres das placas veiculares

Dando continuidade no processo de criação da base de dados foi realizado a rotulação das imagens de forma visual, identificando os caracteres da placa do veículo correspondente a cada frame. Com isso é determinado N classes (placas de veículos) diferentes nas imagens detectadas do processo anterior. Porém, as classes que tinham uma quantidade de frames inferior a 10 foram retiradas da base de dados.

4.5 Aumento de dados utilizando degradações

Para aplicação do aumento de dados na base de dados original foi utilizada uma série de tipos de degradações que serão apresentadas a seguir. Existem várias maneiras de lidar com complicações associadas a dados limitados no aprendizado de

máquina. O aumento de dados é uma técnica útil na construção de redes neurais convolucionais que podem aumentar o tamanho do conjunto de treinamento sem a aquisição de novas imagens. A ideia é simples, será preciso duplicar as imagens com algum tipo de variação para que o modelo possa aprender com mais exemplos. Idealmente, podemos aumentar a imagem de maneira a preservar os recursos essenciais para fazer previsões, mas reorganizar os pixels o suficiente para adicionar algum ruído. O aumento será contraproducente se produzir imagens muito diferentes daquilo em que o modelo será testado, portanto, esse processo deve ser executado com cuidado. Os objetivos principais de aplicar aumento de dados neste trabalho é aumentar a quantidade de imagens no banco de imagens e testar a robustez do sistema proposto. Neste trabalho foi utilizado os seguintes tipos de degradação:

- Rotação: esta degradação gira aleatoriamente a imagem no sentido horário ou anti-horário por um determinado número de graus [93]. Na equação (4.1) apresenta-se como é realizada uma rotação de ângulo θ da imagem. A transformada é aplicada ponto a ponto na imagem sendo que a intensidade do pixel na posição x, y é colocado na posição u, v .

$$\begin{aligned} u &= x \cos(\theta) + y \sin(\theta) \\ v &= -x \sin(\theta) + y \cos(\theta) \end{aligned} \tag{4.1}$$

A rotação normalmente rotaciona alguns dos pixels para fora do quadro da imagem e deixa áreas do quadro sem dados de pixel que devam ser preenchidos, com isso haveria a possibilidade de perda de pixels dos caracteres da placa. Devido a isso, foi aplicado um método de rotação que não produz perda dos dados, para isto é realizado um aumento no tamanho da imagem para que nenhum dado seja perdido. Na Figura 4.4 mostra-se um exemplo de imagem rotacionada no sentido horário.



Figura 4.4: Exemplo da aplicação de rotação em uma imagem.

- Gaussiano: O ruído gaussiano é um ruído estatístico com uma função de densidade de probabilidade igual à distribuição normal, também conhecida como distribuição gaussiana [94]. Na equação (4.2) apresenta-se a distribuição do ruído gaussiano nas imagens.

$$P(g) = e^{\sqrt{\frac{1}{2\pi\sigma^2}} - \frac{(g-\mu)^2}{2\sigma^2}} \quad (4.2)$$

Na equação (4.2) a variável g representa a intensidade do pixel, σ representa o desvio padrão e μ a média. Esta degradação foi aplicada de forma randômica com desvios padrões entre 1 e 5. Um exemplo da aplicação dessa degradação pode visto na Figura 4.5



Figura 4.5: Exemplo da aplicação de ruído gaussiano em uma imagem.

- Gamma contraste: Esta degradação é aplicada utilizando uma operação não linear para alterar valores de luminância em imagens. Ela é utilizada principalmente para realizar ajuste de contraste em imagens através da equação (4.3), aplicada ponto a ponto, onde I é valor de intensidade original e O é valor de intensidade depois da mudança de um específico γ .

$$O = \left(\frac{I}{255} \right)^\gamma \times 255 \quad (4.3)$$

para valores de γ menores do que 1 é observado um aumento da luminância da imagem, já para valores de γ maiores que 1 é observado uma diminuição da luminância da imagem. Na Figura 4.6 é realizado a aplicação da degradação de contraste com um valor de γ menor do que 1.



Figura 4.6: Exemplo da aplicação de mudança de gamma em uma imagem.

- **Escala:** Esta degradação realiza o aumento ou diminuição do tamanho do objeto na imagem. Para esta pesquisa foi utilizada a interpolação bilinear para a realização do redimensionamento da imagem. Segundo o autor Lenza [95] a interpolação bilinear pode ser usada para estimar o valor de um pixel desconhecido após uma operação geométrica em uma imagem. Ela também é uma técnica básica de resampling em processamento de imagens. A interpolação bilinear considera uma vizinhança 2×2 em volta do pixel de valor desconhecido. A partir desta vizinhança, atribui-se um peso a cada posição e calcula-se a média. Este processo é definido através da equação (4.4), aplicada ponto a ponto na imagem, onde R_1 e R_2 são as médias ponderadas dos pontos vizinhos; e y , y_1 e y_2 representam valores cartesianos.

$$P = \frac{y_2 - y}{y_2 - y_1} \times R_1 + \frac{y - y_1}{y_2 - y_1} \times R_2 \quad (4.4)$$

No cenário do nosso sistema proposto o ideal é trabalhar apenas com a diminuição do objeto para que não seja perdido informações relevantes para a etapa de reconhecimento como pode ser observado através da Figura 4.7.



Figura 4.7: Exemplo da aplicação de mudança de escala em uma imagem.

Para manter a imagem formada por esse processo com o mesmo tamanho das outras imagens é aplicado o processo de zero *padding* [93].

- **Poisson:** Esta degradação adiciona ruído amostrado de distribuições de Poisson elemento a elemento nas imagens como pode ser observada através da

equação (4.5), que descreve a probabilidade de k eventos ocorrerem dentro de uma taxa λ de ocorrência por unidade medida [94]. Como as distribuições de Poisson produzem apenas números positivos, o sinal dos valores amostrados são invertidos aleatoriamente.

$$P(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (4.5)$$

Na Figura 4.8 é mostrado um exemplo deste tipo de degradação.



Figura 4.8: Exemplo da aplicação do ruído Poisson em uma imagem.

- Laplace: Esta degradação adiciona ruído amostrado de distribuições de Laplace elemento a elemento nas imagens definida através da equação (4.6), sendo x a intensidade do pixel, μ o pico da distribuição e λ o decaimento exponencial [94].

$$f(x; \mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|x - \mu|}{\lambda}\right) \quad (4.6)$$

A distribuição de Laplace produz resultados semelhantes à distribuição gaussiana, mas coloca mais peso na cauda longa da distribuição. Portanto, esse ruído adicionará mais outliers (valores muito altos / baixos). É algo entre o ruído gaussiano e o ruído de sal e pimenta como pode ser observado na Figura 4.9.



Figura 4.9: Exemplo da aplicação do ruído laplaciano em uma imagem.

Capítulo 5

Experimentos e Resultados

Nesta Seção, serão demonstradas as etapas experimentais e os resultados obtidos de modo a verificar a metodologia proposta para reconhecimento de placas veiculares. Para efeito de comparação serão realizadas duas abordagens distintas, sendo uma baseada em análise de subespaço e a outra baseada em redes neurais. Na Seção 5.1, é destacado o *setup* dos experimentos identificando as ferramentas utilizadas nessa implementação. Na Seção 5.2 destacam-se os principais questionamentos que esta pesquisa visa responder. Na Seção 5.3, apresentam-se os experimentos realizados neste trabalho de pesquisa. Na Seção 5.4, descreve-se a métrica empregada para avaliar o método proposto. Por fim, na Seção 5.5, realiza-se a análise dos resultados obtidos.

5.1 Setup

Os experimentos desta dissertação foram implementados utilizando o sistema operacional Windows 10 com processador Intel(R) core(TM) i7-10750H CPU @ 2,60Ghz, 16Gb de memória RAM. Para desenvolvimento dos algoritmos, empregamos a linguagem de programação Python [96]. A IDE utilizada foi o PyCharm em conjunto com as bibliotecas OpenCV na versão 4.4.0.46, Numpy na versão 1.19.2 e TensorFlow na versão 2.3.0 [97], [98], [99]. Importante ressaltar que não foi utilizada GPU durante o desenvolvimento desta pesquisa.

5.2 Questionamentos principais

Em nossos experimentos, pretende-se abordar os seguintes questionamentos abaixo:

- Qual o desempenho do método de subespaço considerando-se o problema de reconhecimento de placas veicular?
- Comparando-se com um método de referência que utiliza redes neurais, o desempenho do método de subespaço é superado?
- Qual o desempenho do método de subespaço para o problema de reconhecimento de placas quando considera-se uma entrada com degradação?

5.3 Procedimentos Experimentais

O primeiro passo é particionar a base de dados em três conjuntos disjuntos denominados por: treinamento, validação e testes. A partição de treinamento é usada para treinar o modelo, ou seja, onde as características de cada classe da base de dados será aprendida. A partição de validação é usada para comparar e avaliar se o processo de treinamento atingiu resultados satisfatórios ou se é necessário treinamento adicional. A partição de teste é usada para avaliação de desempenho do modelo. Este particionamento foi utilizado de forma exatamente igual tanto no sistema proposto para reconhecimento de placas veiculares que utiliza o método de subespaço quanto para o método de referência baseado em redes neurais.

A base de dados original possui 32 classes (placas veiculares) e 937 imagens não distribuídas uniformemente como pode ser visto na Tabela 5.5. A base de dados original contempla três resoluções diferentes (426×240 pixels, 854×480 pixels e 1280×720 pixels) e todas utilizaram o mesmo tipo de câmera. Neste trabalho, utiliza-se métodos/algoritmos para aumento de dados [100]. O aumento de dados é utilizado para ampliar a quantidade e diversidade das amostras da base de dados. Com o aumento de dados cada classe passou a possuir 300 imagens e total de imagens foi 9600. Como base de dados de imagens foi dividida em três partes iguais (treinamento, validação e teste), cada parte ficou com 3200 imagens, correspondente a 100 imagens de cada classe. No total, 25 bases de dados de imagens

Tabela 5.1: Degradações aplicadas nas partições de cada base de imagens.

Base de dados	Treinamento	Validação	Teste
1	Gaussiano, Rotação	Poisson, Escala	Contraste, Laplaciano
2	Poisson, Escala	Gaussiano, Rotação	Contraste, Laplaciano
3	Gaussiano, Rotação	Contraste, Laplaciano	Poisson, Escala
4	Poisson, Escala	Contraste, Laplaciano	Gaussiano, Rotação
5	Contraste, Laplaciano	Poisson, Escala	Gaussiano, Rotação
6	Contraste, Laplaciano	Gaussiano, Rotação	Poisson, Escala
7	Poisson	Poisson	Poisson
8	Contraste	Contraste	Contraste
9	Gaussiano	Gaussiano	Gaussiano
10	Laplaciano	Laplaciano	Laplaciano
11	Rotação	Rotação	Rotação
12	Escala	Escala	Escala
13	Todos	Todos	Todos
14	Gaussiano nível 1	Gaussiano nível 1	Gaussiano nível 1
15	Gaussiano nível 2	Gaussiano nível 2	Gaussiano nível 2
16	Gaussiano nível 3	Gaussiano nível 3	Gaussiano nível 3
17	Gaussiano nível 1	Gaussiano nível 2	Gaussiano nível 3
18	Gaussiano nível 1	Gaussiano nível 3	Gaussiano nível 2
19	Gaussiano nível 2	Gaussiano nível 3	Gaussiano nível 1
20	Gaussiano nível 2	Gaussiano nível 1	Gaussiano nível 3
21	Gaussiano nível 3	Gaussiano nível 2	Gaussiano nível 1
22	Gaussiano nível 3	Gaussiano nível 1	Gaussiano nível 2
23	Gaussiano nível 1	Gaussiano nível 2, 3	Gaussiano nível 2, 3
24	Gaussiano nível 2	Gaussiano nível 1, 3	Gaussiano nível 1, 3
25	Gaussiano nível 3	Gaussiano nível 1, 2	Gaussiano nível 1, 2

foram criadas com diferentes variações do tipo de aumento de dados aplicado nas imagens como pode ser observado na Tabela 5.1.

Em todas as bases de dados foram aplicados métodos de aumento de dados para se ter uma quantidade suficiente de dados para realizar as técnicas propostas neste trabalho. As degradações aplicadas (nas imagens) no aumento de dados foram: rotação [93], ruído Gaussiano [94], mudança de escala [95], ruído de Poisson [94], mudança de contraste e ruído Laplaciano [94].

Observa-se através da Tabela 5.1 que nas bases de imagens de 1 a 6 foram aplicados tipos de aumento de dados diferentes para cada partição (treinamento, validação e teste). O objetivo de introduzir degradações diferentes em cada partição da base de dados é verificar se as características aprendidas para o reconhecimento das classes são de fato as características principais da classe ou características da degradação aplicada na classe de treinamento.

Nas bases de imagens de 7 a 13 da Tabela 5.1 foi utilizado o mesmo tipo de

degradação nas três partições da base de dados. O objetivo nessas bases de dados é analisar o comportamento dos métodos propostos para cada tipo de degradação para concluir se os métodos utilizados são robustos o suficiente em todos as degradações testadas. Destaque para a base de imagens 13 que utilizou todas os seis tipos de degradações para a realização do aumento de dados.

Para a realização da base de imagens da 14 até a 25 foi escolhido uma das degradações para realizar uma análise mais aprofundada de como a intensidade da degradação pode alterar os resultados obtidos nos métodos propostos. A degradação escolhida foi o ruído Gaussiano como pode ser observado na Tabela 5.1. O nível 1 acrescenta de 0,5% a 4% de ruído Gaussiano na imagem. O nível 2 adiciona de 4% a 8% de ruído Gaussiano na imagem. No nível 3 é acrescentado entre 8% e 12% de ruído Gaussiano na imagem. Logo o nível 1 é o que apresenta a menor intensidade de degradação na base de dados e o nível 3 é o que possui maior intensidade de degradação.

5.4 Métricas de desempenho

Considerou-se para a avaliação do desempenho do modelo proposto a métrica de acurácia. A acurácia indica um desempenho geral do modelo como é mostrado na equação (5.1).

$$A = \frac{\text{número de frames classificados corretamente}}{\text{número total de frames}} \quad (5.1)$$

Esta métrica é mais indicada em base de dados com a mesma proporção de exemplos para cada classe, e quando as penalidades de acerto e erro para cada classe forem as mesmas, logo se encaixa bem nos experimentos realizados nesta pesquisa.

5.5 Análise dos resultados

5.5.1 Resultados da técnica baseada em subespaços

A Tabela 5.2 apresenta os resultados obtidos utilizando o método de subespaço nas bases de dados. A coluna parâmetro mostra a dimensionalidade de entrada

Tabela 5.2: Resultados da técnica baseada em subespaços.

Base de dados	Parâmetros	Acurácia de validação	Acurácia de teste
1	7	0,39	0,50
2	17	0,39	0,57
3	14	0,54	0,37
4	19	0,58	0,39
5	6	0,40	0,41
6	5	0,42	0,39
7	7	0,56	0,52
8	8	0,60	0,55
9	8	0,55	0,51
10	8	0,57	0,51
11	11	0,52	0,49
12	19	0,59	0,56
13	10	0,52	0,50
14	6	0,56	0,52
15	10	0,55	0,50
16	8	0,56	0,51
17	14	0,55	0,52
18	11	0,56	0,52
19	11	0,56	0,50
20	8	0,56	0,51
21	8	0,55	0,52
22	8	0,55	0,51
23	14	0,56	0,52
24	8	0,55	0,51
25	8	0,54	0,52

do método de subespaço que apresentou melhor resultado na base de validação durante o treinamento. Os parâmetros foram variados entre os valores 5 e 20.

O melhor resultado de acurácia (igual a 60%) para as bases de validação foi obtido para a base de dados 8, sendo que nesta base de dados foi aplicada degradação por mudança de contraste nas três partições (treinamento, validação, teste). O melhor resultado de acurácia (igual a 57%) para as bases de teste foi obtido para a base de dados 2, sendo que nesta base de dados foi aplicada: degradação por mudança de escala e ruído de Poisson na base de treinamento; degradação por rotação e ruído Gaussiano na base de validação; e degradação de mudança de contraste e ruído laplaciano na base de testes. Em contrapartida a base de dados 2 apresentou juntamente com a base de dados 1 o pior resultado na acurácia (igual 39%) na base de validação. Já a base de dados 3 foi a que apresentou o pior resultado

na acurácia (igual 37%) na base de teste.

Nos resultados da Tabela 5.2 foi observado que as bases de dados 1, 2, 3 e 4 foram as que apresentaram maior discrepância entre a acurácia da base de validação e da base de testes. Como exemplo a base de dados 4 apresentou acurácia na base de validação de 58% e acurácia na base de testes de 39%, tendo assim uma diferença de 19% entre os dois resultados. Já era esperado que as 6 primeiras base de dados apresentassem as maiores discrepâncias devido ao fato que para cada partição da base de dados utilizou-se degradações diferentes, com isso aumentando a complexidade do problema.

Entre as bases de dados 7 e 25 da Tabela 5.2 foi constatado uma notável proximidade entre a acurácia da base de validação e a acurácia da base de testes. Teoricamente isso já era esperado que essas bases de dados apresentassem resultados mais próximos devido ao fato que foram utilizados os mesmos tipos de degradações em todas as partições das bases. É observado também que a acurácia da base de validação é levemente superior a acurácia da base de testes, o que comprova que a escolha do parâmetro com melhor resultado no processo de validação não garante o melhor caso de treinamento executado, mas apresenta resultados bem próximos disso.

Importante destacar os resultados obtidos na base 13 da Tabela 5.2 que aplicou todos os tipos de degradações disponíveis e ainda assim apresentou performance de precisão condizente com as outras bases de dados, comprovando a capacidade da solução proposta de extrair as características importantes da base de dados. Os resultados obtidos nas bases de dados entre 17 e 25 da Tabela 5.2 também comprovam esta capacidade de abstração das informações principais já que mesmo utilizando níveis diferentes de ruído nas partições das bases de dados, os resultados obtidos de acurácia se mantiveram constantes.

Cada predição levou em média 0,33ms. Levando em consideração que o método mais rápido dentro dos trabalhos relacionados teve tempo de predição de 13ms, podemos concluir que o método proposto neste trabalho é cerca de 50x mais veloz que os métodos atuais utilizados. Ainda é importante destacar que os trabalhos relacionados utilizaram GPU na etapa de predição enquanto que no método proposto nesta pesquisa foi utilizado apenas CPU.

5.5.2 Resultados da técnica baseada em redes neurais

A tabela 5.3 apresenta os resultados obtidos utilizando a técnica baseada em redes neurais. Em todos os bancos de imagens os modelos foram treinados durante 50 épocas e foram salvos os pesos da época que apresentaram melhor resultado.

O melhor resultado de acurácia (igual a 94%) para as bases de validação foi obtido para a base de dados 2, sendo que nesta base de dados foi aplicada: degradação por mudança de escala e ruído de Poisson na base de treinamento; degradação por rotação e ruído Gaussiano na base de validação; e degradação de mudança de contraste e ruído laplaciano na base de testes. Para as bases de teste o melhor resultado de acurácia (igual a 99%) obtido foi para a base de dados 11, sendo que nesta base de dados foi aplicada degradação de rotação nas três partições (treinamento, validação, teste).

Tabela 5.3: Resultados da técnica baseada em redes neurais.

Base de dados	Acurácia de validação	Acurácia de teste
1	0,73	0,84
2	0,94	0,95
3	0,93	0,86
4	0,93	0,86
5	0,84	0,96
6	0,90	0,71
7	0,77	0,88
8	0,86	0,81
9	0,82	0,77
10	0,84	0,86
11	0,91	0,99
12	0,91	0,93
13	0,92	0,98
14	0,85	0,90
15	0,68	0,78
16	0,80	0,92
17	0,84	0,91
18	0,84	0,90
19	0,83	0,82
20	0,70	0,71
21	0,82	0,86
22	0,85	0,94
23	0,80	0,86
24	0,81	0,81
25	0,88	0,91

A base de dados 15 da Tabela 5.3 apresentou o pior resultado de acurácia (igual a 68%) na base de validação, sendo que nesta base de dados foi aplicada degradação de ruído gaussiano nas três partições (treinamento, validação, teste). As bases de dados 6 e 20 da Tabela 5.3 tiveram os piores resultados de acurácia (igual a 71%) na base de testes. Ainda sim o pior resultado das técnicas baseadas em redes neurais apresentou acurácia maior do que o melhor resultado das técnicas baseadas em subespaços.

A base de dados 5 foi a base que mostrou a maior discrepância positiva (foi de 84% para 96%) entre a acurácia de validação e a acurácia de testes, enquanto que a base de dados 6 foi a base que mostrou a maior discrepância negativa (foi de 90% para 71%) entre a acurácia de validação e a acurácia de testes.

As bases de dados entre 14 e 25 da Tabela 5.3 possuem apenas degradações do tipo gaussiano (em variados níveis) os resultados das acurácias foram no geral abaixo do que nas outras bases de dados, comprovando que variações de degradações contribui para uma melhor aprendizagem das técnicas baseadas em redes neurais.

Cada predição levou em média 200ms. Isto é quase 600x mais lento do que a predição utilizando o método do subespaço. Esta predição também foi consideravelmente mais lento que os tempos de predição dos trabalhos relacionados. Porém é importante frisar novamente que os métodos desenvolvidos nesta pesquisa não utilizaram GPU, o que explica esse desempenho pior no tempo de predição do método baseado em redes neurais.

5.5.3 Resultados por classe com variações nos tipos de degradações

Para um melhor entendimento dos resultados obtidos utilizando o método do subespaço foram escolhidas algumas bases de dados para se realizar uma análise das acurácias por cada classe da base de dados.

Na Tabela 5.4 foram analisadas as bases de dados na qual foi aplicada individualmente as seguintes degradações: Rotação (base de dados 8), contraste (base de dados 9), ruído gaussiano (base de dados 11) e todas as degradações (base de dados 13). Os valores de precisão mostrados nesta tabela são relativos a partição de testes dos bancos de imagens.

Tabela 5.4: Análise dos resultados por classe de variados tipos de degradações.

Classe	Quantidade de frames	Acurácia nas bases de dados			
		Base de dados 13	Base de dados 11	Base de dados 9	Base de dados 8
1	12	0,52	0,41	0,32	0,41
2	11	0,95	0,95	0,54	0,85
3	21	0,42	0,45	0,60	0,63
4	13	0,75	0,84	0,76	0,99
5	19	0,21	0,41	0,40	0,65
6	30	0,89	0,90	1,00	1,00
7	15	0,09	0,09	0,00	0,30
8	21	0,39	0,28	0,26	0,16
9	23	0,55	0,66	0,73	0,68
10	17	0,52	0,41	0,56	0,48
11	19	0,85	0,86	0,86	1,00
12	10	0,01	0,00	0,00	0,00
13	77	0,74	0,73	0,90	0,79
14	19	0,68	0,51	0,67	0,63
15	27	0,64	0,61	0,83	0,61
16	19	0,06	0,14	0,01	0,16
17	26	0,64	0,79	0,89	0,86
18	230	0,96	0,86	0,99	0,96
19	11	0,02	0,03	0,01	0,01
20	11	0,92	0,95	1,00	1,00
21	13	0,02	0,00	0,00	0,00
22	12	0,34	0,24	0,00	0,39
23	19	0,62	0,82	1,00	0,98
24	34	0,75	0,71	0,85	0,66
25	20	0,36	0,20	0,22	0,18
26	75	0,45	0,31	0,64	0,54
27	12	0,02	0,01	0,01	0,01
28	22	1,00	1,00	1,00	1,00
29	38	0,74	0,68	1,00	1,00
30	25	0,23	0,09	0,12	0,32
31	24	0,25	0,34	0,14	0,29
32	12	0,36	0,24	0,01	0,01

A classe 28 obteve os melhores resultados nas bases de dados mostradas na Tabela 5.4, alcançando 100% de acurácia em todas as 4 bases de dados analisadas, enquanto que a classe 12 obteve os piores resultados, tendo valores de acurácia próximos de 0% em todas as bases de dados analisadas.

Importante notar que no método de subespaço a qualidade das amostras impactam mais os resultados do que a quantidade de amostras. Isso fica evidente ao verificar na Tabela 5.4 que a classe 18 que tem a maior quantidade de frames (230 amostras) possui aproximadamente 10x mais amostras do que a classe 28 que apresentou os melhores resultados. Já a classe com a menor quantidade de frames foi a 12, com apenas 10 frames, que também foi a que teve os piores resultados, mas podemos ver que isso não é determinante analisando a classe 20 que possui apenas um frame a mais, porém a precisão está acima de 90% em todas as 4 bases de dados analisadas (inclusive alcançando 100% em duas delas). Portanto podemos concluir que o método de subespaço precisa de poucas amostras da classe pra atingir resultados significativos, tornando uma boa opção para aplicações onde não é possível criar grandes bases de dados.

Analisando os resultados de acurácia por classes na Tabela 5.4 fica claro que as bases de dados possuem notável oscilação de qualidade nas suas amostras, pois houve de forma recorrente resultados com alta, intermediária e baixa acurácia nos resultados obtidos. No geral os valores de acurácia das classes se mantiveram próximos mesmo com as variações de degradações em cada base de dados, mas vale destacar que a classe 5 foi a classe que mostrou maior variação em seus resultados, apresentando acurácia de 21% na base de dados 13 e 65% na base de dados 8, ou seja, 44% de flutuação nos resultados.

5.5.4 Resultados por classe com variações nos níveis de degradações

Na Tabela 5.5 foi feita uma análise por classe dos resultados obtidos utilizando a mesma degradação (ruído gaussiano) mas com níveis diferentes de intensidade. Sendo pertinente lembrar que o ruído gaussiano de nível 1, da base de dados 14, possui variação de ruído entre 0% e 4%; o ruído gaussiano de nível 2, da base de dados 15, possui variação de ruído entre 4% e 8%; e o ruído gaussiano de nível 3, da

base de dados 16, possui variação de ruído entre 8% e 12%.

Tabela 5.5: Análise dos resultados por classe de variados níveis de degradação do ruído Gaussiano.

Classe	Quantidade de frames	Acurácia nas bases de dados		
		Base de dados 14	Base de dados 15	Base de dados 16
1	12	0,40	0,42	0,34
2	11	0,58	0,58	0,53
3	21	0,68	0,59	0,47
4	13	0,99	0,99	0,99
5	19	0,42	0,33	0,38
6	30	1,00	1,00	1,00
7	15	0,00	0,00	0,22
8	21	0,48	0,16	0,25
9	23	0,65	0,64	0,71
10	17	0,45	0,51	0,52
11	19	0,90	0,90	0,98
12	10	0,00	0,00	0,00
13	77	0,87	0,98	0,96
14	19	0,62	0,65	0,75
15	27	0,63	0,79	0,56
16	19	0,01	0,01	0,10
17	26	0,94	0,77	0,78
18	230	0,99	0,97	0,99
19	11	0,01	0,01	0,01
20	11	1,00	1,00	1,00
21	13	0,00	0,00	0,00
22	12	0,09	0,04	0,07
23	19	0,98	0,98	1,00
24	34	0,74	0,80	0,75
25	20	0,29	0,23	0,25
26	75	0,68	0,51	0,64
27	12	0,01	0,01	0,00
28	22	1,00	1,00	1,00
29	38	1,00	1,00	1,00
30	25	0,31	0,15	0,01
31	24	0,17	0,29	0,32
32	12	0,01	0,00	0,00

Os melhores resultados obtidos foram encontrados nas classes: 6, 20, 28 e 29; na qual obteve-se 100% de acurácia nestas classes nas três bases de dados analisadas na Tabela 5.5. A quantidade de amostras nestas classes com melhores resultados variou de 11 a 38 imagens, mostrando novamente que não é necessário uma grande quantidade de amostras para que o método de subespaço atinja resultados robustos. Em contrapartida, os piores resultados foram encontrados nas classes 12 e 21 que obtiveram acurácia de 0% nas três bases de dados analisadas, destacando-se também

as classes 7, 27 e 32 que atingiram valores de acurácia bem próximos de 0%.

Observando os melhores e piores resultados por classe na Tabela 5.5 fica claro que ou as classes apresentaram ótimos resultados ou fracos resultados, obtendo como resultado final valores intermediários devido a média ter sido utilizada para obter a acurácia das bases de dados, mas se fosse analisar por outras métricas que utilizam o desvio padrão então se notaria que os resultados obtidos nas bases da Tabela 5.5 foram inferiores aos resultados obtidos na Tabela 5.4.

A classe 8 apresentou a maior variação entre os seus resultados, tendo 16% de acurácia na base de dados 15 e 48% de acurácia na base de dados 14, ou seja, obtendo 32% de flutuação nos resultados. Interessante notar que classe 12 se manteve como uma das classes que apresentaram pior resultados e a classe 20 se manteve como umas das classes que apresentaram melhores resultados, mostrando que o nível de degradação não afeta de forma significativa os resultados obtidos com o método do subespaço.

Capítulo 6

Conclusão

Neste trabalho, propomos conceber um modelo para a realização do reconhecimento de placas veiculares brasileiras em cenários complexos. Para alcançar este objetivo foi utilizada a abordagem do método do subespaço mútuo. Como *baseline* foi empregada uma técnica baseada em redes neurais convolucionais para o reconhecimento de placas veiculares. Na abordagem do método de subespaço utilizou-se de forma a realizar a classificação de placas de licenciamento veiculares em ambientes complexos através dos ângulos canônicos entre os subespaços não lineares, que são gerados a partir dos padrões de imagem de cada classe. Na abordagem de redes neurais convolucionais seguiu-se a técnica de aprendizado supervisionado para a classificação de placas de licenciamento veiculares em cenários complexos.

Foi concebido uma base de imagens de placas veiculares para servir como apoio para a pesquisa proposta. Estes foram os seguintes passos para a sua criação: emprego do detector de objetos YOLO para detecção de automóveis nas imagens; utilização do detector de objetos YOLO para detecção de placas veiculares nas imagens; rotulação manual dos caracteres das placas de cada imagem; e aplicação de técnicas de aumento de dados para diversificar e ampliar a base de imagens. No total, 25 bases de dados de imagens foram criadas com diferentes variações do tipo de aumento de dados aplicado nas imagens. Cada base de dados ficou com 32 classes, cada classe possui 300 imagens, totalizando 9600 imagens. As bases de dados foram divididas igualmente em três partes (treinamento, validação e testes) com 3200 em cada partição. Os tipos de degradações aplicados foram: ruído gaussiano, ruído de Poisson, ruído laplaciano, mudança de escala, rotação e mudança de contraste.

Em termos de resultados, a primeira abordagem utilizando o método do subespaço teve como melhor resultado a acurácia de 57% e tempo médio de predição de 0,33ms, enquanto que a segunda abordagem baseada em redes neurais obteve como melhor resultado a acurácia de 94% e tempo médio de predição de 200ms. Podemos concluir que a abordagem utilizando redes neurais obteve valores de acurácia consideravelmente maiores, porém o método de subespaço teve como grande vantagem o tempo de predição que foi em média 600 vezes menor.

A acurácia do método do subespaço nas 25 base de imagens se mantiveram no geral de forma constante, comprovando a capacidade desta solução proposta de extrair as características importantes da base de imagens, eliminando as degradações. No geral os resultados de acurácia obtidos com o método de redes neurais foram superiores ao método de subespaço, porém foi observado que em bases de imagens na qual não havia grande variações de degradações a técnica baseada em redes neurais teve dificuldade em manter a mesma performance que obteve em base de imagens com grandes variações de degradações, comprovando que este método necessita de grandes bases de dados. Foi interessante notar que no método de subespaço a qualidade das amostras impactam muito mais os resultados do que a quantidade de amostras, pois analisando os resultados por classe observou-se que a classe que obteve melhor performance possuía 10 vezes menos amostra do que a classe com maior quantidade de amostras. Outro ponto importante observado ao analisar os resultados por classe foi que o nível de degradação não afeta de forma significativa os resultados obtidos com o método do subespaço.

Com estes resultados concluímos que o método de subespaço é uma boa opção em aplicações: onde possuímos limitação de hardware, que são em tempo real, onde não é exigido grande acurácia, onde não é possível criar grandes bases de dados. Enquanto que a abordagem com redes neurais é uma boa opção em aplicações: onde contém hardware de alta performance (GPUs), onde existe uma grande e variada base de imagens, onde é exigido grande acurácia.

Como trabalhos futuros é vislumbrado algumas possibilidades:

- Aplicar um outro método para detecção de placas veiculares com menor custo computacional do que a YOLO, como por exemplo, utilizando a técnica de super resolução proposta por Luiz Junior [22];
- Testar a solução proposta em outras bases de imagens, não se restringindo apenas a placas veiculares brasileiras. Entre as bases de dados estudadas neste projeto de pesquisa destaca-se a UFPR-ALPR que é composta de placas veiculares brasileiras e a SSIG-SegPlate que possui placas veiculares europeias;
- Aumentar os tipos e níveis de degradações aplicadas nas bases de imagens, podendo em alguns casos misturar tipos de degradações para aumentar a complexidade do problema;
- Empregar redundância temporal, de modo que se processa cada quadro de forma independente e, em seguida, combina-se os resultados para criar uma previsão mais robusta para cada veículo;
- Investigar a utilização de outros métodos baseados em subespaço como, por exemplo, o *constrained mutual subspace method* (CMSM) que descarta vetores de bases que possuem grandes variações ou o *orthogonal mutual subspace method* onde a relação entre os subespaços de classe são modificado para utilizar a ortogonalização na fase de aprendizagem.

Referências Bibliográficas

- [1] SZELISKI, R., *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [2] MARQUES FILHO, O., NETO, H. V., *Processamento digital de imagens*. Brasport, 1999.
- [3] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., *Deep learning*. MIT press, 2016.
- [4] VIJAYAKUMAR, T., “Neural network analysis for tumor investigation and cancer prediction”, *Journal of Electronics*, v. 1, n. 02, pp. 89–98, 2019.
- [5] GUANG, X., GAO, Y., LEUNG, H., et al., “An autonomous vehicle navigation system based on inertial and visual Sensors”, *Sensors*, v. 18, n. 9, pp. 2952, 2018.
- [6] OLIVEIRA, B. A., CARDOSO, G. M., DIAS, F. P., et al., “Sistema para Classificação de Graos de Café: Uma Comparação Entre Redes Neurais Convolucionais e Processamento Digital de Imagens com MLP”, 2019.
- [7] YOGHEEDHA, K., NASIR, A., JAAFAR, H., et al., “Automatic vehicle license plate recognition system based on image processing and template matching approach”. In: *Proc. Int. Conf. on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pp. 1–8, 2018.
- [8] XIE, L., AHMAD, T., JIN, L., et al., “A new CNN-based method for multi-directional car license plate detection”, *IEEE Transactions on Intelligent Transportation Systems*, v. 19, n. 2, pp. 507–517, 2018.

- [9] ASHTARI, A. H., NORDIN, M. J., FATHY, M., “An Iranian license plate recognition system based on color features”, *IEEE transactions on intelligent transportation systems*, v. 15, n. 4, pp. 1690–1705, 2014.
- [10] NETO, E. C., GOMES, S. L., REBOUÇAS FILHO, P. P., et al., “Brazilian vehicle identification using a new embedded plate recognition system”, *Measurement*, v. 70, pp. 36–46, 2015.
- [11] XIE, F., ZHANG, M., ZHAO, J., et al., “A robust license plate detection and character recognition algorithm based on a combined feature extraction model and BPNN”, *Journal of Advanced Transportation*, v. 2018, 2018.
- [12] LAROCA, R., SEVERO, E., ZANLORENSI, L. A., et al., “A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector”. In: *Proc. Int. Conf. on Neural Networks (IJCNN)*, pp. 1–10, July 2018.
- [13] LAROCA, R., MENOTTI, D., “Automatic License Plate Recognition: An Efficient and Layout-Independent System Based on the YOLO Detector”. In: *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, pp. 15–21, SBC: Porto Alegre, RS, Brasil, 2020.
- [14] LI, H., WANG, P., YOU, M., et al., “Reading car license plates using deep neural networks”, *Image and Vision Computing*, v. 72, pp. 14–23, 2018.
- [15] FUKUI, K., “Subspace Methods”, *Computer Vision: A Reference Guide*, pp. 1–5, 2020.
- [16] GATTO, B. B., DOS SANTOS, E. M., DA SILVA, W. S., “Orthogonal hankel subspaces for applications in gesture recognition”. In: *Proc. Int. Conf. on Graphics, Patterns and Images (SIBGRAPI)*, pp. 429–435, 2017.
- [17] TAN, H., GAO, Y., MA, Z., “Regularized constraint subspace based method for image set classification”, *Pattern Recognition*, v. 76, pp. 434–448, 2018.
- [18] YAGI, Y., KANG, S. B., KWEON, I. S., et al., *Computer Vision-ACCV 2007: 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18–22, 2007, Proceedings, Part II*. v. 4844. Springer, 2007.

- [19] PEDRINI, H., SCHWARTZ, W. R., *Análise de imagens digitais: princípios, algoritmos e aplicações*. Cengage Learning, 2008.
- [20] GONZALEZ, R. C., WOODS, R. C., *Processamento digital de imagens*. Pearson Educação, 2009.
- [21] TOSTA, T. A. A., *Método computacional para segmentação não supervisionada de imagens histológicas de linfoma*, Dissertação de Mestrado, Universidade Federal de Uberlândia, 2016.
- [22] JUNIOR, G., DA SILVA, L. C., OTHERS, *Detecção de Placas de Licenciamento Veicular Utilizando Imagens com Diferentes Resoluções e em Ambiente Não Controlado Empregando Técnicas de Super Resolução*, Dissertação de Mestrado, Universidade Federal do Amazonas, 2019.
- [23] DE OLIVEIRA, D. L. L., DO NASCIMENTO, M. Z., NEVES, L. A., et al., “Unsupervised segmentation method for cuboidal cell nuclei in histological prostate images based on minimum cross entropy”, *Expert Systems with Applications*, v. 40, n. 18, pp. 7331–7340, 2013.
- [24] PRATT, W., *Digital image processing: Paks scientific inside*. wiley-interscience, john wiley & sons, inc. 2007.
- [25] GONÇALVES, J. C., *Reconhecimento de dígitos em imagens de medidores de consumo de gás natural utilizando técnicas de visão computacional*, Dissertação de Mestrado, Universidade Tecnológica Federal do Paraná, 2016.
- [26] GANESAN, P., SAJIV, G., “A comprehensive study of edge detection for image processing applications”. In: *Proc. Int. Conf. on innovations in information, embedded and communication systems (ICIIECS)*, pp. 1–6, 2017.
- [27] ARNAB, A., ZHENG, S., JAYASUMANA, S., et al., “Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction”, *IEEE Signal Processing Magazine*, v. 35, n. 1, pp. 37–52, 2018.

- [28] ABBAS, A., ABDELSAMEA, M. M., GABER, M. M., “Classification of COVID-19 in Chest X-ray Images Using DeTraC Deep Convolutional Neural Network”, *Applied Intelligence*, v. 51, n. 2, pp. 854–864, 2021.
- [29] MARTINS, R. A. S., *Sistema de contagem de pessoas na biblioteca da FEUP utilizando visão computacional*, Dissertação de Mestrado, 2021.
- [30] BAY, H., TUYTELAARS, T., VAN GOOL, L., “Surf: Speeded up robust features”. In: *European conference on computer vision*, pp. 404–417, 2006.
- [31] ROSTEN, E., DRUMMOND, T., “Machine learning for high-speed corner detection”. In: *European conference on computer vision*, pp. 430–443, 2006.
- [32] RUBLEE, E., RABAUD, V., KONOLIGE, K., et al., “ORB: An efficient alternative to SIFT or SURF”.
- [33] GAZALBA, I., REZA, N. G. I., OTHERS, “Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification”. In: *Proc. Int. Conf. on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 294–298, 2017.
- [34] VAPNIK, V., IZMAILOV, R., “Rethinking statistical learning theory: learning using statistical invariants”, *Machine Learning*, v. 108, n. 3, pp. 381–423, 2019.
- [35] LIU, S., WANG, X., LIU, M., et al., “Towards better analysis of machine learning models: A visual analytics perspective”, *Visual Informatics*, v. 1, n. 1, pp. 48–56, 2017.
- [36] PRANCKEVIČIUS, T., MARCINKEVIČIUS, V., “Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification”, *Baltic Journal of Modern Computing*, v. 5, n. 2, pp. 221, 2017.
- [37] YUAN, J., CHEN, C., YANG, W., et al., “A survey of visual analytics techniques for machine learning”, *Computational Visual Media*, v. 7, n. 1, pp. 3–36, 2021.

- [38] TAIGMAN, Y., YANG, M., RANZATO, M., et al., “Deepface: Closing the gap to human-level performance in face verification”. In: *Proc. Int. Conf. on computer vision and pattern recognition*, pp. 1701–1708, 2014.
- [39] FERNÁNDEZ-DELGADO, M., SIRSAT, M. S., CERNADAS, E., et al., “An extensive experimental survey of regression methods”, *Neural Networks*, v. 111, pp. 11–34, 2019.
- [40] CHANDOLA, V., BANERJEE, A., KUMAR, V., “Anomaly detection: A survey”, *ACM computing surveys (CSUR)*, v. 41, n. 3, pp. 1–58, 2009.
- [41] MCCULLOCH, W. S., PITTS, W., “A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysics*, v. 5, n. 4, pp. 115–133, 1943.
- [42] DE OLIVEIRA, R. A., OTHERS, *Agente de aprendizado por reforço tabular para negociação de ações*, Dissertação de Mestrado, Universidade Federal de Minas Gerais, 2020.
- [43] BILAL, A., JOURABLOO, A., YE, M., et al., “Do convolutional neural networks learn class hierarchy?” *IEEE transactions on visualization and computer graphics*, v. 24, n. 1, pp. 152–162, 2017.
- [44] LECUN, Y., BOTTOU, L., BENGIO, Y., et al., “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324, 1998.
- [45] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E., “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, v. 25, pp. 1097–1105, 2012.
- [46] SZEGEDY, C., LIU, W., JIA, Y., et al., “Going deeper with convolutions”. In: *Proc. Int. Conf. on computer vision and pattern recognition*, pp. 1–9, 2015.
- [47] SITAULA, C., HOSSAIN, M. B., “Attention-based VGG-16 model for COVID-19 chest X-ray image classification”, *Applied Intelligence*, v. 51, n. 5, pp. 2850–2863, 2021.

- [48] HE, K., ZHANG, X., REN, S., et al., “Deep residual learning for image recognition”. In: *Proc. Int. Conf. on computer vision and pattern recognition*, pp. 770–778, 2016.
- [49] FERNANDES, E., *Posicionamento de caminhões autônomos para colheita de cana-de-açúcar: uma abordagem baseada em visão computacional*, Dissertação de Mestrado, Centro Universitário FEI, São Bernardo do Campo, 2020.
- [50] OOSTWAL, E., STRAAT, M., BIEHL, M., “Hidden unit specialization in layered neural networks: ReLU vs. sigmoidal activation”, *Physica A: Statistical Mechanics and its Applications*, v. 564, pp. 125517, 2021.
- [51] SHORTEN, C., KHOSHGOFTAAR, T. M., “A Survey on Image Data Augmentation for Deep Learning”, *Journal of Big Data*, v. 6, n. 1, pp. 1–48, 2019.
- [52] CLARO, M., VOGADO, L., SANTOS, J., et al., “Utilização de Técnicas de Data Augmentation em Imagens: Teoria e Prática”, *Sociedade Brasileira de Computação*, 2020.
- [53] OLIVATTO, T. F., *Identificação automática de rampas de acessibilidade apoiada por visão computacional a partir de imagens panorâmicas street-level*, Dissertação de Mestrado, Universidade Federal de São Carlos, 2021.
- [54] SUMA, V., “Computer vision for human-machine interaction-review”, *Journal of trends in Computer Science and Smart technology (TCSST)*, v. 1, n. 02, pp. 131–139, 2019.
- [55] MARQUES, L. C., ULSON, J. A. C., “A APLICAÇÃO DE REDES NEURAIAS PROFUNDAS PARA DETECÇÃO E CLASSIFICAÇÃO DE PLANTAS DANINHAS E SEU ESTADO DA ARTE”, *REGRAD-Revista Eletrônica de Graduação do UNIVEM-ISSN 1984-7866*, v. 11, n. 01, pp. 391–403, 2018.

- [56] REDMON, J., DIVVALA, S., GIRSHICK, R., et al., “You only look once: Unified, Real-time Object Detection”. In: *Proc. Int. Conf. on computer vision and pattern recognition*, pp. 779–788, 2016.
- [57] DU, J., “Understanding of object detection based on CNN family and YOLO”. In: *Journal of Physics: Conference Series*, v. 1004, n. 1, p. 012029, 2018.
- [58] VIEIRA FILHO, J., JASKOWIAK, P. A., “Comparação de Métodos de Deep Learning Pré-Treinados da Biblioteca OpenCV para Detecção de Pessoas em Ambientes Internos”, *Revista Eletrônica de Iniciação Científica em Computação*, v. 18, n. 4, 2020.
- [59] OLIVO, R. Y., DE PAULA FILHO, P. L., JUNIOR, A. C., “Uma abordagem neural na identificação de objetos em imagens para auxílio na manutenção de rede elétrica”. In: *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, pp. 179–182, 2020.
- [60] FUKUI, K., MAKI, A., “Difference subspace and its generalization for subspace-based methods”, *IEEE transactions on pattern analysis and machine intelligence*, v. 37, n. 11, pp. 2164–2177, 2015.
- [61] OJA, E., *Subspace methods of pattern recognition*. v. 6. Research Studies Press, 1983.
- [62] FUKUNAGA, K., KOONTZ, W. L., “Application of the Karhunen-Loeve expansion to feature selection and ordering”, *IEEE Transactions on computers*, v. 100, n. 4, pp. 311–318, 1970.
- [63] ALIPOURFARD, T., AREFI, H., MAHMOUDI, S., “A novel deep learning framework by combination of subspace-based feature extraction and convolutional neural networks for hyperspectral images classification”. In: *Proc. Int. Symp. geoscience and remote sensing*, pp. 4780–4783, 2018.
- [64] MOURA, L., *Agrupamento espectral através de grafos Laplacianos e uma aplicação no cultivo da soja.*, Dissertação de Mestrado, Universidade Estadual Paulista (UNESP), 2018.

- [65] DA SILVA, L. R., LUIZ, R. D. A., FARIAS, V. J. D. C., et al., “Aplicação da decomposição em valores singulares e análise de componentes independentes em dados de fMRI”, *Proc. of the Brazilian Society of Computational and Applied Mathematics*, v. 3, n. 1, 2015.
- [66] SILVA, S. M., JUNG, C. R., “Real-time license plate detection and recognition using deep convolutional neural networks”, *Journal of Visual Communication and Image Representation*, v. 71, pp. 102773, 2020.
- [67] AGARWAL, M., OTHERS, “A Study on Image Analysis and Recognition Using Learning Methods: CNN as the Best Image Learner”, In: *Data Analytics and Management*, pp. 23–30, Springer, 2021.
- [68] SUDHAKARAN, S., ESCALERA, S., LANZ, O., “Learning to Recognize Actions on Objects in Egocentric Video with Attention Dictionaries”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [69] ZHU, Y., LIU, J., CHEN, S., “Semi-random Subspace Method for Face Recognition”, *Image and vision computing*, v. 27, n. 9, pp. 1358–1370, 2009.
- [70] YANG, J.-M., KUO, B.-C., YU, P.-T., et al., “A Dynamic Subspace Method for Hyperspectral Image Classification”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 48, n. 7, pp. 2840–2853, 2010.
- [71] HEFETZ, E. F., CHIEN, E., WEBER, O., “A Subspace Method for Fast Locally Injective Harmonic Mapping”. In: *Computer Graphics Forum*, v. 38, n. 2, pp. 105–119, 2019.
- [72] WU, X., SAHOO, D., HOI, S. C., “Recent Advances in Deep Learning for Object Detection”, *Neurocomputing*, v. 396, pp. 39–64, 2020.
- [73] WANG, W., LAI, Q., FU, H., et al., “Salient Object Detection in The Deep Learning Era: An in-depth Survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [74] XIAO, B., KANG, S.-C., “Development of an Image Data Set of Construction Machines for Deep Learning Object Detection”, *Journal of Computing in Civil Engineering*, v. 35, n. 2, pp. 05020005, 2021.

- [75] HECK, L., KÜNAS, C., PADOIN, E., “Análise de Desempenho de Redes Neurais Convolucionais Aplicadas ao Reconhecimento de Emoções”. In: *Anais da XXI Escola Regional de Alto Desempenho da Região Sul*, pp. 41–44, 2021.
- [76] YONEKURA, D., GUEDES, E., “Uma Abordagem Híbrida para Detecção de Moedas de Real Baseada em Algoritmos de Localização e Redes Neurais Convolucionais”. In: *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional*, pp. 140–151, 2020.
- [77] SOUZA, V., SILVA, L., SANTOS, A., et al., “Análise Comparativa de Redes Neurais Convolucionais no Reconhecimento de Cenas”, *Anais do Computer on the Beach*, v. 11, n. 1, pp. 419–426, 2020.
- [78] KHUMAIIDI, A., YUNIARNO, E. M., PURNOMO, M. H., “Welding Defect Classification Based on Convolution Neural Network (CNN) and Gaussian Kernel”. In: *Int. Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 261–265, 2017.
- [79] JEFF, H., GOODFELLOW, I., BENGIO, Y., et al., “Deep Learning”. pp. 305–307, Springer, 2018.
- [80] TORTELLI, L. M., LORETO, A. B., DE AGUIAR, M. S., “Extensão Intervalar da Camada de Convolução de Redes Neurais Convolucionais”, *Proc. of the Brazilian Society of Computational and Applied Mathematics*, 2017.
- [81] VARGAS, A. C. G., PAES, A., VASCONCELOS, C. N., “Um Estudo Sobre Redes Neurais Convolucionais e Sua Aplicação em Detecção de Pedestres”. In: *Proc. Int. Conf. on Graphics, Patterns and Images*, v. 1, n. 4, 2016.
- [82] WU, J., “Introduction to Convolutional Neural Networks”, *National Key Lab for Novel Software Technology. Nanjing University. China*, v. 5, pp. 23, 2017.
- [83] LI, Z., WANG, T., LI, W., et al., “Deep Multilayer Ffusion Dense Network for Hyperspectral Image Classification”, *IEEE Journal of Selected Topics in*

Applied Earth Observations and Remote Sensing, v. 13, pp. 1258–1270, 2020.

- [84] EBERMAM, E., KROHLING, R., “Uma Introdução Compreensiva às Redes Neurais Convolucionais: Um Estudo de Caso para Reconhecimento de Caracteres Alfabéticos”, *Revista de Sistemas de Informação da FSMA*, v. 22, pp. 49–59, 2018.
- [85] WU, F., WANG, J., LIU, J., et al., “Vulnerability Detection with Deep Learning”. In: *Proc. Int. Conf. on Computer and Communications (ICCC)*, pp. 1298–1302, 2017.
- [86] WANG, Q., YUAN, Y., “Learning to Resize Image”, *Neurocomputing*, v. 131, pp. 357–367, 2014.
- [87] AKYAMA, M. T., *Interpolação de Imagens Baseada em Clustering*, Dissertação de Mestrado, Universidade Tecnológica Federal do Paraná, 2010.
- [88] DOURADO, W. B., *Avaliação de Técnicas de Interpolação de Imagens Digitais*, Dissertação de Mestrado, Universidade Estadual Paulista (UNESP), 2014.
- [89] DE ANDRADE, G. C., SANTOS, S. A., “Decomposição em Valores Singulares e Técnicas de Compressão de Dados”, *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 7, n. 1, 2020.
- [90] EN, L., “Matrix Methods in Data Mining and Pattern Recognition (Fundamentals of Algorithms)”, *Society for Industrial and Applied Mathematics, Philadelphia, PA, USA*, 2007.
- [91] SAKANO, H., MUKAWA, N., NAKAMURA, T., “Kernel Mutual Subspace Method and its Application for Object Recognition”, *Electronics and Communications in Japan (Part II: Electronics)*, v. 88, n. 6, pp. 45–53, 2005.
- [92] FUKUI, K., YAMAGUCHI, O., “The Kernel Orthogonal Mutual Subspace Method and its Application to 3D Object Recognition”. In: *Asian Conference on Computer Vision*, pp. 467–476, 2007.

- [93] PFLUGFELDER, D., SCHARR, H., “Practically Lossless Affine Image Transformation”, *IEEE Transactions on Image Processing*, v. 29, pp. 5367–5373, 2020.
- [94] ABRAMOWITZ, M., STEGUN, I. A., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. v. 55. US Government printing office, 1964.
- [95] LENZA, T. P., “Um Algoritmo Direcionado a Bordas Que Utiliza Segmentação Para Interpolação de Imagens Digitais”, 2017.
- [96] HILL, C., *Learning scientific programming with Python*. Cambridge University Press, 2020.
- [97] SIGUT, J., CASTRO, M., ARNAY, R., et al., “OpenCV basics: a mobile application to support the teaching of computer vision concepts”, *IEEE Transactions on Education*, v. 63, n. 4, pp. 328–335, 2020.
- [98] HARRIS, C. R., MILLMAN, K. J., VAN DER WALT, S. J., et al., “Array programming with NumPy”, *Nature*, v. 585, n. 7825, pp. 357–362, 2020.
- [99] SINGH, P., MANURE, A., “Introduction to tensorflow 2.0”, In: *Learn TensorFlow 2.0*, pp. 1–24, Springer, 2020.
- [100] HUSSAIN, Z., GIMENEZ, F., YI, D., et al., “Differential data augmentation techniques for medical imaging classification tasks”. In: *AMIA annual symposium proceedings*, v. 2017, p. 979, 2017.