



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Classificação de Sinais de Trânsito em Direção Autônoma por Redes Neurais
Convolucionais

Nicomar Fernandes de Oliveira

Orientadores

Carlos Alberto Vieira Campos
Pedro Nuno de Souza Moura

RIO DE JANEIRO, RJ - BRASIL
MARÇO de 2020

Classificação de Sinais de Trânsito em Direção Autônoma por Redes Neurais
Convolucionais

Nicomar Fernandes de Oliveira

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Em conformidade com a Resolução nº5.257 de 25/03/2020, esta ata vai somente por mim assinada, atestando que a defesa ocorreu com a participação dos componentes abaixo listados.

Aprovada por:



D.Sc. Carlos Alberto Vieira Campos - UNIRIO

D.Sc. Pedro Nuno de Souza Moura - UNIRIO

D.Sc. Sidney Cunha de Lucena - UNIRIO

Ph.D. Daniel Sadoc Menasche - UFRJ

RIO DE JANEIRO, RJ - BRASIL
MARÇO de 2020

Oliveira, Nicomar Fernandes de.
B118 Classificação de Sinais de Trânsito em Direção Autônoma por Redes Neurais Con-
volucionais /
Nicomar Fernandes de Oliveira, 2020,
73f.

Orientador: Carlos Alberto Vieira Campos
Coorientador: Pedro Nuno de Souza Moura
Dissertação (Mestrado em Informática) - Universidade Federal do
Estado do Rio de Janeiro, Rio de Janeiro, Programa de Pós-Graduação
em Informática, 2020.

1. Aprendizado de Máquina 2. Redes Neurais Convolucionais
3. Classificação de Sinais de Trânsito. I. Vieira Campos, Carlos Alberto,
orient. II. Moura, Pedro Nuno de Souza, coorient. III. Título.

CDD - 004.678

Dedico essa dissertação aos meus pais e filhos.

Agradecimentos

Agradeço aos meus pais Gumerindo e Maria, pela educação que me deram. Eles me apoiaram em todas minhas etapas escolares durante a minha vida, sempre priorizando a educação.

Agradeço a minha esposa e aos meus filhos que continuam me incentivando.

Agradeço ao meu professor e orientador, Carlos Alberto Vieira Campos, pelo ensino de excelência, pela dedicação, orientação, incentivo e compreensão durante este curso de mestrado.

Agradeço ao meu co-orientador, Pedro Nuno de Souza Moura, pelo apoio na conclusão da minha pesquisa, em que o tema abordado é especialista.

Agradeço aos professores da Unirio pelo ensino das matérias acadêmicas, ao professor Sidney Cunha de Lucena, em que eu passei a me interessar por esta pesquisa e ao meu colega de classe Elton Figueiredo de Souza Soares, pelo seu apoio durante todo curso.

Na empresa de trabalho agradeço a Joao Carlos Tosta e a Eduardo Fritzen pelo incentivo, a Tiago Vieira Melo, Luciana Perciliano, Patricia Zudio e Anderson Leite pelo incentivo e apoio.

Agradeço a comunidade científica pela fonte de pesquisa.

Oliveira, Nicomar Fernandes de. **Classificação de Sinais de Trânsito em Direção Autônoma por Redes Neurais Convolucionais**. UNIRIO, 2020. 70 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Sinais de trânsito são elementos fundamentais que servem de alerta e orientação à direção autônoma de veículos. Dado isso, um dos principais problemas de pesquisa abordados na literatura relacionada é o de classificação automática de sinais de trânsito. A alta variabilidade de placas e regras de trânsito dificultam o desenvolvimento de soluções genéricas, tornando necessária a aplicação de algoritmos de aprendizado de máquina do estado da arte. Além disso, uma vez que veículos autônomos podem ser utilizados em diferentes países, onde os sinais de trânsito podem ter diferentes cores, símbolos e significados, é necessário que os classificadores sejam capazes de se adaptar de acordo com o conjunto de sinais de trânsito de cada país. Dado isso, este trabalho propõe o treinamento de modelos de aprendizado de máquina utilizando datasets públicos de placas de trânsito de diferentes países, com o objetivo de gerar classificadores suficientemente robustos para utilização em diferentes países. Assim, são apresentadas diferentes estratégias para treinamento dos modelos de classificação de modo a permitir a sua avaliação e comparação. Nesse sentido, é apresentada uma avaliação experimental das diferentes estratégias de treinamento, incluindo treinamento com dataset de cada país, combinação de datasets 2 a 2, e por fim, treinamento com um dataset gerado a partir da união de todos os datasets. Além disso, são realizadas comparações entre os resultados obtidos e também com outros trabalhos. Como resultado deste trabalho, foi possível obter um mapeamento das equivalências entre as classes de sinais de trânsito de diferentes países e a identificação da melhor estratégia para treinamento de classificadores de sinais de trânsito de vários países.

Palavras-chave: Classificação, Sinais de Trânsito, Aprendizado Profundo, Direção Autônoma, Aprendizado de Máquina.

ABSTRACT

Traffic signs are fundamental elements that serve as a warning and guidance to the autonomous driving of vehicles. Given this, one of the main research problems addressed in the related literature is the automatic classification of traffic signs. The high variability of signs and traffic rules hinder the development of generic solutions, making it necessary to apply state-of-the-art machine learning algorithms. In addition, since autonomous vehicles can be used in different countries, where traffic signs can have different colors, symbols and meanings, it is necessary that classifiers be able to adapt according to the set of traffic signs of each country. Given this, this work proposes the training of machine learning models using public datasets of traffic signs from different countries, in order to generate classifiers sufficiently robust for use in different countries. Thus, different strategies are presented for training the classification models in order to allow their evaluation and comparison. An experimental evaluation of the different training strategies is presented, including training with datasets from each country, combination of datasets 2 to 2, and finally, training with a dataset generated from the union of all datasets. In addition, comparisons are made between the results obtained and also with other works. As a result of this work, it was possible to obtain a mapping of the equivalences between the traffic sign classes of different countries and the identification of the best strategy for training traffic sign classifiers in several countries.

Keywords: Classification, Traffic Signs, Deep Learning, Autonomous Driving, Convolutional Neural Networks.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Problema	2
1.3	Objetivo	3
1.4	Contribuição	3
1.5	Organização	4
2	Conceitos Preliminares	5
2.1	Inteligência Artificial e Aprendizado de Máquina	5
2.2	Redes Neurais	8
2.2.1	Proposta de McCulloch-Pitts	8
2.2.2	Perceptron	9
2.3	Redes Neurais Artificiais	9
2.3.1	<i>Backpropagation</i>	10
2.3.2	Data Augmentation	10
2.3.3	Overfitting	11
2.3.4	Regularização	11
2.3.5	Hiperparâmetros	11
2.4	Redes Neurais Convolucionais	18
2.4.1	Camadas	19
2.4.2	Classificação	22
3	Trabalhos relacionados	23
3.1	Conceitos Relevantes	27
3.1.1	Aprendizado por Máquina Adversária	28
3.1.2	Adaptação de Domínio e Aprendizado por Transferência	29
4	Método Proposto	36
4.1	Pré-processamento	36
4.2	Modelo CNN	37
4.3	Treinamento	38
5	Experimentos	40
5.1	Ambiente Computacional	40
5.2	Datasets	41
5.2.1	Dataset Alemão	42
5.2.2	Dataset Belga	42
5.2.3	Dataset Croata	43
5.3	Hiperparâmetros	44

5.4	Métricas	45
5.5	Estratégias de treinamento	45
5.6	Experimento 1 - Treinamento Individual	46
5.6.1	Dataset Alemão	47
5.6.2	Dataset Belga	48
5.6.3	Dataset Croata	49
5.6.4	Resumo	51
5.7	Experimento 2 - Treinamento com datasets para classes equivalentes simples	51
5.7.1	Alemão x Belga	53
5.7.2	Alemão x Croata	55
5.7.3	Belga x Alemão	56
5.7.4	Belga x Croata	58
5.7.5	Croata x Alemão	59
5.7.6	Croata x Belga	60
5.7.7	Resumo	62
5.8	Experimento 3 - Treinamento com datasets para classes equivalentes completas	63
5.8.1	Alemão x Belga	63
5.8.2	Alemão x Croata	65
5.8.3	Belga x Alemão	66
5.8.4	Belga x Croata	68
5.8.5	Croata x Alemão	70
5.8.6	Croata x Belga	71
5.8.7	Resumo	73
5.9	Experimento 4 - Treinamento do dataset único	74
5.9.1	Teste das bases Alemã, Belga, Croata	74
5.9.2	Teste da base Alemã	77
5.9.3	Teste da base Belga	78
5.9.4	Teste da base Croata	80
5.10	Análise e discussão de resultados	81
6	Conclusão	84
7	Apêndice	93
7.1	Matrizes Confusão da Estratégia Individual	93
7.2	Matrizes Confusão. Equivalência Simples	97
7.3	Matrizes Confusão. Equivalência Completa	99
7.4	Matrizes Confusão. Estratégia Única	101

Lista de Figuras

2.1	Visão global da Inteligência Artificial. Baseado em [55].	6
2.2	Visão do aprendizado de máquina com seus tipos e respectivos algoritmos. Baseado em [38].	7
2.3	Exemplos de redes neurais. Baseado em [49].	8
2.4	Perceptron	9
2.5	Fluxo de computação de uma Rede Neural Artificial. Extraído de [27].	10
2.6	Modelos underfit, good fit e overfit. Baseado em [27].	11
2.7	Uso de dropout em uma rede neural. Baseado em [61].	12
2.8	Função Sigmoid e Derivada. Baseado em [9].	14
2.9	Função TanH e Derivada. Baseado em [9].	15
2.10	Função ReLU e Derivada. Baseado em [9].	16
2.11	Gráfico da descida do gradiente procurando o mínimo global de uma função que corresponde ao menor erro.	17
2.12	Convergência da descida de gradiente e da descida de gradiente estocástico.	17
2.13	Influência da taxa de aprendizado na convergência do gradiente.	18
2.14	Arquitetura de uma Rede Neural Convolucional.	19
2.15	Camada Convolutiva.	20
2.16	Max Pooling. Baseado em [27].	20
2.17	Camada Flattening.	21
2.18	Camada Densa.	21
2.19	Função Softmax. Extraído de [3].	22
3.1	Adaptação de Dominio. Extraído de [1].	30
3.2	Aprendizado por Transferência. Extraído de [2].	30
3.3	DeepSign. Extraído de [45].	31
3.4	Tabela de configuração de aprendizado por transferência. Extraído de [52].	32
3.5	Aprendizado por Transferência. Extraído de [52].	32
3.6	Diferença entre Aprendizado de Máquina e Aprendizado por Transferência. Extraído de [52].	33
3.7	Transferência baseado em instâncias. Extraído de [64].	34
3.8	Transferência baseado em mapeamento. Extraído de [64].	34
3.9	Transferência baseado em redes. Extraído de [64].	34
3.10	Transferência baseado em adversário. Extraído de [64].	34
3.11	Transferência baseado em subespaço. Extraído de [66].	35

4.1	Ilustração do pré-processamento de algumas imagens. Extraído de [30].	37
4.2	Modelo CNN, baseado em [43].	38
5.1	Padrão de sinais de trânsito europeu extraído de [59].	41
5.2	Conjunto de sinais de trânsito alemão contendo 43 classes.	42
5.3	Conjunto de sinais de trânsito belga contendo 62 classes.	43
5.4	Conjunto de sinais de trânsito croata contendo 31 classes.	44
5.5	Arquitetura de treinamento individual	46
5.6	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Alemão.	48
5.7	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Belga	49
5.8	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Croata.	51
5.9	Modelo de treinamento de classes equivalentes.	52
5.10	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Belga. . .	55
5.11	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Croata .	56
5.12	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Alemão. . .	58
5.13	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Croata. . .	59
5.14	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Alemão. .	60
5.15	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Belga. . .	62
5.16	Modelo de treinamento de classes equivalentes total.	63
5.17	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Belga. .	65
5.18	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Croata	66
5.19	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Alemão .	68
5.20	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Croata. .	70
5.21	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Croata x Alemão.	71
5.22	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Croata x Belga. .	73
5.23	Modelo de treinamento único	74
5.24	Conjunto único de sinais de trânsito (Alemão, Belga e Croata). . . .	74
5.25	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão x Belga x Croata	77
5.26	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão.	78

5.27	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Belga	80
5.28	Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Croata	81

Lista de Tabelas

3.1	Comparativo entre artigos e datasets utilizados. Abreviações usadas: Al (Alemanha), Be (Bélgica), Cr (Croácia), Su (Suécia), Ho (Holanda), Ch (China) e Fr (França).	27
4.1	Descrição das camadas e hiperparâmetros usados pelo modelo CNN proposto	38
5.1	Visão dos datasets de trânsito	44
5.2	Modelo de tabela de treinamento	46
5.3	Acurácia do treinamento do dataset alemão	47
5.4	Métricas da melhor acurácia (alemão)	47
5.5	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Alemão.	47
5.6	Acurácia do treinamento do dataset belga	48
5.7	Métricas da melhor acurácia (belga)	49
5.8	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Belga	49
5.9	Acurácia do treinamento do dataset croata	50
5.10	Métricas da melhor acurácia (croata)	50
5.11	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Croata.	50
5.12	Resumo da acurácia dos datasets	51
5.13	Tabela de equivalência entre as classes dos datasets dos países	53
5.14	Classes equivalentes Alemão x Belga	54
5.15	Tabela de acurácia Alemão x Belga	54
5.16	Métricas da acurácia Alemão x Belga	54
5.17	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Belga.	54
5.18	Classes equivalentes Alemão x Croata	55
5.19	Tabela de acurácia Alemão x Croácia	55
5.20	Métricas da acurácia Alemão x Croácia	56
5.21	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Croata	56
5.22	Classes equivalentes Belga x Alemão	57
5.23	Acurácia Belga x Alemão	57
5.24	Métricas da acurácia Belga x Alemão	57
5.25	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Alemão.	57
5.26	Classes equivalentes Belga x Croata	58

5.27	Acurácia do treinamento do dataset Belga x Croata	58
5.28	Métricas da acurácia Belga x Croata	58
5.29	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Croata.	59
5.30	Classes equivalentes Croata x Alemão	59
5.32	Métricas da acurácia Croata x Alemão	59
5.31	Acurácia Croata x Alemão	60
5.33	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Alemão.	60
5.34	Classes equivalentes Croata x Belga	61
5.35	Acurácia Croata x Belga	61
5.36	Métricas da acurácia Croata x Belga	61
5.37	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Belga.	61
5.38	Resumo da acurácia do treinamento dos datasets	62
5.39	Todas as classes Alemão x Belga	63
5.40	Acurácia Alemão x Belga completa	64
5.41	Métrica Alemão x Belga completa	64
5.42	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Belga.	64
5.43	Todas as classes Alemão x Croata	65
5.44	Acurácia Alemão x Croata completa	66
5.45	Métrica Alemão x Croata completa	66
5.46	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Croata.	66
5.47	Todas as classes Belga x Alemão	67
5.48	Acurácia Belga x Alemão completa	67
5.49	Métrica Belga x Alemão completa	67
5.50	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Alemão.	68
5.51	Todas as classes Belga x Croata	68
5.52	Acurácia Belga x Croata completa	69
5.53	Métrica Belga x Croata completa	69
5.54	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Croata.	69
5.55	Todas as classes Croata x Alemão	70
5.56	Acurácia Croata x Alemão completa	71
5.57	Métricas Croata x Alemão completa	71
5.58	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Croata x Alemão.	71
5.59	Todas as classes Croata x Belga	72
5.60	Acurácia Croata x Belga completa	72
5.61	Métricas Croata x Belga completa	72
5.62	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Croata x Belga.	72
5.63	Resumo da acurácia do treinamento dos datasets	73
5.64	Resumo da acurácia do treinamento do dataset único em uma base de teste única.	75

5.65	Métricas AlemaoxBelga x Croata	75
5.66	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão x Belga x Croata.	76
5.67	Resumo da acurácia do treinamento do dataset único	77
5.68	Métricas Alemã	78
5.69	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão.	78
5.70	Resumo da acurácia do treinamento do dataset único em uma base de teste belga.	79
5.71	Métricas Belga	79
5.72	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Belga.	79
5.73	Resumo da acurácia do treinamento do dataset único em uma base de teste croata.	80
5.74	Métricas Croata	81
5.75	Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Croata.	81
5.76	Resumo da acurácia dos datasets.	82
5.77	Resumo da acurácia do treinamento combinado dos datasets para classes equivalentes.	82
5.78	Resumo da acurácia do treinamento combinado dos datasets para todas classes.	82
5.79	Resumo dos testes com dataset único	83
6.1	Estratégias por países.	85
7.1	Matriz Confusão. Estratégia Individual - Dataset Alemão (1)	93
7.2	Matriz Confusão. Estratégia Individual - Dataset Alemão (2)	94
7.3	Matriz Confusão. Estratégia Individual - Dataset Belga (1)	95
7.4	Matriz Confusão. Estratégia Individual - Dataset Belga (2)	96
7.5	Matriz Confusão. Estratégia Individual - Dataset Croata	97
7.6	Matriz Confusão. Equivalência Simples - Alemão x Belga	97
7.7	Matriz Confusão. Equivalência Simples - Alemão x Croata	97
7.8	Matriz Confusão. Equivalência Simples - Belga x Alemão	98
7.9	Matriz Confusão. Equivalência Simples - Belga x Croata	98
7.10	Matriz Confusão. Equivalência Simples - Croata x Alemão	98
7.11	Matriz Confusão. Equivalência Simples - Croata x Belga	99
7.12	Matriz Confusão. Equivalência Completa - Alemão x Belga	99
7.13	Matriz Confusão. Equivalência Completa - Alemão x Croata	99
7.14	Matriz Confusão. Equivalência Completa - Belga x Alemão	100
7.15	Matriz Confusão. Equivalência Completa - Belga x Croata	100
7.16	Matriz Confusão. Equivalência Completa - Croata x Alemão	100
7.17	Matriz Confusão. Equivalência Completa - Croata x Belga	101
7.18	Matriz Confusão. Estratégia Única - Alemão (1)	101
7.19	Matriz Confusão. Estratégia Única - Alemão (2)	102
7.20	Matriz Confusão. Estratégia Única - Belga (1)	103
7.21	Matriz Confusão. Estratégia Única - Belga (2)	104
7.22	Matriz Confusão. Estratégia Única - Croata (1)	105
7.23	Matriz Confusão. Estratégia Única - Croata (2)	105

7.24	Matriz Confusão. Estratégia Única - Alemão x Belga x Croata (1)	. 106
7.25	Matriz Confusão. Estratégia Única - Alemão x Belga x Croata (2)	. 107
7.26	Matriz Confusão. Estratégia Única - Alemão x Belga x Croata (3)	. 108
7.27	Matriz Confusão. Estratégia Única - Alemão x Belga x Croata (4)	. 109

Lista de Nomenclaturas

ADAM	<i>Adaptive Momentum</i>
AP	<i>Aprendizado Profundo</i>
BTSC	<i>Belgium Traffic Sign Classification</i>
CAVs	<i>Connected and automated vehicles</i>
CNN	<i>Convolucional Neural Network</i>
CUDA	<i>Compute Unified Device Architecture</i>
DNN	<i>Deep Neural Network</i>
FPN	<i>Feature Pyramid Network</i>
GD	<i>Gradient Descent</i>
GPU	<i>Graphics Processing Unit</i>
GTSRB	<i>German Traffic Sign Recognition Benchmarks</i>
HOG	<i>Histogram of Gradients</i>
IA	<i>Inteligência Artificial</i>
MASTIF	<i>Mapping and Assessing the State of Traffic InFrastructure</i>
MSER	<i>Maximally Stable Extremal Regions</i>
MTSD	<i>Mapillary Traffic Sign Dataset</i>
PPM	<i>Portable Pixmap</i>
ReLU	<i>Rectified Linear Units</i>
RGB	<i>Red Green Blue</i>
RNA	<i>Redes Neurais Artificiais</i>
R-CNN	<i>Region Convolucional Neural Network</i>
SAE	<i>Society of Automotive Engineers</i>
SGD	<i>Stochastic Gradient Descent</i>
TanH	<i>Tangente Hiperbólica</i>
TT	<i>Tsinghua-Tencent</i>
VGG	<i>Visual Geometry Group</i>

Capítulo 1

Introdução

1.1 Motivação

A Inteligência Artificial (IA) simula a inteligência humana através de mecanismos e/ou dispositivos. No passado, técnicas de IA eram vistas com desconfiança, mas hoje vem se tornando cada vez mais presentes nas vidas das pessoas. O progresso nesta área vem sendo possível por alguns avanços específicos como o aumento da capacidade de processamento das máquinas e técnicas de multiprocessamento como *Graphics Processing Unit* (GPU) que permitem multiplicação de matrizes em paralelo. Além disso, fabricantes de softwares desenvolveram bibliotecas de código aberto que simplificam a leitura, conversão e processamento dos dados, cálculo de métricas de desempenho, assim como o treinamento e execução dos modelos de classificação e predição.

Graças a estes avanços, aplicações inovadoras de IA se tornaram possíveis como, por exemplo: assistentes virtuais inteligentes para atendimento de usuários, análise de perfil de clientes e produtos relacionados, robôs exercendo funções médicas e trabalhos mecânicos.

Na IA foram aperfeiçoadas técnicas e algoritmos que simulam o aprendizado do cérebro humano através de máquinas, o aprendizado de máquinas. Neste segmento, as redes neurais são os principais componentes. Elas nos permitem reconhecer imagens, que neste trabalho são placas de sinais de trânsito de estradas e classificá-las [70] com alto grau de acurácia. Dado isso, a motivação deste trabalho vem do fato de que estas redes são amplamente utilizadas no reconhecimento de sinais de trânsito por veículos autônomos e a possibilidade de utilização destes veículos em diferentes países requer adaptações nas técnicas de treinamento, para que os modelos gerados sejam capazes de se adaptar aos sistemas de sinalização de cada país.

A direção autônoma [54] divide-se em 6 níveis de autonomia, de acordo com a Sociedade de Engenheiros da Mobilidade (SAE):

0. Sem direção autônoma;
1. Assistência à direção;
2. Direção parcialmente autônoma;
3. Direção autônoma condicional;

4. Direção altamente autônoma; e
5. Direção completamente autônoma.

Para os veículos atingirem os níveis mais altos de autonomia, a tecnologia que vem sendo implementada com sucesso é a Rede Neural Convolutiva, que é um tipo de Rede Neural Profunda. Esta técnica é formada por camadas convolucionais que tratam a imagem como uma matriz de *pixels* e aplica equações matemáticas lineares e não-lineares, gerando um modelo de variáveis e pesos otimizados. Neste trabalho, ela permite treinar um modelo inicial, a partir de um conjunto de sinais de trânsito de um país, gerar um modelo otimizado que possibilita a classificação destes sinais.

Diante dos problemas do trânsito, algumas soluções vêm sendo implementadas com sucesso nos últimos anos. Exemplos incluem motores de carros mais econômicos, carros elétricos, aumento de oferta de bicicletas, aplicativos de trânsito, sensores e radares transmitindo informações do trânsito. Das soluções propostas, os veículos autônomos vêm se destacando como uma das mais promissoras, tendo recebido uma atenção crescente da comunidade científica, de grandes empresas de tecnologia e governantes de grandes centros urbanos.

Avanços nas técnicas de Inteligência Artificial (IA) vem possibilitando o desenvolvimento de veículos com níveis de autonomia cada vez maiores, permitindo-lhes controlar a velocidade, estacionar, controlar a distância para outros veículos, trocar de faixa em estradas, detectar objetos na pista, entre outras ações que normalmente requerem a intervenção de um condutor humano.

Uma das principais funções desempenhadas por veículos autônomos é a detecção e identificação de objetos à sua volta. Estes objetos podem ser outros veículos, pedestres, placas de sinalização, entre outros. Em especial, a capacidade de identificar e interpretar placas de sinalização é essencial para que os veículos autônomos sejam capazes de evitar infrações de trânsito e acidentes. Dado isso, o problema de classificação de sinais de trânsito é abordado neste trabalho através da aplicação do aprendizado de máquina, um ramo da IA focado em identificação de padrões a partir de exemplos, que ganhou destaque nos últimos anos devido ao avanço na capacidade de processamento das máquinas e nas melhorias das técnicas de treinamento.

1.2 Problema

Tomando como exemplo a União Europeia ou o Mercosul, em que um veículo é autorizado a trafegar em todos os países que fazem parte daquela comunidade, tem-se um problema que é a alta variabilidade de placas e regras de trânsito entre países. Há, então, uma preocupação com a capacidade de um veículo autônomo possuindo um modelo treinado em um determinado país de conseguir trafegar por todos os demais daquele grupo.

Além disso, assumindo-se que já se tem à disposição um modelo treinado em um determinado país dessa comunidade, existe um custo associado de retreinar ou atualizar esse modelo para a configuração de placas dos outros países daquela comunidade. É, portanto, um problema cujo estudo é justificado.

1.3 Objetivo

O objetivo deste trabalho é verificar qual a melhor estratégia de treinamento de modelos para classificação de sinais de trânsito em diferentes países.

Dado isso, vislumbra-se responder as seguintes perguntas de pesquisa neste trabalho:

- Dada a diversidade de placas de sinais de trânsito de diferentes países, que podem possuir placas iguais com a mesma semântica, placas diferentes com semânticas iguais e, ainda, placas cujo símbolo seja parecido com semânticas diferentes, um carro com um modelo de classificação de sinais de trânsito treinado em um país x de um conglomerado de países é capaz de transitar por países y ou z sem causar acidentes ou infringir a legislação de trânsito destes países ?
- Supondo que dispõem-se de dados das placas de diferentes países, qual é o impacto de diferentes estratégias de treinamento de um modelo de classificação de sinais de trânsito baseado em diferentes conjuntos de dados de placas ?

1.4 Contribuição

As contribuições técnicas e científicas deste trabalho são:

1) Um mapeamento das classes de placas de trânsito de datasets públicos da Alemanha, Bélgica e Croácia, que permite que não só este como trabalhos futuros avaliem e comparem o desempenho de modelos de classificação de sinais em cada um destes países. Comparando [37] de Juri et al 2015 com este trabalho verificou-se que este último apresentou um número maior de classes equivalentes quando feita a comparação entre os datasets destes países e também quando foi criado o dataset único da união dos 3 datasets, verificou-se um número total de classes maior que do artigo em comparação.

2) Uma avaliação experimental extensiva para determinar a melhor estratégia de treinamento de modelos de classificação de sinais de trânsito em múltiplos países. As estratégias utilizadas foram:

- . Treinamento e teste de cada dataset.
- . Treinamento e teste combinando os datasets só para classes equivalentes.
- . Treinamento e teste combinando os datasets para todas as classes.
- . Treinamento de um dataset único.

Os resultados apresentados por estes experimentos mostrados nos treinos e testes cruzados entre países, onde se treinou um país A e usou estes resultados na classificação no dados de teste de um país B .

Os modelos podem ser treinados na fábrica, na central, em um servidor e depois são atualizados nos veículos. Se houver mudança no código de trânsito de um país ou da comunidade europeia, por exemplo, o modelo poderá ser treinado de novo e carregado nos veículos.

Segundo [54] as estatísticas mostram que a maioria dos acidentes de trânsito ocorre devido à falta de tempo de resposta a eventos no deslocamento. Assim,

esses modelos aplicados a veículos autônomos correspondem a uma possibilidade de redução do número de tais acidentes.

1.5 Organização

Esta dissertação está estruturada da seguinte forma: o Capítulo 2 apresenta uma visão geral dos conceitos preliminares para o entendimento deste trabalho. O Capítulo 3 descreve os trabalhos relacionados, analisando e comparando os seus resultados obtidos com este trabalho. Por sua vez, o Capítulo 4 explica a metodologia proposta detalhando o modelo, procedimentos de treinamento e classificação, avaliação dos resultados. O Capítulo 5 apresenta os experimentos usados para responder o objetivo deste trabalho que é verificar qual a melhor estratégia de treinamento de modelos para classificação de sinais de trânsito em diferentes países. Por fim, o Capítulo 6 apresenta a conclusão, uma síntese dos resultados e algumas ideias de trabalhos futuros.

Capítulo 2

Conceitos Preliminares

Alguns conceitos se fazem necessários para entender o processo de classificação de sinais de trânsito. Dessa forma, neste capítulo, serão apresentados alguns conceitos básicos de Aprendizado de Máquina. Inicialmente, serão abordados alguns conceitos mais gerais da área e, posteriormente, serão apresentados os conceitos essenciais concernentes às redes neurais.

2.1 Inteligência Artificial e Aprendizado de Máquina

Os conceitos e terminologias usados na Inteligência Artificial e Aprendizado de Máquina (*Machine Learning*) são desafiadores porque envolvem áreas e subáreas que se interconectam, tal como ilustrado na Figura 2.1. Segundo John McCarthy, considerado como o pai da IA, “A Inteligência Artificial é a ciência e a engenharia de criar máquinas inteligentes, especialmente programas de computador inteligentes” [11]. Por sua vez, Russel e Norvig [56] resumiram a IA em 4 categorias:

- Sistemas que pensam como humanos: abordagem da modelagem cognitiva. Automatização de atividades associadas ao pensamento humano como tomar decisões, resolver problemas e aprender.
- Sistemas que pensam racionalmente: abordagem das leis do pensamento. Usa Raciocínio certo, baseado em premissas corretas.
- Sistemas que agem como humanos: abordagem do Teste de Turing. Máquinas que executam funções que exigem inteligência e que façam tarefas que hoje são melhor desempenhadas por pessoas.
- Sistemas que agem racionalmente: abordagem de agentes racionais. Age para alcançar o melhor resultado, com base naquilo que se acredita.

A IA também pode ser dividida em algumas subáreas [56], entre as quais encontram-se:

- **Aprendizado de Máquina;**
- Robótica;
- Reconhecimento de Voz;

- Processamento de Linguagem Natural;
- Planejamento;
- Sintetização de Voz;
- Visão Computacional; e
- Raciocínio e Tomada de Decisões.

O Aprendizado de máquina é, portanto, uma subárea da IA que foi definida por Mitchell da seguinte forma: “Diz-se que um programa de computador aprende com a experiência E com relação a algumas classes de tarefas T e possui desempenho em medida P , se seu desempenho nas tarefas em T , medido por P , melhorar com a experiência E .” [50].

A maioria das técnicas do estado-de-arte de aprendizado de máquina, diferentemente dos seres humanos, exige uma grande quantidade de dados de treinamento para que sejam capazes de generalizar seu aprendizado.

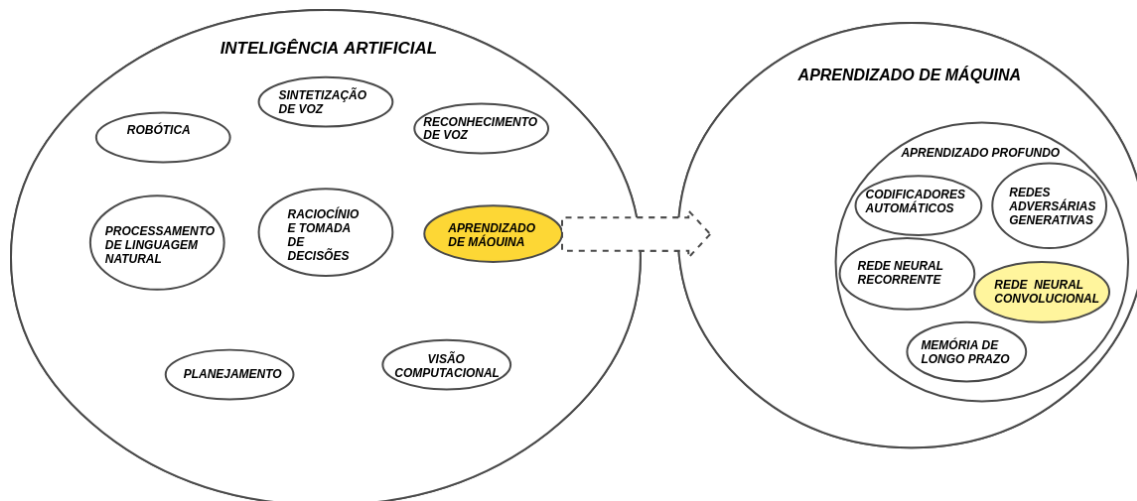


Figura 2.1: Visão global da Inteligência Artificial. Baseado em [55].

Os tipos de aprendizado de máquina [27] e as suas formas de treinamento, ilustrados na Figura 2.2, são:

- **Supervisionado.** Processo de aprendizado de máquina que consiste em mapear uma função matemática que aplicada a um dado de entrada reproduza um resultado esperado [40]. Para que esse aprendizado seja possível, deve haver disponibilidade de exemplos previamente rotulados, isto é, para os quais se saiba o valor desejado de saída. Dessa forma, são feitos ajustes, segundo critérios de otimização, nos parâmetros da função até chegar ao valor ou algo próximo a esse. Este tipo de aprendizado normalmente é utilizado em tarefas/problemas de classificação ou regressão.
- **Não-supervisionado.** O aprendizado de máquina não-supervisionado descobre padrões previamente desconhecidos em dados que não possuem

rótulos. Ele cria agrupamentos a partir de características procurando descobrir similaridades entre os objetos analisados. Algumas formas são usadas como clusterização, compressão de dados e distribuição probabilística [23].

- **Aprendizado por reforço.** Neste tipo de aprendizado, não há dataset, tampouco treinamento. Os elementos são: agente, estado, ações e recompensa. O agente aprende através de tentativa e erro, ao interagir com o ambiente, de modo que o objetivo é maximizar a recompensa, que é modelada segundo uma determinada função objetivo. As informações anteriores servem de aprendizado incremental para as seguintes.

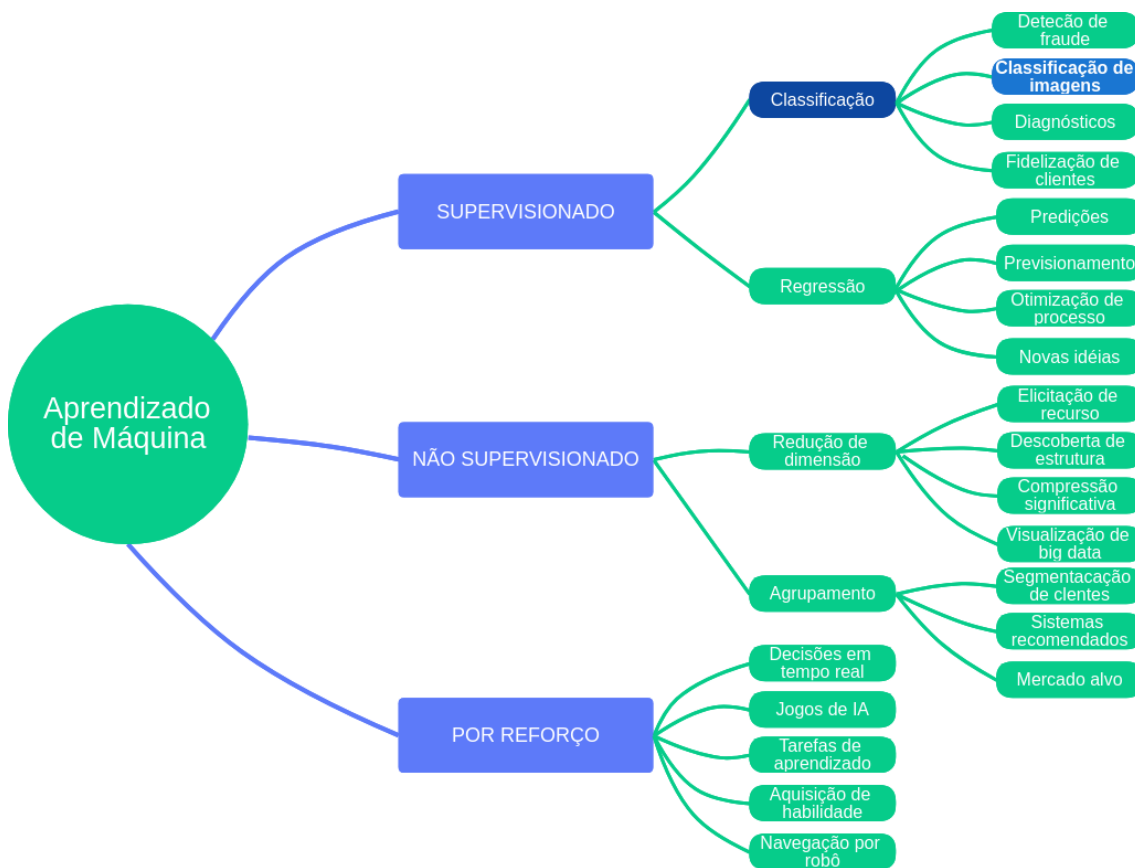


Figura 2.2: Visão do aprendizado de máquina com seus tipos e respectivos algoritmos. Baseado em [38].

Aprendizado Profundo (AP) - *Deep Learning* também conhecido como *Deep Neural Network* (DNN) é uma sub-área da Aprendizagem de Máquina, que emprega algoritmos para processar dados e imitar o processamento feito pelo cérebro humano. Usa camadas de neurônios matemáticos para processar dados, compreender a fala humana e reconhecer objetos visualmente [7].

Segundo [18], o AP compreende alguns métodos de aprendizado supervisionado, não supervisionado e de reforço. Entre alguns métodos deste aprendizado estão:

- **Rede Neural Convolucional** - *Convolutional Neural Network*

- Rede Neural Recorrente - *Recurrent Neural Network*
- Memória de longo prazo - *Long-short Term Memory*
- Codificadores automáticos - *Autoencoder*
 - Autoencoder Denoising
 - Autoencoder empilhado
 - Autoencoder variacional
 - Autoencoder esparso
- Rede Adversarial Generativa - *Generative Adversarial Network*

2.2 Redes Neurais

2.2.1 Proposta de McCulloch-Pitts

Redes neurais são sistemas de computação com nós interconectados que funcionam como os neurônios do cérebro humano. O primeiro modelo matemático de um neurônio artificial foi o *Threshold Logic Unit* apresentado em 1943 por McCulloch - Pitts[49]. A sua forma inicial está ilustrada na Figura 2.3. Eles modelaram suas ideias criando uma rede neural com circuitos lógicos com (e, não, ou).

Para sinais de entrada $X_1, X_2, \dots, X_p$ e pesos $w_1, w_2, \dots, w_p$, limiar (*threshold*) t e sinais booleanos(0,1), ter-se-ia como na Equação 2.1:

$$a = \sum_{i=1}^n w_i * X_i \quad (2.1)$$

A saída y seria a equação 2.2

$$y = \begin{cases} 1, & \text{se } a \geq t \\ 0, & \text{se } a < t \end{cases} \quad (2.2)$$

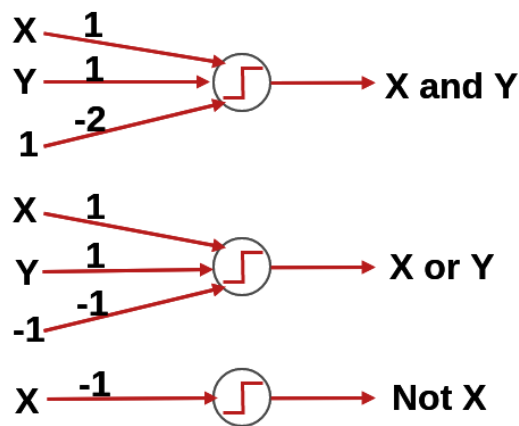


Figura 2.3: Exemplos de redes neurais. Baseado em [49].

2.2.2 Perceptron

Em 1958, Rosenblatt criou o perceptron, que é o modelo de um neurônio básico e corresponde a um separador linear binário. Dessa forma, consistia em um classificador binário, que mapeia a entrada e gera uma saída conforme a equação 2.4. Esse conceito recebe sinais como entrada, no formato de um vetor $x_1, x_2, \dots, x_n$, multiplica estes valores pelo seus pesos $w_1, w_2, \dots, w_n$. É então feito o seguinte somatório:

$$\sum_{i=1}^n x_i * w_i \quad (2.3)$$

ao resultado deste somatório é acrescido o *bias* (viés), um parâmetro adicional usado para ajustar a saída de modo que possa se adaptar melhor aos dados de entrada. Aplica-se então este resultado na função degrau, que corresponde a uma função de ativação conforme será explicado na Seção 2.3.5. Esse processo está ilustrado na Figura 2.4.

$$g(x) = \begin{cases} 1, & \text{se } w * x + b \geq 0 \\ 0, & \text{se } w * x + b < 0 \end{cases} \quad (2.4)$$

em que $w * x$ é produto escalar do vetor de peso pelo vetor de dados da entrada, e b (*bias*).

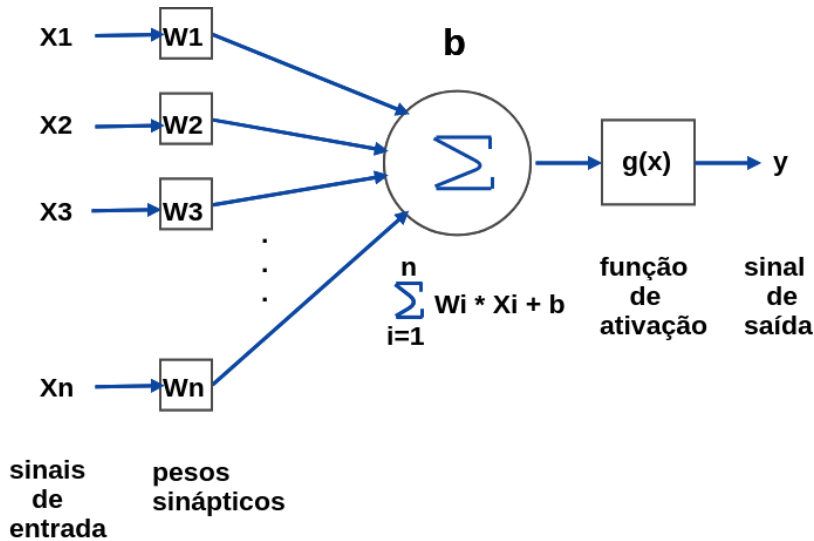


Figura 2.4: Perceptron

2.3 Redes Neurais Artificiais

São redes de neurônios conectados entre si em camadas que aprendem a gerar um resultado para uma determinada tarefa. Corresponde a um modelo matemático que, aplicado a um dado de entrada, gera uma saída que seja o mais próxima possível do resultado esperado. Este modelo é composto de hiperparâmetros cujos valores são obtidos através da etapa de treinamento sobre os dados. Na

Figura 2.5, é mostrado um exemplo de uma rede neural artificial que possui uma camada de entrada, uma camada de saída e camadas intermediárias chamadas de escondidas. Os valores de entrada $x_1, \dots, x_n$ são multiplicados pelos seus respectivos pesos. O resultado do somatório destes valores passa por função de ativação e gera um valor que corresponde a probabilidade do resultado.

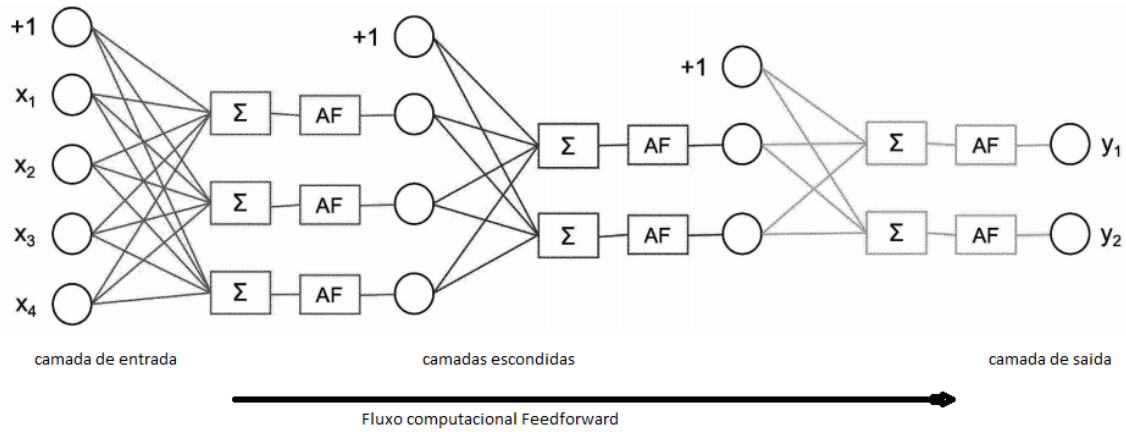


Figura 2.5: Fluxo de computação de uma Rede Neural Artificial. Extraído de [27].

Nas próximas subseções, serão descritos os mais importantes ingredientes das redes neurais.

2.3.1 *Backpropagation*

É um algoritmo de método supervisionado pois se conhece os dados na camada de entrada e o resultado em que se pretende chegar que são os rótulos na camada de saída. Pode ser usado na classificação e regressão. Funções matemáticas aplicadas aos dados de entrada geram uma saída que é comparada aos rótulos. Para a saída chegar ao valor mais próximo dos rótulos precisa-se ajustar os pesos destas equações [35]. As funções de perda calculam esta diferença. *Backpropagation* são procedimentos que fazem o ajuste destas variáveis minimizando esta diferença através do gradiente descendente utilizando uma função de ativação. Usa-se a taxa de aprendizado que indica a velocidade da alteração dos pesos para chegar no erro mínimo. A cada época recalculam-se os pesos que são atualizados recursivamente.

2.3.2 *Data Augmentation*

Esta técnica [67] tem como objetivo contornar uma baixa disponibilidade de dados para uma determinada tarefa, de modo a criar novos dados a partir dos dados de treinamento já existentes. Quando a tarefa sendo abordada envolve imagens, alguns artifícios comuns costumam ser a aplicação de operações como rotação, inclinação, translação, espelhamento, recortes, ruídos, conversões para tons de cinza entre outras e outros. Isto diminui a chance de ocorrer um *overfitting* e melhora o desempenho da rede.

2.3.3 Overfitting

É o nome do fenômeno que ocorre quando um modelo[42] obtém uma performance excelente no dataset de treinamento, mas não consegue reproduzi-la no dataset de teste. Normalmente isto se atribui ao fato de o modelo não ter aprendido um padrão generalizável a partir dos dados de treinamento. Logo, diz-se que o modelo memorizou os dados de treinamento, o que não é necessariamente verdade. Redes neurais são altamente suscetíveis ao overfitting, pois são capazes de aprender funções extremamente complexas. Um ajuste exaustivo dos pesos dessas redes durante o treinamento pode levá-las a aprender a função geradora dos dados de treinamento com perfeição, o que em muitos casos faz com que a capacidade de realizar inferências em novas amostras, que não estavam representadas no dataset de treinamento, seja prejudicada. Em contrapartida, quando um modelo não é capaz de obter uma boa performance no treinamento, considera-se que ocorreu um *underfitting*, pois o modelo foi incapaz de extrair um padrão dos dados de treinamento. Este fenômeno normalmente ocorre quando há um baixo número de amostras disponíveis ou quando o modelo utilizado não é adequado ou não foi suficientemente ajustado para o problema de aprendizado sendo abordado. A Figura 2.6 mostra exemplos de *underfit*, *good fit* e *overfit* em que se divide um base de dados em duas partes (70/30) 70% para treinamento e 30% para validação [27].

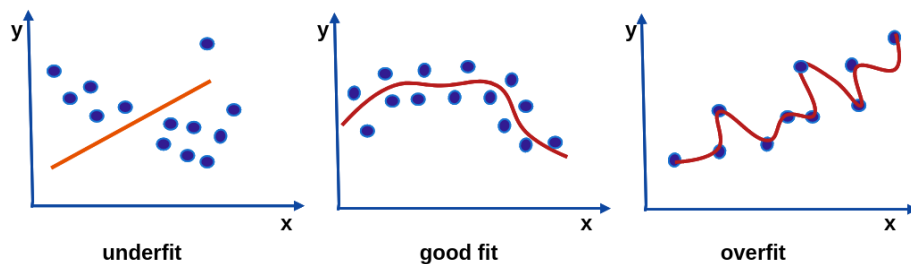


Figura 2.6: Modelos underfit, good fit e overfit. Baseado em [27].

2.3.4 Regularização

A regularização pode ser definida como “qualquer modificação feita em um algoritmo de aprendizado de máquina com o objetivo de reduzir o erro de generalização, mas não necessariamente o erro de treinamento” [26]. Entre estas modificações, podem ser citadas algumas técnicas para controlar *overfitting* e consequentemente melhorar a generalização, tais como *data augmentation* e *dropout* [27], este último mostrado na seção de hiperparâmetros.

2.3.5 Hiperparâmetros

Uma rede neural possui diversos parâmetros de controle do modelo que são pré-definidos e são seus valores fixados no início do processo. Entre os parâmetros relacionados à arquitetura têm-se: número de camadas, número de nós por camada e função de ativação. Já entre os parâmetros do processo de aprendizado, têm-se: método de otimização, taxa de aprendizado, estratégia de regularização e parâmetros associados.

A configuração adequada de valores dos hiperparâmetros pode ser obtida manualmente ou através de alguma abordagem de automatizada. Os principais tipos de ajustes automatizados são: *Random search*, que faz pesquisa aleatória por uma configuração dos parâmetros, *Grid search*, que pesquisa uma combinação de valores dos parâmetros, e *Model-based optimization*, que constrói um modelo do erro e executa um algoritmo de otimização nesse modelo, de maneira a minimizá-lo. Como era de esperar, tal ajuste automático tem custo computacional alto [27].

No que se segue, descrevem-se os principais hiperparâmetros de uma rede neural:

Epoch

Quantidade de vezes que o modelo dá uma passada por todo o dataset. A cada *epoch*, tem-se a garantia de que todos os exemplos de treino foram apresentados ao modelo. Uma época é composta de um mais lotes (*batches*).

Batch size

É um hiperparâmetro definido pelo arquiteto da rede que define o tamanho de lote, isto é, o número de exemplos de treinamento que serão processados antes que os parâmetros internos do modelo sejam atualizados.

Validation_split

Percentual dos dados de treino que é separado para validação.

Dropout

Consiste em uma técnica proposta por [61] para eliminar ou reduzir o *overfitting* durante a aprendizagem do modelo no treinamento sobre os dados, de maneira que é desligado aleatoriamente um percentual p de neurônios da rede que normalmente se encontra no intervalo de 0.5 a 0.8. Para $p = 1$, que corresponde a 100%, não há *dropout* [61]. Isso obriga os demais neurônios da rede a aprenderem a parte que era devida aos neurônios que estão desligados. A Figura 2.7 mostra um exemplo de rede neural antes e depois de aplicar um *dropout*.

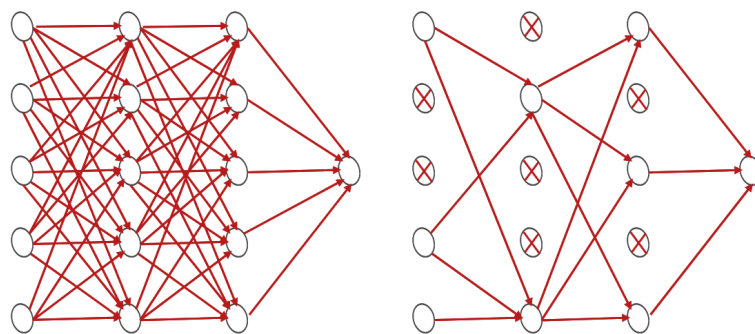


Figura 2.7: Uso de dropout em uma rede neural. Baseado em [61].

Funções de Ativação

São funções que servem para decidir ou não a ativação dos valores dos neurônios da camada seguinte com base em um dado conjunto de entrada. Uma rede neural sem função de ativação é apenas um modelo de regressão linear que faz apenas uma transformação linear nos pesos e *bias*, não resolvendo problemas complexos como classificação de imagens e processamento de linguagem natural [8]. Essas funções reduzem o custo computacional do gradiente nas funções de otimização e possibilitam o *backpropagation*.

Assumindo que seja utilizado uma abordagem baseada em gradiente descendente como otimizador da rede, faz-se imperioso lidar com funções de ativação que se encaixem com essa abordagem. Dessa maneira, segundo [44], os seguintes critérios devem ser satisfeitos para uma função de ativação:

1. A função deve ser contínua e derivável em toda parte (ou quase toda parte);
2. A derivada da função não deve saturar, isto é, tornar-se muito pequena e tender a zero, nos valores do intervalo esperado, pois isto tende a perturbar o aprendizado; e
3. A derivada não deve explodir, isto é, tornar-se muito grande e tender ao infinito, visto que isso leva a problemas de estabilidade numérica.

A função degrau, descrita na Seção 2.2.2, não satisfaz às condições (2) e (3), pois sua derivada explode em 0 e é 0 em todo o resto. Portanto, a função degrau não combina com o método gradiente descendente e não é uma boa escolha para redes profundas [44]. Entre as funções mais usadas estão Sigmoid, Tangente Hiperbólica (TanH) e Unidade Linear Retificada (ReLU).

Função Sigmoid

Esta função representada pela Equação 2.5 é uma alternativa à função degrau, que não pode ser utilizada pelos motivos elencados na seção anterior. Dessa forma, a função Sigmoid retorna valores no intervalo de 0 a 1 conforme a Figura 2.8 e foi muito usada pelas redes neurais no passado, pois tal função simula os neurônios do cérebro que funcionam de forma binária (ativado ou não-ativado) e é eficiente quando se tenta classificar uma classe para um determinado valor.

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \quad (2.5)$$

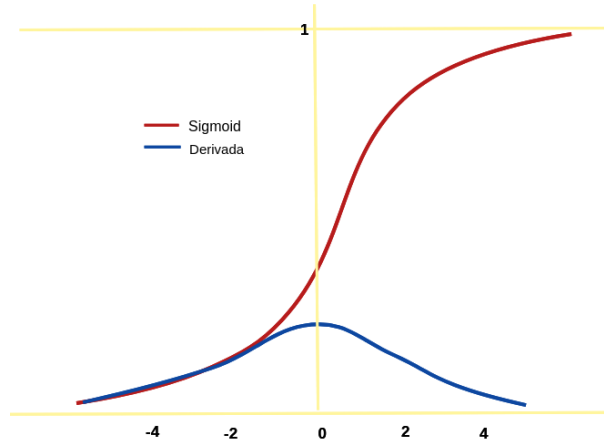


Figura 2.8: Função Sigmoid e Derivada. Baseado em [9].

No entanto, um problema conhecido como *Vanishing Gradient* acontece quando a sua derivada representada pela Equação 2.6 tende a zero para valores muito baixos ou muito altos, fazendo com que a propagação do gradiente desvaneça nesta região, gerando dificuldades no treinamento ao aplicar o *backpropagation*. Outro problema consiste no fato de as saídas serem sempre no intervalo de 0 a 1, porém, por vezes, não é necessário enviar valores positivos para o próximo neurônio.

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (2.6)$$

Desse modo, esta função costuma ser usada na saída de uma rede neural para modelar variáveis binárias, em redes neurais recorrentes e alguns modelos não-supervisionados.

Função Tangente Hiperbólica(Tanh)

Esta função tem semelhança com a função sigmoid, mas retorna valores entre -1 e 1. Nesta comparação, também possui mais valores próximos de 1 conforme Figura 2.9, o que torna-se uma melhor opção na ativação das camadas ocultas das RNAs. Sua função e a derivada estão nas Equações 2.7 e 2.8. Verifica-se ainda saturação, mas a função Tanh tem valores de saída distribuídos em um domínio maior que o domínio da função sigmoid.

$$\tanh(x) = \sigma(2x) - 1 \quad (2.7)$$

$$\tanh'(x) = 1 - \tanh^2(x) \quad (2.8)$$

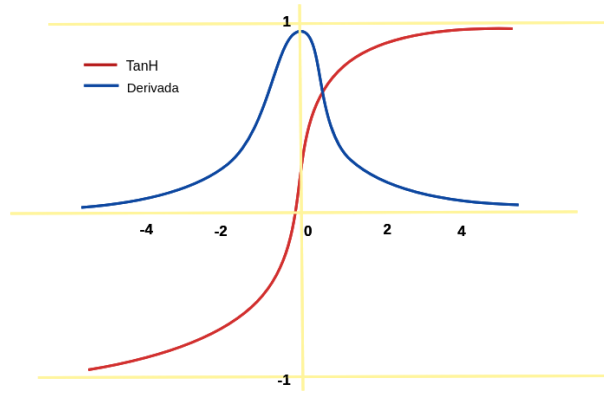


Figura 2.9: Função TanH e Derivada. Baseado em [9].

Função ReLU

A função *rectified linear unit*, ou ReLU, é a função de ativação (Equação 2.9) mais amplamente utilizada atualmente em redes profundas. Como é possível ver pela sua definição, a função não é diferenciável em 0, mas é derivável em toda outra parte, incluindo os pontos próximos a 0. Na prática, a derivada em 0 é definida como 0 (derivada à esquerda) ou 1 (derivada à direita) [44]. O gráfico da sua derivada é exibido na Figura 2.10.

Atualmente, ReLU tem sido mais utilizada do que a Sigmoid como a escolha de função de ativação, pois esta possui os problemas descritos na Seção 2.3.5. Segundo [44], a sua popularidade advém de duas propriedades:

1. O gradiente da ReLU permanece constante e nunca satura para valores positivos de x , acelerando o treinamento. Foi encontrado na prática que redes que usam a ReLU como função de ativação oferecem uma aceleração natural no treinamento em comparação à Sigmoid; e
2. Ambas a função e sua derivada podem ser computadas usando operações matemáticas elementares e eficientes, já que não envolvem exponenciação.

Contudo, a ReLU sofre de um problema relacionado à saturação de sua derivada quando $x < 0$ [44]. Quando o valor de entrada de um nó se torna negativo, é possível que a saída do nó fique “presa” em 0 durante o resto do processo de treinamento. Tal fato é conhecido como *dying ReLU problem*.

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (2.9)$$

$$\text{ReLU}'(x) = \begin{cases} 1, & \text{se } x > 0 \\ 0, & \text{se } x \leq 0 \end{cases} \quad (2.10)$$

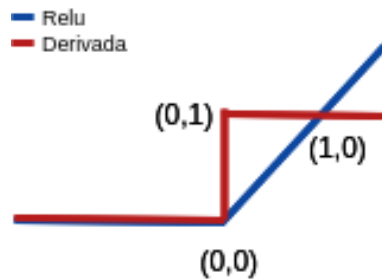


Figura 2.10: Função ReLU e Derivada. Baseado em [9].

Funções de Otimização

São funções que participam do processo de treinamento para reduzir o erro da rede neural, fazendo com que a rede apresente os resultados desejados. Algumas funções importantes são listadas abaixo:

1) Gradiente Descendente

É um algoritmo iterativo que tenta encontrar o mínimo global de uma função ajustando a configuração da rede após cada ponto de treinamento ilustrado na Figura 2.11.

Realiza iterativamente pequenos ajustes nos pesos sinápticos para que o resultado da saída convirja para o valor esperado. Calcula a derivada parcial dos pesos e multiplica pela taxa de aprendizado e são subtraídas do peso sináptico.

2) Gradiente Descendente Estocástico

Utiliza o mesmo princípio visto na Figura 2.11. Enquanto a Descida de Gradiente tradicional calcula o custo para todo o dataset, esta função particiona os dados e calcula o erro com base nestas partições (batches). A convergência ocorre de uma forma mais aleatória como pode ser visto na Figura 2.12

O particionamento dos dados permite que o algoritmo funcione para volumes de dados muito grandes além de poder paralelizar melhor o processamento e de ter a menor probabilidade de ficar preso em um mínimo local, encontrando assim um mínimo mais global, fazendo com que o desempenho da rede treinada fique melhor.

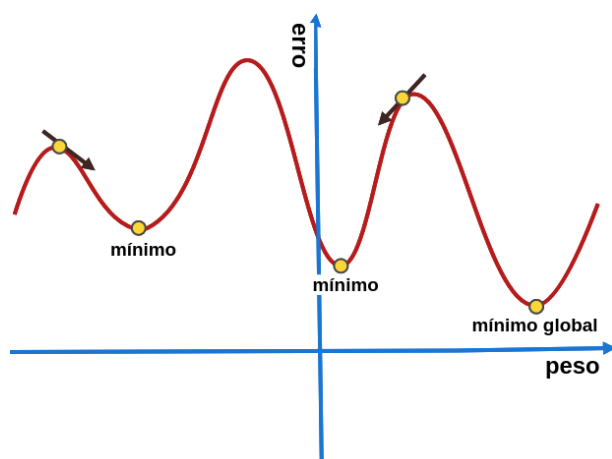


Figura 2.11: Gráfico da descida do gradiente procurando o mínimo global de uma função que corresponde ao menor erro.

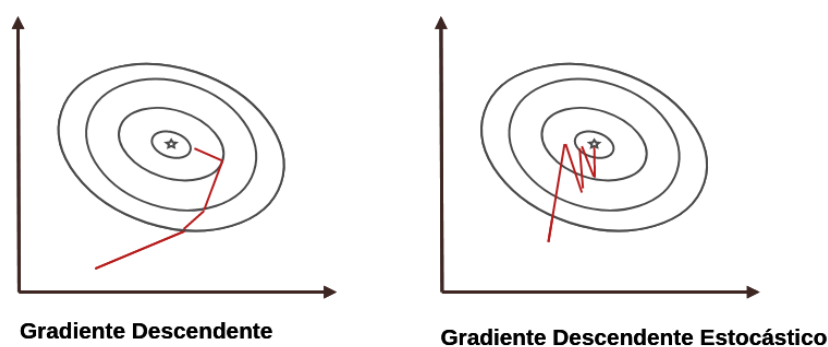


Figura 2.12: Convergência da descida de gradiente e da descida de gradiente estocástico.

Taxa de Aprendizado

A taxa de aprendizado é uma constante que corresponde à velocidade de aprendizado da rede. Se o valor for pequeno, pode levar muito tempo para atingir o mínimo global, por outro lado se o valor for grande ele pode nunca chegar a atingir este valor, conforme pode ser visto na Figura 2.13.

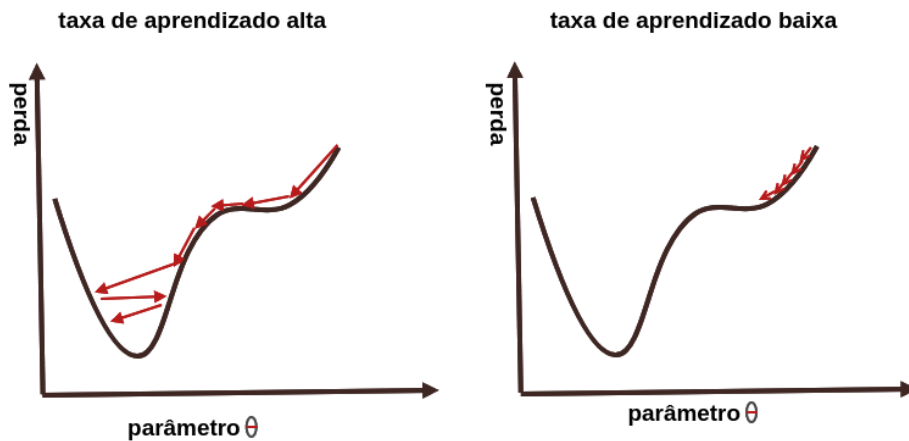


Figura 2.13: Influência da taxa de aprendizado na convergência do gradiente.

2.4 Redes Neurais Convolucionais

Conhecidas também como Convnets ou CNNs (*Convolutional Neural Network*), possuem os princípios das redes neurais artificiais com uma diferença. Nas RNAs, toda entrada representada por neurônios através de matriz é multiplicada por uma matriz de pesos gerando uma saída. Nas CNNs as convoluções são feitas em regiões específicas utilizando filtros e máscaras.

Assim as CNNs têm como princípio básico a convolução que são operações de funções matemáticas com dados de camadas anteriores em forma de matrizes. Essas redes permitem extrair mapa de características, filtrar informações, reduzir número de parâmetros. Possuem grande aplicação para imagens.

A Figura 2.14 mostra a arquitetura de uma rede neural convolucional composta por camadas de entradas, de convolução, de agrupamento, as completamente conectadas e as de saída. A camada inicial recebe as imagens. As camadas intermediárias extraem as características através de funções matemáticas. Inicialmente filtra linhas, curvas e bordas. Nas camadas seguintes, retiram os detalhes mais complexos. Este processo é feito percorrendo a imagem, selecionando matrizes de pixels, aplicando filtros formados por pesos que vão se atualizando através dos treinamentos. A última camada gera uma saída de tamanho igual ao número de classes treinadas que contém o percentual de acerto da imagem em cada classe.

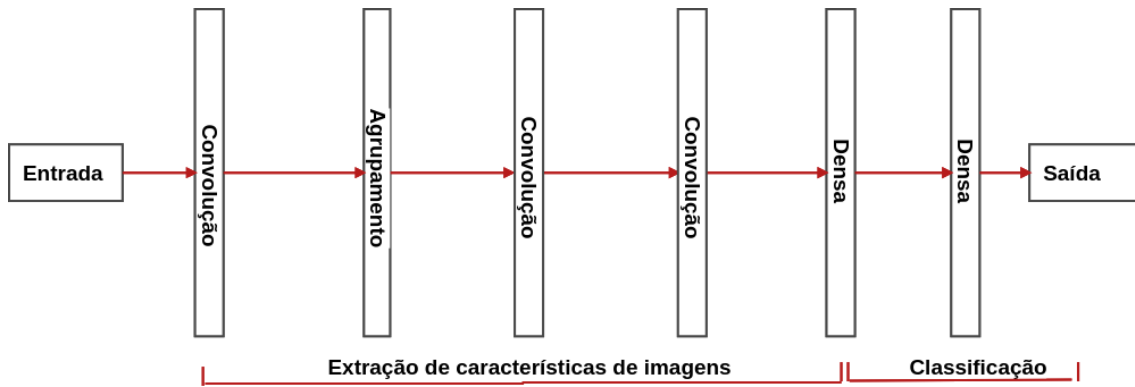


Figura 2.14: Arquitetura de uma Rede Neural Convolucional.

Os principais hiperparâmetros utilizados em uma rede convolucional são:

- *Kernel*: É uma matriz que funciona como filtro na geração do mapa de características.
- *Stride*. Número de pixels que a matriz filtro (*kernel*) se desloca sobre a matriz de entrada.
- *Padding*. Número de zeros que são preenchidos na borda da matriz de entrada para controlar a saída. Permite manter o tamanho da imagem de saída da convolução igual ao tamanho original. O cálculo do padding é feito através da Equação 2.11, em que W é o tamanho (altura) da entrada, K é o tamanho do filtro, P o tamanho do padding e S o stride.

$$\text{Padding} = \frac{W - K + 2P}{S} + 1 \quad (2.11)$$

2.4.1 Camadas

São etapas correlacionadas, usadas para treinar um modelo de redes neurais convolucionais de forma iterativa, onde são feitas extrações de atributos de um conjunto de dados, fazendo ajustes no modelo e, na última etapa, é feita a classificação.

Camada Convolucional

Camada utilizada na extração de características (*feature extraction*) utilizada para extrair mapas de características (*feature maps*) em que são usados n filtros (*kernel*) representados por matrizes de duas dimensões para percorrer os pixels de uma imagem (I) gerando n imagens novas. Neste processo chamado convolução são feitas operações matemáticas entre a matriz da imagem de entrada (I) e filtro (K) gerando mapa de características ($I * K$) ilustrado na Figura 2.15.

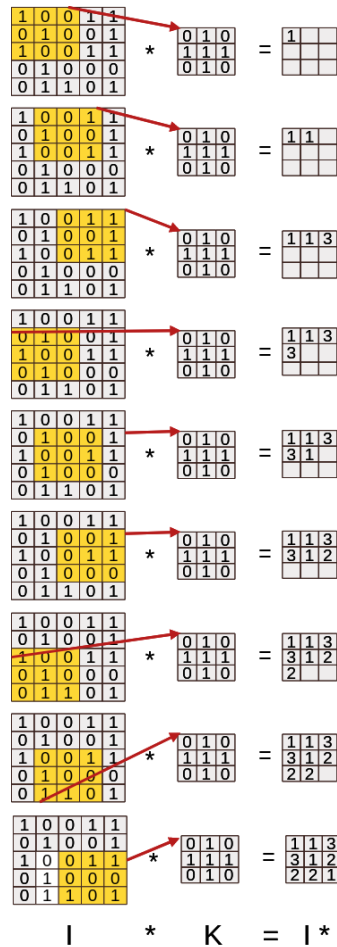


Figura 2.15: Camada Convolutiva.

Camada de Agrupamento

Camada de agrupamento (*pooling*) que gera subamostras (*subsampling*), usada para filtrar as informações mais importantes da camada anterior [27]. Isto reduz o número de parâmetros a ser processados e o custo computacional. Se por um lado ele pode evitar overfitting, por outro lado, se for muito usado, pode perder algumas características importantes.

A função de agrupamento (*pooling*) mais usada é a *Max Pooling* que, após transitar na camada anterior, percorrendo uma matriz de filtro de dimensão n , seleciona os maiores valores conforme é mostrada na Figura 2.16.

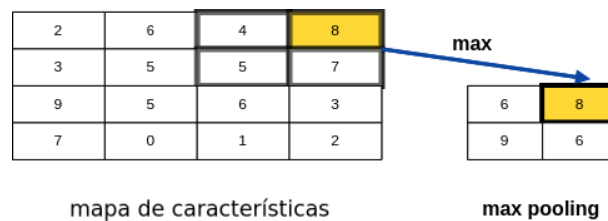


Figura 2.16: Max Pooling. Baseado em [27].

Camada de flattening

Camada também conhecida como achatamento em que é feito uma concatenação de todos os dados recebidos da última camada convolucional em um vetor, enviando para a camada densa, ilustrado na Figura 2.17.

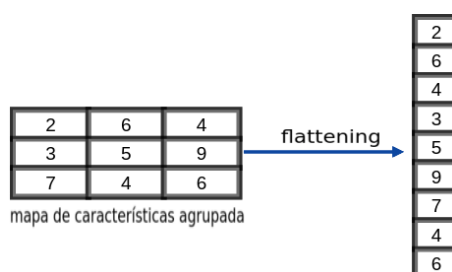


Figura 2.17: Camada Flattening.

Camada Densa

Também conhecida como camada totalmente conectada como pode ser visto na Figura 2.18. Após atravessar as camadas de convolução chega-se nesta camada que contém as características mais importantes relacionadas com a classe do objeto. Esta camada conecta todos os neurônios da camada anterior a todos os seus neurônios, tornando possível a classificação feita na camada de saída que normalmente utiliza a função softmax.

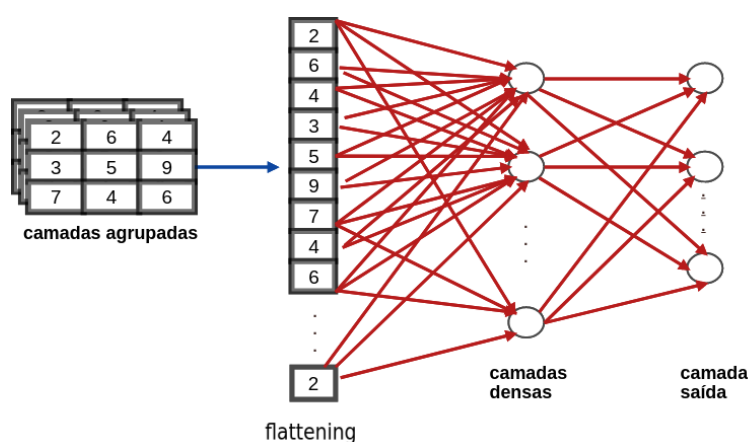


Figura 2.18: Camada Densa.

Camada ReLU

Entre as camadas de ativação, esta é a mais usada atualmente, pois zera os valores negativos e mantém os positivos [21]. Esta camada usa a função de ativação ReLU detalhada na Seção 2.3.5.

Camada de SoftMax

É a última camada após a camada densa ou totalmente conectada, que utiliza a função softmax do tipo sigmóide em que transforma a saída de cada classe em valores entre 0 e 1.

2.4.2 Classificação

Classificação, em um sentido amplo, é a capacidade de associar um objeto a uma classe dentro de um conjunto de classes exclusivas. Neste trabalho, especificamente, pode-se definir classificação como procedimento estatístico em que se atribui às classes a probabilidade de uma imagem pertencê-las. Uma forma muito utilizada para este cálculo é através da função softmax que embora retorne apenas valores entre 0 e 1, divide-se este valor pela soma dos valores obtidos pelas outras saídas, resultando a probabilidade da entrada estar em determinada classe, conforme mostrado na Figura 2.19.

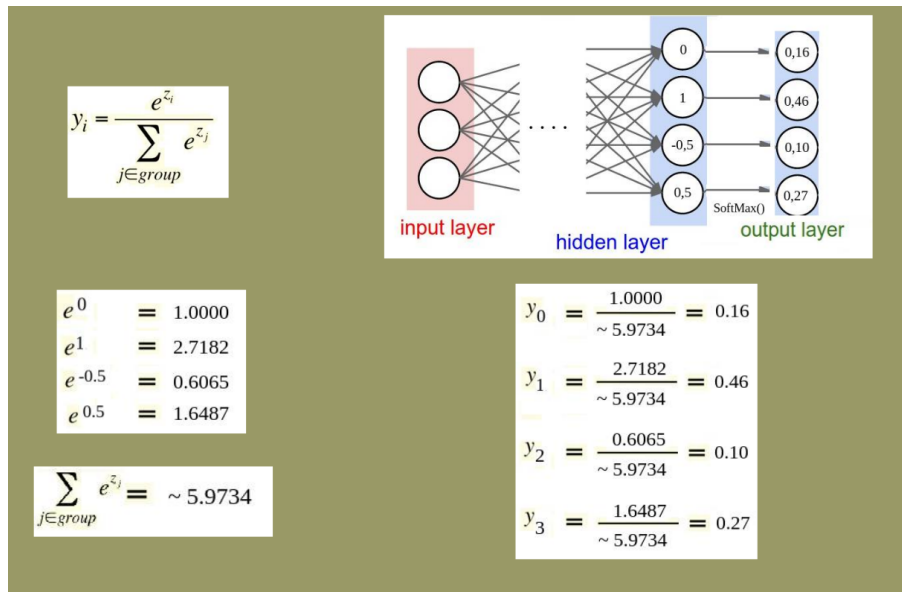


Figura 2.19: Função Softmax. Extraído de [3].

Capítulo 3

Trabalhos relacionados

A Inteligência Artificial tem estado cada vez mais presente em vários setores da sociedade. Entre os avanços, destacam-se o desenvolvimento de técnicas que permitem reconhecer padrões e analisar dados. O aprendizado de máquinas, uma subárea da IA, vem desenvolvendo algoritmos que aperfeiçoam estas técnicas, conforme apresentado no Capítulo 2. A partir destes conhecimentos, a comunidade científica vem apresentando trabalhos na área de mobilidade urbana que permitem aos veículos identificarem os sinais de trânsito, entre outras atividades, utilizando informações como alerta e orientação. Uma área que vem se beneficiando destes conhecimentos é a direção autônoma. A seguir, são apresentados trabalhos que têm relação com esta dissertação, seja de maneira direta ou indireta.

Em CIREŞAN *et al.* [30], é proposto uma arquitetura para classificação de sinais de trânsito em que tinham como problema placas com dificuldade de reconhecimento. A arquitetura foi chamada MCDNN (*Multi-Colum Deep Neural Network*) que combinou uma DNN tradicional utilizando como entrada, colunas de pré-processamento. Para isto, o dataset foi pré-processado antes de começar o treinamento, contemplando a normalização de contraste em que a intensidade dos pixels das imagens foi distribuída mais uniformemente e foi feito redimensionando das imagens que possuíam diversos tamanhos para um tamanho padronizado de 48×48 pixels. Isso permitiu melhorar o reconhecimento das imagens, mesmo aquelas com pouca iluminação ou contraste. Durante o treinamento, todas as imagens originais e pré-processadas foram aleatoriamente distorcidas usando translação, rotação e escala, antes de cada época. A imagem de entrada foi pré-processada por n pré-processamentos $P_0, P_1, \dots, P_{n-1}$ e um número de colunas foi treinado para cada entrada preprocessada. As previsões finais foram obtidas calculando-se a média individual das previsões de cada DNN. O treinamento foi feito utilizando-se o dataset alemão GTSRB[4]. O modelo obteve uma acurácia de 99,46%, mas, no pré-processamento, foi dada ênfase ao contraste das imagens e levantada a dúvida se teria os mesmos resultados para outros tipos de imagens de outros datasets. No trabalho apresentado também foi feito treinamento com dataset alemão GTSRB [4], mas, o pré-processamento foi mais generalizado e obteve relativamente resultados próximos.

Em JURISÍĆ *et al.* [37], é apresentado um modelo CNN de classificação de sinais de trânsito chamado OneCNN. Para o treinamento deste modelo, os autores criaram um dataset único, composto da união dos dataset alemão GTSRB[4], dataset belga BTSC[5] e o dataset croata rMASTIF[6]. Para este dataset único os

autores mantiveram as classes da base de treino que possuem imagens correspondentes e criaram classes novas para as imagens que não tinham correspondentes. Também foram feitos treinamentos de modelo combinando datasets de 2 países. No presente trabalho, conforme será explicado no Capítulo 5 também treinaram-se modelos utilizando as duas estratégias que foram dataset único e combinação de dataset 2 a 2, sendo que, para esta última estratégia, este trabalho obteve uma melhor equivalência de classes que tinham imagens correspondentes entre os datasets destes países. Foram usados os mesmos datasets e os resultados foram semelhantes.

Em SERNA *et al.* [59], é apresentado um modelo de classificação de sinais de trânsito. Para o treinamento deste modelo, os autores criaram um dataset único com os dados públicos dos países Bélgica, Croácia, França, Alemanha, Holanda e Suécia. Este dataset possui 80.000 imagens divididas em 164 classes, que por sua vez distribuem-se em 4 categorias (Perigo/Aviso, Regulador, Informativo, Outras). Também foi realizada uma análise e comparação do desempenho de classificação de cinco arquiteturas de redes neurais convolucionais. No presente trabalho, também foi usado treinamento com dataset único além de mais três estratégias: treinamento individual, treinamento de datasets com classes equivalentes e treinamento de datasets para todas classes. Também usou-se uma classe "outras" para definir aquelas que não tinham imagens correspondentes entre as classes, mas no artigo em comparação foi usado essa classe como uma categoria. Outra semelhança foi o uso de imagens redimensionadas com tamanho $48 \times 48 \times 3$. Os resultados obtidos para o dataset alemão GTSRB[4] ficaram próximos assim como os valores obtidos com *data augmentation*.

Em ERTLER *et al.* [33], é realizado um estudo de detecção e classificação de sinais de trânsito em que os autores criaram o dataset chamado *Mapillary Traffic Sign Dataset* (MTSD) que cobre os 6 continentes com 313 classes e 100.000 imagens (52.000 de sinais de trânsito, 36.000 de treinamento, 5.000 de validação e 11.000 de teste). Foi utilizado o algoritmo *Faster R-CNN* e feitas comparações dos resultados com o dataset chinês Tsinghua-Tencent (TT100K) (100.000 imagens, sendo 10.000 de sinais de trânsito e 90.000 imagens de fundo sem sinais de trânsito). Os resultados obtidos das acurácias foram: MTSD: 81.1% (FPN50 + classificador), TT100K: 89.9% (FPN50 + classifier), em que FPN (*Feature Pyramid Network*) é um extrator de características. É importante citar que esse modelo teve uma cobertura muito ampla de imagens. Por outro lado, tem os mesmos problemas quando reúne muitas imagens de vários países: semelhanças de imagens, oclusão, nitidez. O nosso modelo também usou a estratégia de dataset único e enfrentou os mesmos problemas do artigo em comparação. Os resultados do artigo ficaram um pouco abaixo pois ele fez detecção e classificação. O nosso trabalho apresentou outras estratégias de classificação de sinais de trânsito como combinação de treinamento e teste entre datasets.

Em SWAMINATHAN *et al.* [62], é apresentado um experimento de reconhecimento e classificação de sinais de trânsito com carros autônomos, em tempo real, usando uma rede neural convolucional. Este experimento foi realizado com um protótipo de carro controlado por uma placa microcontroladora arduíno [25] que permitiu programar a maneira como suas entradas e saídas interagem com dispositivos externos. Apesar de obter bons resultados, o trabalho ficou restrito ao dataset belga BTSC [5] e deixa dúvida sobre como seria o resultado usando

este modelo em outros países. No nosso trabalho também foi usada a estratégia de treinamento do dataset belga BTSC [5], em que se obteve acurácia próxima e também foi feito treinamentos para outros países.

Em CIREŞAN *et al.* [29], é proposto um modelo de redes neurais convolucionais que usa uma arquitetura CNN com múltiplas camadas, parametrizada para executar na GPU. Devido à alta variação de contraste entre as imagens, foi realizada normalização de contraste de 8 maneiras diferentes, totalizando 8 conjuntos de teste: 4 conjuntos com normalização em escala cinza e 4 conjuntos com normalização em cores. Nestes processos os pixels são dimensionados por valores proporcionais a média das suas intensidades. Além disso, foi utilizado o dataset público alemão GTSRB [4], obtendo acurácia de 99,15%, mas as taxas de erros foram altas, entre 3 – 4%, maiores que as taxas de erro apresentadas por modelos CNNs propostos pela literatura para o mesmo dataset. A limitação do modelo é que o mesmo foi treinado de maneira restrita a um conjunto de dados contendo placas de apenas um país, o que levanta dúvidas quanto à sua aplicabilidade a outros países.

Em JIN *et al.* [36], apresentam o método *Hinge Loss Stochastic Gradient Descent* (HLSGD) para treinar CNNs que usam o método SGD como otimizador, apresentando uma convergência rápida e estável. Para treinar e avaliar o modelo, foi utilizado o dataset alemão GTSRB [4]. Assim como discutido acima, no contexto do problema tratado neste trabalho, essa abordagem incorre no problema de ficar restrita às placas de apenas um país. No nosso modelo, dentre as estratégias apresentadas, uma delas também usou CNN e otimizador como descida de gradiente estocástico, obtendo bom desempenho para o dataset alemão GTSRB citedatasetalemao e obteve resultados semelhantes. A principal vantagem do nosso modelo é que ele pode ser usado em outros datasets, conforme será discutido no Capítulo 5.

Em AGHDAM *et al.* [22], propuseram uma arquitetura de redes neurais convolucionais que permite reduzir o consumo de CPU no processamento de um modelo CNN para classificação de sinais de trânsito. Em seguida, propuseram um método que criou um conjunto de CNN que foram comparados com 2 conjuntos de CNNs avançados. Nesta comparação, o modelo proposto obteve redução no número de operações aritméticas em 88% e 73% comparados com estes dois conjuntos. Ele foi 0.1% mais preciso do que um dos conjuntos, e 0.04% menos preciso do que o outro. Além disso, obteve redução no número de multiplicações de 95% e 88%, e obteve acurácia de 0,2% e 0,4% a menos que estes 2 conjuntos. Foram utilizados o dataset alemão GTSRB [4] e belga BTSC [5]. Todas estas comparações foram feitas para um mesmo conjunto de dados. O artigo cita que o método usado obteve os melhores resultados de classificação com o dataset alemão GTSRB [4] mas ele não generaliza bem para outros datasets. No nosso trabalho, obteve-se performance semelhante com um maior número de datasets.

Em YANG *et al.* [73], apresentaram um trabalho de reconhecimento de sinal de trânsito em 2 etapas: identificação de um sinal em uma imagem e posterior classificação desse sinal. A detecção é feita usando detector de região *Maximally Stable Extremal Regions* (MSER) que faz separação de imagens consideradas válidas, pois existem falsas negativas. O MSER se baseia no formato e cor das placas dependendo da categoria do país, de maneira que, por exemplo, para o dataset chinês as cores possuíam a seguinte associação: (i) proibido: redondo vermelho;

(ii) perigo: triângulo amarelo; e (iii) mandatário: redondo azul. Assim sendo, para a classificação foi feito um pré-processamento baseado na probabilidade de cor e um HOG (*Histogram of Oriented Gradient*) que fornece um histograma de cores. Em seguida, foi utilizada usou uma rede convolucional para classificar os sinais detectados. As experiências foram feitas com os seguintes datasets: *Chinese Traffic Sign Dataset* (CTSD), *German Traffic Sign Detection Benchmark* (GTSDDB) e *German Traffic Sign Recognition Benchmark* (GTSRB). Comparando esse artigo com o presente trabalho, verificou-se que ambos usam CNNs e também efetuam pré-processamento nas imagens, tais como tornar em escala de cinza, equalização de histograma, resize de imagem e obtiveram acurácias próximas, mas este trabalho utilizou mais estratégias de treinamento e mais países com datasets de treinamento na classificação de sinais de trânsito.

Em SAHA *et al.* [57], propuseram três CNNs para classificação de sinais de trânsito que tiveram uma acurácia de 99,92%. As redes usadas foram: uma rede padrão GoogLeNet [63], uma rede padrão VGG [60] e uma terceira que aprende a escolher qual das duas redes anteriores vai ser usada na classificação de uma imagem, de modo a fazer um *ensemble learning*. Para isto, ele usa os atributos das imagens que correspondem à decisão probabilística neste processo de escolha e foi usada a base de dados de sinais de trânsito alemã GTSRB [4]. Dessa forma, esse trabalho levanta uma dúvida de como seria o desempenho em outro dataset, uma vez que ele informou que o melhor resultado foi usando o dataset alemão GTSRB [4] e verifica-se em muitos artigos que esta dataset apresenta bons resultados. O nosso modelo apresentou valores próximos aos obtidos por esse trabalho para o dataset alemão GTSRB [4], além do modelo também ter sido treinado para outros datasets.

CAO *et al.* [28] apresentaram um algoritmo de detecção e reconhecimento de sinais de trânsito, em que consideraram o baixo desempenho em situações causadas pelo ambiente, quando usado aprendizado profundo. O artigo aponta um problema no uso do padrão HSV de cores para identificar as placas de sinais, uma vez que são detectadas efetivamente pela forma. O artigo propôs então um modelo CNN LeNet-5 clássico, usando um kernel especial (Gabor) como kernel convolucional inicial, adicionando normalização em lote após a camada de agrupamento e usa algoritmo de otimização Adam [39]. Analogamente ao artigo anterior, este artigo deixa a dúvida de como seria o seu desempenho com outros datasets, com outros tipos de imagens e que possuem diferentes qualidades. Será que usar este mesmo pré-processamento teria os mesmos resultados? O nosso modelo também usa pré-processamento de normalização de cores para imagens, além de ser treinado para outros datasets, de maneira a validar a sua performance e aplicabilidade.

QAYYUM *et al.* [54] descrevem os Veículos Autônomos Conectados (*Connected Autonomous Vehicle* - CAVs) e falam sobre a sua importância nos Sistemas de Transporte Inteligente. São apresentadas 6 categorias em que cada uma é proporcional ao nível de autonomia do veículo e é ressaltada a importância do aprendizado de máquina para tais veículos. Por outro lado, também são elencados os problemas de segurança em relação ao aprendizado de máquina que podem ocasionar vítimas no trânsito. Por fim, são apresentadas soluções para estes problemas e os desafios a serem enfrentados.

Por sua vez, COHEN *et al.* [31], fizeram um estudo sobre a importância dos

CAVs nos transportes e uma visão de como podem ser as cidades no futuro. Mostraram os impactos positivos e negativos dessa abordagem nas cidades, destacando consequências em empregabilidade do turismo, oportunidades socioeconômicas e etc, assim como realçaram os problemas associados e os desafios a serem resolvidos. Os autores apresentaram este estudo como base de pesquisa para novos planejamentos e novas políticas.

A Tabela 3.1 mostra um comparativo entre os artigos supracitados e os datasets usados. A coluna Grupo corresponde aos artigos que criaram um dataset único através da união dos outros datasets ou criaram um dataset separado. Todos os artigos desta tabela usaram modelos baseados em Redes Neurais Convolucionais.

Neste trabalho foi proposto um modelo que usou as melhores técnicas da literatura científica, distribuídas em estratégias, para obter um melhor desempenho de classificação encontrado nos artigos relacionados, assim como foram usados datasets consolidados. Assim foi possível gerar estratégias de classificação de sinais de trânsito. Entre estas técnicas foram utilizadas equalização de histogramas e normalização, como em [30], [29], [57] para o pré-processamento dos dados. Foi usado o dataset alemão GTSRB [4] que é uma referência para a classificação de sinais de trânsito. Na estratégia de combinação de datasets, alguns artigos serviram de referência e aperfeiçoamento deste trabalho, tal como [37]. Na combinação de datasets sem equivalências de classes, define-se para este caso uma classe “outras” no treinamento, assim como em [59]. Na estratégia de dataset único foram utilizadas técnicas presentes nos trabalhos de [37], [59] e [33]. Os resultados obtidos neste trabalho, assim como os melhores resultados obtidos pelos artigos elencados, para o dataset alemão GTSRB [4] foram acima na faixa de 99%. Para outros datasets também obteve-se bons resultados.

Tabela 3.1: Comparativo entre artigos e datasets utilizados. Abreviações usadas: Al (Alemanha), Be (Bélgica), Cr (Croácia), Su (Suécia), Ho (Holanda), Ch (China) e Fr (França).

Artigo	Al	Be	Cr	Su	Ho	Ch	Fr	Grupo
[30]	X							
[37]	X	X	X					X
[59]	X	X	X	X	X		X	X
[33]						X		X
[62]		X						
[29]	X							
[36]	X							
[22]	X							
[57]	X							
[28]	X							
Nosso trabalho	X	X	X					X

3.1 Conceitos Relevantes

Alguns conceitos adicionais relevantes na área de redes neurais são apresentados:

- Aprendizado por Máquina Adversária

- Adaptação de Domínio e Aprendizado por Transferência

3.1.1 Aprendizado por Máquina Adversária

O Aprendizado por Máquina Adversária, *Adversarial Machine Learning*, é uma área de pesquisa que permite utilizar com segurança técnicas de aprendizado de máquinas em ambiente contraditório. Ataques são executados causando alterações a uma imagem que é imperceptível ao ser humano mas permitem que uma imagem seja classificada incorretamente por classificadores baseados em redes neurais e sistemas legados de visão computacional. A seguir, são apresentados alguns artigos sobre este assunto:

[10] apresenta uma ferramenta web interativa chamada AdVis.js que abre uma opção para você entrar com uma imagem e ela gera uma imagem contraditória imperceptível para uma pessoa, mas que engana o classificador. Esta ferramenta atualiza dinamicamente, em tempo real, as pontuações de classificação através de um mapa de ativação de classe (CAM) que é uma técnica que visualiza as regiões de entrada importantes para previsões de modelos. O CAM usa informações do método *Fast Gradient Sign* (FGSM) da camada final de uma CNN.

[19] mostra um vídeo comparando uma situação normal de trânsito e uma mesma situação com um ataque. [20] apresenta um vídeo desta ferramenta mostrando 2 métodos de ataque e 2 métodos de defesa conforme a seguir :

- Attack #1 - Fast Gradient Sign Method
- Defense #1 - Adversarial Training
- Attack #2 - Backward Pass Differentiable Approximation (Torna frágil a defesa do FGSM)
- Defense #2 - The Barrage of Random Transforms

O Artigo [51] apresentou um pipeline que permite enganar um sistema TSR instalado em um veículo, através de impressão de sinais de trânsito contraditórios. Este ataque usou um classificador próprio para gerar sinais contraditórios no modo de caixa branca e, em seguida, testou o efeito destes sinais em um classificador diferente no modo de caixa preta.

Modo caixa preta é uma configuração de cenário em que o invasor não tem acesso ao classificador atacado. Modo caixa branca o invasor tem acesso total ao classificador atacado, à sua arquitetura, seus parâmetros e suas saídas.

O Artigo [76] apresentou um ataque a detectores de objetos em tempo real de última geração, como YOLO V3 e Faster RCNN, para distâncias entre 1m e 25m. Os experimentos mostraram um alto nível de transferência das AEs (Adversarial examples) geradas para outros detectores de objetos de caixa preta. O Artigo mostra que não existe nenhum mecanismo de defesa geral para ataques adversários contra detectores de objetos. Ele apresenta alguns mecanismos de defesa.

- Modificar as entradas para reduzir ou remover as perturbações criadas, usando compressão JPEG [32], randomização [71], filtro mediano [72], redimensionamento de imagem [34], etc. Para este caso, este método não obtem sucesso, pois as EAs foram geradas com várias transformações e ruídos aleatórios.

- Melhorar os modelos, como o treinamento adversário [65], destilação de defesa [53] e ofuscação por gradiente [24]. Contudo, essa defesa é limitada ao re-ataque e ataque transferível.
- Derrotar EAs com GAN(*generative adversarial network*) em que se gera um dado novo com o padrão do dado treinado. Um classificador pode ser treinado para distinguir se a entrada é contraditória ou não usando o GAN [58]. No entanto, esse GAN precisa ser treinado com base nas EAs geradas pelo artigo para derrotar o seu ataque.

O artigo [75] propôs um ataque chamado RoLMA(*Robust Light Mask Attacks*), contra sistemas LPR - *License Plate Recognition* (Reconhecimento de placas de carro) baseado em aprendizado profundo. Foi usado tecnologias de iluminação em que se cria uma placa adversária com vários pontos de luz com ruídos para perturbar as imagens das placas capturadas pelos sistemas LPR, em vez de técnicas de alterações de imagens.

LPR usa normalmente a técnica OCR (*optical character recognition*) que reconhece a placa de um veículo. Atualmente no OCR, em vez de usar reconhecimento caracter a caracter, alguns modelos de aprendizado profundo têm sido aplicados como RNNs, Hidden Markov Models, LSTMs e Gated Recurrent Units, que reconhece o texto todo. O HyperLPR [77] é um sistema LPR de ponta, mas mesmo assim o RoLMA consegue enganá-lo.

Após a apresentação deste artigos verificou-se que não existe uma solução efetiva para os ataques adversários. No artigo [47] criou-se uma imagem contraditória de sinal de trânsito e foi mostrado uma sequência em que após uma defesa é criado um novo ataque. No artigo [51] foi feito uma impressão que enganou o classificador tanto no modo caixa branca quanto preta. No Artigo [76] foi apresentado algumas soluções que reduzem o ataque a uma detecção de objetos em tempo real, mas não existe um mecanismos de defesa geral. No final foi apresentado o artigo [75] que, em vez de fazer alterações nas imagens, criou placas adversárias com alterações de luz, brilho que conseguiu enganar o HyperLPR que é um sistema LPR de ponta.

3.1.2 Adaptação de Domínio e Aprendizado por Transferência

Adaptação de Domínio

Adaptação do domínio consiste na transferência de conhecimento do domínio de origem D_s para o alvo D_t . O objetivo é então aprender h (a partir de amostras rotuladas ou não identificadas provenientes dos dois domínios), de modo que cometa o mínimo de erros possível no domínio de destino D_t [1]. conforme Figura 3.1.

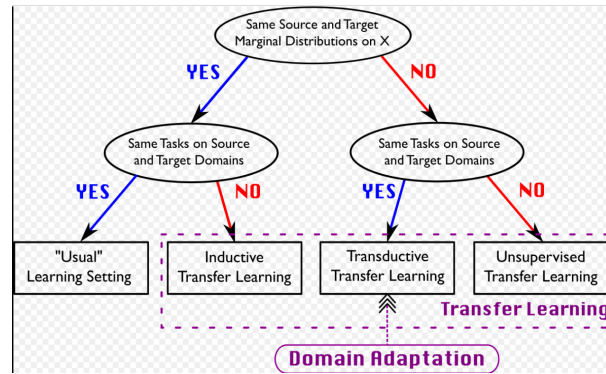


Figura 3.1: Adaptação de Domínio. Extraído de [1].

O Artigo [68] apresenta Transferable Attention for Domain Adaptation (TADA) que mostra um método de transferência de domínio. Ele tem 2 níveis de transferência: por imagem e por região.

- domínio em nível de região para destacar regiões transferíveis.
- domínio no nível de imagem única para destacar imagens transferíveis.

Aprendizado por Transferência

O Aprendizado por Transferência é uma técnica que pega uma parte de um modelo que já foi treinado e reutiliza-o em um novo modelo, conforme Figura 3.2. Neste processo extrai-se informações relevantes de imagens visando melhorias nas classificações de outros classificadores. Sendo assim retira-se as camadas que não deseja de uma rede pronta e coloca novas camadas sobre esta rede, desta forma treina-se apenas estas novas camadas.

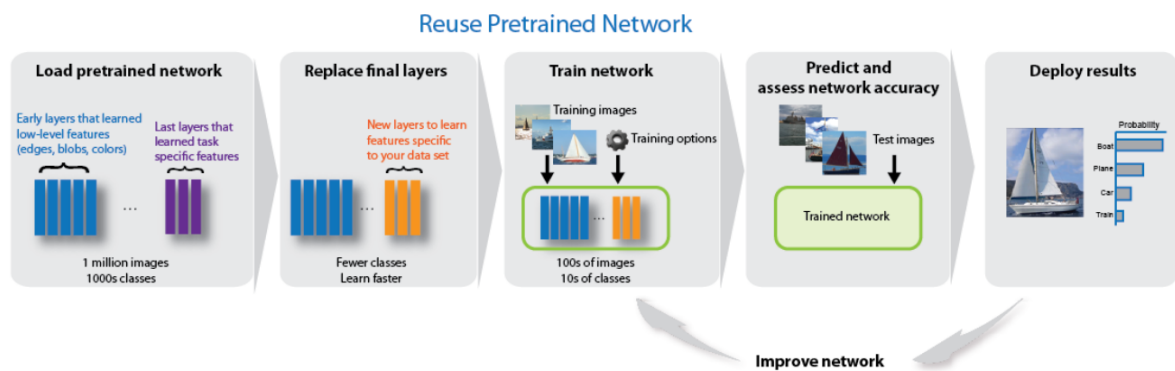


Figura 3.2: Aprendizado por Transferência. Extraído de [2].

O Artigo [74] apresenta um método que calcula o nível de transferência de características de cada camada de rede neural, e também mostra a sua generalidade e especificidade. Durante a divisão da rede, apresenta os problemas de transferência de características quando são transferidos da parte inferior, média ou superior da rede. Contudo a inicialização com recursos transferidos pode melhorar o desempenho da generalização, consequentemente da rede.

O Artigo [59] usou uma parte do conceito de *Transfer Learning* em que utiliza os pesos de um pré-treinamento como inicializadores para os novos procedimentos de treinamento.

O Artigo [22] também usou uma parte deste conceito em que treinou um modelo com um dataset alemão e testou com um dataset Belga.

Até este momento não verificou-se um modelo que utilizasse o conceito de *Transfer Learning*, que permitisse um VA trafegar entre países, utilizando o treinamento do modelo com dataset de um país e ao entrar em outro país pudesse aprender com novos sinais de trânsito.

O Artigo [45] propôs um algoritmo de reconhecimento de sinais de trânsito que utiliza aprendizado profundo chamado DeepSign, dividido em três módulos, mostrado na Figura 3.3:

- Um módulo de detecção Position Network(PosNet) para localizar o sinal de trânsito em uma imagem estática.
- Um módulo de classificação Patch Network (PatchNet) para classificar o patch da imagem detectado.
- Um filtro temporal para corrigir os resultados deste reconhecimento.

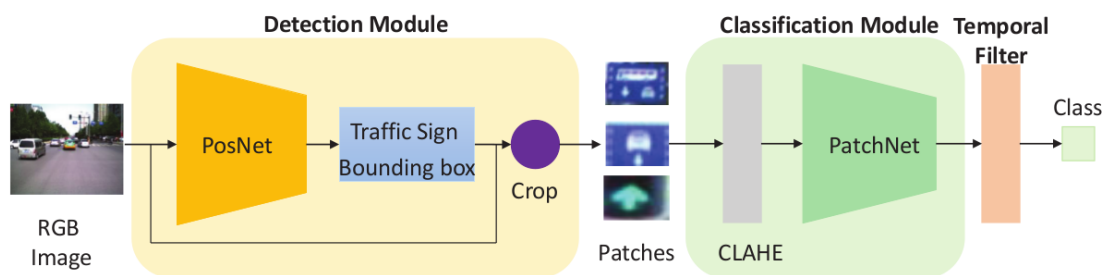


Figura 3.3: DeepSign. Extraído de [45].

PosNet é uma CNN de detecção de objetos, primeiro detecta os sinais de trânsito em uma imagem RGB. Depois de cortar a região detectada, os patches da imagem são convertidos em imagens cinza e processados pelo Contrast Limited Adaptive Histogram Equalization (CLAHE) e enviado para a rede de correção (PatchNet) para classificação. Finalmente, o filtro temporal é empregado para melhorar o resultado do reconhecimento. O módulo de detecção foi o Faster R-CNN. O dataset usado foi o CIVFC de 2017 com 133 videos(100 treinamentos, 33 validacao). Foram utilizadas 65 classes de sinais de trânsito.

O Artigo cita como trabalho futuro o uso de rede neural LSTM como forma de explorar melhor as informações temporais.

O Artigo [52] apresenta um *survey* de aprendizado por tranferencia em que verifica-se sua importância quando tem-se que expandir a classificação em outros domínios que podem ter distribuições de dados diferentes. Uma vantagem é economia de processamento, uma vez que economiza etapas de treinamento, evitando esforços de rotulagem de dados.

Nesta pesquisa, discute-se a relação entre o aprendizado de transferência e outras técnicas de aprendizado de máquina relacionadas, como adaptação de domínio, aprendizado multitarefas, viés de seleção de amostras e covariância.

A Figura 3.4 e a Figura 3.5 mostram a relação entre as diferentes configurações da transferência de aprendizado e as áreas relacionadas.

Figura 3.4: Tabela de configuração de aprendizado por transferência. Extraído de [52].

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
<i>Inductive Transfer Learning</i>	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
<i>Transductive Transfer Learning</i>	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
<i>Unsupervised Transfer Learning</i>		Unavailable	Unavailable	Clustering, Dimensionality Reduction

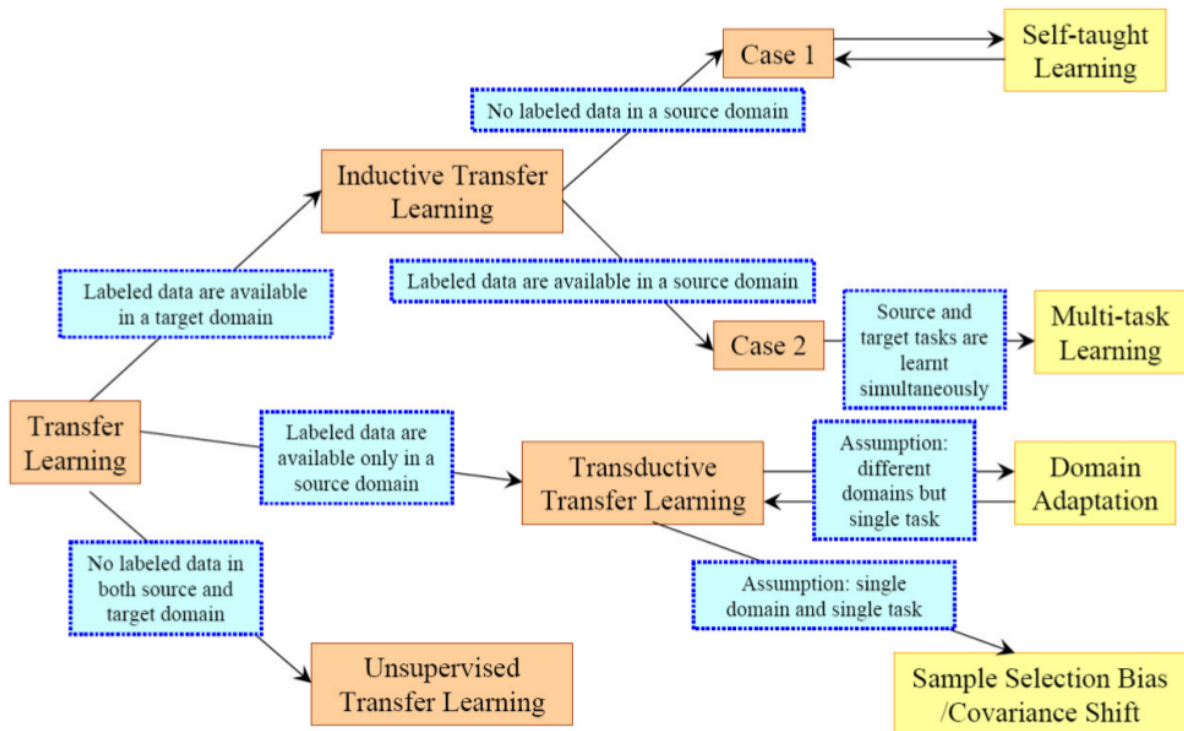


Figura 3.5: Aprendizado por Transferência. Extraído de [52].

Como podemos ver, as técnicas tradicionais de aprendizado de máquina tentam aprender cada tarefa do zero, enquanto as técnicas de aprendizado de transferência tentam transferir o conhecimento de algumas tarefas anteriores para uma tarefa de destino quando esta possui menos dados de treinamento de alta qualidade.

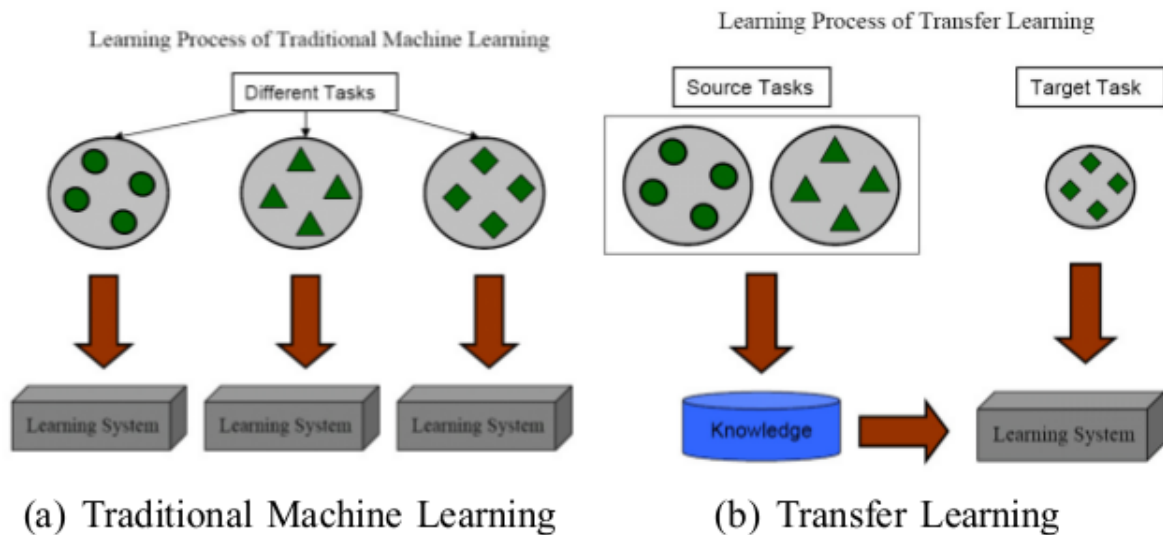


Figura 3.6: Diferença entre Aprendizado de Máquina e Aprendizado por Transferência. Extraído de [52].

O Artigo [64] mostrou que o aprendizado de transferência é uma ferramenta importante no sentido que ela tenta transferir o conhecimento do domínio de origem para o domínio de destino e apresenta algumas vantagens como:

- Os dados de treinamento não precisam ser independentes e distribuídos de forma idêntica com os dados de teste.
- o modelo no domínio de destino não precisa ser treinado a partir do zero, o que pode reduzir significativamente a demanda de dados de treinamento e o tempo de treinamento no domínio de destino.

Conceitos importantes: Domínio e Tarefa.

Domínio. $D = \{\chi, P(X)\}$

- χ - espaço de características.
- $P(X)$ - Distribuição de probabilidade da borda. $X = \{x_1, \dots, x_n\} \in \chi$.

Tarefa. $T = y, f(x)$.

- y - espaço do rótulo.
- $f(x)$ - função de previsão do alvo. Também pode ser considerado como uma função de probabilidade condicional $P(y | x)$.

Este artigo classificou o aprendizado por transferência em 4 categorias:

- Baseado em instâncias. Utiliza instâncias no domínio de origem pelo peso apropriado. Figura 3.7
- Baseado em mapeamento. Mapeia instâncias de dois domínios para um novo espaço de dados com melhor semelhança. Figura 3.8

- Baseado em Redes . Reutiliza a parte parcial da rede pré-treinada no domínio de origem. Figura 3.9
- Baseado em Adversário. Usa a tecnologia adversária para encontrar recursos transferíveis adequados para dois domínios. Figura 3.10

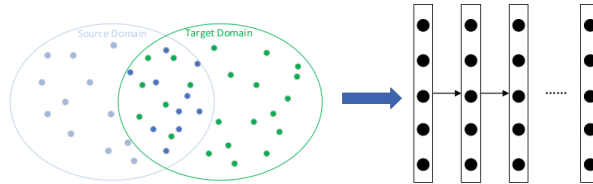


Figura 3.7: Transferência baseado em instâncias. Extraído de [64].

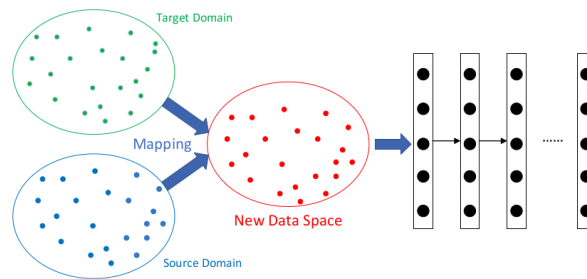


Figura 3.8: Transferência baseado em mapeamento. Extraído de [64].

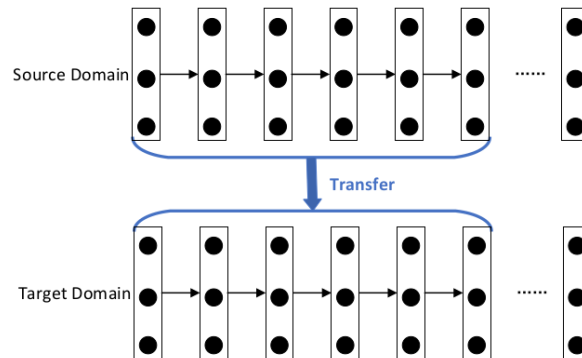


Figura 3.9: Transferência baseado em redes. Extraído de [64].

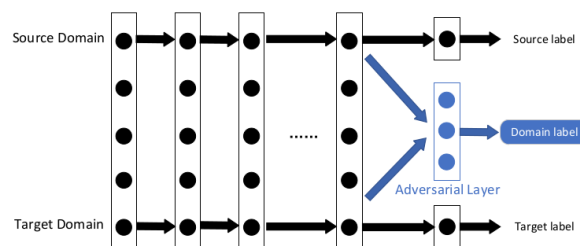


Figura 3.10: Transferência baseado em adversário. Extraído de [64].

O Artigo [66] propôs um algoritmo de detecção de veículo baseado em um modelo de distribuição de características em múltiplos subespaços usando aprendizado de transferência on-line, conforme a Figura 3.11. É estabelecido um modelo em que várias Restricted Boltzmann Machines (RBMs) são usadas para criar múltiplos subespaços com as características das primeiras camadas e uma *Deep Belief Network* (DBN) é usada para criar uma super estrutura. Depois foi aplicado um método de extração de recursos não supervisionado, com base em restrições esparsas. Foi proposta uma estrutura de aprendizado de transferência com uma etapa de geração de amostra on-line e o método de treinamento baseado no modelo apresentado.

Neste experimento foi usado o conjunto de dados experimento KITTI que contém 7481 fotos e inclui 35.000 veículos, enquanto o kit de teste KITTI contém 7518 fotos e inclui aproximadamente 27.000 veículos.

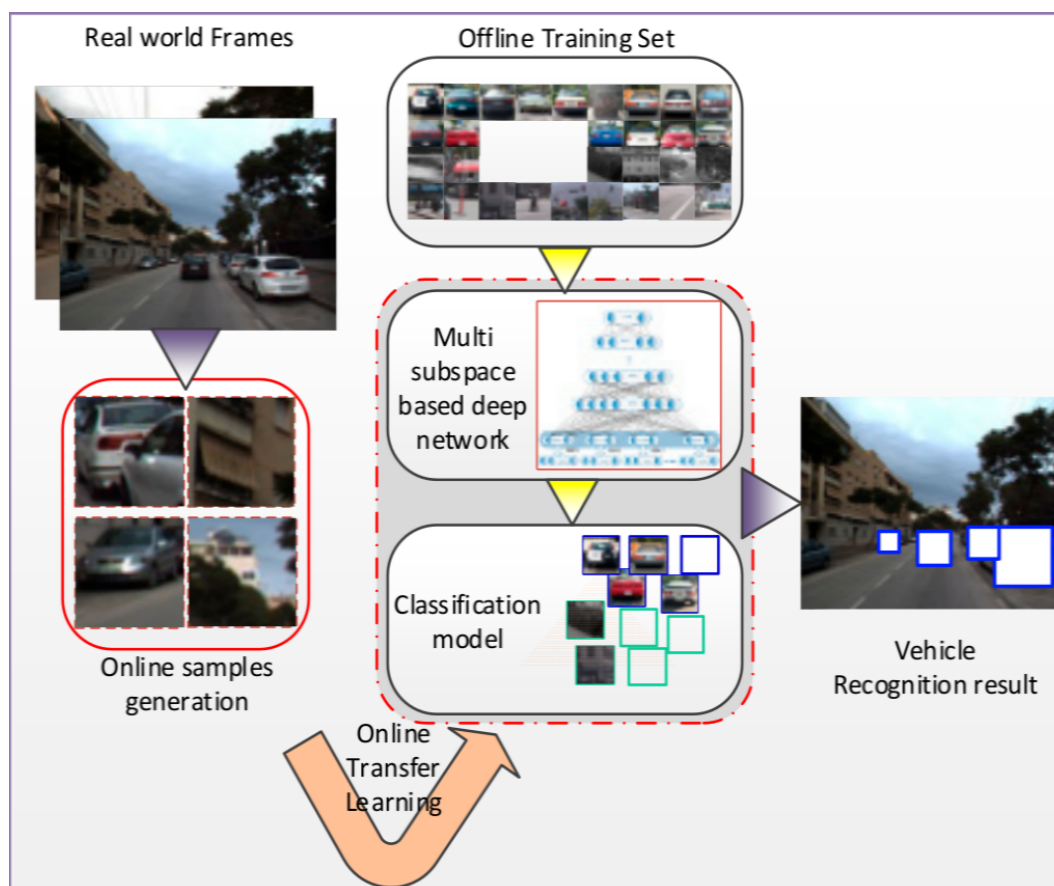


Figura 3.11: Transferência baseado em subespaço. Extraído de [66].

Capítulo 4

Método Proposto

Neste capítulo, descreve-se o método proposto que é composto inicialmente por uma etapa de pré-processamento, em que as imagens são normalizadas e redimensionadas. Na seção seguinte é apresentado o modelo CNN proposto, em que detalha a arquitetura usada. Por fim, é descrito o treinamento usado de acordo com o modelo CNN.

4.1 Pré-processamento

O método de pré-processamento utilizado inclui um conjunto de procedimentos para limpeza e formatação dos dados. Isso reduz o custo computacional na etapa posterior e melhora a identificação de características das imagens. Para a classificação, as cores das imagens não são fatores indispensáveis na extração de características, pois, sinais são feitos para pessoas com deficiência na detecção de cores [48].

Mais especificamente, as transformações aplicadas e suas justificativas estão descritas abaixo:

- A escala de cinzas possibilita criar uma uniformidade entre as imagens e identificar suas semelhanças e diferenças, além de reduzir o número de canais do formato RGB de três para um, melhorando os resultados em alguns casos [48].
- A normalização retira o excesso de cor, contraste e brilho. Isto permite que a imagem tenha um padrão que facilita identificar as características.
- O contraste permite identificar melhor as imagens de fundo.
- O redimensionamento reduz o tamanho original das imagens. Em [59] é demonstrado que as dimensões de 48×48 permite obter bons resultados em se tratando da classificação de imagens de datasets de sinais de trânsito.
- A centralização ajuda o treinamento do modelo deslocando a imagem para o eixo central.
- O *augmentation* é um método usado para aumentar os dados do treinamento para que a rede consiga aprender os padrões que se deseja. Para isto, é usado artifícios como translação, rotação, inclinação, ruídos, variação de brilho,

saturação, zoom entre outras. Esta técnica permite reduzir o overfitting e melhorar os resultados da rede sem adicionar novas características.

A Figura 4.1 exibe alguns resultados destes procedimentos que permitem reduzir o número de parâmetros das redes neurais, reduzindo o custo computacional na classificação e tornando o modelo mais eficiente. Essas operações permitem reduzir também os custos de armazenamento das imagens e de processamento computacional sem afetar a acurácia da classificação.



Figura 4.1: Ilustração do pré-processamento de algumas imagens. Extraído de [30].

4.2 Modelo CNN

O modelo CNN proposto tem configuração conforme a Tabela 4.1 em que as camadas são ilustradas na Figura 4.2. Neste modelo são aplicadas convoluções de tamanho 32, 64 e 128. A cada *maxpooling*, é reduzida a dimensão das imagens pela metade, pois, cada matriz (2, 2) calcula o valor máximo. O *dropout* quando usado não altera o tamanho das imagens utilizadas. Em seguida, é feito um *flatten* (achatamento) convertendo a saída em um vetor. A seguir, é executado uma camada densa com 512 convoluções. Por fim, executa-se uma camada densa com a função *softmax* que gera a classificação. Nas últimas linhas, observa-se o resultado específico para cada país.

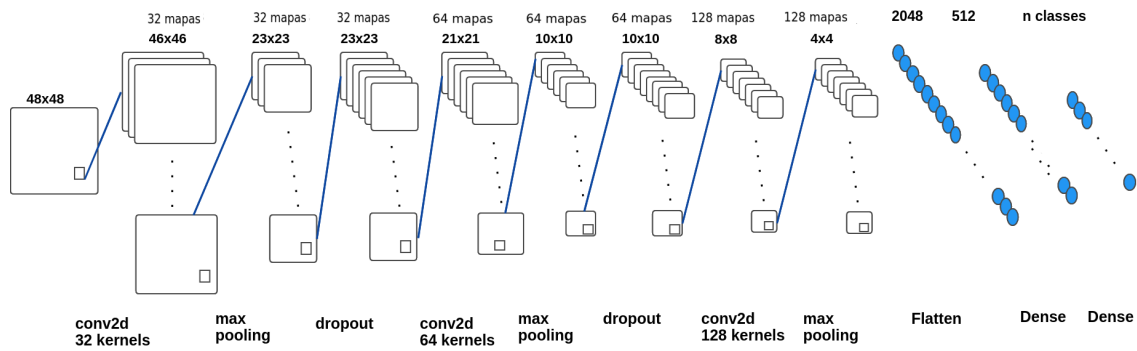


Figura 4.2: Modelo CNN, baseado em [43].

Tabela 4.1: Descrição das camadas e hiperparâmetros usados pelo modelo CNN proposto

Layer (type)	Output Shape	Param#
conv2d_1 (Conv2D)	(None, 32, 48, 48)	896
conv2d_2 (Conv2D)	(None, 32, 46, 46)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 32, 23, 23)	0
dropout_1 (Dropout)	(None, 32, 23, 23)	0
conv2d_3 (Conv2D)	(None, 64, 23, 23)	18496
conv2d_4 (Conv2D)	(None, 64, 21, 21)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 64, 10, 10)	0
dropout_2 (Dropout)	(None, 64, 10, 10)	0
conv2d_5 (Conv2D)	(None, 128, 10, 10)	73856
conv2d_6 (Conv2D)	(None, 128, 8, 8)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 128, 4, 4)	0
dropout_3 (Dropout)	(None, 128, 4, 4)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dropout_4 (Dropout)	(None, 512)	0
dense_2 (Dense) Alemão	(None, 43)	22059
dense_2 (Dense) Belga	(None, 62)	31806
dense_2 (Dense) Croata	(None, 31)	15903

4.3 Treinamento

O treinamento é feito lendo cada diretório do treino que corresponde ao nome da classe correspondente. As imagens lidas são inicialmente pré-processadas e depois armazenadas em um *array*. Outro *array* armazena as classes correspondentes.

Em seguida, é instanciado o modelo CNN. Esse modelo adota *categorical_crossentropy* como função de erro, pelo fato do problema ser de classificação. Por sua vez, o otimizador utilizado é o *Stochastic Gradient Descent*. Neste treinamento, são ado-

tados *epoch* = 30 e *validation_split* de 20% e 30%, por que observou-se que estes valores obtiveram as melhores acurácias.

Após a instanciação do modelo, é executado o treinamento de várias formas, variando e/ou combinando alguns hiperparâmetros como: *batch_size*, *augmentation* e *dropout* para descobrir a combinação que gera a melhor acurácia.

Após a etapa de treinamento, é realizada a etapa de teste. O modelo CNN é então aplicado aos dados de teste. Dessa forma, são obtidas as métricas acurácia, *precision*, *recall* e *f1 score*. Com o resultado constata-se o desempenho do modelo, verificando também se houve *underfitting* ou *overfitting*.

Capítulo 5

Experimentos

Neste capítulo serão descritos os experimentos realizados. O objetivo desses experimentos é verificar como pode-se avaliar a classificação de sinais de trânsito por veículos autônomos, fazendo uso de algoritmos de aprendizado de máquina e se deslocando por diferentes países.

Assim, inicialmente é descrito o ambiente computacional em que foram feitos os experimentos, assim como o conjunto de dados utilizados, valores de hiperparâmetros utilizados e métricas usadas. Por fim, serão mostrados os experimentos e os resultados obtidos, bem como uma análise sobre os mesmos.

5.1 Ambiente Computacional

O ambiente computacional usado na implementação foi um Notebook Gamer Dell com 16 GB de memória, processador i7-8750H, placa gráfica Nvidia GeForce GTX 1060 com 6GB e o sistema operacional Ubuntu 16 Linux.

Na parte de software, a implementação dos programas foi feita na linguagem Python 3.7 com interface CUDA para GPU, de maneira que foi possível executar todas as rotinas do código de forma rápida através da placa gráfica e o uso de processamento em paralelo. Mais especificamente, foram utilizadas as bibliotecas *keras* [12] para rotinas de aprendizado de máquina e *Numpy* [14] para operações de álgebra linear. Por sua vez, a geração dos gráficos foi realizada através da biblioteca *Matplotlib* [13] e para o cálculo das métricas usou-se *scikit-learn* [17]. Ademais, para tratamento de imagens foi usado o *scikit-image* [16].

A seguir será dada uma breve descrição das principais ferramentas utilizadas na implementação:

- CUDA. Tecnologia desenvolvida pela NVIDIA. Permite utilizar uma GPU como um multiprocessador paralelo. Isto é base do processamento de imagens nas redes neurais em que usa processamento de matrizes em paralelo na identificação de características.
- SCIKIT-LEARN. Biblioteca usada para aprendizado de máquina. Possui rotinas para classificação, regressão, cálculo de métricas e matriz de confusão.
- KERAS. Biblioteca desenvolvida pela google, de alto nível, que roda sobre o Tensor flow, utilizada para aprendizado de máquinas. Permite criar, con-

figurar, treinar redes neurais. Muito usado na identificação de imagens ou fotos. Pode ser usado com CPU ou GPU.

- NUMPY. Biblioteca utilizada para operações com matrizes. Permite operações de entrada, saída, dimensionamento destas matrizes e uso em funções matemáticas e estatísticas.
- MATPLOTLIB. Biblioteca para plotar gráficos 2D. Permite flexibilidade de forma, cor, visualização, através de adição de parâmetros.
- SCIKIT-IMAGE. Biblioteca usada no tratamento de imagens. Permite conversão de cores, normalização e redimensionamento.

Os programas em Python estão em ¹.

5.2 Datasets

Os datasets Alemão, Belga e Croata de sinais de trânsito utilizados nos experimentos do continente europeu seguem um padrão formado por categorias que definem o formato das placas de sinais como é mostrado na Figura 5.1.

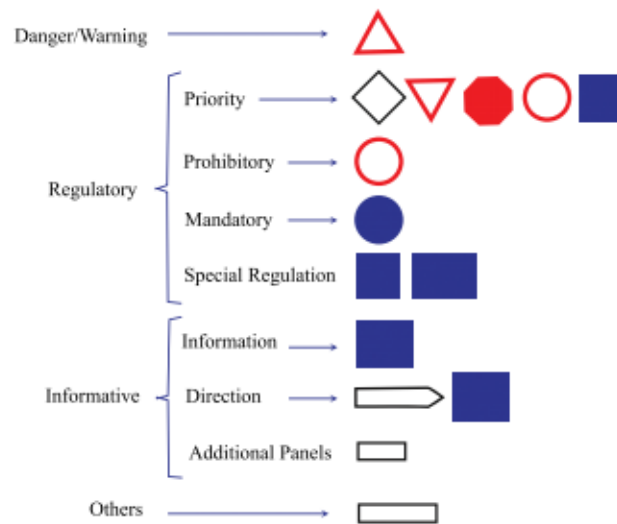


Figura 5.1: Padrão de sinais de trânsito europeu extraído de [59].

Estes 3 conjuntos de dados são públicos e podem ser usados livremente em pesquisas acadêmicas. Os dados de treinamento e teste são divididos em diretórios que possuem o nome da classe correspondente. As imagens estão em formato “ppm” (*Portable Pixmap*), sendo de fácil armazenamento e manipulação de imagens coloridas.

¹<https://github.com/nicomarf/MultiTrafficClassification>

5.2.1 Dataset Alemão

O conjunto de sinais de trânsito alemão, conhecido como *German Traffic Sign Recognition Benchmarks* (GTSRB) [4], é composto por um conjunto de dados de treinamento com 31.367 imagens, uma base de teste com 12.630 imagens, sendo que cada base de dados contém 43 classes distintas. A Figura 5.2 ilustra uma imagem de cada classe existente, discriminando também a quantidade de elementos em cada categoria.



Figura 5.2: Conjunto de sinais de trânsito alemão contendo 43 classes.

5.2.2 Dataset Belga

Por sua vez, o dataset *Belgium Traffic Sign Classification* (BTSC) [5] reúne os sinais de trânsito da Bélgica, sendo composto por um conjunto de treinamento contendo 4.591 imagens, um de teste contendo 2.534 imagens e cada base de dados contém 62 classes. A Figura 5.3 ilustra uma imagem de cada classe existente no dataset, além do quantitativo de elementos em cada categoria.



Figura 5.3: Conjunto de sinais de trânsito belga contendo 62 classes.

5.2.3 Dataset Croata

O conjunto de imagens representativas de sinais de trânsito da Croácia rMASTIF [6] foi criado pelo MASTIF (*Mapping and Assessing the State of Traffic InFrastructure*), sendo composto por um conjunto de treinamento contendo 4.044 imagens e uma de teste contendo 1.784 imagens. Cada base de dados contém 31 classes. A Figura 5.4 ilustra uma imagem de cada classe, além do quantitativo de elementos em cada categoria.



Figura 5.4: Conjunto de sinais de trânsito croata contendo 31 classes.

A Tabela 5.1 exibe um resumo de cada conjunto de dados.

Tabela 5.1: Visão dos datasets de trânsito

País	Dados	Classes	Treino	Teste	Total
Alemanha	GTSRB	43	31367	12630	43997
Bélgica	BTSC	62	4591	2534	7125
Croácia	rMASTIF	31	4044	1784	5828

5.3 Hiperparâmetros

Neste trabalho, foram realizados diversos experimentos para determinar quais hiperparâmetros obtinham os melhores resultados nos treinamentos e testes. Dessa forma, foram testados os seguintes valores de *batch size*: 8, 16, 24, 32, 40 e 48. Para o dataset alemão, as melhores acurácias foram obtidas com os maiores valores de *batch size*, pois esta base de dados contém muitas imagens. Isso se opõe ao resultado obtido para os demais datasets, que tiveram melhores resultados com os menores *batch sizes*. Foram usadas 30 épocas onde se atingiu os melhores resultados em todos os treinamentos.

O método utilizado para otimização foi o *Stochastic Gradiente Descendent* (SGD). Nesta função, usaram-se os seguintes parâmetros: *learning rate* de 0.01, taxa de decaimento do aprendizado foi *decay*= $1e-6$. Por seu turno, para o parâmetro que acelera este decaimento, foi usado *momentum* = 0.9. E para *nesterov*, usou-se *True*. Segundo Keras [15], os valores apresentados obtêm resultados ótimos ou vão se ajustando com as interações.

Além disso, foram também definidos 3 importantes hiperparâmetros: a função de erro *loss*='categorical_crossentropy', a função de otimização SGD e a métrica *accuracy*.

Para o treinamento foram configurados os seguintes hiperparâmetros: *validation_split*, *shuffle*, *callback*. *Validation_split* define o percentual das imagens de treinamento usado para validação. Nos nossos testes foram utilizados os valores: 0.2 e 0.3. *Shuffle=True* embaralhou os dados a cada passada. Para o parâmetro *Callback* foi utilizada uma função de escalonamento do aprendizado *LearningRateScheduler* que reduziu o aprendizado proporcionalmente ao número de épocas.

5.4 Métricas

As métricas de avaliação utilizadas neste trabalho são funções que avaliam a performance do modelo no treinamento e no teste. Hiperparâmetros como *dropout*, *batch size* e métodos artificiais como *augmentation* tendem a uma possibilidade de melhoria dos valores das métricas. Entre as métricas importantes estão: *Accuracy*, *Precision*, *Recall* e F1-Score. No que se segue, serão descritas tais métricas, em que a seguinte convenção foi adotada: vp (verdadeiro positivo), vn (verdadeiro negativo), fp (falso positivo) e fn (falso negativo).

Accuracy permite fazer uma análise completa comparando as classificações corretas em relação a todas as classificações. A Equação 5.1 define esta métrica:

$$\text{Accuracy} = \frac{vp + vn}{vp + vn + fp + fn} \quad (5.1)$$

Precision permite calcular o percentual de classificações positivas calculadas pelo modelo em relação a quantidade positiva correta. A Equação 5.2 define esta métrica:

$$\text{Precision} = \frac{vp}{vp + fp} \quad (5.2)$$

Recall permite calcular a relação entre todos que tiveram situação positiva e as corretas. A Equação 5.3 define esta métrica:

$$\text{Recall} = \frac{vp}{vp + fn} \quad (5.3)$$

F1-Score também conhecida como F-Score ou F-measure, calcula a média harmônica entre *Precision* e *Recall*. A Equação 5.4 define esta métrica:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.4)$$

5.5 Estratégias de treinamento

Diversas estratégias de treinamento foram abordadas neste trabalho. Dessa forma, os resultados obtidos são expostos através de tabelas em que a primeira coluna representa os valores de hiperparâmetros e a primeira linha descreve os *batch sizes* usados. Este formato é apresentado na Tabela 5.2.

Tabela 5.2: Modelo de tabela de treinamento

Hiperparâmetros	Descrição
batch size	batch size usados nos treinos.
sem drop	acurácias obtidas sem dropout no modelo CNN adotado.
augm 20 + valid 20	acurácias obtidas com augmentation de 20%, validation de 20%
augm 20 + valid 30	acurácias obtidas com augmentation de 20%, validation de 30%
drop 0,2	acurácias obtidas com dropout de 20%
drop 0,2 + augm 20	acurácias obtidas com dropout de 20% e augmentation de 20%

Para cada experimento a ser descrito nas próximas subseções, são mostrados o seguinte:

- A tabela de acurácia de cada treinamento.
- A tabela com as métricas da melhor acurácia desta tabela.
- A Matriz de Confusão dos 5 maiores percentuais de erro de cada treinamento.
 - A coluna Erros mostra o total de erros de cada classe.
 - A coluna Total mostra o total de imagens de cada classe.
 - A coluna Perc mostra o percentual de Erros sobre Total.
- Placas de sinais de trânsito desta Matriz de Confusão.

No Apêndice é mostrado a Matriz de Confusão completa de cada treinamento.

5.6 Experimento 1 - Treinamento Individual

Esta primeira estratégia de treinamento foi usada para responder à questão da pesquisa que procura saber qual é o impacto de diferentes estratégias de treinamento do modelo proposto em cima de diferentes conjuntos de dados de placas. Assim, este treinamento é constituído de experimentos feitos para os datasets de treino e teste de cada país individualmente. Dessa forma, primeiramente é feita a leitura do dataset de treino passando por pré-processamento para deixar os dados normalizados e redimensionados. Em seguida, o modelo CNN faz o treinamento gerando um arquivo contendo o valor dos parâmetros obtidos e que é usado na etapa de teste para obter as métricas de avaliação, conforme a Figura 5.5.

Nas subseções seguintes, são apresentados os resultados do treinamento individual dos datasets de sinais de trânsito da Alemanha, Bélgica e Croácia que seguem o formato apresentado na Tabela 5.2.

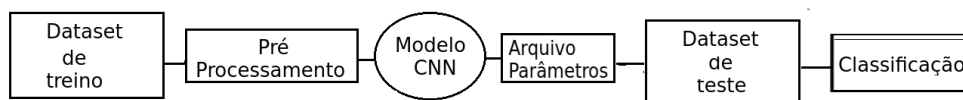


Figura 5.5: Arquitetura de treinamento individual

5.6.1 Dataset Alemão

Foi feito o treinamento de um modelo sobre o dataset alemão, seguindo os passos apresentados na Seção 5.6.

As especificações deste dataset estão na Subseção 5.2.1. A Tabela 5.3 mostra os hiperparâmetros usados no treinamento com dataset de treino e a acurácia obtida quando aplicado este modelo no dataset de teste. A melhor acurácia foi de 98.89% obtida quando usou *batch size* de 24, *dropout* de 0.2 e *augmentation* de 20%. Neste caso, *precision* foi de 98.49%, *recall* foi de 98.74% e *f1 score* foi de 98.58% conforme Tabela 5.4. A Tabela 5.5 mostra a tabela da matriz de confusão resumida dos 5 maiores percentuais de erros deste treinamento. A Figura 5.6 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.3: Acurácia do treinamento do dataset alemão

batch size	16	24	32	48
sem drop	93.94	95.70	95.24	94.95
augm 20 + val 20	97.77	98.17	98.15	97.78
augm 20 + val 30	97.96	98.13	97.74	97.71
drop 0,2	94.53	97.15	97.25	97.15
drop 0,2 + augm 20	97.67	98.89	98.67	98.36

Tabela 5.4: Métricas da melhor acurácia (alemão)

Batch size	24
Acurácia	98.89
Precision	98.49
Recall	98.74
F1 Score	98.58

Tabela 5.5: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Alemão.

Class	2	5	6	8	11	13	15	18	20	21	22	23	25	26	28	29	30	31	34	36	40	42	Erros	Total	Perc
6	0	6	120	4	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	30	1500	20
11	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	20	420	4.76
18	1	0	0	0	0	0	1	364	0	3	6	0	1	6	1	0	0	7	0	0	0	0	26	390	6.67
30	0	2	0	1	1	0	0	0	7	0	0	1	1	0	0	1	134	2	0	0	0	0	16	150	10.67
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	84	0	6	90	6.67

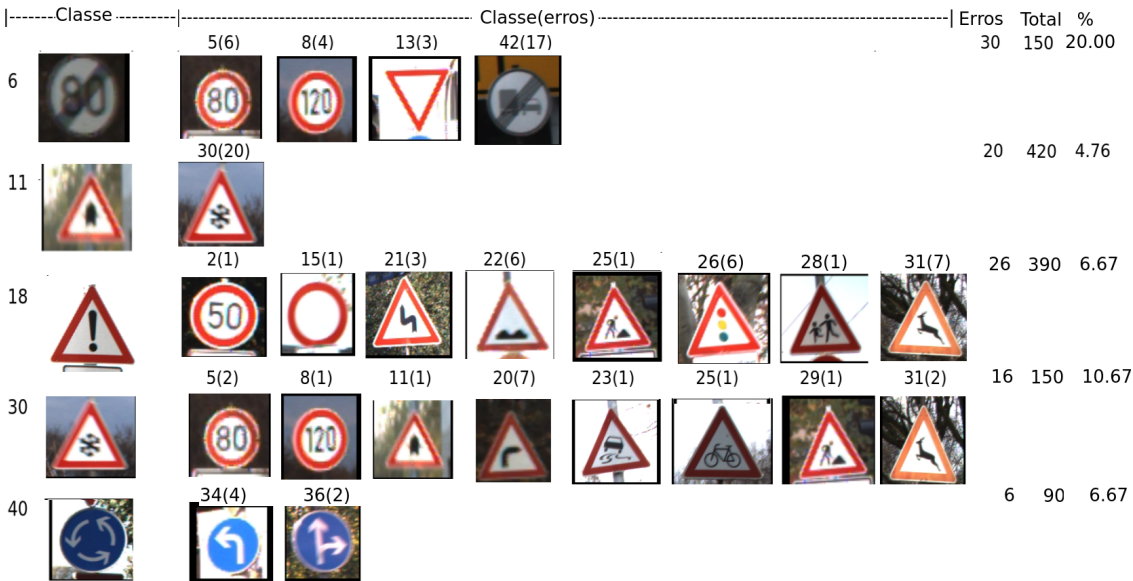


Figura 5.6: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Alemão.

5.6.2 Dataset Belga

Usando o dataset belga, foi feito o treinamento de um modelo, seguindo os passos apresentados na Seção 5.6.

A subseção 5.2.2 descreve detalhadamente este dataset e a Tabela 5.6 mostra os hiperparâmetros usados no treinamento com dataset de treino e a acurácia obtida quando aplicado este modelo no dataset de teste. A melhor acurácia foi de 97.77% obtida quando usou *batch size* de 8, *dropout* de 0.2 e *augmentation* de 20%. Neste caso, *precision* foi de 87.80%, *recall* foi de 89.06% e *f1 score* foi de 98.58% conforme Tabela 5.7. A Tabela 5.8 mostra a tabela da matriz de confusão resumida dos 5 maiores percentuais de erros deste treinamento. A Figura 5.7 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.6: Acurácia do treinamento do dataset belga				
batch size	8	16	24	32
sem drop	93.96	96.66	96.38	96.70
augm 20 + val 20	93.80	96.42	95.59	94.04
augm 20 + val 30	97.61	97.46	96.38	96.54
drop 0,2	95.95	96.62	96.34	96.07
drop 0,2 + augm 20%	97.77	97.02	96.19	95.67

Tabela 5.7: Métricas da melhor acurácia (belga)

Batch size	8
Acurácia	97.77
Precision	87.80
Recall	89.06
F1 Score	88.00

Tabela 5.8: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Belga

Class	3	8	42	43	45	46	47	49	51	52	56	57	Erros	Total	Perc
3	4	2	0	0	0	0	0	0	0	0	0	0	2	6	33.33
42	0	0	5	4	0	0	0	0	0	0	0	0	4	9	44.44
46	0	0	0	0	3	3	0	0	0	0	0	0	3	6	50.00
49	0	0	0	0	0	0	3	0	0	0	0	0	3	3	100.00
51	0	0	0	0	0	0	0	0	2	1	0	0	1	3	33.33
57	0	0	0	0	0	0	0	0	0	0	13	28	13	41	31.71



Figura 5.7: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Belga

5.6.3 Dataset Croata

Seguindo os procedimentos da Seção 5.6, e usando o dataset croata, foi realizado o treinamento do modelo para este dataset.

A configuração deste dataset está na Subseção 5.2.3 e a Tabela 5.9 mostra os hiperparâmetros usados no treinamento com dataset de treino e a acurácia obtida quando aplicado este modelo no dataset de teste. A melhor acurácia foi de 97.36% obtida quando usou *batch size* de 16, *augmentation* de 20% e *validation* de 30%, conforme Tabela 5.10. A Tabela 5.11 mostra a tabela da matriz de confusão

resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.8 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.9: Acurácia do treinamento do dataset croata

batch size	8	16	24	32
sem drop	91.42	93.94	94.73	94.22
augm 20 + val 20	96.13	95.90	97.02	95.17
augm 20 + val 30	94.61	97.36	96.58	91.81
drop 0,2	95.57	96.74	95.79	95.29
drop 0,2 + augm 20	96.58	96.80	94.95	94.84

Tabela 5.10: Métricas da melhor acurácia (croata)

Batch size	16
acurácia	97.36
Precision	96.95
Recall	96.08
F1 Score	96.36

Tabela 5.11: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Croata.

Class	0	1	2	5	6	7	9	10	11	13	17	18	19	20	23	25	26	29	30	Erros	Total	Perc
0	33	0	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	7	40	17.50
13	0	0	0	0	2	0	2	0	1	11	0	0	0	0	0	0	0	0	0	5	16	31.25
19	0	0	0	0	0	0	0	0	0	0	0	0	22	2	0	0	0	0	0	2	24	8.33
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	18	0	0	20	20	10.00
29	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	26	0	2	28	7.14

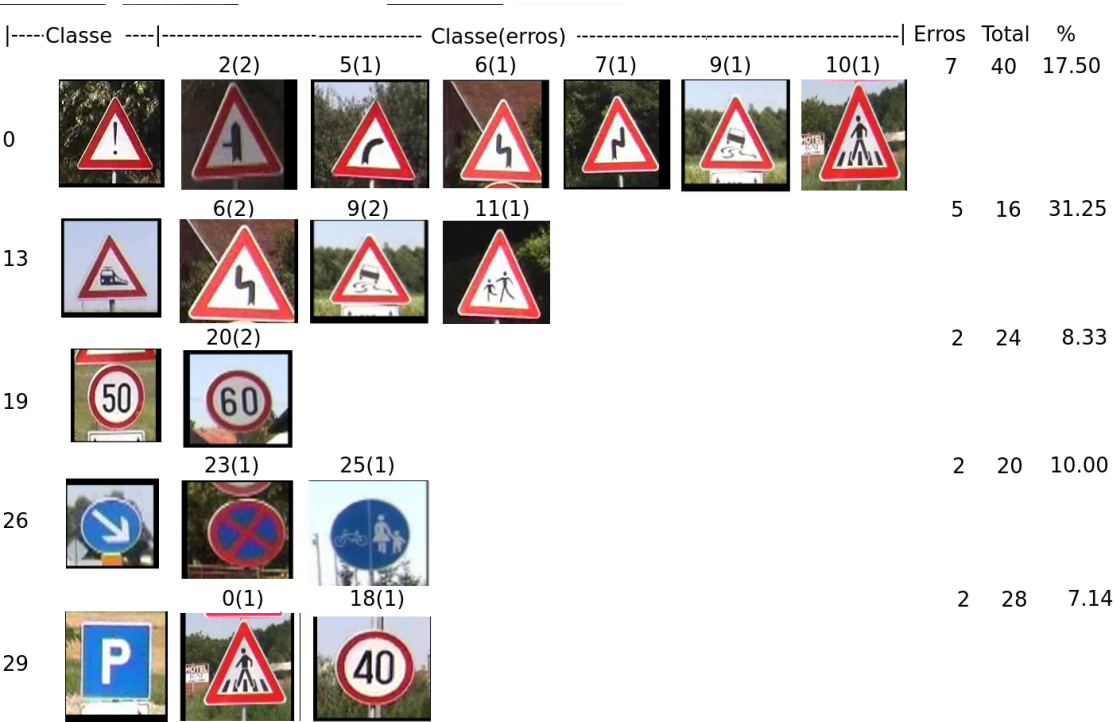


Figura 5.8: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Individual - Dataset Croata.

5.6.4 Resumo

A Tabela 5.12 mostra o resumo do treinamento individual dos datasets alemão, belga e croata.

Tabela 5.12: Resumo da acurácia dos datasets					
Dataset	Acur	Prec	Rec	BS	Valores dos Hiperparâmetros
Alemão	98.89	98.49	98.74	24	drop 0,2 + augm 20% + valid 20%
Belga	97.77	87.80	89.06	8	augm 20% + valid 20%
Croata	97.36	96.95	96.08	16	augm 20% + valid 30%

BS = Batch Size

Conclusão: Em todos os casos, o *Augmentation* esteve presente e a acurácia maior foi no dataset alemão em que as imagens tiveram mais representatividade. Verificou-se que o modelo usado no treinamento dos 3 datasets teve uma acurácia ótima, sendo acima de 97%.

5.7 Experimento 2 - Treinamento com datasets para classes equivalentes simples

Esta estratégia de treinamento foi usada para responder à questão da pesquisa que procura saber qual é o impacto de diferentes estratégias de treinamento de

modelos propostos em cima de diferentes conjuntos de dados de placas. Neste treinamento, os experimentos são feitos dois a dois com os datasets alemão, belga e croata. É feita uma combinação de treinamento e teste entre os datasets, de maneira que se treina o primeiro dataset com o modelo CNN adotado e realiza-se o teste sobre o segundo dataset, totalizando 6 execuções como é mostrado na Figura 5.9.

Os datasets têm classes em comum com rótulos diferentes. Por isso, foi preciso reclassificar estas classes para torná-las equivalentes, gerando assim uma base de treino do primeiro dataset e uma base de teste com o segundo dataset.

A Tabela 5.13 mostra uma visão geral dos datasets e suas classes correspondentes. Para cada experimento são mostradas as classes equivalentes entre os datasets executados. Todo o procedimento segue a metodologia proposta no Capítulo 4.

Nas subseções seguintes, são apresentados uma descrição e os resultados obtidos do treinamento de cada combinação destes datasets. Esses resultados seguem o formato apresentado na Tabela 5.2.

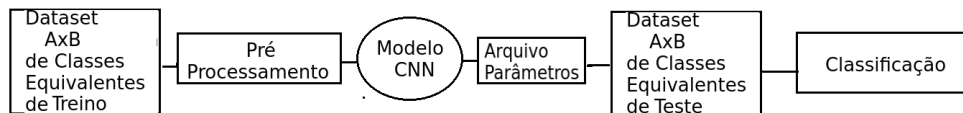


Figura 5.9: Modelo de treinamento de classes equivalentes.

Tabela 5.13: Tabela de equivalência entre as classes dos datasets dos países
O valor “**” significa que não tem classe correspondente.

Alemão	Belga	Croata	Descrição
1	**	17	Speed limit (30km/h)
2	**	19	Speed limit (50km/h)
3	**	20	Speed limit (60km/h)
4	**	21	Speed limit (70km/h)
9	31	22	No passing
11	17	1	Right-of-way at the next intersection
12	61	16	Priority road
13	19	14	Yield
14	21	15	Stop
**	14	8	Road narrow on both sides
15	28	**	No vehicles
16	25	**	Vehicles over 3.5 metric tons prohibited
17	22	**	No entry
18	13	0	General caution
19	3	4	Dangerous curve to the left
20	4	5	Dangerous curve to the right
21	5	6	Double curve (left)
22	0	**	Bumpy road
23	2	9	Slippery road
24	16	**	Road narrows on the right
25	10	**	Road work
26	11	**	Traffic signals
27	**	10	Pedestrians
28	7	11	Children crossing
29	8	**	Bicycles crossing
31	**	12	Wild animals crossing
35	34	**	Ahead only
36	36	**	Go straight or right
**	38	24	Bicycles only
**	39	25	shared use path
38	**	26	Keep right
40	37	**	Roundabout mandatory
41	**	27	End of no passing
**	6	7	Double curve (right)
**	41	23	No stopping
**	45	29	Parkin place
**	56	28	Pedestrian crossing

5.7.1 Alemão x Belga

Para responder à questão de pesquisa onde quer-se avaliar o impacto do treinamento de datasets com classes equivalentes entre o dataset alemão e belga, fez-se o relacionamento de classes conforme a Tabela 5.14, em que a primeira linha cor-

responde às classes do dataset alemão e a segunda linha corresponde às classes do dataset belga. Com isso, foi adotado o dataset alemão como base de treino e o belga, como base de teste.

Seguindo os procedimentos da Seção 5.7 para estes datasets criados, obtiveram-se os resultados conforme a Tabela 5.15. A melhor acurácia foi de 97.58% que foi obtida com *batch size* 16, *dropout* de 0.2 e *augmentation* de 20% com o tempo de 235s. A Tabela 5.17 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.10 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.14: Classes equivalentes Alemão x Belga

9	11	12	13	14	15	16	17	18	19	20
31	17	61	19	21	28	25	22	13	3	4

21	22	23	24	25	26	28	29	35	36	40
5	0	2	16	10	11	7	8	34	36	37

Tabela 5.15: Tabela de acurácia Alemão x Belga

batch size	16	24	32	48
sem drop	86.44	89.49	87.39	87.81
augm 20 + val 20	97.26	96.32	97.26	95.27
augm 20 + val 30	94.85	95.48	96.42	95.58
drop 0,2	92.22	89.81	92.75	91.07
drop 0,2 + augm 20	97.58	97.26	96.63	96.53

Tabela 5.16: Métricas da acurácia Alemão x Belga

Accuracy	97.58
Precision	91.63
Recall	89.68
F1 score	89.04

O tempo de treinamento desta melhor acurácia foi de 235s.

Tabela 5.17: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Belga.

Class	9	11	13	14	15	19	24	28	29	35	40	Erros	Total	Perc
14	0	0	1	43	0	0	0	0	0	0	1	2	45	4,44
15	1	0	0	1	49	0	0	0	0	0	0	2	51	3,92
19	0	0	0	0	0	5	1	0	0	0	0	1	6	16,67
29	0	3	0	0	0	0	1	6	2	0	0	10	12	83,33
40	0	0	0	0	0	0	0	0	0	1	30	1	31	3,23

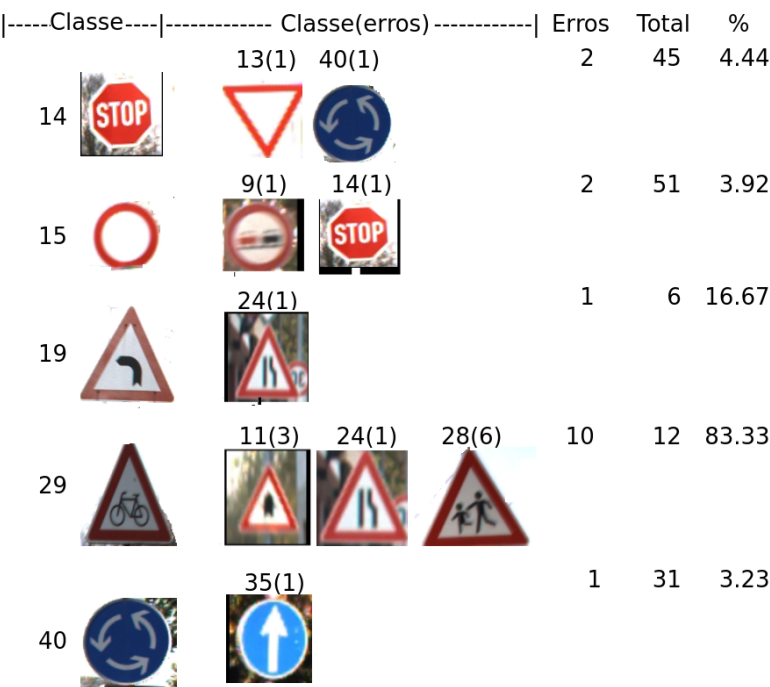


Figura 5.10: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Belga.

5.7.2 Alemão x Croata

Este treinamento avalia o impacto do treinamento feito com as classes equivalentes como mostra a Tabela 5.18 que foi usada para criar a base de treino alemã e a base de teste croata. Seguindo os procedimentos da Seção 5.7, obtiveram-se os resultados que estão apresentados na Tabela 5.19. A melhor acurácia foi 91.98% que foi obtida com *batch size* 24, *dropout* de 0.2 e *augmentation* de 20% com o tempo de 271s. A Tabela 5.21 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.11 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.18: Classes equivalentes Alemão x Croata

1	2	3	4	9	11	12	13	14	18	19	20	21	23	27	28	31	38	41
17	19	20	21	22	1	16	14	15	0	4	5	6	9	10	11	12	26	27

Tabela 5.19: Tabela de acurácia Alemão x Croácia

batch size	16	24	32	48
sem drop	85.27	84.14	83.62	83.44
augm 20 + val 20	91.81	89.54	87.80	89.45
augm 20 + val 30	89.54	90.50	88.85	88.32
drop 0,2	83.79	84.05	87.63	84.40
drop 0,2 + augm 20	91.02	91.98	90.41	88.32

Tabela 5.20: Métricas da acurácia Alemão x Croácia

Accuracy	91.98
Precision	91.48
Recall	90.60
F1 score	90.19

O tempo de treinamento desta melhor acurácia foi de 271s.

Tabela 5.21: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Croata

Class	1	2	3	11	12	14	18	20	23	27	28	31	Erro	Total	Percen
2	0	20	0	0	1	3	0	0	0	0	0	0	4	24	16.67
3	0	21	19	0	0	0	0	0	0	0	0	0	21	40	52.50
18	0	0	0	0	1	0	33	3	0	3	0	0	7	40	17.50
23	0	0	0	0	0	0	0	2	35	0	3	0	5	40	12.50
27	1	0	0	2	0	0	12	0	1	60	2	2	20	80	25.00

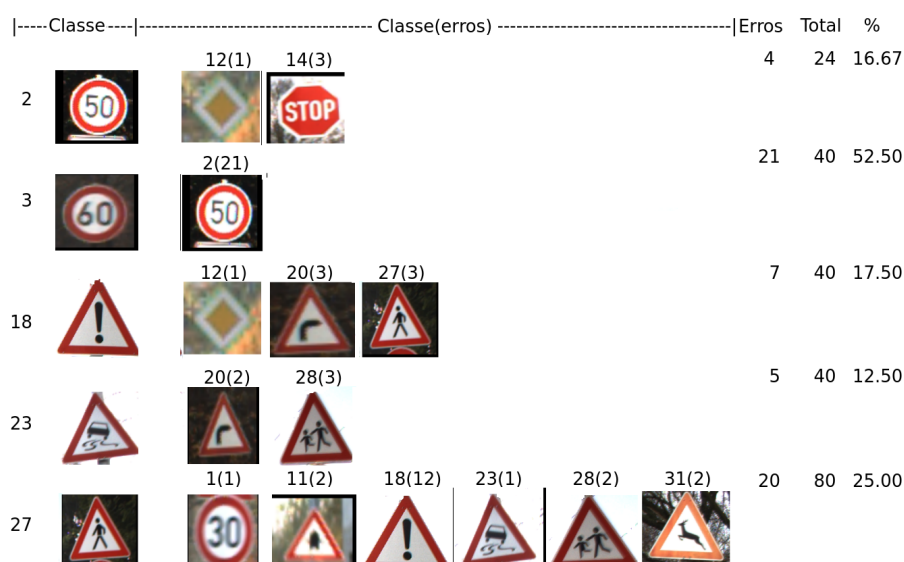


Figura 5.11: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Alemão x Croata

5.7.3 Belga x Alemão

Este treinamento avalia o impacto do treinamento feito com as classes equivalentes como mostra a Tabela 5.22, que foi usada para criar a base de treino belga e a base de teste alemã. Seguindo os procedimentos da Seção 5.7, obtiveram-se os resultados que estão na Tabela 5.23. A melhor acurácia foi 65% que foi obtida com *batch size* 32, *augmentation* de 20% e *validation* 30%, com o tempo de 45s. A Tabela 5.25 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.12 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.22: Classes equivalentes Belga x Alemão

0	2	3	4	5	7	8	10	11	13	16
22	23	19	20	21	28	29	15	26	18	24

17	19	21	22	25	28	31	34	36	37	61
11	13	14	17	16	15	9	35	36	40	12

Tabela 5.23: Acurácia Belga x Alemão

batch size	16	24	32	48
sem drop	50.41	58.37	56.00	47.92
augm 20 + val 20	64.37	60.67	61.95	63.59
augm 20 + val 30	63.59	62.95	65.00	64.52
drop 0,2	57.37	60.00	55.95	54.32
drop 0,2 + augm 20	63.25	62.83	61.27	59.33

Tabela 5.24: Métricas da acurácia Belga x Alemão

Accuracy	65.00
Precision	56.45
Recall	55.20
F1 score	51.94

O tempo de treinamento desta melhor acurácia foi de 45s.

Tabela 5.25: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Alemão.

Class	0	2	3	4	5	7	8	10	11	13	16	17	22	34	36	37	Erros	Total	Percent
2	0	30	1	9	35	31	0	0	0	4	0	0	0	28	0	12	120	150	80.00
4	0	40	0	8	2	27	0	0	0	4	0	2	0	1	0	6	82	90	91.11
8	10	5	29	10	15	12	0	1	0	0	6	0	2	0	0	0	90	90	100.00
11	0	11	2	1	0	2	0	0	34	111	0	8	11	0	0	0	146	180	81.11
16	1	11	26	2	2	4	4	2	1	8	13	1	2	6	1	6	77	90	85.56

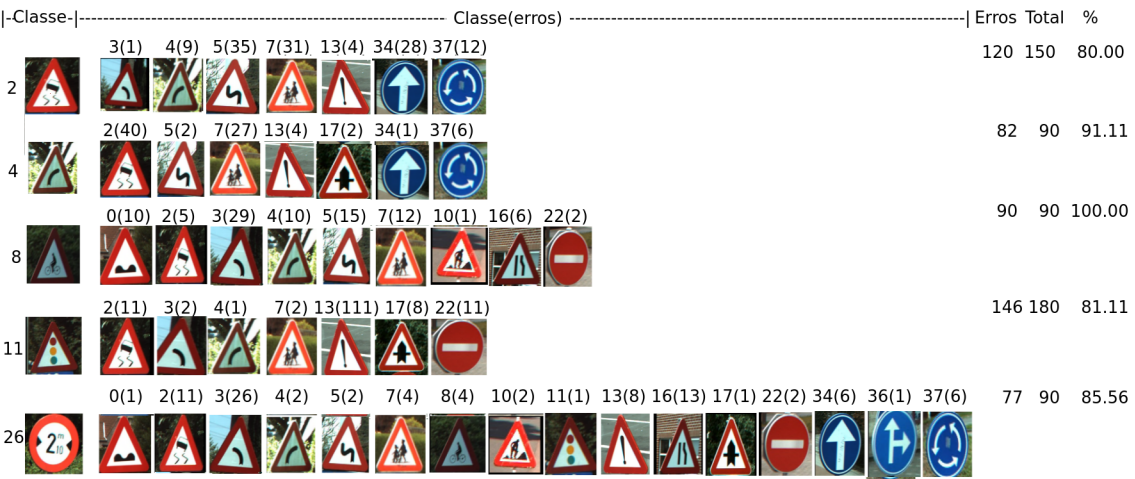


Figura 5.12: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Alemão.

5.7.4 Belga x Croata

Este treinamento avalia o impacto do treinamento feito com as classes equivalentes como mostra a Tabela 5.26 que foi usada para criar a base de treino belga e a base de teste croata. Seguindo os procedimentos da Seção 5.7, obtiveram-se os resultados que estão na Tabela 5.27. A melhor acurácia foi 61.99% que foi obtida com *batch size* 24 e *augmentation* de 20% com o tempo de 30s. A Tabela 5.29 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.13 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.26: Classes equivalentes Belga x Croata

02	03	04	05	13	14	17	19	21	31	38	39	61	07	06	41	45	56
09	04	05	06	00	08	01	14	15	22	24	25	16	11	07	23	29	28

Tabela 5.27: Acurácia do treinamento do dataset Belga x Croata

batch size	16	24	32	48
sem drop	50.92	49.66	44.93	42.22
augm 20 + val 20	51.35	61.99	57.68	61.06
augm 20 + val 30	49.74	57.85	56.08	54.72
drop 0,2	53.46	49.83	51.68	53.20
drop 0,2 + augm 20	56.75	57.17	52.02	54.56

Tabela 5.28: Métricas da acurácia Belga x Croata

Accuracy	61.99
Precision	66.20
Recall	55.63
F1 score	52.56

O tempo de treinamento desta melhor acurácia foi de 30s.

Tabela 5.29: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Croata.

Class	2	3	4	5	6	7	13	14	17	19	21	31	38	39	41	56	61	Erros	Total	Perc
5	3	9	4	2	0	2	13	9	0	0	8	1	1	0	0	1	7	58	60	96.67
6	1	1	17	0	8	16	1	2	0	1	8	1	0	0	0	1	15	64	72	88.89
17	0	0	1	0	0	4	13	4	4	2	0	1	0	0	0	2	17	44	48	91.67
41	0	0	0	0	0	0	0	0	0	0	0	0	19	0	8	0	1	20	28	71.43
45	0	0	0	0	0	0	0	0	0	0	0	0	14	3	0	3	0	20	28	71.43

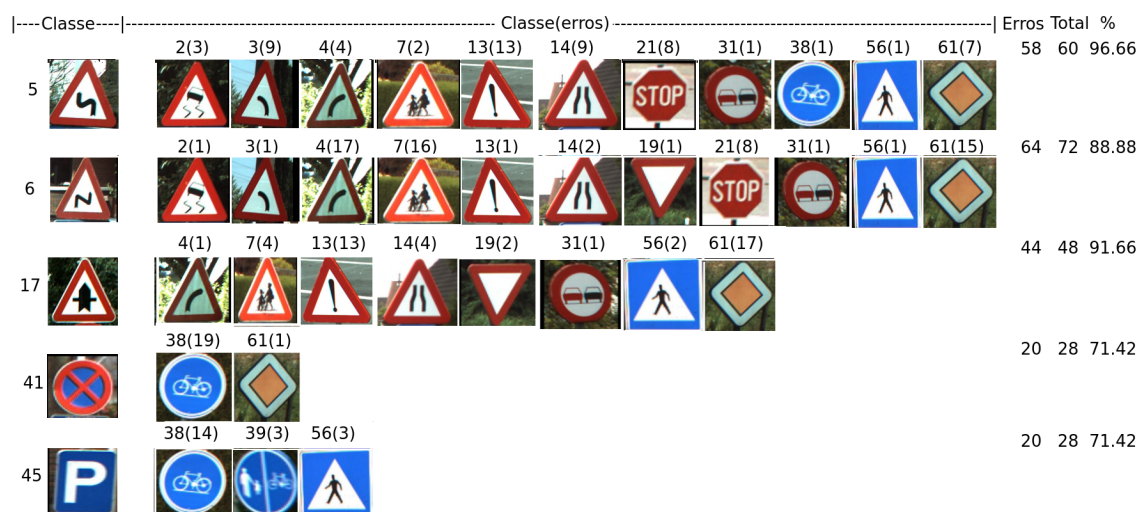


Figura 5.13: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Belga x Croata.

5.7.5 Croata x Alemão

Este treinamento foi feito com as classes equivalentes como mostra a Tabela 5.30. O Resultado da acurácia está na Tabela 5.31. A melhor acurácia foi 56.67% que foi obtida com *batch size* 32 e *augmentation* de 20% com o tempo de 67s. A Tabela 5.33 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.14 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.30: Classes equivalentes Croata x Alemão

0	1	4	5	6	9	10	11	12	14	15	16	17	19	20	21	22	26	27
18	11	19	20	21	23	27	28	31	13	14	12	01	02	03	04	09	38	41

Tabela 5.32: Métricas da acurácia Croata x Alemão

Accuracy	56.67
Precision	52.23
Recall	64.04
F1 score	51.05

Tabela 5.31: Acurácia Croata x Alemão

batch size	16	24	32	48
sem drop	50.27	49.70	50.40	49.69
augm 20 + val 20	55.27	52.86	56.67	55.01
augm 20 + val 30	52.62	48.78	52.37	54.03
drop 0,2	52.50	49.51	48.31	47.36
drop 0,2 + augm 20	52.17	49.58	46.62	51.84

O tempo de treinamento desta melhor acurácia foi de 67s.

Tabela 5.33: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Alemão.

Class	0	1	4	5	6	9	10	11	12	14	15	16	17	19	20	21	22	26	27	Erros	Total	Perc
10	15	13	0	2	0	0	26	0	4	0	0	0	0	0	0	0	0	0	0	34	60	56.67
17	17	374	9	9	133	16	1	11	183	6	0	6	649	29	0	34	10	9	64	911	1560	58.40
19	6	0	0	0	9	32	0	79	0	16	1	687	48	641	36	12	10	0	133	1069	1710	62.51
20	3	1	0	0	7	142	5	1	0	725	177	1	37	92	294	8	32	3	2	1236	1530	80.78
23	0	0	0	0	4	142	0	1	0	1	0	0	0	0	1	0	0	1	0	150	150	100.00

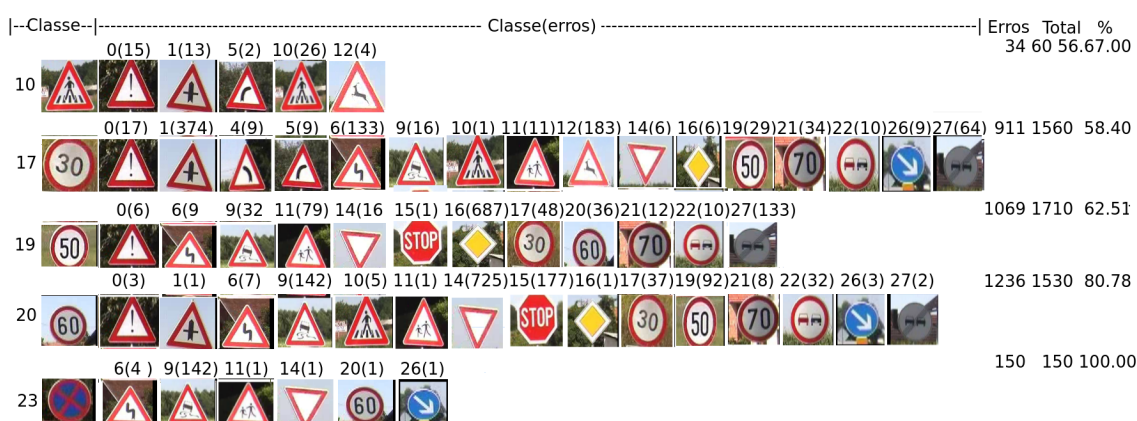


Figura 5.14: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Alemão.

5.7.6 Croata x Belga

Este treinamento avalia o impacto do treinamento feito com as classes equivalentes como mostra a Tabela 5.34 que foi usada para criar a base de treino croata e a base de teste belga. Seguindo os procedimentos da Seção 5.7, obtiveram-se os resultados que estão na Tabela 5.35. A melhor acurácia foi 91.08% que foi obtida com *batch size* 48, *augmentation* de 20% e *validation* 30%, com o tempo de 34s. A Tabela 5.37 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.15 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.34: Classes equivalentes Croata x Belga

0	1	4	5	6	7	8	9	11	14	15	16	22	23	24	25	28	29
13	17	3	4	5	6	14	2	7	19	21	61	31	41	38	39	56	45

Tabela 5.35: Acurácia Croata x Belga

batch size	16	24	32	48
sem drop	79.91	69.50	72.25	71.00
augm 20 + val 20	87.83	89.75	82.25	87.00
augm 20 + val 30	87.91	87.50	87.66	91.08
drop 0,2	83.08	74.66	76.16	76.25
drop 0,2 + augm 20	82.41	84.25	80.25	74.66

Tabela 5.36: Métricas da acurácia Croata x Belga

Accuracy	91.08
Precision	80.00
Recall	85.37
F1 score	79.58

O tempo de treinamento desta melhor acurácia foi de 34s.

Tabela 5.37: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Belga.

Class	0	1	4	5	6	7	9	11	15	22	23	24	25	28	29	Erros	Total	Perc
0	19	3	6	0	5	1	0	0	0	0	0	0	0	5	0	20	39	51.28
7	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	6	6	100.00
11	0	6	0	0	1	26	6	50	0	0	0	0	0	1	0	40	90	44.44
23	0	0	0	0	0	0	3	0	0	0	7	0	0	0	1	4	11	36.36
25	0	0	0	0	0	0	0	0	1	4	0	3	88	0	3	11	99	11.11

Classe	Classe(erros)	Erros	Total	%
0	1(3) 4(6) 6(5) 7(1) 28(5)	20	39	51.28
7	11(6)	6	6	100.00
11	1(6) 6(1) 7(26) 9(6) 28(1)	40	90	44.44
23	9(3) 29(1)	4	11	36.36
25	15(1) 22(4) 24(3) 29(3)	11	99	11.11

Figura 5.15: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Simples - Croata x Belga.

5.7.7 Resumo

A Tabela 5.38 lista as melhores acurácias obtidas nos treinamentos dos datasets da combinação dos países.

Tabela 5.38: Resumo da acurácia do treinamento dos datasets

Países	Acur	Prec	Rec	BS	Valores dos Hiperparâmetros
Alemão x Belga	97.58	91.63	89.68	16	drop 0,2+aug 20%
Alemão x Croata	91.98	91.48	90.60	24	drop 0,2+aug 20%
Belga x Alemão	65.00	56.45	55.20	32	augm 20% + valid 30%
Belga x Croata	61.99	66.20	55.63	24	augm 20% + valid 20%
Croata x Alemão	56.67	52.23	64.04	32	augm 20% + valid 20%
Croata x Belga	91.08	80.00	85.37	48	augm 20% + valid 30%

BS = Batch Size

Conclusão: Os modelos treinados a partir do dataset alemão obtiveram melhor acurácia quando aplicados a um segundo dataset na etapa de teste. Isto provavelmente se deve ao fato de que o dataset alemão tem muitas imagens de boa qualidade, de modo que os filtros do modelo CNN permitem uma boa variedade de características e, portanto, permitem uma melhor classificação. Por outro lado, quando o modelo foi treinado em outro dataset e testado sobre o alemão, é obtida uma acurácia baixa. Isso se justifica, pois os outros datasets têm menos características e, portanto, não permitem à CNN extrair características da mesma maneira, implicando uma acurácia ruim.

5.8 Experimento 3 - Treinamento com datasets para classes equivalentes completas

Esta estratégia de treinamento foi adotada para responder à questão da pesquisa que procura saber qual é o impacto de diferentes estratégias de treinamento do modelo proposto em cima de diferentes conjuntos de dados de placas. Este experimento é uma alteração do Experimento 2, descrito na Seção 5.7, em que um dataset é adotado como base de treino e outro, como de teste, e equivalência entre as classes é realizada. Entretanto, neste caso, tem mais uma classe adicional denominada de “outros”, que busca reunir as classes que não possuem equivalências.

Nas subseções seguintes são apresentados uma descrição e os resultados obtidos do treinamento de cada combinação destes datasets. Estes resultados seguem o formato apresentado na Tabela 5.2.

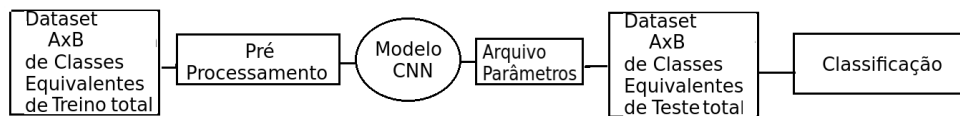


Figura 5.16: Modelo de treinamento de classes equivalentes total.

5.8.1 Alemão x Belga

Neste caso, foi criada equivalência entre as classes de ambos os datasets alemão e belga, tal como descrito na Tabela 5.14, e agruparam-se as classes que não tinham equivalência na classe artificial “outros” como mostra a Tabela 5.39.

Seguindo os procedimentos da Seção 5.8, obtiveram-se os resultados conforme a Tabela 5.40. A melhor acurácia foi de 91.07% que foi obtida com *batch size* 32, *dropout* de 0,2 e *augmentation* de 20% com o tempo de 472s. A Tabela 5.42 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.17 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.39: Todas as classes Alemão x Belga
O valor “**” significa que não tem classe correspondente.

Alemão	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
Belga	**	**	**	**	**	**	**	**	**	31	**	17	61	19	21	28	25	22

Alemão	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Belga	13	03	04	05	00	02	16	10	11	**	07	08	**	**	**	**	**	34

Alemão	36	37	38	39	40	41	42	**	**	**	**	**	**	**	**	**	**	**
Belga	36	**	**	**	37	**	**	01	06	09	12	14	15	18	20	23	24	26

Alemão	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
Belga	27	29	30	32	33	35	38	39	40	41	42	43	44	45	46	47	48	49

Alemão	**	**	**	**	**	**	**	**	**	**	**
Belga	50	51	52	53	54	55	56	57	58	59	60

Tabela 5.40: Acurácia Alemão x Belga completa

batch size	16	24	32	48
sem drop	82.10	82.61	80.75	83.53
augm 20 + val 20	88.05	88.25	85.91	90.35
augm 20 + val 30	62.22	90.51	90.51	86.50
drop 0,2	78.13	85.11	83.69	81.23
drop 0,2 + augm 20	87.42	90.99	91.07	86.19

Tabela 5.41: Métrica Alemão x Belga completa

Accuracy	91.07
Precision	70.71
Recall	86.21
F1 score	74.58

O tempo de treinamento desta melhor acurácia foi de 472s.

Tabela 5.42: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Belga.

Class	9	18	24	28	29	40	99	Erros	Total	Perc
9	82	0	0	0	0	0	4	4	86	4.65
18	0	38	1	0	0	0	0	1	39	2.56
28	0	0	0	87	0	0	3	3	90	3.33
29	0	0	0	3	2	0	7	10	12	83.33
40	0	0	0	0	0	29	2	2	31	6.45

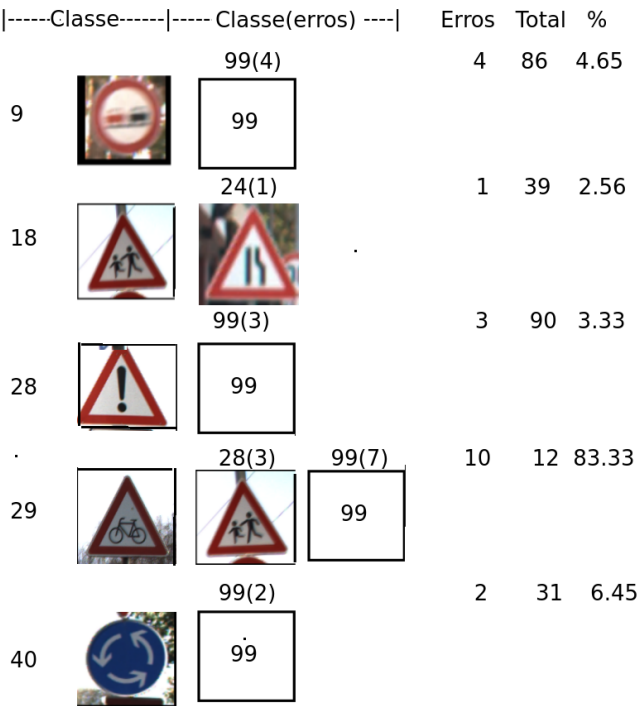


Figura 5.17: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Belga.

5.8.2 Alemão x Croata

Analogamente, neste caso, definiram-se equivalências entre as classes dos datasets alemão e croata, de maneira que a Tabela 5.18 expressa isso. Agruparam-se as classes que não tinham equivalência na classe artificial “outros” como mostra a Tabela 5.43.

Seguindo os procedimentos da Seção 5.8, obtiveram-se os resultados conforme a Tabela 5.44. A melhor acurácia foi 78,81% que foi obtida com *batch size* 24 e *augmentation* de 20% com o tempo de 478s. A Tabela 5.46 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.18 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.43: Todas as classes Alemão x Croata
O valor “**” significa que não tem classe correspondente.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
18	11	**	**	19	20	21	**	**	23	27	28	31	**	13	14	12	01	**

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
0	4	5	6	**	9	**	**	**	10	11	**	**	12	**	**	**	**

36	37	38	39	40	41	42
**	**	26	**	**	27	**

Tabela 5.44: Acurácia Alemão x Croata completa

batch size	16	24	32	48
sem drop	71.80	69.56	65.97	70.34
augm 20 + val 20	75.89	78.81	74.32	72.98
augm 20 + val 30	77.74	72.08	74.21	74.88
drop 0,2	72.92	72.02	68.32	70.90
drop 0,2 + augm 20	72.25	77.52	76.73	77.24

Tabela 5.45: Métrica Alemão x Croata completa

Accuracy	78.81
Precision	80.61
Recall	84.65
F1 score	81.61

O tempo de treinamento desta melhor acurácia foi de 478s.

Tabela 5.46: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Croata.

Class	1	2	3	9	11	13	14	18	20	23	27	28	31	41	99	Erros	Total	Perc
3	0	4	1	0	0	0	0	0	0	0	0	0	0	0	35	39	40	97.50
18	0	0	0	0	0	0	1	32	2	0	0	0	0	0	5	8	40	20.00
23	0	0	0	0	0	0	0	0	0	31	0	0	0	0	9	9	40	22.50
27	0	0	0	0	0	0	3	0	3	0	46	0	0	0	31	34	80	42.50
99	3	2	39	33	59	1	0	32	16	1	8	38	1	3	400	236	636	37.11

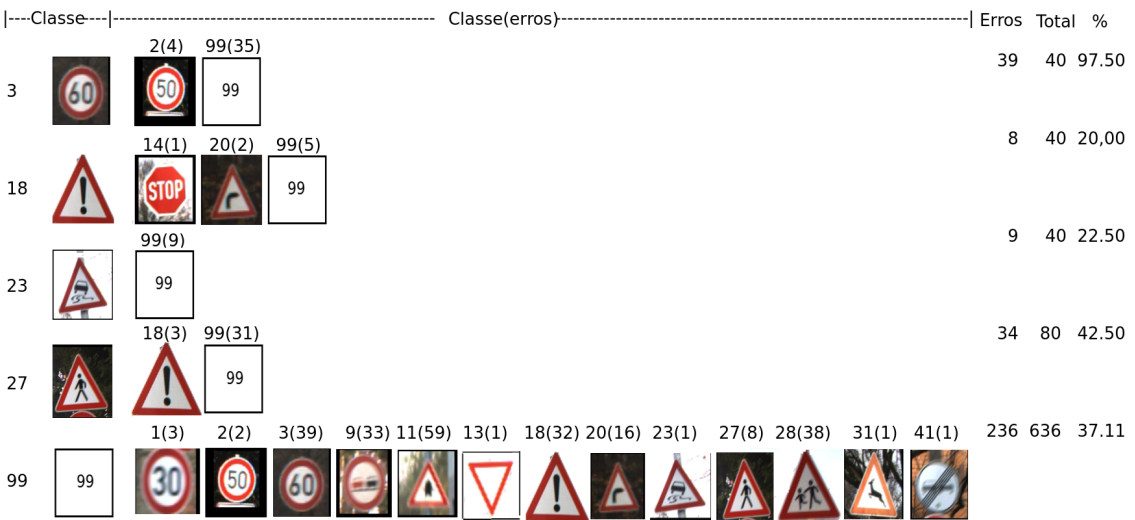


Figura 5.18: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Alemão x Croata

5.8.3 Belga x Alemão

Adotando-se o dataset belga como base de treinamento e o alemão, como teste, foram criados relacionamentos entre as classes de ambos, conforme a Tabela 5.22

e agruparam-se aquelas que não tinham equivalências na classe “outros” como mostra a Tabela 5.47.

Seguindo os procedimentos da Seção 5.8, obtiveram-se os resultados conforme a Tabela 5.48. A melhor acurácia foi 64,51% que foi obtida com *batch size* 16 e *augmentation* de 20% com o tempo de 128s. A Tabela 5.50 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.19 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.47: Todas as classes Belga x Alemão
O valor “**” significa que não tem classe correspondente.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
22	**	23	19	20	21	28	**	29	**	25	26	**	18	**	**	24	11

18	19	20	21	22	23	24	25	26	27	28	28	30	31	32	33	34	35
**	13	**	14	17	**	**	16	**	**	15	**	**	9	**	**	35	**

36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
36	40	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**

54	55	56	57	58	59	60	61
**	**	**	**	**	**	**	12

Tabela 5.48: Acurácia Belga x Alemão completa

batch size	16	24	32	48
sem drop	57.11	59.82	57.85	55.86
augm 20 + val 20	64.51	63.74	62.11	60.85
augm 20 + val 30	63.71	62.82	59.40	60.05
drop 0,2	62.51	62.53	59.32	60.06
drop 0,2 + augm 20	62.74	61.19	61.72	61.58

Tabela 5.49: Métrica Belga x Alemão completa

Accuracy	64.51
Precision	35.98
Recall	52.09
F1 score	40.27

O tempo de treinamento desta melhor acurácia foi de 128s.

Tabela 5.50: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Alemão.

Class	3	4	5	7	13	16	17	21	25	28	31	37	61	99	Erro	Total	Perce
2	0	0	1	3	0	0	0	0	0	0	0	0	0	146	150	150	100.00
5	1	0	11	0	0	3	1	0	0	0	0	0	0	74	79	90	87.78
8	0	3	8	1	0	5	0	1	0	0	0	0	0	72	90	90	100.00
16	0	0	0	7	1	14	0	0	0	0	0	2	0	66	76	90	84.44
31	26	3	8	1	0	5	0	1	1	76	224	0	4	371	496	720	68.89

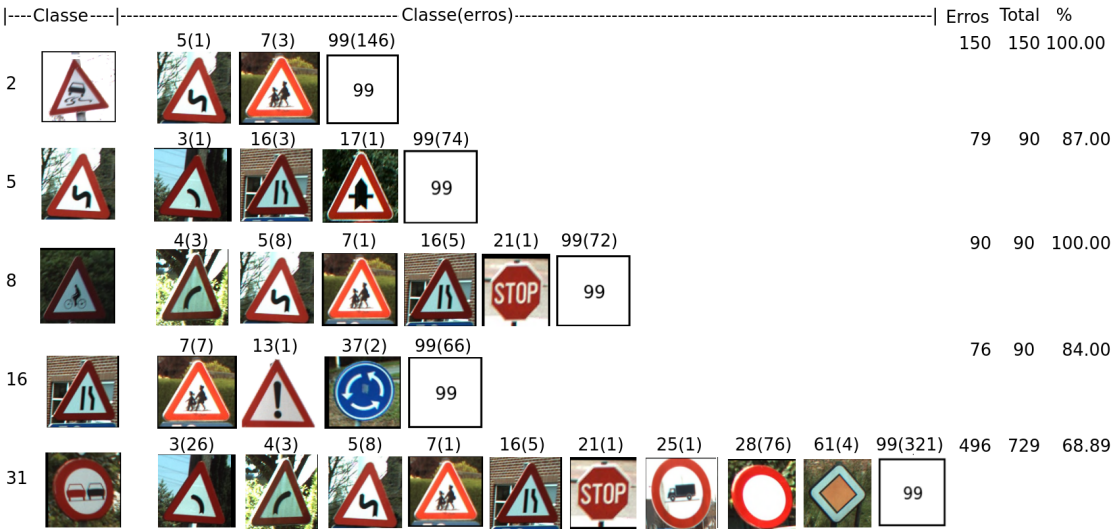


Figura 5.19: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Alemão

5.8.4 Belga x Croata

Similarmente, foi aplicado usar o dataset belga como base de treino e o croata como base de teste, buscando avaliar o impacto disso na performance do modelo, conforme expressa a Tabela 5.26. Além da equivalência, agruparam-se as classes que não tinham correspondência na classe “outros”, como mostra a Tabela 5.51.

Seguindo os procedimentos da Seção 5.8, obtiveram-se os resultados conforme a Tabela 5.52. A melhor acurácia foi 63.73% que foi obtida com *batch size* 16, *augmentation* de 20% e *validation* 30%, com o tempo de 57s. A Tabela 5.54 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.20 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.51: Todas as classes Belga x Croata
O valor “**” significa que não tem classe correspondente.

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
**	**	09	04	05	06	07	11	**	**	**	**	**	00	08	**	**	01	**

19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
14	**	15	**	**	**	**	**	**	**	**	**	22	**	**	**	**	**	**

38	39	40	41	42	43	44	45	46	47	48	49	56	61
24	25	**	23	**	**	**	29	**	**	**	**	28	16

Tabela 5.52: Acurácia Belga x Croata completa

batch size	16	24	32	48
sem drop	56.95	54.87	51.73	55.66
augm 20 + val 20	57.95	57.23	60.48	56.27
augm 20 + val 30	63.73	60.65	58.18	56.44
drop 0,2	60.20	58.23	59.36	55.43
drop 0,2 + augm 20	57.79	55.32	55.04	54.37

Tabela 5.53: Métrica Belga x Croata completa

Accuracy	63.73
Precision	53.66
Recall	51.45
F1 score	50.46

O tempo de treinamento desta melhor acurácia foi de 57s.

Tabela 5.54: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Croata.

class	2	3	4	5	6	7	13	14	17	21	38	45	56	61	99	Erro	Total	Perc
2	7	0	2	1	1	8	0	1	0	5	0	0	0	3	12	33	40	82.50
5	3	14	0	0	0	0	0	7	0	8	0	0	0	0	28	60	60	100.00
6	1	0	17	0	2	18	0	2	0	8	0	0	0	2	22	70	72	97.22
17	0	0	0	0	0	10	1	1	4	0	0	0	2	2	28	44	48	91.67
45	0	0	0	0	0	0	0	0	0	0	2	1	1	0	24	27	28	96.43

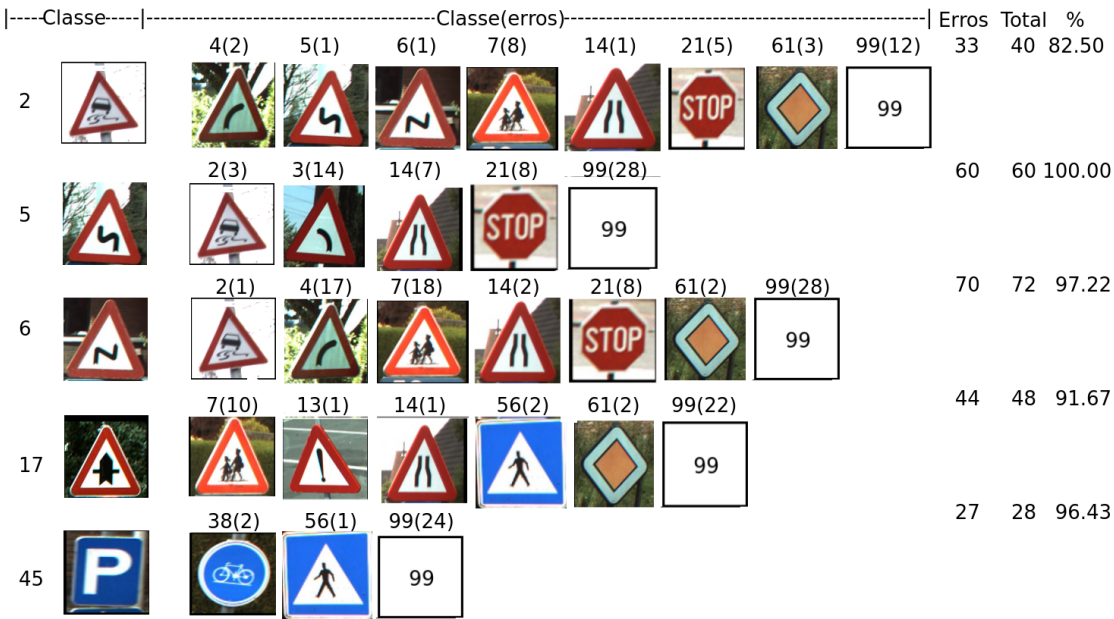


Figura 5.20: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Belga x Croata.

5.8.5 Croata x Alemão

Neste caso, foi usado o dataset croata como base de treino e o alemão, como teste. Criaram-se os relacionamentos entre classes conforme a Tabela 5.30, assim como agruparam-se aquelas que não tinham equivalências na classe “outros”, como mostra a Tabela 5.55.

Seguindo os procedimentos da Seção 5.8, obtiveram-se os resultados conforme a Tabela 5.56. A melhor acurácia foi 50.62% que foi obtida com *batch size* 16 sem *dropout* e com o tempo de 115s. A Tabela 5.58 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.21 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.55: Todas as classes Croata x Alemão

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
18	11	**	**	19	20	21	**	**	23	27	28	31	**	13	14	12	01	**

19	20	21	22	23	24	25	26	27	28	29	30
02	03	04	09	**	**	**	38	41	**	**	**

Criaram-se os relacionamentos entre classes conforme a Tabela 5.34 e agruparam-se aquelas que não tinham equivalências na classe “outros”, como mostra a Tabela 5.59.

Seguindo os procedimentos da Seção 5.8, obtiveram-se os resultados conforme a Tabela 5.61. A melhor acurácia foi 68.33% que foi obtida com *batch size* 32 com *dropout* de 0,2 e com o tempo de 53s. A Tabela 5.62 mostra a matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.22 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.59: Todas as classes Croata x Belga

O valor “**” significa que não tem classe correspondente.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
13	17	**	**	3	4	5	6	14	2	**	7	**	**	19	21	61	**	**

19	20	21	22	23	24	25	26	27	28	29	30
**	**	**	31	41	38	39	**	**	56	45	**

Tabela 5.60: Acurácia Croata x Belga completa

batch size	16	24	32	48
sem drop	60.99	62.34	57.85	68.33
augm 20 + val 20	64.84	67.10	65.83	67.77
augm 30	64.56	67.26	63.21	65.07
drop 0,2	65.63	65.27	61.46	63.65
drop 0,2 + aug 20	64.88	62.06	63.88	61.54

Tabela 5.61: Métricas Croata x Belga completa

Accuracy	68.33
Precision	51.17
Recall	64.72
F1 score	49.61

O tempo de treinamento desta melhor acurácia foi de 53s.

Tabela 5.62: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Croata x Belga.

Class	0	1	6	7	8	11	14	15	23	25	28	29	99	Erro	Total	Perc
0	6	0	0	0	3	2	0	0	0	0	0	0	28	33	39	84.62
1	0	75	0	4	0	0	0	0	0	0	0	0	104	108	183	59.02
11	0	0	1	29	0	44	0	0	0	0	2	0	14	46	90	51.11
23	0	0	0	0	0	1	2	1	1	0	0	3	3	10	11	90.91
29	0	0	0	0	0	35	0	14	0	13	0	17	5	67	84	79.76

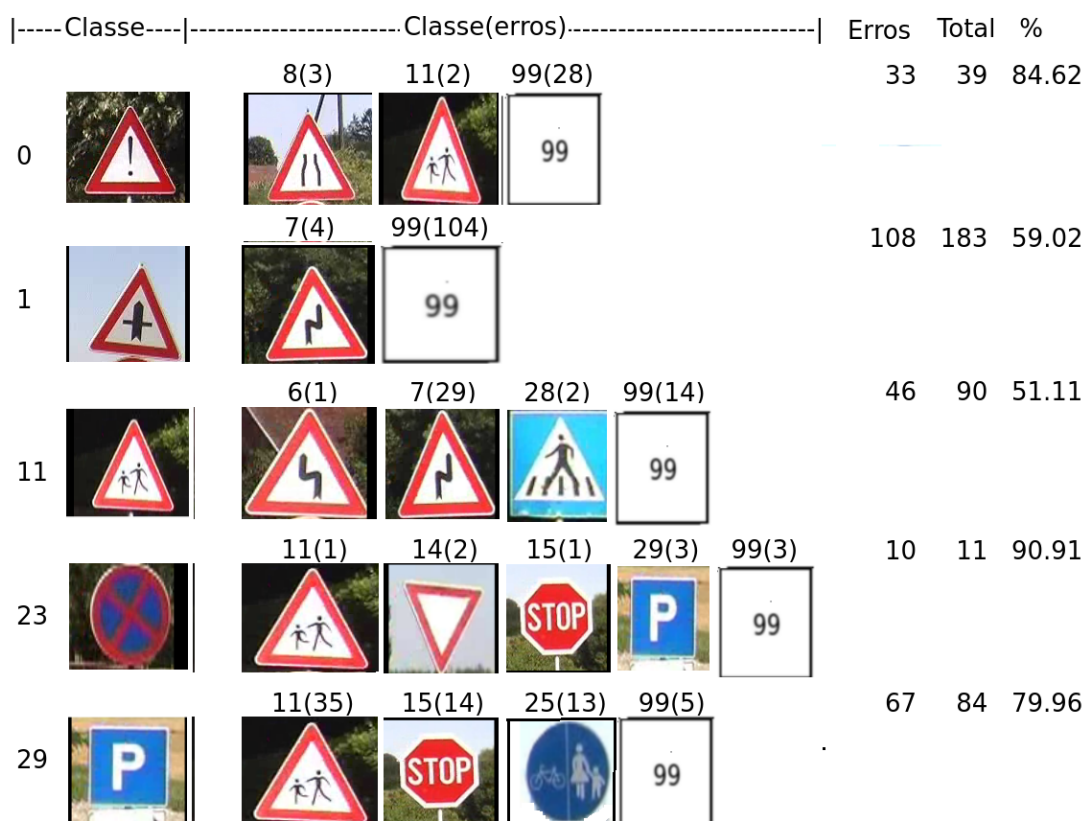


Figura 5.22: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia de Equivalência Completa - Croata x Belga.

5.8.7 Resumo

A Tabela 5.63 lista as melhores acurácias obtidas nos treinamentos dos datasets da combinação dos países em que foi criado uma classe "outras" para aquelas que não tinham correspondências.

Conclusão: Os resultados obtidos nesta estratégia teve o mesmo comportamento descrito na conclusão do experimento 2 em que foi feito o treinamento com datasets para classes equivalentes simples,

Tabela 5.63: Resumo da acurácia do treinamento dos datasets

Países	Acur	Prec	Rec	BS	Valores dos Hiperparâmetros
Alemão x Belga	91.07	70.71	86.21	32	srop de 0,2 + augm 20%
Alemão x Croata	78.81	80.61	84.65	24	augm 20% + valid 20%
Belga x Alemão	64.51	35.98	52.09	16	augm 20% + valid 20%
Belga x Croata	63.73	53.66	51.45	16	augm 20% + valid 30%
Croata x Alemão	50.62	40.27	61.99	16	sem dropout
Croata x Belga	68.33	51.17	64.72	48	sem dropout

BS = Batch Size

5.9 Experimento 4 - Treinamento do dataset único

Esta estratégia corresponde à quarta estratégia usada para responder à questão da pesquisa em que busca-se saber o impacto de diferentes estratégias de treinamento do modelo proposto em cima de diferentes conjuntos de dados de placas. Neste treinamento, foi criado um dataset único de treino com a união das classes equivalentes dos datasets alemão, belga e croata, mostrado na Figura 5.24. Inicialmente criou-se um dataset a partir da base alemã. Após isso, neste novo dataset, foram importadas as imagens das classes Belga e Croata que tinham correspondência com este dataset. Para as imagens que não tinham correspondência com este dataset foi criada uma classe especial. Por fim, o dataset final ficou com 88 classes.

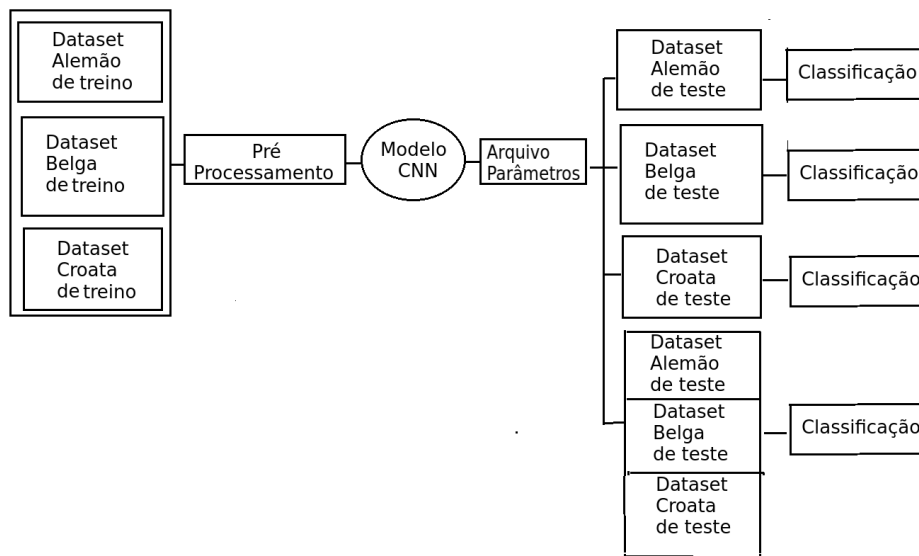


Figura 5.23: Modelo de treinamento único



Figura 5.24: Conjunto único de sinais de trânsito (Alemão, Belga e Croata).

5.9.1 Teste das bases Alemã, Belga, Croata

Neste treinamento foi usada a base única para as etapas de treino e teste, fazendo a devida separação. Os resultados estão mostrados na Tabela 5.64. O tempo de

treinamento da melhor acurácia foi de 666s. A melhor configuração foi usando *batch size* igual a 24 e *augmentation* de 20% em que obteve-se acurácia de 98.37%. A Tabela 5.66 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.25 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.64: Resumo da acurácia do treinamento do dataset único em uma base de teste única.

batch size	16	24	32	48
drop	93.10	96.58	96.97	96.01
augm 20 + val 20	95.60	98.37	98.22	98.26
augm 20 + val 30	97.07	97.91	97.56	97.67
drop 0,2	91.54	96.25	96.25	96.16
drop 0,2 + aug 20	95.05	98.13	98.20	98.32

Tabela 5.65: Métricas AlemaoxBelga x Croata

Accuracy	98.37
Precision	94.61
Recall	96.82
F1 score	95.20

Tabela 5.66: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão x Belga x Croata.

Class	5	6	11	13	15	18	20	21	23	25	26	27	30	31	32	35	38	39	42			57	59		64	80	83	84	Erros	Total	Perc
6	12	130	0	1	0	0	0	0	0	0	0	0	0	0	2	0	1	0	4	0	0	0	0	0	0	0	0	0	20	150	13.33
18	0	0	4	0	1	431	0	5	0	2	4	3	0	5	0	0	0	0	0	2	0	0	0	0	0	0	7	5	38	469	8.1
30	1	0	2	0	0	0	4	0	1	0	0	0	134	6	0	0	0	0	0	1	1	0	0	0	0	0	0	0	16	150	10.67
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	11	1	0	0	0	0	0	2	13	15.38
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	0	0	0	0	0	5	9	55.56	
64	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	17	18	0	0	0	0	0	0	112	1	0	0	42	154	27.27

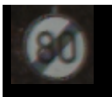
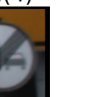
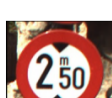
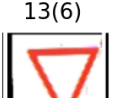
---Classe---	Classe(errores)										Erros	Total	%
6		5(12) 	13(1) 	32(2) 	38(1) 	42(4) 					20	150	13.33
30		5(1) 	11(2) 	20(4) 	23(1) 	31(6) 	ND(1)	ND(1)	16	150	10.67		
57		35(1) 	59(1) 						2	13	15.38		
59		57(5) 							5	9	55.56		
64		13(6) 	38(17) 	39(18) 	80(1) 				42	154	27.27		

Figura 5.25: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão x Belga x Croata

5.9.2 Teste da base Alemã

Neste treinamento foi usada a base única para a etapa de treino e a base alemã para a etapa de teste. Os resultados são mostrados na Tabela 5.67. O tempo de treinamento da melhor acurácia foi de 621s. A melhor configuração foi usando *batch size* igual a 48, *dropout* de 0.2 e *augmentation* de 20%, em que obteve-se acurácia de 98.59%. A Tabela 5.69 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.26 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.67: Resumo da acurácia do treinamento do dataset único

batch size	16	24	32	48
drop	93.27	96.19	95.60	95.35
augm 20 + val 20	97.62	98.40	98.24	98.13
augm 20 + val 30	97.74	98.25	97.80	97.68
drop 0,2	94.18	97.24	96.96	96.50
drop 0,2 + aug 20	96.56	98.55	98.56	98.59

Tabela 5.68: Métricas Alemã

Accuracy	98.59
Precision	80.98
Recall	81.26
F1 score	81.26

Tabela 5.69: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão.

C\I	5	6	8	13	18	20	21	22	26	27	30	31	33	34	36	37	39	40	42		49	84	84	Erros	Total	Perc
6	7	128	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	22	150	14,67
18	0	0	0	0	366	1	1	0	3	2	0	8	0	0	0	0	0	0	0	0	0	4	5	24	390	6,15
22	0	0	0	0	0	0	0	113	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	7	120	5,83
39	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	85	0	0	3	0	0	0	5	90	5,56
40	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	1	0	85	0	0	1	0	0	5	90	5,56

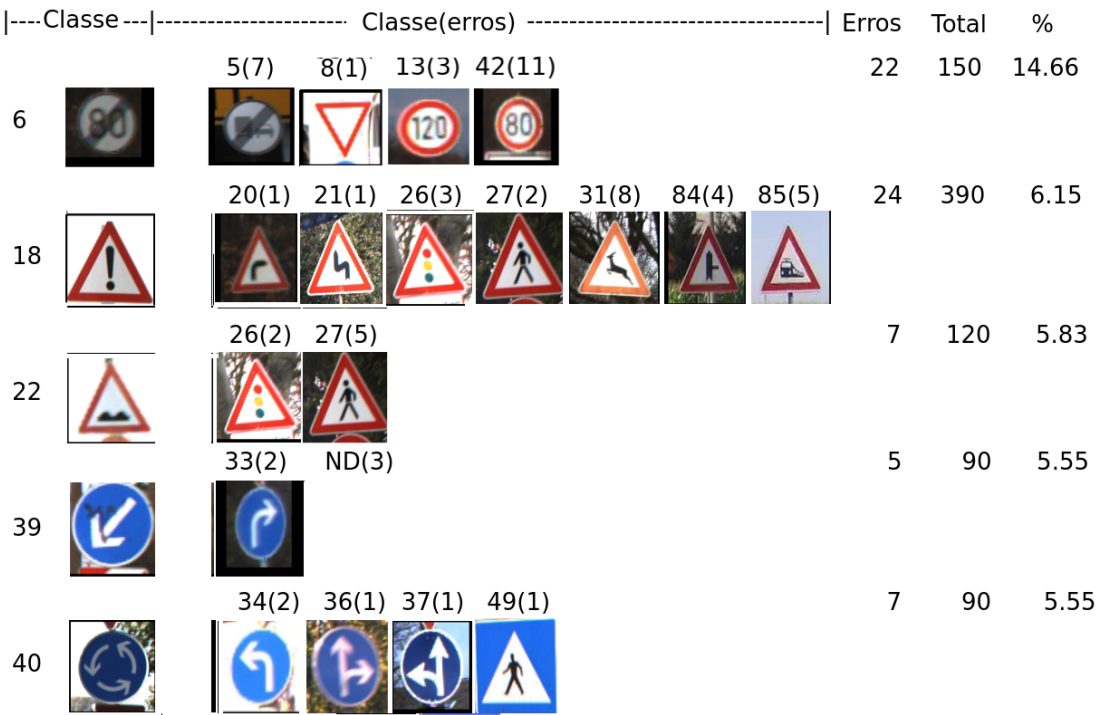


Figura 5.26: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Alemão.

5.9.3 Teste da base Belga

Analogamente, neste experimento, foi utilizada a base única para realizar o treinamento e a base belga, para o teste. Os resultados são mostrados na Tabela 5.70. O tempo de treinamento da melhor acurácia foi de 584s. A melhor configuração foi usando *batch size* igual a 24, *dropout* de 0.2 e *augmentation* de 20%, em que obteve-se acurácia de 97.18%. A Tabela 5.72 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.27 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.70: Resumo da acurácia do treinamento do dataset único em uma base de teste belga.

batch size	16	24	32	48
drop	90.59	92.38	91.86	93.21
augm 20 + val 20	96.19	97.14	96.94	96.54
augm 20 + val 30	96.62	97.06	96.50	96.74
drop 0,2	91.34	93.80	94.20	94.00
drop 0,2 + aug 20	91.94	97.18	96.74	97.02

Tabela 5.71: Métricas Belga

Accuracy	97.18
Precision	85.11
Recall	84.65
F1 score	84.45

Tabela 5.72: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Belga.

	12	13	15	25			48	57	59	64	69	82	Erros	Total	Perc
57	0	0	2	0	0	0	0	10	1	0	0	0	3	13	23.08
59	0	0	0	0	0	0	0	2	7	0	0	0	2	9	22.22
64	0	1	1	0	8	15	0	0	0	129	0	0	25	154	16.23
69	0	0	0	0	0	0	3	0	0	0	3	0	3	6	50.00
82	3	0	0	1	0	0	0	0	0	0	0	7	4	11	36.36

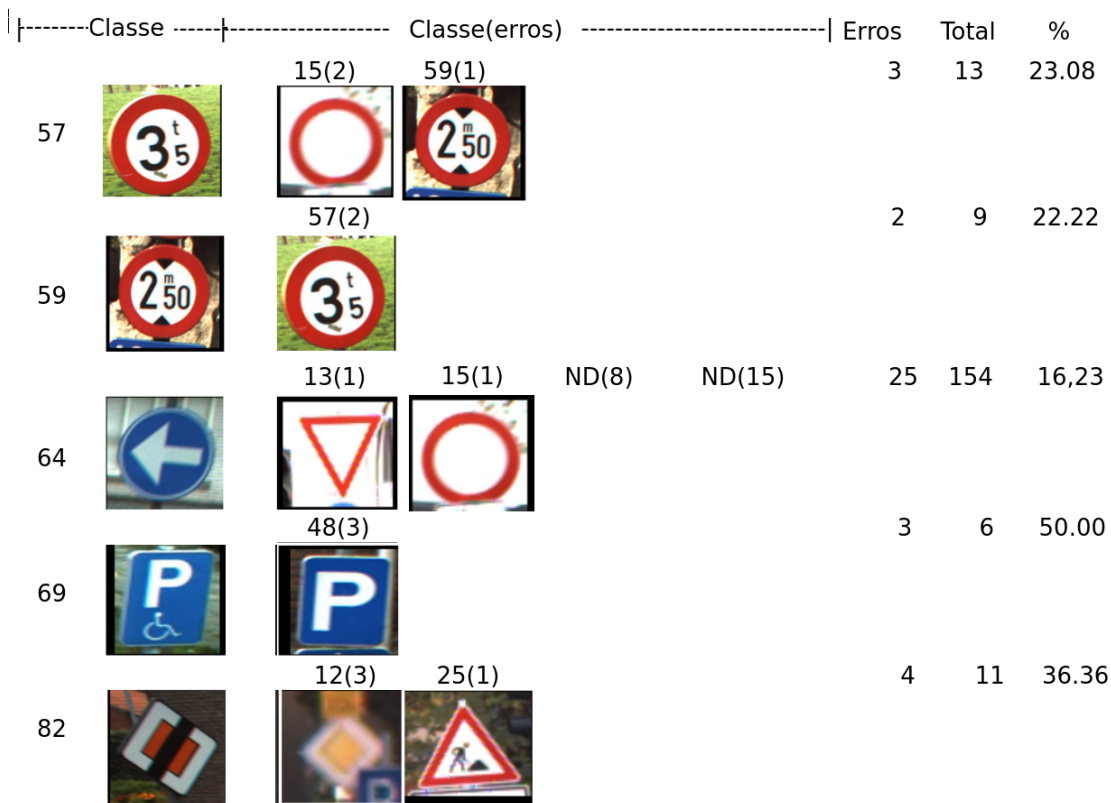


Figura 5.27: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Belga

5.9.4 Teste da base Croata

Por sua vez, neste experimento, foi feito a mesma coisa, mas trocando-se a base de teste para o dataset Croata. Os resultados são mostrados na Tabela 5.73. O tempo de treinamento da melhor acurácia foi de 486s. A melhor configuração foi usando *batch size* igual a 32 e *augmentation* de 20%, em que obteve-se acurácia de 98.82%. A Tabela 5.75 mostra a tabela da matriz de confusão resumida deste treinamento dos 5 maiores percentuais de erros e a Figura 5.28 mostra os sinais de trânsito correspondentes desta tabela.

Tabela 5.73: Resumo da acurácia do treinamento do dataset único em uma base de teste croata.

batch size	16	24	32	48
drop	91.98	95.01	95.06	95.12
augm 20 + val 20	96.74	98.43	98.82	98.31
augm 20 + val 30	96.18	98.65	98.31	98.43
drop 0,2	91.36	96.63	96.91	96.13
drop 0,2 + aug 20	93.77	98.15	98.76	98.76

Tabela 5.74: Métricas Croata

Accuracy	98.82
Precision	92.60
Recall	92.24
F1 score	92.36

Tabela 5.75: Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Croata.

Clas	1	2	3	4	11	13	21	23		31	43	48	70	79	85	Erros	Total	Perc
2	1	22	1	0	0	0	0	0	0	0	0	0	0	0	0	2	24	8.33
4	2	0	0	26	0	0	0	0	0	0	0	0	0	0	0	2	28	7.14
31	0	0	0	0	0	0	1	2	0	48	0	0	0	0	1	4	52	7.69
43	0	0	0	0	0	0	0	0	2	0	30	0	0	0	0	2	32	6.25
48	0	0	0	0	1	1	0	0	0	0	0	23	1	2	0	5	28	17.86

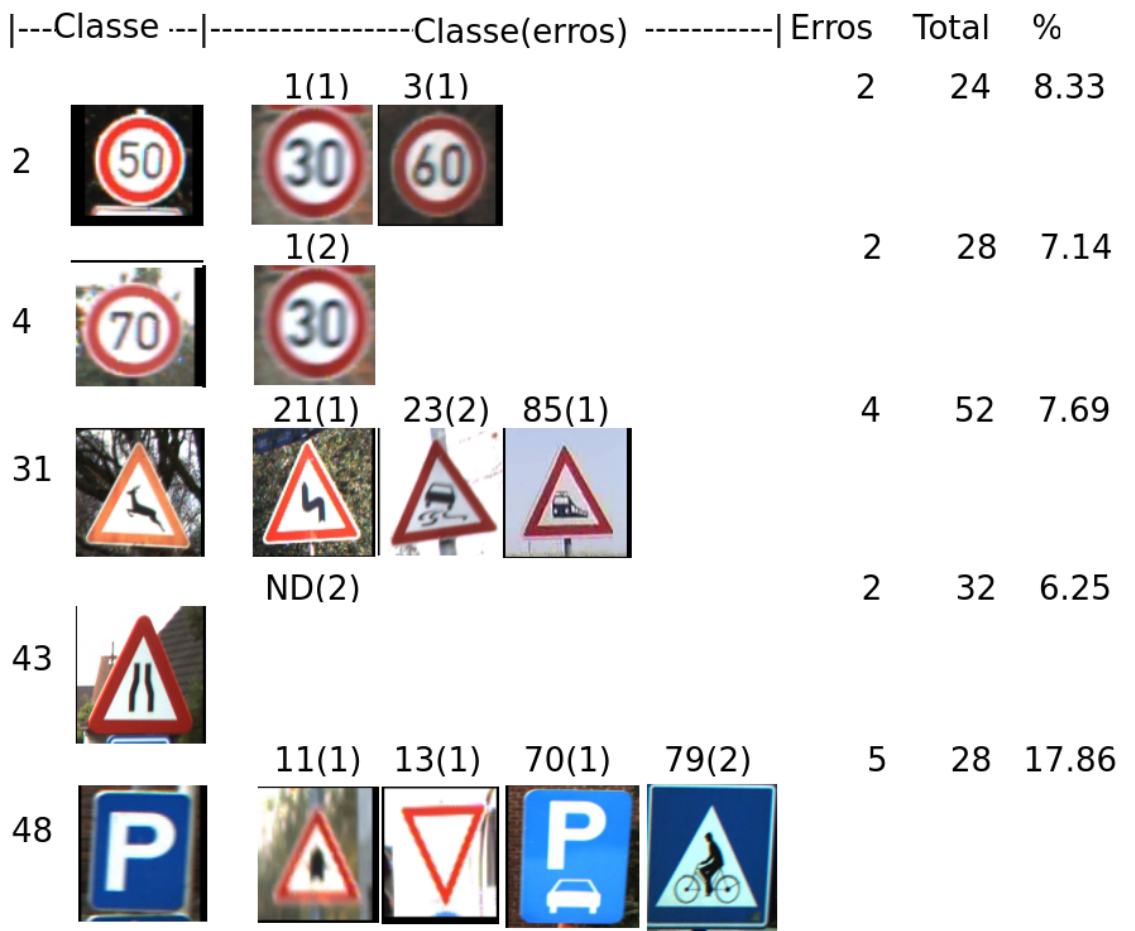


Figura 5.28: Placas de sinais da Matriz de Confusão dos 5 maiores percentuais de erro da Estratégia Única - Croata

5.10 **Análise e discussão de resultados**

Esta seção faz um resumo dos resultados obtidos pelas estratégias usadas, análise e discussão destes resultados.

Nos experimentos em que foi usada a mesma base individual para treino e teste, obtiveram-se resultados conforme a Tabela 5.76.

Tabela 5.76: Resumo da acurácia dos datasets.

Dataset	Acur	Prec	Rec	BS	Valores dos Hiperparâmetros
Alemão	98.89	98.49	98.74	24	drop 0,2 + Augm 20%
Belga	97.77	87.80	89.06	8	augm 20% + valid 20%
Croata	97.36	96.95	96.08	16	augm 20% + valid 30%

BS = Batch Size

A Tabela 5.77 apresenta os resultados obtidos com a classificação dos sinais de trânsito usando a combinação de datasets 2 a 2, utilizando somente as classes correspondentes e descartando as demais.

Tabela 5.77: Resumo da acurácia do treinamento combinado dos datasets para classes equivalentes.

Países	Acur	Prec	Rec	BS	Valores dos Hiperparâmetros
Alemão x Belga	97.58	91.63	89.68	16	drop 0,2+aug 20%
Alemão x Croata	91.98	91.48	90.60	24	drop 0,2+aug 20%
Belga x Alemão	65.00	56.45	55.20	32	augm 20% + valid 30%
Belga x Croata	61.99	66.20	55.63	24	augm 20% + valid 20%
Croata x Alemão	56.67	52.23	64.04	32	augm 20% + valid 20%
Croata x Belga	91.08	80.00	85.37	48	augm 20% + valid 30%

BS = Batch Size

A Tabela 5.78 mostra os resultados obtidos com a combinação 2 a 2 de datasets, de modo que as classes que não tem correspondentes são agrupadas em uma classe “outros” tanto na base de treino quanto na base de teste.

Tabela 5.78: Resumo da acurácia do treinamento combinado dos datasets para todas classes.

Países	Acur	Prec	Rec	BS	Valores dos Hiperparâmetros
Alemão x Belga	91.07	70.71	86.21	32	Drop de 0,2 + Augm 20%
Alemão x Croata	78.81	80.61	84.65	24	augm 20% + valid 20%
Belga x Alemão	64.51	35.98	52.09	16	augm 20% + valid 20%
Belga x Croata	63.73	53.66	51.45	16	augm 20% + valid 30%
Croata x Alemão	50.62	40.27	61.99	16	sem dropout
Croata x Belga	68.33	51.17	64.72	48	sem dropout

BS = Batch Size

O último experimento feito foi em cima de uma base única de treino e teste. Neste caso, foram agrupadas todas as classes correspondentes dos datasets. Para as classes que não tinham correspondentes foram criadas classes novas.

A Tabela 5.79 apresenta o resultado obtido do treinamento do dataset único, aplicando-se o modelo para a base de testes de cada país. Tais resultados mostram que obtém-se melhor acurácia quando se tem uma base de treino rica e diversa em imagens com boa qualidade e variações, como foi apontado por [41], visto que isto permite cobrir uma maior quantidade de características e permite ao modelo aprender a extraí-las.

Tabela 5.79: Resumo dos testes com dataset único

treino	teste	acur	prec	rec	treino	BS	Valores dos Hiperparâmetros
AlBeCr	AlBeCr	98.37	94.61	96.82	666s	24	drop 0.2+aug20%
AlBeCr	Alemão	98.59	80.98	81.26	621s	32	drop 0.2+aug20%
AlBeCr	Belga	97.18	85.11	84.65	584s	32	drop 0.2+aug20%
AlBeCr	Croata	98.82	92.60	92.24	486s	48	drop 0.2+aug20%

AlBeCr=AlemaoBelgaCroata BS=Batch Size

Fica claro, pois, que a classificação de sinais de um único país obteve bons resultados, conforme a Tabela 5.76, e na comparação das Tabelas 5.77, 5.78 e 5.79, quando se analisa a classificação de mais de um país ao mesmo tempo, vemos que os resultados exibidos pela Tabela 5.79 obteve as melhores acurácias. Isso indica que a estratégia de se adotar um dataset único contendo imagens de sinais de trânsito dos países de uma determinada comunidade é a melhor opção, respondendo à pergunta de pesquisa levantada.

Fica claro, também, de acordo com as matrizes de confusão dos 5 maiores percentuais de erro das estratégias apresentadas, que por maior que seja os valores das métricas obtidas, se houver um erro, pode causar um acidente em um veículo, dependendo da placa de sinal.

Capítulo 6

Conclusão

Ao final deste trabalho, após as estratégias de treinamento do modelo para classificação de sinais de trânsito, concluiu-se que em um treinamento individual de um dataset utilizando um mesmo pré-processamento e um mesmo modelo, existem fatores que contribuem para uma melhor acurácia, como foi visto no dataset alemão que apresentou 98,89%, superior ao belga com 97,77% e croata com 97,36%. O que contribui para isto foi um maior número de imagens representativas por classes tanto na base de treinamento quanto na de teste, que permitiu identificar mais características e melhorar o modelo.

Devido aos três datasets já terem bases de treinamento e teste consolidadas, verificou-se que os valores das métricas foram altos. Também verificou-se que, na comparação com outros artigos que fizeram treinamentos usando redes CNN com estes datasets, os valores das métricas obtidas por eles, ficaram muito próximos independente de pré-processamento, modelo e hiperparâmetro. No pré-processamento, alguns utilizaram escala cinza, normalização, contraste, redimensionamento de formas diferentes. Em outros modelos variaram os hiperparâmetros.

Quando foram feitos treinamentos em que combinou-se os datasets, objetivou-se saber se um modelo que foi treinado com o dataset de um país e testado este mesmo modelo com o dataset de outro país, qual seriam as métricas obtidas, uma vez que os datasets possuem equivalências de sinais? Os experimentos realizados mostraram que os melhores resultados foram obtidos quando se treinava o modelo com dataset alemão e testava com os outros datasets. Os piores resultados foram observados quando o dataset alemão foi usado como dataset de teste. Isto aconteceu pois o treinamento do modelo com uma base de dados mais representativa permite classificar melhor uma outra base de teste, e o contrário também acontecia.

No treinamento combinado, quando se treinou o modelo com o dataset alemão e testou no dataset belga e croata, estes 2 últimos obtiveram as melhores acurácias, respectivamente 97,78% e 96,50%. Quando treinou-se o modelo usando os datasets belga e testou no dataset alemão, obteve-se a pior acurácia para este caso. Quando treinou-se o modelo com o dataset alemão e treinou-se com o dataset croata, obteve-se a pior acurácia para este caso.

Quando se criou um dataset único a partir da união dos datasets, treinou-se o modelo, e testou-se em cada dataset. Foi obtida uma alta acurácia para os três datasets: Alemão - 98,40% Belga - 97,30%, Croata - 98,76%. Estes resultados

ficaram próximos aos dos artigos comparados.

Por fim, verificou-se que na estratégia com dataset único, os resultados foram tão bons quanto nos treinamentos individuais, com a vantagem de poder usar em mais de um país, e foram melhores que os treinamentos com datasets combinados. Por outro lado, o tempo de treinamento foi maior que os outros devido ao tamanho do dataset.

Concluimos que o melhor modelo foi o treinamento com dataset único, correspondente à união de todos os datasets, que teve bons resultados em todas as combinações.

Verificou-se também que o *augmentation* permitiu aumento dos valores das métricas na maioria dos casos e que a CNN esteve presente em todos os trabalhos relacionados.

Após apresentar as estratégias de modelos de classificação que permite um VA trafegar entre países, é mostrado na Tabela 6.1, uma referência cruzada de países (origem x destino) e as melhores estratégias destes pares.

Tabela 6.1: Estratégias por países.

	Alemanha	Bélgica	Croácia
Alemanha	I,U	E,C,U	E,U
Bélgica	U	I,U	U
Croácia	U	E,U	I,U

- I - Treinamento Individual, E - Treinamento com equivalência de classes
- C - Treinamento com equivalência completa, U - Treinamento com dataset único

Não existe ainda um modelo que consiga, para todas as imagens de trânsito de diversos países, métricas de acurácia, *precision* e *recall* próximas de 100%.

Contudo, verificou-se que não existe uma modelo de classificação de sinais de trânsito que consiga melhor acurácia para todas as imagens do trânsito que apresentam problemas de oclusão, conservação, nitidez. Alguns problemas são encontrados:

- Modelos que dão ênfase na acurácia, criando arquiteturas com muitas camadas e algoritmos sofisticados, apresentam um tempo de classificação mais elevado.
- Modelos com arquitetura mais simples, com tempo de resposta rápido, apresentam acurácia inferior.
- Modelos que utilizam mais de um dataset enfrentam o problema de imagens semelhantes com semânticas diferentes e imagens diferentes com a mesma semântica.

Na direção autônoma precisa-se de classificação de sinais com valores de métricas altas e baixos tempos de respostas. Como trabalhos futuros, consideramos uma investigação aprofundada do trade-off entre estas duas métricas, com o objetivo de identificar um ponto de equilíbrio ideal. Além disso, outros desafios podem ser citados como: criar um conjunto de dados de treinamento que possa

atender um número máximo de países na classificação de modelos de sinais de trânsito e o estudo de técnicas que possibilitem reconhecer os sinais de trânsito a maiores distâncias.

Outro trabalho futuro, seria o uso de Aprendizado por Transferência, que vem sendo explorado em diversas aplicações de IA e ITS. Este algoritmo permite usar o conhecimento adquirido no aprendizado de novas tarefas, dessa forma acumula conhecimento ao longo do tempo e aprende com novas informações. Quando se implementa este algoritmo se utiliza uma CNN já treinada que permite extrair características de novas imagens, gerando um classificador que possibilita reconhecer novos padrões e adicionar conhecimento ao modelo. Isto se aplica ao presente trabalho em que se treinando um modelo com uma base de dados de sinais de trânsito de um país e circulando em um outro país, este modelo teria uma base de dados treinada e iria aprender com novas imagens de sinais de trânsito de outro país.

Ainda não foi apresentado um trabalho que use este aprendizado que obtenha o estado da arte na classificação de sinais de trânsito. O artigo [69] usou este aprendizado na detecção de sinais de trânsito, usando uma R-CNN, o dataset CIFAR-10 como treinamento e o dataset GTSDb como teste. O artigo [46] apresentou um método de reconhecimento e classificação de sinais de trânsito com este aprendizado, usando o Belgium Traffic Sign Dataset(BTSD) mas, não foi usado um outro dataset que seja treinado e incorpore o aprendizado do primeiro dataset treinado.

Referências Bibliográficas

- [1] Adaptação de domínio. https://en.wikipedia.org/wiki/Domain_adaptation. Accessed: 29/03/2020.
- [2] Aprendizado por transferência. <https://www.mathworks.com/help/deeplearning/deeplearning-with-images.html>. Accessed: 29/03/2020.
- [3] Arquitetura de uma rede neural convolucional. <https://developer.nvidia.com/discover/convolutional-neural-network>. Accessed: 29/03/2020.
- [4] Dataset alemão de sinais de trânsito. <http://benchmark.ini.rub.de/?section=gtsrbsubsection=dataset>. Accessed: 29/03/2020.
- [5] Dataset belga de sinais de trânsito. <https://btsd.ethz.ch/shareddata/>. Accessed: 29/03/2020.
- [6] Dataset croata de sinais de trânsito. <http://www.zemris.fer.hr/kalfa/Datasets/rMASTIF/>. Accessed: 29/03/2020.
- [7] Deep neural network. <https://www.techopedia.com/definition/32902/deep-neural-network>. Accessed: 29/03/2020.
- [8] Função de ativação. <http://deeplearningbook.com.br/funcao-de-ativacao/>. Accessed: 29/03/2020.
- [9] Funções de ativação. <https://matheusfacure.github.io/2017/07/12/activ-func/>. Accessed: 29/03/2020.
- [10] Github do advis.js. <https://github.com/jaxball/advis.js>. Accessed: 29/03/2020.
- [11] Inteligência artificial. <http://www-formal.stanford.edu/jmc/whatisai.pdf>. Accessed: 12/11/2007.
- [12] Keras documentation. <https://keras.io/>. Accessed: 29/03/2020.
- [13] Matplotlib documentation. <https://matplotlib.org/>. Accessed: 13/05/2020.
- [14] Numpy documentation. <https://numpy.org/>. Accessed: 13/05/2020.
- [15] Otimizador keras. <https://keras.io/optimizers/>. Accessed: 29/03/2020.
- [16] Scikit-image documentation. <https://scikit-image.org/>. Accessed: 13/05/2020.

- [17] Scikit-learn documentation. <https://scikit-learn.org/stable/>. Accessed: 13/05/2020.
- [18] Subáreas do aprendizado de máquina. <https://www.datacamp.com/community/tutorials/machine-deep-learning>. Accessed: 29/03/2020.
- [19] Video de ataque adversário sobre sinais de trânsito. http://kaichen.org/datas/CCS2019_HA_IE.mp4. Accessed : 29/03/2020.
- [20] Vídeo do advis.js. https://www.youtube.com/watch?v=JjIV62_kGUc. Accessed: 29/03/2020.
- [21] AF Agarap. An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification. arxiv 2017. *arXiv preprint arXiv:1712.03541*.
- [22] Hamed Habibi Aghdam, Elnaz Jahani Heravi, and Domenec Puig. A practical and highly optimized convolutional neural network for classifying traffic signs in real-time. *International Journal of Computer Vision*, 122(2):246–269, 2017.
- [23] Amparo Albalade and Wolfgang Minker. *Semi-supervised and unsupervised machine learning: novel strategies*. John Wiley & Sons, 2013.
- [24] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.
- [25] Yusuf Abdullahi Badamasi. The working principle of an arduino. In *2014 11th international conference on electronics, computer and computation (ICECCO)*, pages 1–4. IEEE, 2014.
- [26] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press, 2017.
- [27] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep learning techniques for music generation - A survey. *CoRR*, abs/1709.01620, 2017.
- [28] Jingwei Cao, Chuanxue Song, Silun Peng, Feng Xiao, and Shixin Song. Improved traffic sign detection and recognition algorithm for intelligent vehicles. *Sensors*, 19(18):4021, 2019.
- [29] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. A committee of neural networks for traffic sign classification. In *The 2011 international joint conference on neural networks*, pages 1918–1921. IEEE, 2011.
- [30] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural networks*, 32:333–338, 2012.
- [31] Scott A Cohen and Debbie Hopkins. Autonomous vehicles and the future of urban tourism. *Annals of tourism research*, 74:33–42, 2019.

- [32] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.
- [33] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, and Yubin Kuang. Traffic sign detection and classification around the world. *arXiv preprint arXiv:1909.04422*, 2019.
- [34] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.
- [35] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.
- [36] Junqi Jin, Kun Fu, and Changshui Zhang. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):1991–2000, 2014.
- [37] Fran Jurišić, Ivan Filković, and Zoran Kalafatić. Multiple-dataset traffic sign classification with onecnn. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 614–618. IEEE, 2015.
- [38] Ki-Jo Kim and Ilias Tagkopoulos. Application of machine learning in rheumatic disease research. *The Korean journal of internal medicine*, 34(4):708, 2019.
- [39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [40] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.
- [41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [42] Steve Lawrence and C Lee Giles. Overfitting and neural networks: conjugate gradient and backpropagation. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 1, pages 114–119. IEEE, 2000.
- [43] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [44] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, USA, 3rd edition, 2020.
- [45] Dong Li, Dongbin Zhao, Yaran Chen, and Qichao Zhang. Deepsign: Deep learning based traffic sign recognition. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–6. IEEE, 2018.

- [46] Chunmian Lin, Lin Li, Wenting Luo, Kelvin CP Wang, and Jiangang Guo. Transfer learning based traffic sign recognition using inception-v3 model. *Periodica Polytechnica Transportation Engineering*, 47(3):242–250, 2019.
- [47] Jason Lin and Dilara Soylu. AdVis: Visualizing and Attributing ML Attacks to Adversarial Examples. Technical report, Georgia Institute of Technology, 2018. Accessed: [Insert date here].
- [48] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition—how far are we from the solution? In *The 2013 international joint conference on Neural networks (IJCNN)*, pages 1–8. IEEE, 2013.
- [49] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [50] Tom M Mitchel. *machine learning*. McGraw-Hill, 1997.
- [51] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *arXiv preprint arXiv:1907.00374*, 2019.
- [52] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [53] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.
- [54] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. *arXiv preprint arXiv:1905.12762*, 2019.
- [55] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.
- [56] Stuart Jonathan Russell and Peter Norvig. *A Modern Approach*. Prentice Hall, 2010.
- [57] Sourajit Saha, Md Saiful Islam, Md Asif Bin Khaled, and Suraiya Tairin. An efficient traffic sign recognition approach using a novel deep neural network selection architecture. In *Emerging Technologies in Data Mining and Information Security*, pages 849–862. Springer, 2019.
- [58] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.
- [59] Citlalli Gámez Serna and Yassine Ruichek. Classification of traffic signs: The european dataset. *IEEE Access*, 6:78136–78148, 2018.
- [60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- [61] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [62] Vaibhav Swaminathan, Shrey Arora, Ravi Bansal, and R Rajalakshmi. Autonomous driving system with road sign recognition using convolutional neural networks. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–4. IEEE, 2019.
- [63] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [64] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.
- [65] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.
- [66] Hai Wang, Yijie Yu, Yingfeng Cai, Long Chen, and Xiaobo Chen. A vehicle recognition algorithm based on deep transfer learning with a multiple feature subspace distribution. *Sensors*, 18(12):4109, 2018.
- [67] Jason Wang and Luis Perez. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, page 11, 2017.
- [68] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5345–5352, 2019.
- [69] Liu Wei, Lu Runge, and Liu Xiaolei. Traffic sign detection and recognition via transfer learning. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 5884–5887. IEEE, 2018.
- [70] Sholom M Weiss and Casimir A Kulikowski. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann Publishers Inc., 1991.
- [71] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.
- [72] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.
- [73] Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu. Towards real-time traffic sign detection and classification. *IEEE Transactions on Intelligent transportation systems*, 17(7):2022–2031, 2015.

- [74] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.
- [75] Mingming Zha, Guozhu Meng, Chaoyang Lin, Zhe Zhou, and Kai Chen. Rolma: A practical adversarial attack against deep learning-based lpr systems. In *International Conference on Information Security and Cryptology*, pages 101–117. Springer, 2019.
- [76] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn’t believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1989–2004, 2019.
- [77] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. Invisible mask: Practical attacks on face recognition with infrared. *arXiv preprint arXiv:1803.04683*, 2018.

Capítulo 7

Apêndice

7.1 Matrizes Confusão da Estratégia Individual

Tabela 7.1: Matriz Confusão. Estratégia Individual - Dataset Alemão (1)

Classe	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	720	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	749	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	439	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	8	0	0	650	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	2	1	0	619	0	6	2	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	6	120	0	4	0	0	0	0	3	0	0	0	0	0	0	0
7	0	1	0	1	0	1	0	445	2	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	3	0	4	0	2	441	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	658	0	0	1	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	674	0	0	16	0	0	0	0	0	0
13	0	0	0	0	0	0	1	0	0	0	0	0	0	718	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	268	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	210	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	356	0	0	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	364	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	9
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	2	0	0	1	0	0	1	0	0	0	0	0	0	0	0	7
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	9
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 7.2: Matriz Confusão. Estratégia Individual - Dataset Alemão (2)

Classe	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	Erros	Total	Perc
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	720	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	750	0.13
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	450	2.44
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	10	660	1.52
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	630	1.75
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	30	150	20
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	450	1.11
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	450	2
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	660	0.3
11	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	20	420	4.76
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	690	2.32
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	720	0.28
14	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	270	0.74
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	210	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	0	4	360	1.11
18	3	6	0	0	1	6	0	1	0	0	7	0	0	0	0	0	0	0	0	0	0	0	26	390	6.67
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0
21	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0
22	0	116	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	120	3.33
23	0	0	148	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	150	1.33
24	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0
25	0	0	0	0	468	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	480	2.5
26	0	0	0	0	0	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0
27	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	60	1.67
28	0	0	0	0	0	0	0	147	3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	150	2
29	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0
30	0	0	1	0	1	0	0	0	1	134	2	0	0	0	0	0	0	0	0	0	0	0	16	150	10.67
31	0	0	0	0	0	0	0	0	0	0	260	0	0	0	0	0	0	0	0	0	0	0	10	270	3.7
32	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	60	0
33	0	0	0	0	0	0	0	0	0	0	0	1	209	0	0	0	0	0	0	0	0	0	1	210	0.48
34	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	120	0
35	0	0	0	0	0	0	0	0	0	0	0	0	1	0	389	0	0	0	0	0	0	0	1	390	0.26
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	120	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	60	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	681	0	0	0	0	9	690	1.3
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	90	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	2	0	0	0	84	0	0	6	90	6.67
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58	1	2	60	3.33
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	90	0

[illegible]

[illegible]

Tabela 7.5: Matriz Confusão. Estratégia Individual - Dataset Croata

Class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Erros	Total	Perc
0	33	0	2	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	40	17.50	
1	0	45	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	48	6.25	
2	2	1	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	56	5.36	
3	0	0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0.00	
4	0	0	0	0	61	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	64	4.69	
5	0	0	0	0	0	71	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	72	1.39	
6	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0.00	
7	0	0	0	0	0	0	0	70	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	72	2.78	
8	0	0	0	0	0	1	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	32	3.13	
9	0	0	0	0	0	0	0	0	3	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	40	7.50	
10	1	0	1	0	0	0	0	0	0	0	78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	80	2.50	
11	0	0	0	0	0	1	0	0	3	0	0	84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	88	4.55	
12	0	0	0	0	0	0	2	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	52	3.85	
13	0	0	0	0	0	0	2	0	0	2	0	1	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	16	31.25	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0.00	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	0.00	
16	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	118	0	0	0	0	0	0	0	0	0	0	0	0	0	2	120	1.67	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0.00	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	55	0	0	0	0	0	0	0	0	0	0	0	0	1	56	1.79	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	2	0	0	0	0	0	0	0	0	0	2	24	8.33	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	40	0.00	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	28	0.00	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	95	0	0	0	0	0	0	0	0	1	96	1.04	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	28	0.00
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	40	0.00	
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	16	0.00	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	18	0	0	0	0	0	2	20	10.00	
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	0	0	0	0	56	0.00	
28	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	0	1	164	0.61	
29	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	26	0	2	28	7.14	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	72	0	72	0.00	

7.2 Matrizes Confusão. Equivalência Simples

Tabela 7.6: Matriz Confusão. Equivalência Simples - Alemão x Belga

Classe	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	28	29	35	40	Erros	Total	Perc
9	84	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	86	2.33
11	0	178	0	0	0	0	0	0	1	0	0	0	0	0	1	0	3	0	0	0	0	5	183	2.73
12	0	0	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	105	0.00
13	0	0	0	163	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	0.00
14	0	0	0	1	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	45	4.44
15	1	0	0	0	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	51	3.92
16	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0.00
17	0	0	0	0	0	0	0	61	0	0	0	0	0	0	0	0	0	0	0	0	0	61	0.00	
18	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	39	0.00	
19	0	0	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0	0	0	0	0	1	6	16.67
20	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	12	0.00
21	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	3	0.00	
22	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	6	0.00	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	7	0.00	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	12	0.00	
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	28	0.00	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	90	0.00	
29	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	6	2	0	0	10	12	83.33
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	9	0.00	
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	30	1	31	3.23

Tabela 7.7: Matriz Confusão. Equivalência Simples - Alemão x Croata

Classe	1	2	3	4	9	11	12	13	14	18	19	20	21	23	27	28	31	38	41	Erro	Percen	Total
1	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	44
2	0	20	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	4	16.67	24
3	0	21	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	52.50	40
4	1	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3.57	28
9	0	0	0	0	92	0	0	0	4	0	0	0	0	0	0	0	0	0	0	4	4.17	96
11	0	0	0	0	0	47	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2.08	48
12	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	120
13	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0.00	40
14	0	0	0	0	0	0	1	0	135	0	0	0	0	0	0	0	0	0	0	1	0.74	136
18	0	0	0	0	0	0	1	0	0	33	0	3	0	0	3	0	0	0	0	7	17.50	40
19	0	0	0	0	0	0	0	0	0	0	56	8	0	0	0	0	0	0	0	8	12.50	64
20	0	0	0	0	0	0	0	0	0	0	0	70	0	0	0	1	0	0	1	2	2.78	72
21	0	0	0	0	0	0	0	0	0	0	1	3	54	0	1	1	0	0	0	6	10.00	60
23	0	0	0	0	0	0	0	0	0	0	2	0	0	35	0	3	0	0	0	5	12.50	40
27	1	0	0	0	0	2	0	0	0	12	0	0	0	1	60	2	2	0	0	20	25.00	80
28	0	0	0	0	0	0	0	0	0	0	4	0	1	0	83	0	0	0	0	5	5.68	88
31	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	50	0	0	2	3.85	52
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0.00	20
41	0	1	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	51	5	8.93	56

Tabela 7.8: Matriz Confusão. Equivalência Simples - Belga x Alemão

Classe	0	2	3	4	5	7	8	10	11	13	16	17	19	21	22	25	28	31	34	36	37	61	Erro	Total	Perc
0	110	0	1	2	0	0	0	1	0	4	0	0	1	0	0	0	0	0	1	0	0	0	10	120	8.33
2	0	30	1	9	35	31	0	0	0	4	0	0	0	0	0	0	0	0	28	0	12	0	120	150	80.00
3	1	7	17	2	20	0	0	0	0	4	0	0	0	0	0	0	0	0	9	0	0	0	43	60	71.67
4	0	40	0	8	2	27	0	0	0	4	0	2	0	0	0	0	0	0	1	0	6	0	82	90	91.11
5	1	16	19	3	44	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	46	90	51.11
7	2	3	10	20	3	100	10	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	50	150	33.33
8	10	5	29	10	15	12	0	1	0	0	6	0	0	0	2	0	0	0	0	0	0	0	90	90	100.00
10	2	8	23	8	6	42	7	263	0	25	4	8	7	0	3	1	0	1	29	18	8	17	217	480	45.21
11	0	11	2	1	0	2	0	0	34	111	0	8	0	0	11	0	0	0	0	0	0	0	146	180	81.11
13	3	6	29	4	0	46	21	0	4	222	0	19	0	0	0	0	0	0	1	3	30	2	168	390	43.08
16	1	11	26	2	2	4	4	2	1	8	13	1	0	0	2	0	0	0	6	1	6	0	77	90	85.56
17	0	2	8	4	4	97	0	1	0	2	0	228	4	3	5	0	0	2	0	1	30	29	192	420	45.71
19	0	0	0	0	0	0	0	0	0	0	0	0	683	2	1	0	0	0	30	3	0	1	37	720	5.14
21	0	0	0	0	0	1	0	0	0	0	0	0	0	0	259	0	7	0	0	0	0	3	11	270	4.07
22	0	0	0	0	0	0	0	0	0	0	0	0	0	4	341	0	0	0	0	0	0	15	19	360	5.28
25	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	97	0	12	0	0	38	1	53	150	35.33
28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	185	0	24	0	0	0	25	210	11.90
31	11	12	46	15	38	12	0	2	0	4	6	0	14	0	32	18	58	273	166	6	7	0	447	720	62.08
34	0	0	0	5	0	0	0	5	0	0	0	0	3	0	1	0	58	0	275	28	7	8	115	390	29.49
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	16	0	19	68	3	0	52	120	43.33
37	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	9	0	2	9	52	16	38	90	42.22
61	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	30	0	14	0	25	618	72	690	10.43

Tabela 7.9: Matriz Confusão. Equivalência Simples - Belga x Croata

Classe	2	3	4	5	6	7	13	14	17	19	21	31	38	39	41	45	56	61	Erros	Total	Perc
2	2	0	6	1	3	7	2	4	0	0	3	0	0	0	0	0	0	12	38	40	95.00
3	2	25	0	1	0	1	8	3	0	2	4	0	0	0	0	0	6	12	39	64	60.94
4	0	0	36	0	1	7	2	4	4	2	9	0	0	0	0	0	0	7	36	72	50.00
5	3	9	4	2	0	2	13	9	0	0	8	1	1	0	0	0	1	7	58	60	96.67
6	1	1	17	0	8	16	1	2	0	1	8	1	0	0	0	0	1	15	64	72	88.89
7	0	5	33	1	14	24	2	1	1	1	4	1	0	0	0	1	0	0	64	88	72.73
13	0	2	0	0	0	3	27	2	0	0	4	0	0	0	0	0	0	2	13	40	32.50
14	1	0	8	1	1	3	1	8	0	0	4	0	0	0	0	0	0	5	24	32	75.00
17	0	0	1	0	0	4	13	4	4	2	0	1	0	0	0	0	2	17	44	48	91.67
19	0	0	0	0	0	0	0	0	0	37	0	0	1	1	0	0	0	1	3	40	7.50
21	0	0	0	0	0	0	0	0	0	0	105	0	4	0	0	0	0	27	31	136	22.79
31	0	0	0	0	3	0	0	0	0	7	0	70	8	0	0	0	0	8	26	96	27.08
38	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	40	0.00
39	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	16	0.00
41	0	0	0	0	0	0	0	0	0	0	0	0	19	0	8	0	0	1	20	28	71.43
45	0	0	0	0	0	0	0	0	0	0	0	0	14	3	0	8	3	0	20	28	71.43
56	0	0	0	0	0	0	0	0	0	0	0	0	15	1	0	0	148	0	16	164	9.76
61	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	117	3	120	2.50

Tabela 7.10: Matriz Confusão. Equivalência Simples - Croata x Alemão

Classe	0	1	4	5	6	9	10	11	12	14	15	16	17	19	20	21	22	23	26	27	Erro	Total	Perc
0	302	45	0	2	14	3	7	0	1	0	0	3	0	0	0	0	2	0	0	11	88	390	22.56
1	0	374	3	5	12	2	1	3	0	1	0	5	0	0	0	0	0	0	2	12	46	420	10.95
4	0	0	48	0	2	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	60	20.00
5	1	0	0	67	1	5	7	9	0	0	0	0	0	0	0	0	0	0	0	0	23	90	25.56
6	0	0	1	0	78	1	0	8	1	0	0	0	0	0	0	0	0	0	0	1	12	90	13.33
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
10	15	13	0	2	0	0	26	0	4	0	0	0	0	0	0	0	0	0	0	0	34	60	56.67
11	0	0	2	0	1	5	0	141	0	0	0	1	0	0	0	0	0	0	0	0	9	150	6.00
12	17	0	5	4	39	13	0	0	182	2	0	0	0	0	0	0	1	0	7	0	88	270	32.59
14	0	0	0	0	2	0	0	0	0	717	0	0	0	0	0	0	0	0	0	1	3	720	0.42
15	0	0	0	0	0	0	1	0	0	0	269	0	0	0	0	0	0	0	0	0	1	270	0.37
16	0	0	0	0	0	0	0	0	0	4	0	686	0	0	0	0	0	0	0	0	4	690	0.58
17	17	374	9	9	133	16	1	11	183	6	0	6	649	29	0	34	10	0	9	64	911	1560	58.40
19	6	0	0	0	9	32	0	79	0	16	1	687	48	641	36	12	10	0	0	133	1069	1710	62.51
20	3	1	0	0	7	142	5	1	0	725	177	1	37	92	294	8	32	0	3	2	1236	1530	80.78
21	1	5	2	8	65	6	4	13	0	20	280	6	80	14	9	503	28	0	95	1	637	1140	55.88
22	0	0	49	0	2	38	5	48	27	9	54	2	2	0	0	0	473	0	11	0	247	720	34.31
23	0	0	0	0	4	142	0	1	0	1	0	0	0	0	1	0	0	0	1	0	150	150	100.00
26	0	0	0	0	0	0	0	0	0	10	48	3	3	9	0	3	0	0	610	4	80	690	11.59
27	0	0	0	0	0	0	0	0	0	2	0	0	12	0	0	0	2	0	0	44	16	60	26.67

Tabela 7.11: Matriz Confusão. Equivalência Simples - Croata x Belga

Classe	0	1	4	5	6	7	8	9	11	14	15	16	22	23	24	25	28	29	Erro	Total	Perce
0	19	3	6	0	5	1	0	0	0	0	0	0	0	0	0	0	5	0	20	39	51.28
1	0	179	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	1	4	183	2.19
4	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0.00
5	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0.00
6	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.00
7	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	6	100.00
8	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	15	0.00
9	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	7	0.00
11	0	6	0	0	1	26	0	6	50	0	0	0	0	0	0	0	1	0	40	90	44.44
14	0	0	0	0	0	0	0	0	0	163	0	0	0	0	0	0	0	0	0	163	0.00
15	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	45	0.00
16	0	0	0	0	0	0	0	0	0	0	0	105	0	0	0	0	0	0	0	105	0.00
22	0	0	0	0	0	0	0	0	0	0	0	0	86	0	0	0	0	0	0	86	0.00
23	0	0	0	0	0	0	0	3	0	0	0	0	0	7	0	0	0	1	4	11	36.36
24	0	0	0	0	0	0	0	0	1	1	8	0	1	0	199	1	0	2	14	213	6.57
25	0	0	0	0	0	0	0	0	0	0	1	0	4	0	3	88	0	3	11	99	11.11
28	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	1	33	3.03
29	0	0	0	0	0	0	0	0	0	0	4	0	0	0	2	1	0	77	7	84	8.33

7.3 Matrizes Confusão. Equivalência Completa

Tabela 7.12: Matriz Confusão. Equivalência Completa - Alemão x Belga

Classe	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	28	29	35	36	40	99	Erros	Total	Perc
9	82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	86	4.65
11	0	180	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	183	1.64
12	0	0	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	105	0.00
13	0	0	0	163	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	0.00
14	0	0	0	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	45	2.22
15	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0.00
16	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.00
17	0	0	0	0	0	0	0	61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	61	0.00
18	0	0	0	0	0	0	0	0	38	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	39	2.56
19	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0.00
20	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0.00
21	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	3	0.00
22	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	6	0.00
23	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	7	0.00
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	12	0.00
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	28	0.00
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	0	0	0	0	0	3	3	90	3.33
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	0	0	0	0	7	10	12	83.33
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	9	0.00
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	2	2	31	6.45
99	4	3	11	5	0	16	2	0	3	0	4	0	29	3	15	1	3	9	8	76	9	0	1367	201	1568	12.82

Tabela 7.13: Matriz Confusão. Equivalência Completa - Alemão x Croata

Classe	1	2	3	4	9	11	12	13	14	18	19	20	21	23	27	28	31	38	41	99	Erro	Total	Perc
1	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0.00
2	1	20	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	4	24	16.67
3	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	39	40	97.50
4	1	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	28	3.57
9	0	0	0	0	92	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	4	96	4.17
11	0	0	0	0	0	44	0	0	0	0	0	0	0	0	2	0	0	0	0	2	4	48	8.33
12	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0.00
13	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0.00
14	0	0	0	0	0	0	1	0	135	0	0	0	0	0	0	0	0	0	0	0	1	136	0.74
18	0	0	0	0	0	0	0	0	1	32	0	2	0	0	0	0	0	0	0	5	8	40	20.00
19	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	1	0	0	4	5	64	7.81
20	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0	0	0	0	0	9	9	72	12.50
21	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	5	5	60	8.33
23	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	9	9	40	22.50
27	0	0	0	0	0	0	0	0	0	3	0	0	0	0	46	0	0	0	0	31	34	80	42.50
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	0	0	0	14	14	88	15.91
31	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	50	0	0	1	2	52	3.85
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	20	0.00
41	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	2	3	56	5.36
99	3	2	39	0	33	59	0	1	0	32	0	16	0	1	8	38	1	0	3	400	236	636	37.11

Tabela 7.14: Matriz Confusão. Equivalência Completa - Belga x Alemão

Classe	0	2	3	4	5	7	8	10	11	13	16	17	19	21	22	25	28	31	34	36	37	61	99	Erros	Total	Perc	
0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	60	61	120	50.83	
2	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	146	150	150	100.00	
3	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	34	60	56.67	
4	0	0	0	0	27	0	3	0	0	0	0	3	7	0	0	0	2	0	0	0	0	0	48	63	90	70.00	
5	0	0	1	0	11	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	74	79	90	87.78	
7	0	0	0	9	0	74	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	2	62	76	150	50.67	
8	0	0	0	3	8	1	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	72	90	90	100.00	
10	0	0	0	0	0	34	17	0	241	0	0	3	0	0	0	0	0	0	0	0	0	0	185	239	480	49.79	
11	0	4	0	13	0	0	0	0	72	59	0	0	0	0	9	0	1	0	0	0	0	1	21	108	180	60.00	
13	0	0	3	6	0	17	10	18	1	258	0	2	0	0	0	0	0	0	0	0	0	0	75	132	390	33.85	
16	0	0	0	0	0	7	0	0	0	1	14	0	0	0	0	0	0	0	0	0	2	0	66	76	90	84.44	
17	0	0	0	0	0	54	18	0	0	1	4	287	1	0	0	0	0	0	0	0	0	0	55	133	420	31.67	
19	0	0	0	0	0	0	0	0	0	0	0	0	618	0	4	0	10	0	0	0	0	0	88	102	720	14.17	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	250	1	0	0	0	0	0	0	2	17	20	270	7.41	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	6	316	0	0	0	0	0	0	0	12	26	44	360	12.22
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	14	0	0	0	36	0	22	72	150	48.00	
28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	186	0	0	0	0	0	22	24	210	11.43	
31	0	0	26	3	8	1	0	0	0	0	5	0	0	1	0	1	76	224	0	0	0	4	371	496	720	68.89	
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	0	1	0	221	222	390	56.92	
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79	0	0	41	41	120	34.17	
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69	0	21	21	90	23.33	
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	52	0	0	0	3	581	51	109	690	15.80	
99	59	4	44	59	50	196	44	259	73	408	69	310	624	299	342	82	542	178	168	105	230	723	9562	4868	14430	33.74	

Tabela 7.15: Matriz Confusão. Equivalência Completa - Belga x Croata

Classe	2	3	4	5	6	7	13	14	17	19	21	31	38	39	41	45	56	61	99	Total	Erro	Perc
2	7	0	2	1	1	8	0	1	0	0	5	0	0	0	0	0	0	3	12	40	33	82.50
3	6	28	1	0	0	0	0	2	0	0	3	0	0	0	1	0	2	4	17	64	36	56.25
4	0	0	53	0	1	0	0	1	0	0	11	0	0	0	0	0	0	0	6	72	19	26.39
5	3	14	0	0	0	0	0	7	0	0	8	0	0	0	0	0	0	0	28	60	60	100.00
6	1	0	17	0	2	18	0	2	0	0	8	0	0	0	0	0	0	2	22	72	70	97.22
7	0	0	1	1	11	18	0	5	0	0	4	0	0	0	0	0	0	0	48	88	70	79.55
13	0	0	2	0	0	1	25	1	0	0	4	0	0	0	0	0	1	0	6	40	15	37.50
14	0	1	0	0	0	0	0	9	0	0	4	0	0	0	0	0	0	0	18	32	23	71.88
17	0	0	0	0	0	10	1	1	4	0	0	0	0	0	0	0	2	2	28	48	44	91.67
19	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	4	40	4	10.00
21	0	0	0	0	0	0	0	0	0	0	118	0	1	0	1	0	0	8	8	136	18	13.24
31	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	4	28	96	32	33.33
38	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	40	0	0.00
39	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	16	0	0.00
41	0	0	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	13	28	21	75.00
45	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	1	0	24	28	27	96.43
56	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0	0	123	0	36	164	41	25.00
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	114	6	120	6	5.00
99	13	3	2	10	4	10	13	12	16	0	3	7	15	4	0	4	5	7	472	600	128	21.33

Tabela 7.16: Matriz Confusão. Equivalência Completa - Croata x Alemão

Classe	0	1	4	5	6	9	10	11	12	14	15	16	17	19	20	21	22	23	26	27	99	Erros	Total	Perc
0	252	0	1	0	20	2	17	0	0	0	4	0	0	0	2	0	0	0	1	0	91	138	390	35.38
1	0	260	0	0	4	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	153	160	420	38.10
4	0	0	8	0	35	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	60	86.67
5	0	0	0	72	0	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0	11	18	90	20.00
6	1	0	0	0	48	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	30	42	90	46.67
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
12	15	0	2	1	35	9	0	10	150	0	0	0	0	0	0	0	0	2	0	46	120	270	44.44	
14	0	0	0	0	2	0	0	1	0	693	3	1	0	0	1	0	1	0	2	0	16	27	720	3.75
15	0	0	0	0	0	0	0	0	0	0	266	0	0	0	0	0	1	0	0	1	2	4	270	1.48
16	0	0	0	0	1	0	0	0	0	2	608	6	0	0	0	4	0	0	0	69	82	690	11.88	
17	16	260	2	1	95	9	1	10	161	2	0	2	556	25	0	9	5	0	3	44	359	1004	1560	64.36
19	20	0	1	2	16	2	5	4	0	5	2	608	57	498	16	0	9	0	0	99	366	1212	1710	70.88
20	0	0	0	2	8	131	2	10	0	695	11	5	0	0	132	0	5	0	3	2	524	1398	1530	91.37
21	0	0	0	0	18	4	2	9	0	3	269	8	90	4	0	247	2	0	81	2	401	893	1140	78.33
22	0	0	8	0	35	32	0	142	1	1	0	0	0	0	0	0	458	0	0	43	262	720	36.39	
23	0	0	0	0	6	131	0	6	0	0	0	0	0	0	0	0	0	0	0	7	150	150	100.00	
26	0	0	0	0	0	0	0	0	0	25	18	2	1	0	2	1	1	0	504	0	136	186	690	26.96
27	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	5	0	0	44	10	16	60	26.67
99	654	10	1	456	66	110	83	428	8	171	255	66	405	80	542	41	1279	0	665	226	4144	5546	9690	57.23

Classe	0	1	4	5	6	7	8	9	11	14	15	16	22	23	24	25	28	29	99	Erro	Total	Perc
0	6	0	0	0	0	0	3	0	2	0	0	0	0	0	0	0	0	0	28	33	39	84.62
1	0	75	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	104	108	183	59.02
4	0	0	5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	6	16.67
5	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0.00
6	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3	33.33
7	0	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0	0	0	0	3	6	50.00
8	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	2	2	15	13.33
9	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	2	3	7	42.86
11	0	0	0	0	1	29	0	0	44	0	0	0	0	0	0	0	2	0	14	46	90	51.11
14	0	1	3	0	0	1	0	0	1	141	0	0	0	8	0	0	0	3	5	22	163	13.50
15	0	0	0	0	0	0	0	0	0	3	41	1	0	0	0	0	0	0	0	4	45	8.89
16	0	0	0	0	0	0	0	0	0	0	0	104	0	1	0	0	0	0	0	1	105	0.95
22	0	0	0	0	0	0	0	0	0	0	0	0	81	0	0	0	0	0	5	5	86	5.81
23	0	0	0	0	0	0	0	0	1	2	1	0	0	1	0	0	0	3	3	10	11	90.91
24	0	0	0	0	0	0	0	0	7	0	35	1	2	6	143	1	0	13	5	70	213	32.86
25	0	0	0	0	0	0	0	2	10	0	2	0	0	5	4	69	0	3	4	30	99	30.30
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	9	9	33	27.27
29	0	0	0	0	0	0	0	0	35	0	14	0	0	0	0	13	0	17	5	67	84	79.76
99	1	4	41	4	20	38	13	24	42	11	15	12	22	47	0	0	54	35	937	383	1320	29.02

7.4 Matrizes Confusão. Estratégia Única

[illegible]

Tabela 7.19: Matriz Confusão. Estratégia Única - Alemão (2)

[illegible]

Tabela 7.20: Matriz Confusão. Estratégia Única - Belga (1)

Classe	2	3	4	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	28	29	35	40	43	44	45	46
9	0	0	0	86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	183	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	163	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	1	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	209	1
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	98	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	4	1	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	8	15	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Tabela 7.21: Matriz Confusão. Estratégia Única - Belga (2)

Classe	43	44	45	46	47	48	49	50	52	54	55	56	57	59	60	61	62	64	65	66	67	68	69	70	72	74	76	77	78	79	80	81	82	84	Erro	Total	Perc
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	183	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	105	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	45	222	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	61	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	39	513	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	28	357	
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	
43	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	
44	0	209	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	213	188	
45	0	0	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	99	1.01	
46	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	
47	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	
48	0	0	0	0	0	0	77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	3	3	0	0	0	0	0	0	0	0	7	84	833
49	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	
50	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	
52	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
54	0	0	0	0	0	0	0	0	0	118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	122	328	
55	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
56	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	
57	0	0	0	0	0	0	0	0	0	0	0	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	13	23.08	
59	0	0	0	0	0	0	0	0	0	0	0	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	9	22.22	
60	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	414	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	422	1.9	
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	154	16.23	
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	3	6	50	
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	31	0	
72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	3	0	
74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	3	0	
76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	24	0		
77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	48	0		
78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	15	0		
79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	
81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	17	0	
82	0																																				

Tabela 7.22: Matriz Confusão. Estratégia Única - Croata (1)

Classe	1	2	3	4	9	11	12	13	14	18	19	20	21	23	27	28		
1	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	136	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	32	0	0	2	0	0	2	0	0
19	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	71	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0
27	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	79	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	1
31	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
86	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 7.23: Matriz Confusão. Estratégia Única - Croata (2)

Classe	31		38	41	43	44	45	46	47	48	49	70	79	83	84	85	86	87	Erros	Total	Perc
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	24	8.33
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	28	7.14
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	48	2.08
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	8	40	20
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0
20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	72	1.39
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
23	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	40	2.5
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	80	1.25
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	88	1.14
31	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	52	7.69
38	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0
41	0	0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0
43	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	2	32	6.25
44	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
45	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	16	0
46	0	0	0	0	0	0	0	72	0	0	0	0	0	0	0	0	0	0	0	72	0
47	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	28	0
48	0	0	0	0	0	0	0	0	0	23	0	1	2	0	0	0	0	0	5	28	17.86
49	0	0	0	0	0	0	0	0	0	0	163	0	0	0	0	0	0	0	1	164	0.61
83	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	0	0	0	0	56	0
84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	0	0	0	56	0
85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	1	16	6.25	
86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	0	2	56	3.57	
87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	72	0	72	0	

Tabela 7.24: Matriz Confusão. Estratégia Única - Alemão x Belga x Croata (1)

Classe	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40					
0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
1	0	761	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	0	3	768	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	478	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	0	1	0	1	679	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	1	4	0	621	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
6	0	0	0	0	0	12	130	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0		
7	0	0	0	0	0	1	0	446	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
8	0	0	0	1	0	7	0	1	441	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
9	0	0	0	0	0	0	0	0	662	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
10	0	0	0	0	0	1	0	0	0	0	656	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
11	0	0	0	0	0	0	0	0	0	0	0	638	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	8	0	0	0	0	0	0	0	0	0	0	0		
12	0	0	0	0	0	0	0	0	0	0	0	0	913	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
13	0	0	0	0	0	0	0	0	0	0	0	0	0	922	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	451	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	260	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	414	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
18	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0	0	431	0	0	5	0	0	0	0	2	4	3	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	173	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	0	0	0	0	0	6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	492	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	138	0	0	0	0	0	0	0	0	0	0	0	0			
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	326	1	0	0	0	0	0	0	0	0	0	0	0			
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0	0	0	0	0	0			
30	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0	0	0	134	6	0	0	0	0	0	0	0	0	0	0		
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
33	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0</						

