



CENTRO UNIVERSITÁRIO SENAI CIMATEC
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL
E TECNOLOGIA INDUSTRIAL

DANIEL DA SILVA MOTTA

**Otimização na classificação automática de mosquitos adultos:
uma investigação com redes neurais convolucionais (CNN)**

Salvador
2019

DANIEL DA SILVA MOTTA

**Otimização na classificação automática de mosquitos adultos:
uma investigação com redes neurais convolucionais (CNN)**

Tese de Doutorado apresentada ao Programa de Pós-Graduação de Modelagem Computacional e Tecnologia Industrial, Curso de Doutorado em Modelagem Computacional e Tecnologia Industrial do Centro Universitário SENAI CIMATEC, como requisito para a obtenção do título de Doutor em Modelagem Computacional e Tecnologia Industrial.

Orientador: Prof. Dr. Alex A. B. Santos
Coorientador: Prof. Dr. Roberto Badaró

**Salvador
2019**

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

M921o Motta, Daniel da Silva

Otimização na classificação automática de mosquitos adultos: uma investigação com redes neurais convolucionais (CNN) / Daniel da Silva Motta. – Salvador, 2019.

148 f. : il. color.

Orientador: Prof. Dr. Alex Álisson Bandeira Santos.
Coorientador: Prof. Dr. Roberto José da Silva Badaró.

Tese (Doutorado em Modelagem Computacional e Tecnologia Industrial) – Programa de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2019.

Inclui referências.


1. Aedes 2. Culex. 3. Aprendizagem de máquina. 4. Aprendizagem profunda. 5. Redes neurais convolucionais. I. Centro Universitário SENAI CIMATEC. II. Santos, Alex Álisson Bandeira. III. Badaró, Roberto José da Silva. IV. Título.

CDD: 614


Doutorado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leu e aprovou a Tese de doutorado, intitulada "**Otimização na classificação automática de mosquitos adultos: uma investigação com redes neurais convolucionais (CNN).**", apresentada no dia 27 de março de 2019, como parte dos requisitos necessários para a obtenção do Título de Doutor em Modelagem Computacional e Tecnologia Industrial.

Orientador:


Prof. Dr. Alex Álisson Bandeira Santos
SENAI CIMATEC


Coorientador:


Prof. Dr. Roberto José da Silva Badaró
FIEB

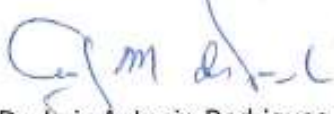
Membro Interno:


Prof. Dr. Marcelo Albano Moret Simões Gonçalves
SENAI CIMATEC


Membro Interno:


Prof. Dr. Valter de Senna
SENAI CIMATEC

Membro Externo:


Prof. Dr. Luiz Antonio Rodrigues de Freitas
UFBA

Membro Externo:


Prof. Dr. Eduardo Mario Dias
USP

Dedico este trabalho à minha família e amigos

AGRADECIMENTOS

À minha família, em especial à esposa e filha, Aline e Isabel, aos meus Pais, Renato e Nelba, e aos meus irmãos, Alexandre e Gustavo, pelo amor e presença diários, que me dão força para enfrentar qualquer desafio.

Aos meus orientadores, Prof. Dr. Alex Santos e Prof. Dr. Roberto Badaró, por me guiarem com paciência e dedicação.

Ao Prof. Dr. Leone Andrade, pelo incentivo para ingressar no programa de doutorado e apoio durante todo o desenvolvimento desta tese.

À Prof^a. Dra Bruna Machado, pelo apoio essencial para publicação dos artigos científicos que deram base ao desenvolvimento deste trabalho.

À Prof^a. Dra Ingrid Winkler, pelo apoio na definição do tema e pesquisa bibliográfica.

À Eduardo Oyama, entomologista do ITS CIMATEC, pelas aulas sobre entomologia e pela aquisição e estruturação do banco de imagens.

À Daniel Imperial e Alexandre Cavancanti, bolsistas Fapesb e estudantes de graduação do SENAI CIMATEC, por manter o trabalho de pesquisa continuamente ativo com as reuniões semanais e pelo apoio no desenvolvimento da pesquisa bibliográfica e dos experimentos.

Ao colega Otávio Filho, por toda a orientação, aula e dedicação no desenvolvimento dos códigos de programa usados neste trabalho.

Aos professores e colegas dos Programas de Pós-Graduação do SENAI CIMATEC, pela contribuição a cada apresentação e durante as aulas, em especial aos professores Dr. Valter de Senna, Dr. Marcelo Moret e Dr Hernane Pereira, e, aos colegas Flavio Marinho e André Oliveira.

Aos colegas do DFKI Bremen – Instituto de Inteligência Artificial da Alemanha -, Frank Kirchner, Sirko Straube, Nina Hoyer e Luis Octavio, por todo o apoio e ensinamentos durante o período de desenvolvimento do trabalho na Alemanha.

Ao SENAI CIMATEC por todo o apoio, pela infraestrutura computacional e disponibilidade de equipe, fundamentais para o desenvolvimento deste trabalho.

Ao Departamento Nacional do SENAI pelo apoio financeiro para desenvolvimento da etapa experimental na Alemanha.

À FAPESB pela disponibilização de bolsas de iniciação científica.

À FIOCRUZ pela cessão de fotos usadas neste trabalho e, em especial, à colegas Maycon Neves e Monique Motta pelo treinamento de classificação de mosquitos usando chaves dicotômicas.

A todos aqueles que contribuíram de forma direta ou indireta. Foram muitos que apoiaram esse desenvolvimento. Meus sinceros agradecimentos.

“Nunca ande por trilhas, pois assim só irá até onde outros já foram”

Alexander Graham Bell

RESUMO

Dengue, chikungunya e Zika são arboviroses transmitidas por mosquitos do gênero *Aedes* com diversos surtos de doenças no mundo nos dez últimos anos, causando um impacto negativo na saúde pública e danos econômicos em todo o mundo. Atualmente, a identificação morfológica de mosquitos inclui um exame visual individual do inseto, o que demanda tempo elevado para a sua classificação, e, além disso, requer expertise na área (vários anos de experiência do profissional). Outra possibilidade também utilizada, é a classificação a partir de técnicas moleculares, que apesar da elevada precisão, requer um alto grau de investimento, limitando o seu uso. Dentro desta problemática, pesquisadores têm desenvolvido um arsenal de ferramentas para reduzir, prever ou eliminar efeitos causados por esses mosquitos. Com base nesse contexto, neste trabalho, foi desenvolvido um modelo computacional baseado em redes neurais convolucionais (CNN's) para extrair características de imagens de mosquitos com o objetivo de classificar as espécies de *Aedes aegypti*, *Aedes albopictus* e de *Culex quinquefasciatus*, bem como para detectar mosquitos do gênero *Aedes*. Dessa forma, para treinar as CNN's, com o intuito de realizar a classificação morfológica automática dos mosquitos, foi usado um banco de imagens que incluiu 7.561 imagens das três espécies-alvo do estudo e 1.187 imagens de outros insetos. As amostras de mosquitos usadas para captura de imagens foram obtidas a partir do Laboratório de Parasitologia da Universidade Federal da Bahia (UFBA) e também coletadas em campo (Salvador, Bahia, Brazil). Diferentes redes neurais, incluindo LeNet, AlexNet, GoogleNet, Xception e DenseNet, foram utilizadas para o desenvolvimento do modelo de classificação automática a partir de imagens. Um processo estruturado de pesquisa aleatória e pesquisa exaustiva foi desenvolvido para otimização dos hiperparâmetros na busca do aumento da acurácia do modelo. Estratégias de eliminação de *overfitting* foram empregadas computacionalmente, visando a maior generalização do modelo proposto. O modelo otimizado foi testado para as atividades de classificação dos mosquitos-alvo do estudo e para detecção do gênero *Aedes*, obtendo o resultado de acurácia balanceada global de 93,5% para classificação e de 97,3% para detecção. Uma investigação sobre a influência do ângulo do mosquito na imagem foi conduzida para orientar sobre a maneira mais adequada para aquisição das imagens. Os resultados fornecem informações fundamentais para a classificação morfológica automática de espécies de mosquitos. A incorporação do modelo em uma poderosa ferramenta entomológica é um recurso valioso e acessível para profissionais de saúde e outros não especialistas para a identificação de insetos que podem transmitir agentes infecciosos para seres humanos.

Palavras-chave: *Aedes*; *Culex*; Classificação; Aprendizagem de Máquina; Aprendizagem Profunda; Redes Neurais Convolucionais.

ABSTRACT

Dengue, chikungunya and Zika are arboviruses transmitted by mosquitos of the genus *Aedes* with several outbreaks in world over the past ten years, causing a negative impact on public health and economic damages worldwide. Currently, morphological identification of mosquitos includes an individual visual examination of the samples, which is time consuming and requires adequately trained professionals with several years of experience. Another possibility also used is the classification from molecular techniques, which, despite its precision, requires a high degree of investment, limiting its use. Within this problem, researchers have developed an arsenal of tools to reduce, predict or eliminate effects caused by these mosquitoes. In this context, in this work a computational model based on convolutional neural network (CNN) was developed to extract features from mosquito images and classify the species *Aedes aegypti*, *Aedes albopictus* and *Culex quinquefasciatus*, as well as to detect mosquitoes of the genus *Aedes*. Thus, to train the CNN's to perform automatic morphological classification of mosquitoes, a dataset was structured, that included 7,561 images of the three target mosquitoes of the study and 1,187 images of other insects. The mosquito samples used for image capture were obtained from the Parasitology Laboratory of the Federal University of Bahia (UFBA) and also collected in the field (Salvador, Bahia, Brazil). Different neural networks, including LeNet, AlexNet, GoogleNet, Xception e DenseNet, were used for the development of the image based automatic classification model. A structured random search and grid search was developed to optimize the hyperparameters in order to increase the accuracy of the model. Strategies to eliminate overfitting were also implemented, aiming at greater generalization of the proposed model. The optimized model was tested for the classification task of the target mosquitoes and for the detection task of the genus *Aedes*, obtaining the overall balanced accuracy of 93.5% for classification and 97.3% for detection. An investigation about the influence of the angle of the mosquito on the image was conducted to guide the appropriate way to acquire the images. The results provide fundamental information for the automatic morphological classification of mosquito species. The use of CNN embedded in an entomological tool is a valuable and accessible resource for health workers and other non-taxonomists for the identification of insects that can transmit infectious agents to humans.

Key-words: *Aedes*; *Culex*; Classification; Machine Learning; Deep Learning, Convolutional Neural Networks

LISTA DE TABELAS

Capítulo 1

Tabela 1. Metas estabelecidas no plano de resposta global no controle de vetores transmissores de doenças de 2017 a 2030.....	21
---	----

Capítulo 4

Table 1. Parameters used as inputs for the full factorial experiments in this study.	76
---	----

Table 2. Dataset structure and sample sizes of the mosquito images used in this study.....	77
--	----

Table 3. Confusion matrix showing the results of the testing phase for the LeNet neural network.....	81
--	----

Table 4. Confusion matrix showing the results of the testing phase for the AlexNet neural network.....	81
--	----

Table 5. Confusion matrix showing the results of the testing phase for the GoogLeNet neural network.....	82
--	----

Capítulo 5

Tabela 1. Número de imagens por classe.....	95
---	----

Tabela 2. Pesquisa de grade – hiperparâmetros para ampliação das imagens no banco de dados.....	100
---	-----

Tabela 3. Pesquisa de grade – CNN para extratação dos atributos das imagens e hiperparâmetros para as camadas de classificação.....	101
---	-----

Tabela 4. Pesquisa aleatória para definição dos hiperparâmetros da pesquisa de grade.....	107
---	-----

Tabela 5. Matriz de confusão da rede DenseNet201 após o processo de otimização dos hiperparâmetros de ampliação do banco de dados.....	116
Tabela 6. Resultado da pesquisa de grade (grid search) dos hiperparâmetros para extração de atributos e classificação.....	117
Tabela 7. Matriz de confusão da rede DenseNet201 após o processo de otimização do modelo.....	122
Tabela 8. Matriz de confusão do modelo otimizado para 2 classes.....	124
Tabela 9. Matriz de confusão do modelo otimizado para 7 classes.....	126

LISTA DE QUADROS

Capítulo 1

Quadro 1. Diferentes técnicas desenvolvidas e aplicadas para contagem, detecção e classificação de insetos.....24

Capítulo 2

Quadro 1. Configuração básica de algumas CNNs.....48

Capítulo 5

Quadro 1. Matriz de Confusão 2x2.....98

Quadro 2. Matriz de Confusão acima de duas classes.....99

Quadro 3: Configuração dos hiperparâmetros obtidos após a otimização para classificação de mosquitos adultos.....121

Quadro 4: Cálculos dos indicadores de desempenho da matriz de confusão 2x2.
.....124

LISTA DE FIGURAS

Capítulo 1

Figura 1. Distribuição global dos principais vetores de transmissão de doenças.
.....20

Figura 2. Representação esquemática do processo de classificação tradicional de mosquitos adultos.....22

Capítulo 2

Figura 1. Ciclo de vida do mosquito (genérico).....33

Figura 2. Representação genérica da arquitetura de uma rede neural convolucional (CNN).....40

Figura 3. Representação de uma camada convolucional.....40

Figura 4. Representação de uma camada de pooling máximo.....41

Figura 5. Representação de uma camada totalmente conectada.....42

Figura 6. Linha do tempo de arquiteturas de redes neurais convolucionais entre 2012 e 2018.....43

Figura 7. Margem de erro-top-5 das arquiteturas de redes vencedoras do desafio ILSVRC entre 2012 e 2016.....44

Figura 8. Representação da arquitetura da DenseNet.....48

Capítulo 3

Figure 1. Human vs. computer: how an image is recognized by each (illustrative).
.....56

Figure 2. Relationship between artificial intelligence, machine learning, and deep learning.....56

Figure 3. The number of papers published from 2000 to 2017 relating AI to vector-borne diseases.....	58
Figure 4. The number of patents granted from 2000 to 2017 relating AI to vector-borne diseases.....	58
Figure 5. Countries with more patents from 2000 to 2017 relating AI to vector-borne diseases.....	58
Figure 6. Combined global distribution of seven major vector-borne diseases: malaria, lymphatic filariasis, leishmaniasis, dengue, Japanese encephalitis, yellow fever, and Chagas disease transmission.....	59
Figure 7. Image of <i>Culex quinquefasciatus</i> from Fiocruz.....	60
Figure 8. Representation of the process using dichotomous keys for the classification of mosquitoes.....	61
Figure 9. Process of entomological characterization of an adult mosquito.....	61
Figure 10. Representation of underfitting and overfitting.....	64

Capítulo 4

Fig 1. Representation of the training (phase 1), validating (phase 2) and testing (phase 3) processes of the mosquito classification model.....	73
Fig 2. Images used for the validation phase during the development of the model used in this study.	74
Fig 3. Effects of the first level parameters on the accuracy of the testing phase (average accuracy percentage).....	78
Fig 4. Effects of the second level parameters on the accuracy of the testing phase (average accuracy percentage).....	78
Fig 5. Histograms representing the outcome data.....	79
Fig 6. Representation of the training process of each CNN:	80

Capítulo 5

Figura 1. Amostra de Imagens utilizadas para treinamento, validação e teste do modelo.....	96
Figura 2. Amostras de imagens com diferentes ângulos dos mosquitos.....	103
Figura 3. Programas elaborados e utilizados para treinamento e avaliação do modelo de classificação.....	104
Figura 4. Evolução da aprendizagem da rede ResNet50.....	108
Figura 5. Evolução da aprendizagem da rede com o uso da VGG16.....	109
Figura 6. Evolução da aprendizagem das redes Xception e DenseNet201.....	111
Figura 7. Evolução da aprendizagem da rede Xception, após a redução da taxa de aprendizagem inicial para 1E-5 (experimento 16).....	112
Figura 8. Evolução da aprendizagem da rede DenseNet, após a ampliação do dataset.....	113
Figura 9. Impacto da variação dos hiperparâmetros de ampliação de dados na acurácia na fase de validação.....	114
Figura 10. Impacto da variação dos hiperparâmetros de ampliação de dados na função perda (cross entropy) na fase de validação.....	114
Figura 11. Função da taxa de aprendizagem (learning rate) ao longo das épocas de treinamento.....	115
Figura 12. Acurácia balanceada global para os experimentos 7 ao 16 (Tabela 6).	118
Figura 13. Variação da acurácia balanceada global dos experimentos 12 e 13.....	119
Figura 14. Acurácia balanceada global com a variação do tamanho de lote.....	120

Figura 15. Evolução da aprendizagem do modelo otimizado para duas classes..	
.....	123
Figura 16. Evolução da aprendizagem do modelo otimizado para sete classes...	
.....	126
Figura 17. Desempenho da acurácia balanceada global e de cada uma das seis classes com diferentes ângulos dos mosquitos.....	127

GLOSSÁRIO E LISTA DE SIGLAS

<i>Acc</i>	Acurácia: consiste no somatório de verdadeiro-positivo e verdadeiro-negativo, sobre o total de amostras avaliadas.
<i>BA</i>	Acurácia Balanceada (<i>Balanced Accuracy</i>): considera o desbalanceamento do banco de dados (<i>dataset</i>). É a média entre a taxa verdadeiro-positivo e a taxa verdadeiro-negativo.
<i>Batch size</i>	Tamanho do lote. Número de exemplos de treinamento usado a época.
<i>Bottlenecks</i>	Vetores dos atributos do banco de dados.
<i>Bottlenecks X</i>	Vetores dos atributos das imagens processados pela rede.
<i>Bottlenecks Y</i>	Conjunto de imagens com seus rótulos.
<i>Callback</i>	Conjunto de funções aplicadas em determinados estágios do processo de aprendizagem
<i>CNN</i>	Rede Neural Convolutiva.
<i>Convolução</i>	Operação matemática em duas funções de um argumento de valor real.
<i>Culicídeos</i>	Família de insetos habitualmente chamados de muriçoca, mosquitos ou pernilongos.
<i>Deep Learning</i>	Aprendizagem profunda.
<i>Dropout</i>	Método de zerar a ativação de determinados neurônios de forma estocástica ao longo das épocas de treinamento
<i>Earlystopping</i>	<i>Callback</i> para interromper o processo de aprendizagem, quando não se observa redução da função perda.
<i>Entomologia</i>	Especialidade da biologia que estuda os insetos.

Épocas	Ciclos de interações na aprendizagem nas fases de treinamento e validação.
Erro-top-5	% de imagens de teste para as quais o rótulo correto não está entre os cinco rótulos considerados mais prováveis pelo modelo.
FP	Falso Positivo
FN	Falso Negativo
GAP	<i>Global Average Pooling.</i>
GPU	Graphics Processing Unit, ou Unidade de Processamento Gráfico.
Hiperparâmetros	Parâmetros de treinamento das CNN's que não são aprendidos automaticamente.
IA	Inteligência Artificial.
ILSVRC	ImageNet Large Scale Visual Recognition Competition (desafio de reconhecimento visual de grande escala).
<i>ImageNet</i>	Banco de dados de imagens organizado de acordo com a hierarquia do WordNet.
<i>Learning Rate</i>	Taxa de aprendizagem.
<i>Machine Learning</i>	Aprendizagem de máquina.
OMS/WHO	Organização Mundial da Saúde.
<i>Overfitting</i>	Efeito resultante de um número limitado de dados, que afeta a correlação entre saída e entrada, mas não correspondem aos dados da vida real. Corresponde à incapacidade de generalização das análises da rede.
PAHO	Organização Pan-Americana da Saúde.
Perda (<i>cross entropy</i>)	Função de custo de entropia cruzada.
Pesquisa aleatória	Busca aleatória de um subconjunto de hiperparâmetros. Conhecida na literatura internacional como <i>random search</i> .
Pesquisa de grade	Busca exaustiva através de um subconjunto de hiperparâmetros especificado manualmente.

	Conhecida na literatura internacional como <i>grid search</i> .
Precisão (PR)	% de acerto na predição da classificação. Representa o número de predições verdadeiras corretas do algoritmo, em relação ao número total de predições feitas.
<i>Reduce LR on Plateau</i>	Função para redução da taxa de aprendizagem, caso o desempenho da rede não melhore após um determinado número de épocas.
<i>Transfer Learning</i>	Utilização de modelos já treinados.
TVN	Taxa Verdadeiro-Negativo. % de predições negativas corretas do algoritmo, em relação ao número total de amostras de cada classe. Representa os casos que o algoritmo classificou como “NÃO” e, de fato, era “NÃO”.
TVP	Taxa Verdadeiro-Positivo. % de predições verdadeiras corretas do algoritmo, em relação ao número total de amostras de cada classe. Representa os casos que o algoritmo classificou como “SIM” e, de fato, era “SIM”.
<i>Underfitting</i>	Efeito resultante de um baixo número de interações que não foi suficiente para o aprendizado da rede.
VN	Verdadeiro Negativo.
VP	Verdadeiro Positivo.

SUMÁRIO

1	INTRODUÇÃO	20
1.1	DEFINIÇÃO DO PROBLEMA.....	20
1.2	OBJETIVOS	26
1.2.1	Objetivo Geral	26
1.2.2	Objetivos Específicos	27
1.3	VALOR AGREGADO DESTE ESTUDO.....	27
1.4	ESTRUTURA DO TRABALHO	28
1.5	REFERÊNCIAS	30
2	REVISÃO BIBLIOGRÁFICA	33
2.1	MÉTODOS PARA CLASSIFICAÇÃO DE MOSQUITOS	33
2.1.1	Métodos baseados na observação de ovos	34
2.1.2	Métodos baseados na observação da fase larvária	35
2.1.3	Métodos baseados na observação de mosquitos adultos ...	35
2.2	REDES NEURAIS CONVOLUCIONAIS – CNN	39
2.2.1	Modelo genérico das CNN's.....	39
2.2.2	Evolução das arquiteturas CNN (um histórico).....	42
2.3	REFERÊNCIAS	50
3	CAPÍTULO DE LIVRO.....	54
4	ARTIGO I.....	68
5	ARTIGO II.....	88
5.1	RESUMO.....	90
5.2	INTRODUÇÃO	91
5.3	MATERIAIS E MÉTODOS.....	94
5.3.1	Declaração de ética.....	94
5.3.2	Coleta de amostras	94
5.3.3	Aquisição das imagens.....	94

5.3.4	Divisão das imagens em treinamento, validação e teste.....	97
5.3.5	Indicadores de avaliação de desempenho do modelo	97
5.3.6	Aumento dos dados (<i>data augmentation</i>).....	99
5.3.7	Definição da arquitetura de CNN e otimização dos hiperparâmetros das camadas de classificação.....	100
5.3.8	Avaliação do modelo otimizado	102
5.3.9	Avaliação da influência do ângulo da imagem	102
5.4	MODELO COMPUTACIONAL.....	104
5.4.1	Programas.....	104
5.5	RESULTADOS E DISCUSSÕES	107
5.5.1	Pesquisa aleatória (<i>random search</i>).....	107
5.5.2	Otimização dos hiperparâmetros para ampliação dos dados (<i>data augmentation</i>)	114
5.5.3	Arquitetura de CNN e otimização dos hiperparâmetros das camadas de classificação	117
5.5.4	Desempenho do modelo para duas classes.....	123
5.5.5	Desempenho do modelo para sete classes.....	125
5.5.6	Avaliação da influência do ângulo da imagem	127
5.6	CONCLUSÕES	129
5.7	REFERÊNCIAS	130
6	CONSIDERAÇÕES FINAIS	133
	ANEXOS	135
	ANEXO I: <i>SPLIT_DATASET.PY</i>	135
	ANEXO II: <i>IMAGE_AUGMENTOR.PY</i>	136
	ANEXO III: <i>GENERATE_BOTTLENECKS.PY</i>	139
	ANEXO IV: <i>TRAIN_TOPLAYER.PY</i>	143
	ANEXO V: <i>TEST_MODEL.PY</i>	147

1 INTRODUÇÃO

1.1 DEFINIÇÃO DO PROBLEMA

As doenças transmitidas por vetores são responsáveis por 17% da carga global estimada de doenças transmissíveis. De acordo com os dados da Organização Mundial da Saúde (OMS), elas causam mais de 700.000 mortes por ano e 80% da população mundial vive em áreas de risco (WHO, 2017). Desta forma, geram altos custos aos programas de saúde pública, principalmente em países subdesenvolvidos ou em desenvolvimento, e, por consequência, impactam de forma significativa no desenvolvimento econômico dos países afetados. Além dos custos gerados por conta da perda de produtividade dos trabalhadores infectados, essas doenças reduzem também receitas trazidas pelo turismo devido às recomendações de evitar áreas com elevada transmissão de arboviroses (WHO, 2017). Na Figura 1, apresenta-se a distribuição global dos principais vetores de transmissão de doenças, e, verifica-se que América do Sul, África e Ásia são os continentes que apresentam maior índice de doenças relacionadas à transmissão por mosquitos.

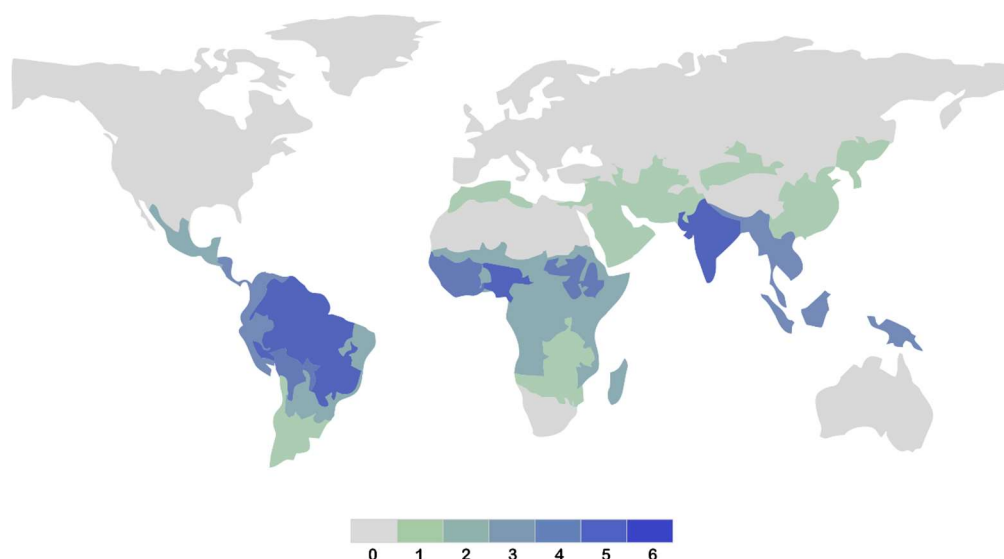


Figura 1. Distribuição global dos principais vetores de transmissão de doenças
(A escala de cor indica o número de doenças que representam risco em um raio de 5 x 5 km.
Cinza indica menor risco e azul escuro maior risco).

Fonte: (WHO, 2017).

Dengue, chikungunya e Zika são as doenças virais mais comuns transmitidas por vetores e causaram várias epidemias na França e na América Latina nos

últimos cinco anos (CAUCHEMEZ et al., 2016; DEL CARPIO-ORANTES; GONZÁLEZ-CLEMENTE; LAMOTHE-AGUILAR, 2018; DING et al., 2018). Essas três doenças virais estão entre os maiores desafios de saúde pública no mundo (PAHO, 2017). A pesquisa entomológica é considerada prioritária pela OMS para o desenvolvimento de ferramentas que possam ser aplicadas para reduzir o número de doenças transmitidas por vetores em todo o mundo (WHO, 2017). Em 2017, essa mesma organização propôs um plano de resposta global no controle de vetores transmissores de doenças de 2017 a 2030 com o objetivo de reduzir casos de doença e de mortalidade, além de erradicar epidemias, conforme apresentado na Tabela 1.

Tabela 1. Metas estabelecidas no plano de resposta global no controle de vetores transmissores de doenças de 2017 a 2030.

Fonte: (WHO, 2017).

Metas	Metas intermediárias		Meta final
	2020	2025	2030
Reduzir a mortalidade devido a doenças transmitidas globalmente por vetores, comparado com 2016	Em no mínimo 30%	Em no mínimo 50%	Em no mínimo 75%
Reduzir casos de ocorrências de doenças transmitidas globalmente por vetores, comparado com 2016	Em no mínimo 25%	Em no mínimo 40%	Em no mínimo 60%
Prevenir epidemias de doenças transmitidas globalmente por vetores (detecção e redução rápidas de surtos, para evitar propagação no mundo)		Em todos os Países sem transmissão em 2016	Em todos os Países

As espécies de mosquitos *Aedes aegypti* e *Aedes albopictus* receberam uma atenção global, uma vez que são vetores eficientes para a transmissão de doenças arbovirais humanas, como Zika, dengue, chikungunya e febre amarela (CEVALLOS et al., 2018; LETA et al., 2018; THANGAMANI et al., 2016). Adicionalmente, estudos controversos indicam o *Culex quinquefasciatus* como um possível transmissor do vírus Zika (GUEDES et al., 2017; GUO et al., 2016a; MAIN et al., 2018). *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus* são mosquitos domiciliares notificados em quase todas as áreas urbanas do território brasileiro (CONSOLI; OLIVEIRA, 1998; PANCETTI et al., 2015). Além disso, os

mosquitos *Aedes* estão presentes durante o ano todo em muitas áreas tropicais nas Américas, na África e na Ásia (ATTAWAY et al., 2017).

Dentro dessa problemática, pesquisadores têm desenvolvido um arsenal de ferramentas mecânicas, químicas, biológicas e educacionais para mitigar os efeitos causados por insetos (BATISTA et al., 2011). Entretanto, a eficiência dessas ferramentas depende do conhecimento da dinâmica espaço temporal dos mesmos. Desta forma, a caracterização entomológica é fundamental para adquirir informações sobre o comportamento dos mosquitos. Em geral, a prática atualmente utilizada para identificar espécies de insetos é feita principalmente por profissionais treinados através de exame microscópico visual. A Figura 2 representa um esquema de fluxo do processo utilizado para a classificação de mosquitos adultos.



Figura 2. Representação esquemática do processo de classificação tradicional de mosquitos adultos.

É importante destacar que esta atividade consome tempo e requer vários anos de experiência dos profissionais que a executam (PARK et al., 2016; WANG et al., 2012; YANG et al., 2015). Embora o interesse geral em documentar a diversidade de espécies tenha crescido exponencialmente ao longo dos anos, o número de taxonomistas e outros profissionais treinados na identificação de espécies tem diminuído constantemente (KARTHIKA et al., 2018; UTSUGI; TOSHIHIDE; MOTOMI, 2011; WANG et al., 2012). Além disso, fatores importantes podem interferir na identificação correta dos mosquitos, como a

preservação de amostras durante o transporte e o uso de equipamentos adequados para capturar os mosquitos sem causar danos na amostra coletada (MINISTÉRIO DA SAÚDE, 2011). Outra possibilidade para a identificação de espécies é utilizar técnicas moleculares, que foram evidenciadas por diferentes estudos, como códigos de barras de DNA (KUMAR et al., 2007), DNA ambiental (SCHNEIDER et al., 2016) e PCR em tempo real (qPCR) (VOSENBERG et al., 2015). Contudo, apesar da eficiência no processo de identificação de espécies quando empregadas técnicas moleculares, os altos custos de infraestrutura e pessoal treinado limita a utilização desses métodos para a maioria dos laboratórios.

Sistemas para classificação automática têm sido desenvolvidos não apenas para mosquitos, mas também para vespas, traças, aranhas, abelhas e outros (YANG et al., 2015). Apesar do avanço no desenvolvimento desses sistemas, alguns problemas permanecem sem solução. Por exemplo, alguns dos sistemas desenvolvidos apresentam uma boa acurácia na classificação, mas tem pouca funcionalidade, uma vez que o código fonte do algoritmo precisa ser reescrito quando novos dados morfológicos são inseridos (YANG et al., 2015). Um segundo problema, também reportado por Yang et al. (2015), está relacionado à baixa acurácia desses sistemas, por não conseguirem capturar toda a variabilidade de atributos morfológicos dos insetos.

Outra limitação está relacionada à dependência laboratorial desses sistemas e sua dificuldade de capturar dados e aplicar essas ferramentas no ambiente real que vivem esses insetos (BATISTA et al., 2011). Neste sentido, taxonomistas têm buscado métodos eficientes para atender aos requisitos de identificação e classificação de insetos no mundo real (WANG et al., 2012).

Recentemente novos modelos para automatizar o processo de contagem, detecção e classificação de mosquitos foram desenvolvidos. Alguns estudos foram desenvolvidos com o intuito de detectar e classificar as espécies com base na frequência e harmônicos do batimento de asa do mosquito (ARTHUR et al., 2014; KISKIN et al., 2017; OUYANG et al., 2015; SILVA et al., 2013). Técnicas baseadas em análise de padrões de imagem foram utilizadas por outros autores como método de contagem, detecção e classificação (DA SILVA; RODRIGUES;

DE ARAUJO, 2012; KESAVARAJU; DICKSON, 2012; LORENZ; SERGIO; SUESDEK, 2015; MUNDADA; GOHOKAR, 2013; REYES et al., 2016; SANCHEZ-ORTIZ et al., 2017; YAO et al., 2012). Alguns desses pesquisadores desenvolveram métodos de detecção e classificação baseados em *machine learning* e técnicas de *deep learning* (LORENZ; SERGIO; SUESDEK, 2015; OUYANG et al., 2015; SANCHEZ-ORTIZ et al., 2017; SILVA et al., 2013). Contudo, os problemas relacionados às questões do tempo entre a colocação de armadilhas, captura dos mosquitos e a análise do espécime, bem como a necessidade de recursos laboratoriais ainda permanecem sem solução. No Quadro 1 são apresentados alguns trabalhos desenvolvidos para a classificação de insetos utilizando diferentes técnicas.

Quadro 1: Diferentes técnicas desenvolvidas e aplicadas para contagem, detecção e classificação de insetos.

Amostra	Técnica Aplicada	Objetivo	Referência
<i>Ae. aegypti</i>	Processamento de imagens	Contagem de ovos	(DA SILVA; RODRIGUES; DE ARAUJO, 2012)
Mosquitos adultos	Processamento de imagens	Contagem de mosquitos em armadilhas	(KESAVARAJU; DICKSON, 2012)
<i>Lepidoptera</i> (praga de arroz)	Processamento de imagens e Máquina de vetores de suporte (SVM)	Detecção e contagem de pragas em plantação de arroz	(YAO et al., 2012)
<i>Ae. aegypti</i> , <i>Anopheles gambiae</i> , <i>Apis mellifera</i> , <i>Cotinis mutabilis</i> , <i>C. quinquefasciatus</i> , <i>Culex tarsalis</i> , <i>Drosophila melanogaster</i> , <i>Musca domestica</i> e <i>Psychodidae díptera</i> .	Ótica, Acústica e <i>Machine learning</i>	Classificação de mosquitos em armadilhas.	(SILVA et al., 2013)
Pragas de plantas	Processamento de imagens e Máquina	Detecção precoce de pragas nas culturas infectadas em estufa	(MUNDADA; GOHOKAR, 2013)

	de vetores de suporte (SVM)		
<i>Ae. aegypti</i>	Frequência e harmônicos do batimento de asas	Classificação de <i>Ae. aegypti</i>	(ARTHUR et al., 2014)
<i>Anopheles, Aedes, e Culex</i>	Processamento de imagens e Rede neural artificial	Classificação da espécie a partir das características morfológicas das asas	(LORENZ; SERGIO; SUESDEK, 2015)
<i>Ae. albopictus, Ae. aegypti</i> e <i>C. quinquefasciatus</i>	Ótica, Acústica e <i>Machine learning</i>	Classificação de mosquitos	(OUYANG et al., 2015)
<i>Ae. Aegypti</i> e não <i>Ae. Aegypti</i> (ex. <i>Culex</i>)	Processamento de imagens e Máquina de vetores de suporte (SVM)	Detecção de <i>Ae. aegypti</i>	(REYES et al., 2016)
<i>Ae. Aegypti</i> e não <i>Ae. Aegypti</i>	Processamento de imagens e Rede neural convolucional (CNN)	Detecção de <i>Ae. aegypti</i>	(SANCHEZ-ORTIZ et al., 2017)
Mosquitos adultos	Acústica e Rede neural convolucional (CNN)	Detecção de mosquitos	(KISKIN et al., 2017)

Ressalta-se que, a aplicação de inteligência artificial (IA) nos últimos anos tem experimentado um rápido crescimento em diversos setores da economia, como evidenciado pelos relatórios anuais de IA, publicado pela Universidade de Stanford em 2017 (SHOHAM et al., 2017) e em 2018 (SHOHAM et al., 2018). Redes neurais artificiais (ANNs) têm sido utilizadas para identificar organismos com base em suas características morfológicas desde 1988 e são cada vez mais utilizadas para classificar organismos biológicos. Métodos baseados em *deep learning* são essenciais para o reconhecimento geral de objetos (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). No maior concurso para reconhecimento de objetos, o *ImageNet* (ILSVRC) - desafio de reconhecimento visual de grande escala, os algoritmos de *deep learning* consagram-se vencedores desde 2012, quando uma rede convolucional venceu a competição, reduzindo a taxa de erro-

top-5 de 26,1% para 15,3% (GOODFELLOW; BENGIO; COURVILLE, 2016). Redes neurais convolucionais (CNN) constituem uma classe de modelos que utiliza conhecimento prévio para compensar os dados que não estão disponíveis (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Neste contexto, a justificativa deste trabalho baseia-se em três pontos principais: (1) A classificação automática de mosquitos adultos em campo será um forte instrumento epidemiológico para a rápida identificação de foco de doenças transmitidas aos humanos, reduzindo assim o tempo entre coleta e análise das amostras e permitindo sua aplicação sem a necessidade de recursos laboratoriais; (2) O desenvolvimento de uma ferramenta que possa ser aplicada por não-especialistas permitirá o envolvimento da comunidade no alerta às autoridades de saúde sobre a circulação de potenciais vetores de transmissão de doenças, ampliando assim o conhecimento da dinâmica espaço temporal desses vetores; (3) Técnicas baseadas em *deep learning* atualmente apresentam desempenho superior à outras técnicas para a atividade de classificação de objetos. Por se tratar de uma atividade de treinamento de máquina supervisionada, esta técnica requer conhecimento prévio do objeto a ser classificado. Neste sentido, a estruturação de um *dataset* robusto de imagens de mosquitos permitirá não só o aperfeiçoamento das técnicas aqui apresentadas, bem como o seu uso por qualquer novo método, baseado em imagens, que venha a ser desenvolvido no futuro.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste trabalho foi o de investigar e determinar métodos baseados em redes neurais convolucionais (CNN) para realizar a classificação automática em campo, a partir de imagens, de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*, bem como para detectar mosquitos do gênero *Aedes*.

1.2.2 Objetivos Específicos

- Elaborar um estudo prospectivo e de revisão utilizando informações científicas e tecnológicas contidas nas bases de artigos e patentes sobre as diferentes técnicas utilizadas para classificação de insetos e sobre a aplicação de IA no controle de arboviroses.
- Avaliar a viabilidade na aplicação de CNN para classificação de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*, a partir de imagens.
- Desenvolver modelos computacionais, utilizando CNN's, para extração de atributos de imagens e para classificação das mesmas, em uma abordagem supervisionada.
- Realizar a otimização dos hiperparâmetros do modelo para classificação de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*, bem como para detectar mosquitos do gênero *Aedes*.
- Avaliar o desempenho do modelo otimizado na classificação de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus* e na detecção do gênero *Aedes*.
- Avaliar a influência do ângulo da imagem no desempenho do modelo para classificação de mosquitos adultos.

1.3 VALOR AGREGADO DESTA ESTUDO

De acordo com o levantamento técnico e científico, este é o primeiro estudo usando um modelo baseado em CNN's dedicado a extrair características de imagens de mosquitos adultos para classificar automaticamente as espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*.

O banco de dados desse trabalho conta com 7.561 imagens das espécies-alvo do estudo, sendo 95% das imagens de aquisição própria. O desenvolvimento de novas arquiteturas de rede tem sido contínuo e, ano após ano, mais complexas e com melhor desempenho. Independente da arquitetura que se utilize, um

banco de dados é fundamental para o processo de aprendizagem das redes neurais. Dessa forma, o banco de dados estruturado neste trabalho permitirá o desenvolvimento de estudos futuros e aperfeiçoamento do processo de classificação automática de mosquitos adultos.

As CNN's exigem a definição de diversos hiperparâmetros que não são ajustados automaticamente durante o processo de aprendizagem. A otimização desses parâmetros precede a etapa de aprendizagem do modelo e, portanto, se constitui em uma fase fundamental para garantir o melhor desempenho do mesmo. Neste trabalho um processo estruturado de investigação dos melhores valores de hiperparâmetros foi realizado, permitindo assim o desenvolvimento de um modelo otimizado para a classificação automática, a partir de imagens, de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e o *C. quinquefasciatus*. Os pesos da rede treinada podem ser usados como ponto de partida para o processo de transferência de aprendizagem (*transfer learning*) em trabalhos similares de classificação.

O processo de caracterização entomológica tradicional é lento e tem um alto custo. Assim, este trabalho sugere o desenvolvimento de uma ferramenta entomológica para uso em campo por especialistas e não especialistas, visando facilitar a classificação das três espécies estudadas. Desta forma, foi proposto um modelo que permitirá maior eficiência e eficácia nas ações de controle de vetores transmissores de doenças, com o objetivo de reduzir casos de doença e de mortalidade. O resultado deste trabalho fornece evidências de que as CNN's podem ser usadas como técnica para o desenvolvimento dessa ferramenta entomológica e constitui o primeiro passo para construção de uma nova biblioteca com imagens de outros mosquitos transmissores de doenças para os seres humanos, incluindo *Anopheles* e outras espécies entre os *Culicídeos*.

1.4 ESTRUTURA DO TRABALHO

Este trabalho foi organizado no formato de capítulos para uma melhor apresentação e entendimento do mesmo, possuindo no total seis capítulos, conforme descrito a seguir:

O **capítulo 1** constitui a introdução, objetivos, valor agregado deste trabalho e a estrutura do documento apresentado. Neste capítulo o contexto geral que sustenta a importância do desenvolvimento do estudo é discutido, evidenciando as principais contribuições científicas.

O **capítulo 2** apresenta a revisão bibliográfica que é dividida em dois tópicos principais. O primeiro discute a diversidade de métodos utilizados com o objetivo de detectar e classificar espécies de mosquitos em diferentes fases do seu ciclo de vida, e o segundo aprofunda sobre conceitos e evolução de algumas arquiteturas das CNN's ao longo do tempo.

O **capítulo 3** refere-se a um capítulo de livro, publicado online em novembro de 2018 pela IntechOpen, como parte do desenvolvimento dos objetivos específicos desse trabalho. Consiste em um estudo prospectivo e de revisão utilizando informações científicas e tecnológicas contidas nas bases científicas (artigos) e tecnológicas (patentes) sobre as diferentes técnicas utilizadas para classificação de insetos e sobre a aplicação de IA no controle de arboviroses.

O **capítulo 4** apresenta um artigo publicado em janeiro de 2019 na revista PloS ONE, como parte do desenvolvimento dos objetivos específicos desse trabalho. Refere-se à avaliação da viabilidade de aplicação de técnicas de *deep learning* para classificação de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*. Os resultados deste trabalho evidenciaram a viabilidade da técnica baseada em CNN e sustenta o aprofundamento do estudo a partir de um processo estruturado de otimização.

O **capítulo 5** consiste no desenvolvimento do modelo computacional, no processo de otimização dos hiperparâmetros e na avaliação de desempenho do modelo proposto. Neste capítulo são apresentadas as etapas do processo de ampliação do banco de imagens, desenvolvimento do modelo, otimização dos hiperparâmetros e os resultados do processo de otimização para classificação de mosquitos adultos.

Por fim, no **capítulo 6** são apresentadas as considerações finais do trabalho e sugestões para trabalhos futuros.

1.5 REFERÊNCIAS

ARTHUR, B. J. et al. Mosquito (*Aedes aegypti*) flight tones: Frequency, harmonicity, spherical spreading, and phase relationships. **The Journal of the Acoustical Society of America**, v. 135, n. 2, p. 933–941, 2014.

ATTAWAY, D. F. et al. Zika virus: Endemic and epidemic ranges of *Aedes* mosquito transmission. **Journal of Infection and Public Health**, v. 10, n. 1, p. 120–123, 1 jan. 2017.

BATISTA, G. E. A. P. A. et al. Towards automatic classification on flying insects using inexpensive sensors. **Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011**, v. 1, n. December, p. 364–369, 2011.

CAUCHEMEZ, S. et al. Association between Zika virus and microcephaly in French Polynesia, 2013–2015: a retrospective study. **Lancet (London, England)**, v. 387, n. 10033, p. 2125–2132, 21 maio 2016.

CEVALLOS, V. et al. Zika and Chikungunya virus detection in naturally infected *Aedes aegypti* in Ecuador. **Acta Tropica**, v. 177, p. 74–80, 2018.

CONSOLI, R. A. G. B.; OLIVEIRA, R. L. DE. **Principais mosquitos de importância sanitária no Brasil**. [s.l.: s.n.].

DA SILVA, M. G. N. M.; RODRIGUES, M. A. B.; DE ARAUJO, R. E. Sistema de aquisição e processamento de imagens de ovitrampas para o combate a dengue. **Revista Brasileira de Engenharia Biomedica**, v. 28, n. 4, p. 364–374, 2012.

DEL CARPIO-ORANTES, L.; GONZÁLEZ-CLEMENTE, M. DEL C.; LAMOTHE-AGUILAR, T. Zika and its vector mosquitoes in Mexico. **Journal of Asia-Pacific Biodiversity**, v. 11, n. 2, p. 317–319, 2018.

DING, F. et al. Mapping the spatial distribution of *Aedes aegypti* and *Aedes albopictus*. **Acta Tropica**, v. 178, p. 155–162, 2018.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [s.l.] MIT Press, 2016.

GUEDES, D. R. et al. Zika virus replication in the mosquito *Culex quinquefasciatus* in Brazil. **Emerging Microbes & Infections**, v. 6, n. 8, 2017.

GUO, X. et al. *Culex pipiens quinquefasciatus*: a potential vector to transmit Zika virus. **Emerging Microbes & Infections**, v. 5, n. 9, p. e102, 2016.

KARTHIKA, P. et al. DNA barcoding of five Japanese encephalitis mosquito vectors (*Culex fuscocephala*, *Culex gelidus*, *Culex tritaeniorhynchus*, *Culex pseudovishnui* and *Culex vishnui*). **Acta Tropica**, v. 183, p. 84–91, 2018.

KESAVARAJU, B.; DICKSON, S. New Technique to Count Mosquito Adults: Using ImageJ Software to Estimate Number of Mosquito Adults in a Trap. **Journal of the American Mosquito Control Association**, v. 28, n. 4, p. 330–333, 2012.

KISKIN, I. et al. Mosquito Detection with Neural Networks: The Buzz of Deep Learning. **CoRR**, n. May, 2017.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Advances In Neural Information Processing Systems**, p. 1–9, 2012.

KUMAR, N. P. et al. DNA Barcodes Can Distinguish Species of Indian Mosquitoes (Diptera: Culicidae). **Journal of Medical Entomology**, v. 44, n. 1, p. 1–7, 1 jan. 2007.

LETA, S. et al. Global risk mapping for major diseases transmitted by *Aedes aegypti* and *Aedes albopictus*. **International Journal of Infectious Diseases**, v. 67, p. 25–35, 1 fev. 2018.

LORENZ, C.; SERGIO, A.; SUESDEK, L. Artificial Neural Network applied as a methodology of mosquito species identification. **Acta Tropica**, v. 152, p. 165–169, 2015.

MAIN, B. J. et al. Vector competence of *Aedes aegypti*, *Culex tarsalis*, and *Culex quinquefasciatus* from California for Zika virus. **PLOS Neglected Tropical Diseases**, v. 12, n. 6, p. e0006524, 21 jun. 2018.

MINISTÉRIO DA SAÚDE. **Guia de vigilância do *Culex quinquefasciatus***, 2011.

MUNDADA, R. G.; GOHOKAR, V. V. Detection and Classification of Pests in Greenhouse Using Image Processing. v. 5, n. 6, p. 57–63, 2013.

OUYANG, T. H. et al. Mosquito vector monitoring system based on optical wingbeat classification. **Computers and Electronics in Agriculture**, v. 118, p. 47–55, 2015.

PAHO. **Tool for the diagnosis and care of patients with suspected arboviral diseases**. [s.l: s.n.].

PANCETTI, F. G. M. et al. Twenty-eight years of *Aedes albopictus* in Brazil: A rationale to maintain active entomological and epidemiological surveillance. **Revista da Sociedade Brasileira de Medicina Tropical**, v. 48, n. 1, p. 87–89, 2015.

PARK, S. I. et al. Species identification of food contaminating beetles by recognizing patterns in microscopic images of elytra fragments. **PLoS ONE**, v. 11, n. 6, p. 1–22, 2016.

REYES, A. M. M. D. L. et al. **Detection of *Aedes Aegypti* mosquito by digital image processing techniques and support vector machine** 2016 IEEE Region 10 Conference (TENCON), 2016.

SANCHEZ-ORTIZ, A. et al. **Mosquito larva classification method based on convolutional neural networks**. 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP). **Anais...2017**

SCHNEIDER, J. et al. Detection of Invasive Mosquito Vectors Using

Environmental DNA (eDNA) from Water Samples. **PLOS ONE**, v. 11, n. 9, p. e0162493, 14 set. 2016.

SHOHAM, Y. et al. **Artificial Intelligence Index 2017 Annual Report**.

SHOHAM, Y. et al. **Artificial Intelligence Index 2018 Annual Report**.

SILVA, D. F. et al. **Applying machine learning and audio analysis techniques to insect recognition in intelligent traps**. Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013. **Anais...2013**

THANGAMANI, S. et al. Vertical Transmission of Zika Virus in *Aedes aegypti* Mosquitoes. **The American Journal of Tropical Medicine and Hygiene**, v. 95, n. 5, p. 1169–1173, 2 nov. 2016.

UTSUGI, J.; TOSHIHIDE, K.; MOTOMI, I. T. O. Current progress in DNA barcoding and future implications for entomology. **Entomological Science**, v. 14, n. 2, p. 107–124, 24 mar. 2011.

VOSENBERG, B. T. L. H. VAN DE et al. Real-Time PCR Tests in Dutch Exotic Mosquito Surveys; Implementation of *Aedes aegypti* and *Aedes albopictus* Identification Tests, and the Development of Tests for the Identification of *Aedes atropalpus* and *Aedes japonicus japonicus* (Diptera: Culicidae). **Journal of Medical Entomology**, v. 52, n. 3, p. 336–350, 1 maio 2015.

WANG, J. et al. A new automatic identification system of insect images at the order level. **Knowledge-Based Systems**, v. 33, p. 102–110, 2012.

WHO. Global Vector Control Response - Background document to inform deliberations during the 70th session of the World Health Assembly. **WHO**, v. 2030, p. 47, 2017.

YANG, H. P. et al. A tool for developing an automatic insect identification system based on wing outlines. **Scientific Reports**, v. 5, p. 1–11, 2015.

YAO, Q. et al. An Insect Imaging System to Automate Rice Light-Trap Pest Identification. **Journal of Integrative Agriculture**, v. 11, n. 6, p. 978–985, 2012.

2 REVISÃO BIBLIOGRÁFICA

2.1 MÉTODOS PARA CLASSIFICAÇÃO DE MOSQUITOS

A identificação correta das espécies de mosquitos é um passo essencial no desenvolvimento de estratégias eficazes de controle de doenças transmitidas por vetores.

A classificação de mosquitos provê informações imediatas sobre as espécies presentes no ambiente e possibilita a obtenção de outras informações relacionadas a seu estado fisiológico, comportamento e infecção vetorial (MELO-SANTOS, 2008). Recentemente, os métodos mais utilizados para classificação de mosquitos estão baseados em critérios morfológicos ou no uso de biologia molecular (NEBBAK et al., 2016). Estudos demonstram a aplicação de métodos de classificação nas diversas fases do ciclo de vida do mosquito, representados na Figura 1.

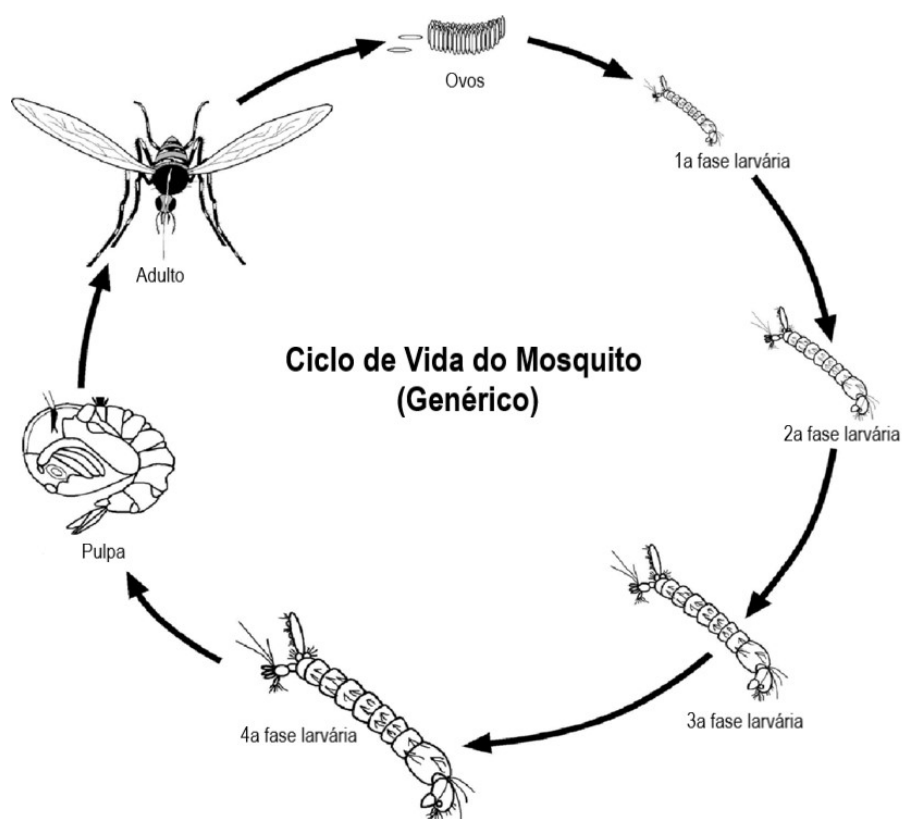


Figura 1. Ciclo de vida do mosquito (genérico).

Fonte: (NTAGEREKA, 2015).

2.1.1 Métodos baseados na observação de ovos

Estudos baseados da observação dos ovos buscam o desenvolvimento de métodos para determinar a densidade populacional dos mosquitos e também para classificação das espécies, a partir de suas características morfológicas.

Com o objetivo de obter a estimativa de fecundidade dos mosquitos, Gaburro et al. (2016) apresentaram um sistema de contagem automática de ovos utilizando uma câmera e um software próprio desenvolvido para contagem. As imagens que representam toda a paleta com ovos depositados em sua superfície são segmentadas em imagens menores tiradas com uma câmera convencional e um estereoscópio. O sistema proposto apresentou resultado semelhante à contagem manual, com uma velocidade três vezes maior, demonstrando assim a sua eficiência.

Para a atividade de classificação, Obenauer, Buss & Kaufman (2009) apresentaram uma proposta de tecnologia para classificação de espécies a partir da melhoria da resolução da imagem de ovos capturados de *Ae. albopictus*, *Aedes triseriatus*, *Orthopodomyia signifera* e *Toxorhynchites rutilus*. Este estudo, contudo, não conseguiu demonstrar a eficácia do método, uma vez que não foi possível gerar uma imagem nítida dos ovos por conta do contínuo movimento destes nas lâminas de coleta e de algumas características como o ressecamento.

No trabalho realizado por Bova, Paulson & Paulson (2016) foram analisadas as características morfológicas entre os ovos das espécies *Aedes japonicus japonicus*, *Ae. triseriatus* e *Ae. albopictus* através de processamento digital de imagens. As figuras dos ovos foram retiradas a partir de câmeras acopladas aos microscópios, e analisadas utilizando um software. Os resultados apontam que a maior parte dos ovos é amendoada, em formato análogo a um charuto. O software utilizado combina diversas imagens em diversos níveis de focos e os autores apontam que o método de classificação permitiu classificar quatro espécies diferentes utilizando apenas um estereoscópio.

2.1.2 Métodos baseados na observação da fase larvária

Outros trabalhos propõem métodos para classificação morfológica na fase larvária. Em seu trabalho, Schaper & Hernández-Chavarría (2006) propuseram um método de classificação morfológica para todos os quatro estágios da fase larvária do mosquito *Ae. aegypti*. O método utiliza um microscópio eletrônico de varredura para avaliação das imagens do segmento abdominal e, a partir delas, identifica cada um dos estágios da larva do mosquito *Ae. aegypti*. Segundo os autores essa classificação é fundamental para compreender a história natural do mosquito, principalmente nos novos territórios afetados.

Em outra aplicação, imagens digitais de larvas de mosquito foram utilizadas em um algoritmo de *machine learning* para identificação de larvas de *Aedes*. Utilizando a CNN AlexNet, Sanchez-Ortiz et al. (2017) desenvolveram um método de identificação de larvas de mosquitos *Aedes* a partir de imagens captadas por celulares. O método de classificação foi dividido nos seguintes passos: 1) Aquisição de imagens; 2) pré-processamento das imagens; 3) treinamento da CNN; e 4) classificação em tempo real. Como resultado, a taxa verdadeiro-positivo para classificação de larvas de *Aedes* foi de 100%, enquanto que a taxa verdadeiro-negativo foi de 70%. Apesar do número baixo de apenas 310 imagens no banco de dados, os autores consideraram o método proposto rápido para identificar larvas de *Aedes*. Contudo, estudos com maior banco de imagens para o treinamento da rede são necessários para dar a confiabilidade ao modelo proposto.

2.1.3 Métodos baseados na observação de mosquitos adultos

Estudos evidenciam que a observação de mosquitos adultos tem como propósito o desenvolvimento de métodos para avaliar a evolução das espécies, a identificação da presença de novas espécies, a avaliação das condições de saneamento básico e a classificação automática de espécies já conhecidas a partir de suas características morfológicas.

Para a avaliação morfológica de mosquitos adultos, alguns autores utilizam chaves dicotômicas, a exemplo das propostas por Consoli & Oliveira (1998) ou

chaves picturais, como a proposta por Huang & Rueda (2015). As chaves dicotômicas/picturais, em sua maioria, existem para classificar as espécies a partir do 4º estágio da fase de larva (SCHAPER; HERNÁNDEZ-CHAVARRÍA, 2006) e não estão disponíveis para todas as espécies já identificadas de insetos (NEBBAK et al., 2016).

Um exemplo prático de chaves picturais foram as utilizadas por Huang & Rueda (2015) para a classificação de oito espécies do subgênero *Zavortinkius*, do gênero *Aedes* presentes na região da África, a partir de fotos do tórax e perna de mosquitos adultos. Os mesmos autores propuseram mais tarde o mesmo processo de chaves picturais para classificação de outras espécies. Para a classificação de nove espécies do subgênero *Diceromyia Theobald*, gênero *Aedes*, presentes na região da África (HUANG; RUEDA, 2016) e para a classificação de oito espécies do subgênero *Finlaya Theobald*, do gênero *Aedes*, também presentes na região da África (HUANG; RUEDA, 2017).

Em alguns casos o estudo morfológico foi realizado com o objetivo de avaliar a evolução das espécies. Vidal, Carvalho & Suesdek (2012), por exemplo, apresentam a evolução da espécie *Ae. albopictus* com o passar dos anos usando o estudo de suas asas como efeito comparativo. Durante quatro anos (2007-2010) mosquitos foram coletados na região de São Paulo – Brasil e a asa esquerda de cada inseto foi analisada utilizando o microscópio estereoscópio. Por estas imagens, 18 pontos de referência foram representados nas interseções das veias da asa. A partir destes pontos foi possível medir as mudanças de distância e tamanho das asas com o passar dos anos, podendo assim, ter o efeito comparativo. Os resultados mostraram que a morfologia da asa do *Ae. albopictus* muda durante períodos menores que um ano. Durante a fase do estudo se tornou perceptível que houve aumento na forma da asa. Outra constatação do estudo é a diferença no padrão de crescimento das asas entre macho e fêmea.

O estudo morfológico é também utilizado com o objetivo de identificar a presença de novas espécies de mosquitos em determinada região. Em 2008, uma espécie de mosquito, não identificado anteriormente, foi encontrada na Bélgica. Com a avaliação morfológica detalhada foi possível analisar e comparar com a

taxonomia do mosquito *Aedes koreicus*, originário da Península coreana. Segundo Pecor et al. (2014), os mosquitos analisados compartilharam muitas características e, portanto, foi possível concluir que se tratavam da mesma espécie. Em outro estudo, Airi & Kaur (2015) avaliaram a taxonomia do *Culex tritaeniorhynchus* (principal transmissor de encefalite japonesa) e do *Culex tritaeniorhynchus summosus* (considerado inicialmente uma subespécie do *C. tritaeniorhynchus*). Amostras de mosquitos adultos das duas espécies foram estudadas e comparadas, evidenciando diferenças morfológicas significativas entre elas. Desta forma, ao invés de subespécies, os resultados comprovaram que se tratava de duas espécies diferentes.

Em alguns estudos realizados, a avaliação de espécies adultas teve como objetivo o monitoramento da fauna e avaliação das condições de saneamento básico em cidades. Santos & Calado (2014) estudaram os *culicídeos* de importância médica encontrados na região do oeste da Bahia, mais precisamente em Barreiras e municípios próximos. Entre fevereiro de 2009 e janeiro de 2010 foram capturados 1.744 *culicídeos*, sendo identificadas diversas espécies de *Aedes*, *Coquilletidia*, *Culex*, *Limatus*, *Mansonia*, *Ochlerotatus*, *Psorophora* e *Uranotaenia*. Baseado na quantidade de *C. quinquefasciatus* encontrados ao longo do período estudado e a quantidade de mosquitos capturados durante as estações mais secas, os autores evidenciaram o impacto da falta de saneamento na região correlacionando-a com o aumento no índice de infestação de mosquitos.

Na busca pela classificação automática de espécies conhecidas, alguns autores utilizam técnicas de processamento de imagem para classificação de mosquitos a partir do estudo de determinadas partes do corpo. Henry et al. (2010) estudam as variações taxonômicas venosas nas asas de mosquitos *Ae. aegypti* e *Ae. albopictus*. Amostras de ambos os sexos para as duas espécies foram coletadas através de armadilhas e suas asas foram fotografadas e digitalizadas para permitir comparações entre seus tamanhos. Utilizando um software desenvolvido para classificação morfológica a partir das imagens, foram encontradas relações fortes entre as espécies dos diversos locais. Os autores indicaram mudanças sutis nos perfis de asas dos insetos devido ao ambiente em que vivem, mas uma conexão forte à herança genética aos padrões de asas.

Técnicas de máquina de vetores de suporte (*Support Vector Machine* - SVM) também foram aplicadas para classificação automática de mosquitos. Reyes et al. (2016) utilizaram o processamento digital de imagens e SVM para detectar o mosquito *Ae. aegypti*. Os autores sugeriram um método de identificação como chave binária de mosquitos a partir da identificação visual de sua morfologia. Uma câmera foi integrada a uma placa de circuito, onde as imagens foram alimentadas em uma máquina de vetores de suporte, correspondendo às características corporais do inseto. Fotos de insetos foram tiradas e, em seguida, entregues à máquina para comparação de dados, onde as propriedades da foto foram avaliadas e, em seguida, equiparadas. Pela construção do equipamento, o sistema só respondeu se o mosquito identificado fosse o *Ae. aegypti* ou não. Para este propósito o sistema apresentou uma taxa verdadeiro positivo acima de 90%.

Ainda com o objetivo de classificar espécies conhecidas, técnicas baseadas na frequência de vibração do batimento de asas dos mosquitos relataram bons resultados em relação à acurácia. De Souza, Silva & Batista (2013) evidenciaram a diferença da frequência do batimento de asas em sete espécies, *Lucidota atra*, *Chauliognathus marginatus*, *Oulema melanopus*, *Drosophila melanogaster*, *C. quinquefascitus*, *Anopheles stephensi*, e *Ae. aegypti*. No entanto, os sensores testados tiveram uma quantidade limitada de memória e não armazenaram todo o fluxo de dados para processamento posterior. Essa mesma técnica foi utilizada para ajudar no reconhecimento de insetos, aplicando *machine learning* para projetar armadilhas inteligentes (SILVA et al., 2013). O dispositivo desenvolvido pelos autores foi capaz de atrair e distinguir os insetos danosos dos benéficos, e dessa forma capturar e matar apenas os insetos-alvo, com o objetivo de fornecer informações para estimar a densidade populacional. Diferentes conjuntos de recursos de análise de áudio e algoritmos de *machine learning* alcançaram acurácia de 98% na classificação de insetos.

Outro exemplo foi o desenvolvimento de um sistema automático de classificação de mosquitos com um dispositivo de gravação por infravermelho para traçar o perfil da frequência das batidas de asas das espécies de mosquitos em voo. Também foi usado um modelo de *machine learning* para classificar o gênero e as espécies dos mosquitos (OUYANG et al., 2015). Para avaliar o desempenho

do sistema, os autores usaram *Ae. albopictus*, machos e fêmeas vivos, *Ae. aegypti* e *C. quinquefasciatus*. Os resultados mostraram que a acurácia do sistema proposto ficou acima de 80% na identificação da espécie e gênero dos mosquitos.

2.2 REDES NEURAIS CONVOLUCIONAIS – CNN

Os conceitos das redes neurais convolucionais foram inspirados pela pesquisa desenvolvida por Hubel & Wiesel (1968), que propuseram uma explicação para a maneira pela qual os mamíferos percebem visualmente o mundo ao seu redor usando uma arquitetura em camadas de neurônios no cérebro. Em 1981, por conta deste estudo, David Hubel e Torsten Wiesel juntamente com Roger Sperry, receberam o Prêmio Nobel de Fisiologia ou Medicina.

Convolução é uma operação em duas funções de um argumento de valor real, tipicamente representada pela equação 1.

$$s(t) = (x * w)(t). \quad (1)$$

Na terminologia de rede convolucional, o primeiro argumento para a convolução é frequentemente chamado de entrada (*input*), nesta equação representado por x e o segundo argumento é o filtro (ou kernel), nesta equação representado por w . A saída (*output*) é às vezes referenciada como o mapa de atributos (*feature map*) (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.2.1 Modelo genérico das CNN's

As CNN's são classificadas como redes de *Deep Learning* e consideradas como uma das mais robustas abordagens para aplicação em visão computacional (GUO et al., 2016b). Em sua arquitetura genérica para extração de atributos incluem uma camada de entrada de dados (*input layer*), diversas camadas intercalando entre convolucionais e de *pooling* e uma camada totalmente conectada (*fully-connected*), como verificado na Figura 2.

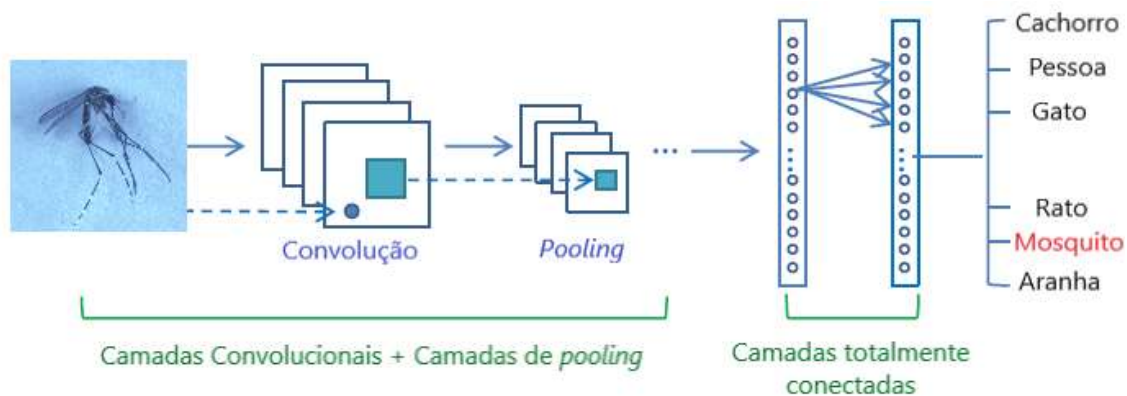


Figura 2. Representação genérica da arquitetura de uma rede neural convolucional (CNN).

Fonte: Adaptado de (GUO et al., 2016b).

Nas camadas convolucionais, a CNN utiliza vários filtros para convolver toda a imagem gerando vários mapas de atributos (*feature maps*), conforme apresentado na Figura 3. As principais vantagens dessa operação são (1) o mecanismo de compartilhamento de peso no mesmo mapa de atributos reduz o número de parâmetros; (2) conectividade local aprende as correlações existentes entre pixels próximos; (3) invariância em relação à localização do objeto (GUO et al., 2016b).

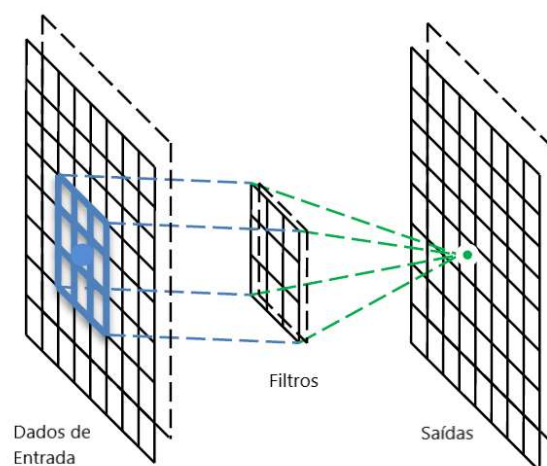


Figura 3. Representação de uma camada convolucional.

Fonte: (GUO et al., 2016b).

A camadas de *pooling* vem normalmente após às de convolução e são utilizadas para reduzir a dimensões dos mapas de atributos (*feature maps*) e dos

parâmetros da rede, tornando a representação invariante a pequenas translações do sinal de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016). Um exemplo de *pooling* máximo (*max pooling*) é apresentado na Figura 4.

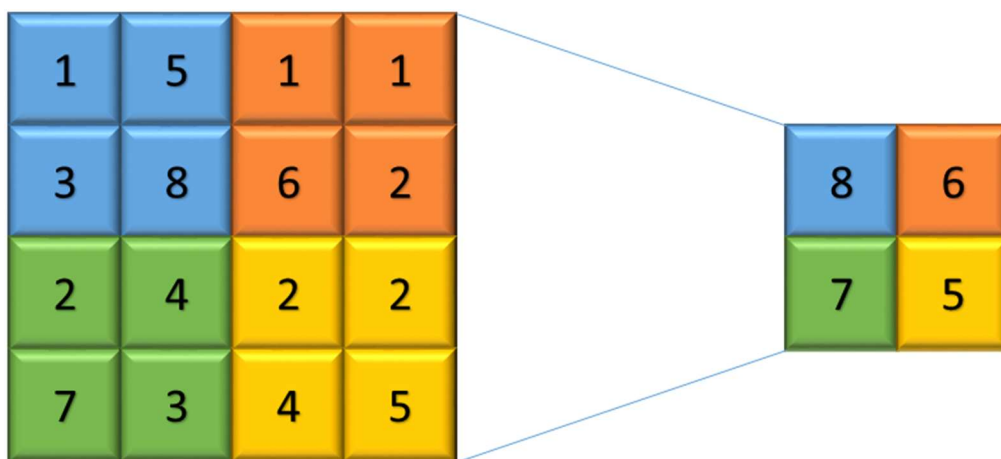


Figura 4. Representação de uma camada de *pooling* máximo.

As estratégias mais comuns utilizadas nas camadas de *pooling* são o *pooling* médio (*average pooling*) e o *pooling* máximo (*max pooling*), que retornam os valores de *pixel* médio do filtro e máximo, respectivamente (GUO et al., 2016b). Uma avaliação de desempenho dessas duas estratégias realizada por Scherer, Muller & Behnke (2010), conclui que o *pooling* máximo é muito superior para capturar invariâncias em dados semelhantes às imagens em comparação com uma operação de *pooling* médio.

Após a última camada de *pooling* vem as camadas totalmente conectadas, convertendo os mapas de atributo 2D em um vetor de atributo 1D. São elas que calculam a pontuação de cada classe dos atributos extraídos de uma camada convolucional nas etapas anteriores e realizam o processo de classificação ou reconhecimento do objeto (ALOM et al., 2018). Essas camadas estão estruturadas de forma semelhante à uma rede neural tradicional e contêm cerca de 90% dos parâmetros de uma CNN (GUO et al., 2016b). Na Figura 5, é apresentada uma representação gráfica das camadas totalmente conectadas.

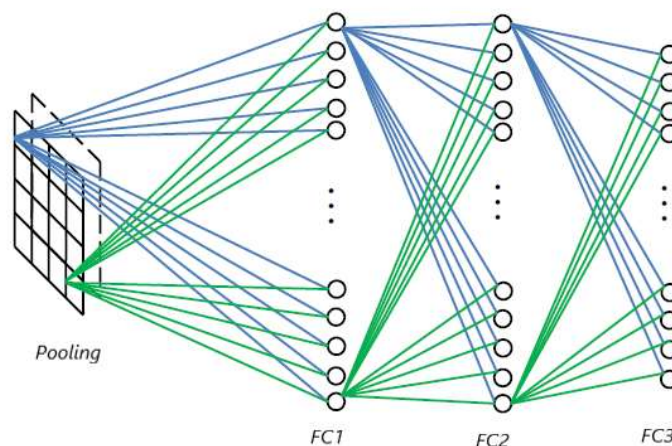


Figura 5. Representação de uma camada totalmente conectada: FC: *Fully-connected*.

Fonte: (GUO et al., 2016b).

Atualmente, uma estratégia utilizada para eliminar o efeito de *overfitting* é a desativação aleatória de alguns neurônios nas camadas de classificação ao longo do treinamento da rede. Essa estratégia é conhecida como *Dropout* e é um método bastante utilizado de regularização em redes neurais convolucionais (WU; GU, 2015). Sua função é zerar a ativação de determinados neurônios de forma estocástica ao longo das épocas de treinamento. O uso do *dropout* tem por objetivo eliminar a co-dependência entre neurônios que pode ser desenvolvida durante o processo de aprendizagem.

2.2.2 Evolução das arquiteturas CNN (um histórico)

Nos anos 80 um modelo de rede neural chamado de *neocognitron* foi utilizado para reconhecimento de padrões visuais de numerais manuscritos (FUKUSHIMA, 1988). Essa foi a primeira vez que uma arquitetura de rede neural convolucional foi proposta. No entanto, este modelo não foi amplamente utilizado devido à limitação de capacidade computacional para treinamento da rede (ALOM et al., 2018).

Nos anos 90, Lecun et al. (1998) aplicaram uma rede convolucional, então chamada de LeNet, no reconhecimento de mais de 60.000 caracteres manuscritos e compararam seus resultados com outras técnicas. Na

comparação em relação à acurácia na classificação, a técnica de máquina suporte a vetor (*Support Vector Machine*) e de aprendizado à distância tangente (*Tangent Distance Learning*) apresentaram desempenho similar à LeNet. Contudo, em relação à demanda de memória computacional, a CNN demandou muito menos memória que as demais. Os autores concluíram que os conceitos apresentados de redes neurais convolucionais eram claramente aplicáveis em diversas situações onde o conhecimento pode ser representado graficamente.

Por um longo período a LeNet foi considerada o estado da arte das CNN's, sendo considerado o primeiro passo para uma real aplicação do método de *deep learning*. Desde então, outras arquiteturas de redes neurais convolucionais têm sido desenvolvidas e testadas. A Figura 6 apresenta uma linha do tempo de algumas arquiteturas comumente utilizadas para resolver questões de reconhecimento de padrões gráficos.

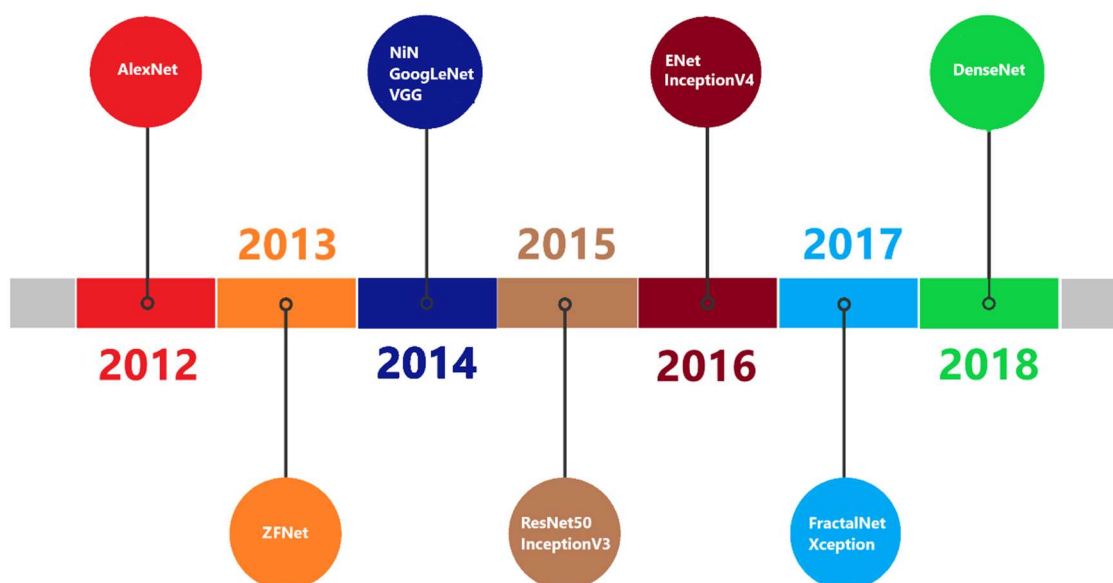


Figura 6. Linha do tempo de arquiteturas de redes neurais convolucionais entre 2012 e 2018.

Fonte: (ALOM et al., 2018; CANZIANI; PASZKE; CULURCIELLO, 2016).

Em 2012, Alex Krizhevsky lançou a AlexNet, uma arquitetura de rede baseada na LeNet com 650 mil neurônios, cinco camadas convolucionais, algumas seguidas de camadas de *pool* máximo, e três camadas totalmente conectadas (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). A AlexNet foi colocada à prova

na competição ILSVRC-2012 - *ImageNet Large-Scale Visual Recognition Challenge* de 2012 - vencendo a competição com uma taxa de erro-top-5 de 15,3%. A rede que obteve a segunda colocação obteve 26,2% de erro-top-5.

A partir de 2012 as tarefas de classificação de imagens e de localização de objeto do ILSVRC foram vencidas por redes neurais convolucionais (RUSSAKOVSKY et al., 2015). A Figura 7 evidencia as redes vencedoras do ILSVRC para classificação de imagens desde 2012. Apesar de ainda não ter sido divulgada a arquitetura da rede vencedora do desafio em 2016, percebe-se que as redes artificiais já superam desde 2015 o desempenho humano nesta atividade. Desde o início, em 2010, deste desafio, houve uma redução de 9,4 vezes a margem de erro na classificação de imagens (de 28,20% para 2,99%), evidenciando o grande progresso que foi feito no reconhecimento de objetos em grande escala nos últimos anos (RUSSAKOVSKY et al., 2015).

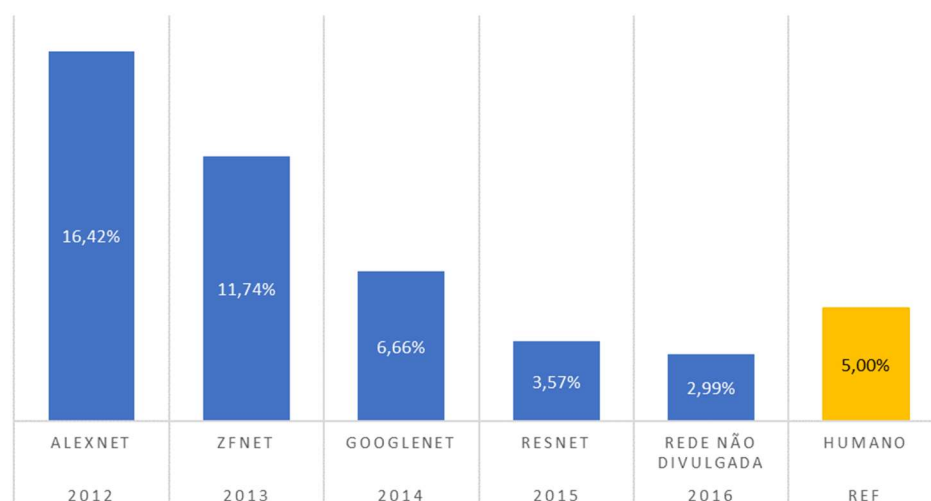


Figura 7. Margem de erro-top-5 das arquiteturas de redes vencedoras do desafio ILSVRC entre 2012 e 2016.

(Em amarelo, para referência, o percentual médio de erro humano)

Fonte: (ALOM et al., 2018; RUSSAKOVSKY et al., 2015) e <http://imagenet.org/challenges/LSVRC/>.

Em 2013, o desafio foi vencido pela rede ZFNet (ou Clarifai). Essa arquitetura foi desenvolvida por Matthew Zeiler e Rob Fergus a partir do modelo da AlexNet (ZEILER; FERGUS, 2014). A principal diferença está na significativa redução nos

números de parâmetros de treinamento da rede que melhorou a acurácia geral na atividade de reconhecimento de padrões (ALOM et al., 2018).

Mais tarde, Lin, Chen & Yan (2014) propuseram um novo modelo de estrutura de rede, chamado de Network in Network (NiN). Este modelo foi construído com micro redes neurais com estruturas mais complexas para abstrair os dados dentro do campo receptivo, diferente das arquiteturas da época, que utilizavam filtros lineares seguidos por uma função de ativação não-linear para digitalizar os dados de entrada. Um segundo conceito importante introduzido pela NiN foi o uso do *Global Average Pooling* (GAP) como uma alternativa de camadas totalmente conectadas para reduzir o número de parâmetros de rede. O GAP altera significativamente a estrutura da rede e ao aplica-lo um vetor de atributo final de baixa dimensionalidade é gerado sem reduzir a dimensão dos mapas de atributos (ALOM et al., 2018). Esta nova abordagem aumentou o poder representacional das redes neurais e serviu de inspiração para outras arquiteturas propostas a partir de então, a exemplo da GoogLeNet (SZEGEDY et al., 2014).

Ainda em 2014, Christian Szegedy, da empresa Google, começou a investigar métodos para reduzir a carga computacional das redes neurais profundas (SZEGEDY et al., 2014). Nesta nova arquitetura, codinome Inception e mais tarde chamada de GoogLeNet, todos os filtros são aprendidos. Além disso, as camadas iniciais são repetidas muitas vezes levando a um modelo profundo de 22 camadas, aumentando a eficiência da rede. A GoogLeNet foi a vencedora do ILSVRC 2014, superando a ZFNet e foi proclamada como o novo paradigma das redes neurais convolucionais.

No desafio do ILSVRC 2014, outra arquitetura obteve o segundo lugar na tarefa de classificação e o primeiro lugar na tarefa de localização de objetos. Foi o time da *Visual Geometry Group* (VGG). A principal contribuição da VGG foi a avaliação completa das redes de maior profundidade usando uma arquitetura com pequenos filtros de convolução (3x3). Essa abordagem representa uma melhoria significativa nas configurações apresentadas anteriormente (SIMONYAN; ZISSERMAN, 2014). Contudo, comparada com outras redes, a VGG é que apresenta os maiores custos operacionais, em termos de número de

operações requeridas a cada época de aprendizado e pelo número de parâmetros da rede (CANZIANI; PASZKE; CULURCIELLO, 2016; SZEGEDY; VANHOUCKE; SHLENS, 2015). A VGG apresenta configurações com números de camadas convolucionais diferentes. A VGG-11 com oito camadas convolucionais, a VGG-16 com 13 camadas convolucionais e a VGG-19 com 16 camadas convolucionais.

He et al. (2015) apresentaram a ResNet (*Residual Network*). O principal desafio deste modelo foi aumentar significativamente a profundidade da rede (número de camadas) sem tornar o treinamento mais difícil. As camadas foram reformuladas como funções de aprendizado residuais com referência aos dados de entrada ao invés de aprender funções não referenciadas. A ResNet apresentou 152 camadas, oito vezes mais profunda que a VGG-19 e sagrou-se campeã do desafio ILSVRC 2015 na categoria de classificação de objetos com a margem de erro-top-5 de 3,57%. A arquitetura mais conhecida é a ResNet50, com 49 camadas convolucionais e uma camada totalmente conectada.

Baseado na arquitetura da GoogLeNet (também conhecida por Inception), Szegedy, Vanhoucke & Shlens (2015) propuseram a arquitetura conhecida como InceptionV3. Nesta configuração a rede tem 42 camadas, o custo operacional é superior à GoogLeNet em 2,5 vezes, mas muito inferior ao da VGG. Na análise de desempenho realizada por Canziani, Paszke & Culurciello (2016), a InceptionV3 obteve melhor acurácia e menor custo operacional que as configurações avaliadas da VGG. A combinação de menor número de parâmetros e a regularização adicional com classificadores auxiliares com *batch-normalizado* permitiram o treinamento de redes de alta qualidade com tamanho modesto de banco de dados (SZEGEDY; VANHOUCKE; SHLENS, 2015).

Em 2016, uma nova arquitetura de rede foi apresentada, a ENet (*efficient neural network*). Esta configuração foi criada especificamente para atividades que requerem operação de baixa latência. O principal objetivo é a segmentação de imagens e aplicações em tempo real que demandem baixo tempo de resposta (PASZKE et al., 2016).

Na busca de uma evolução dos modelos GoogLeNet e InceptionV3, Szegedy et al. (2016) propuseram a arquitetura da InceptionV4 e, baseado nos resultados

alcançados pelo modelo com funções de aprendizado residuais nas camadas, propõem uma combinação das arquiteturas Inception e ResNet. Em seu estudo, os autores concluíram que o modelo que introduziu as ligações residuais aumenta significativamente a velocidade de treinamento comparada com a arquitetura da GoogLeNet. Também determinaram que as novas versões, InceptionV4 e Inception-ResNet-v2, superam o desempenho de todas as redes anteriores avaliadas pelos autores.

Em 2017, o conceito até então de que as funções residuais eram essenciais para a aplicação de redes mais profundas foi questionado por Larsson, Maire & Shakhnarovich (2017). Segundo os autores, a chave pode ser a capacidade de transição, durante o treinamento, entre camadas rasas e profundas. Eles apresentam então a FractaNet, uma rede ultra profunda sem funções residuais e que apresenta desempenho melhor que a ResNet.

A partir da observação de que as camadas da GoogLeNet (Inception) podem ser um passo intermediário entre a convolução regular e a convolução profunda, Chollet (2017) apresentou uma nova arquitetura chamada Xception (*Extreme Inception*). A Xception é baseada inteiramente em camadas de convolução separáveis em profundidade, o que torna a arquitetura fácil de definir e modificar. São necessárias apenas 30 a 40 linhas de código usando uma biblioteca, a exemplo do *Keras* ou *Tensorflow*. Em uma comparação com as redes VGG-16, ResNet-152 e InceptionV3, a Xception apresentou melhor desempenho em relação à acurácia na classificação. Apesar da necessidade de otimização dos parâmetros de treinamento (hiperparâmetros), o autor considera que este conceito pode servir como base para desenvolvimentos futuros.

Huang et al. (2018) apresentaram a DenseNet (*Dense Convolutional Network*). Enquanto nas redes convolucionais as camadas se conectam de maneira subsequente, na DenseNet os mapas de atributos de todas as camadas anteriores são usados como entradas para as camadas subsequentes, conforme verificado na Figura 8. Os resultados evidenciaram que a DenseNet produziu melhorias consistentes na acurácia, sem sinais de *overfitting*. Em comparação com outras arquiteturas, esta alcançou melhor desempenho em diversos

critérios de avaliação. Além disso, a DenseNet demanda menor número de parâmetros e menor capacidade computacional.

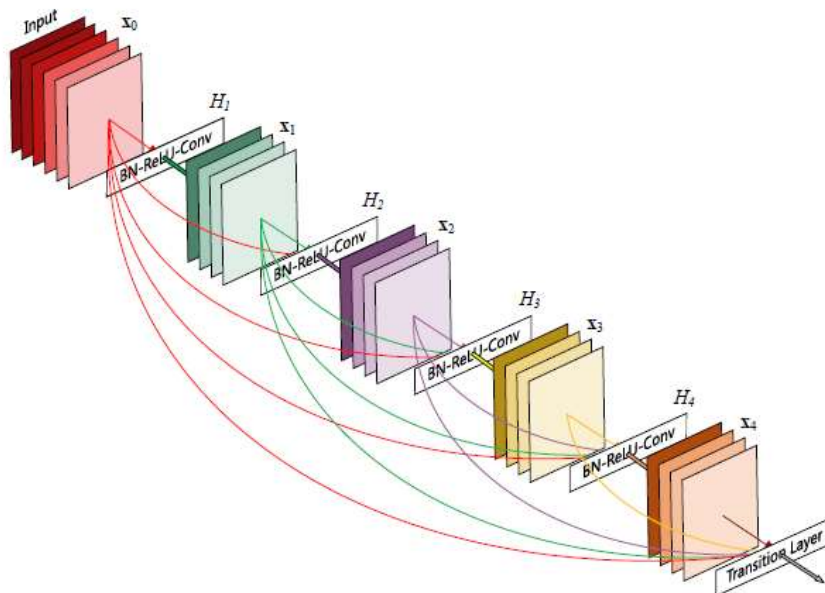


Figura 8. Representação da arquitetura da DenseNet.

Fonte:(HUANG et al., 2018).

O Quadro 1 apresenta a configuração básica de algumas CNNs, evidenciando suas evoluções de arquitetura ao longo dos anos.

Quadro 1: Configuração básica de algumas CNNs.

Ano	CNN	Configuração básica	Referência
1998	LeNet-5	Sete camadas, sem contar a de entrada (duas convolucionais, duas de subamostragem, duas totalmente conectadas e uma conexão Gaussiana).	(LECUN et al., 1998)
2012	AlexNet	Oito camadas, sem contar a de entrada (cinco convolucionais e três totalmente conectadas)	(KRIZHEVSKY; SUTSKEVER; HINTON, 2012)
2013	ZFNet	Oito camadas, sem contar a de entrada (cinco convolucionais e três totalmente conectadas).	(ZEILER; FERGUS, 2014)

		As principais diferenças em relação à AlexNet estão nos filtros e nas formas de conexão usadas entre as camadas.	
2014	NiN	Número flexível de camadas. A arquitetura básica corresponde a três camadas convolucionais MLP (<i>multilayer perceptron</i>) e uma camada GAP (<i>Global Average Pooling</i>), em substituição às camadas totalmente conectadas. Camadas de subamostragem podem ser adicionadas entre as camadas convolucionais.	(LIN; CHEN; YAN, 2014)
2014	GoogLeNet	22 camadas, considerando apenas as camadas com parâmetros de treinamento. Em sua arquitetura inclui camadas convolucionais, <i>max pooling</i> , módulos <i>inception</i> e GAP.	(SZEGEDY et al., 2014)
2014	VGG	Apresenta configurações com 11, 13, 16 e 19 camadas com parâmetros de treinamento. A construção da rede apresenta um número total de 100 camadas. Em sua arquitetura inclui camadas convolucionais, <i>max pooling</i> e totalmente conectadas.	(SIMONYAN; ZISSERMAN, 2014)
2015	ResNet	Apresenta configurações com 18, 34, 50, 101 e 152 camadas com parâmetros de treinamento. Em sua arquitetura inclui camadas convolucionais, <i>max pooling</i> , GAP e totalmente conectadas.	(HE et al., 2015)
2015	Inception-V3	42 camadas, incluindo camadas convolucionais, módulos <i>inception</i> e camadas de <i>pooling</i> .	(SZEGEDY; VANHOUCHE; SHLENS, 2015)
2016	ENet	Apresenta arquitetura básica em dois blocos. No primeiro apresenta camada <i>max pooling</i> e convolucional com 13 filtros. O segundo consiste de três camadas convolucionais. Este segundo bloco é repetido em estágios. A arquitetura termina com uma camada totalmente conectada.	(PASZKE et al., 2016)
2016	Inception-V4	Número flexível de camadas. A arquitetura básica consiste de módulos <i>inception</i> , blocos de redução, GAP e camadas de <i>dropout</i> .	(SZEGEDY et al., 2016)

2017	FractalNet	Apresenta configurações com 20 e 40 camadas com parâmetros de treinamento. Sua arquitetura inclui subcaminhos de diferentes comprimentos, mas não incluem <i>pass-through</i> ou conexões residuais. Cada sinal interno é transformado por um filtro e não-linearidade antes de ser visto por camadas subsequentes.	(LARSSON; MAIRE; SHAKHNAROVICH, 2017)
2017	Xception	Apresenta 36 camadas convolucionais estruturadas em 14 módulos, todos com conexões residuais lineares em torno deles, com exceção do primeiro e último módulos. Sua arquitetura apresenta o mesmo número de parâmetros da Inception V3. O ganho de desempenho não está no aumento de capacidade, mas sim no uso mais eficiente dos parâmetros do modelo.	(CHOLLET, 2017)
2018	DenseNet	Apresenta configurações com 121, 169, 201 e 264 camadas com parâmetros de treinamento, incluindo camadas convolucionais, <i>pooling</i> , blocos densos, camadas de transição, GAP e camadas totalmente conectadas. Os mapas de atributos de todas as camadas anteriores são usados como entradas para as camadas subsequentes.	(HUANG et al., 2018)

2.3 REFERÊNCIAS

AIRI, M.; KAUR, S. Confirmation of *Culex (Culex) tritaeniorhynchus summorosus* (Diptera: Culicidae) as a separate species. **Journal of Vector Borne Diseases**, v. 52, n. 3, p. 219–223, 2015.

ALOM, M. Z. et al. **The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches**. [s.l: s.n.].

BOVA, J. E.; PAULSON, S. L.; PAULSON, G. Morphological differentiation of the eggs of North American container-inhabiting *Aedes* mosquitoes. **Journal of the American Mosquito Control Association**, v. 32, n. 3, p. 244–246, 2016.

CANZIANI, A.; PASZKE, A.; CULURCIELLO, E. An Analysis of Deep Neural Network Models for Practical Applications. p. 1–7, 2016.

CHOLLET, F. **Xception: Deep Learning with Depthwise Separable Convolutions**. [s.l: s.n.].

CONSOLI, R. A. G. B.; OLIVEIRA, R. L. DE. **Principais mosquitos de**

importância sanitária no Brasil. [s.l: s.n.].

DE SOUZA, V. M. A.; SILVA, D. F.; BATISTA, G. E. A. P. A. Classification of data streams applied to insect recognition: Initial results. **Proceedings - 2013 Brazilian Conference on Intelligent Systems, BRACIS 2013**, p. 76–81, 2013.

FUKUSHIMA, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. **Neural Networks**, v. 1, n. 2, p. 119–130, 1 jan. 1988.

GABURRO, J. et al. Assessment of ICount software, a precise and fast egg counting tool for the mosquito vector *Aedes aegypti*. **Parasites & Vectors**, v. 9, n. 1, p. 590, 2016.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [s.l.] MIT Press, 2016.

GUO, Y. et al. Deep learning for visual understanding: A review. **Neurocomputing**, v. 187, p. 27–48, 2016.

HE, K. et al. Deep Residual Learning for Image Recognition. **arXiv:1512.03385v1**, 2015.

HENRY, A. et al. Wing shape of dengue vectors from around the world. **Infection, Genetics and Evolution**, v. 10, n. 2, p. 207–214, 2010.

HUANG, G. et al. Densely Connected Convolutional Networks. **arXiv:1608.06993v5**, 2018.

HUANG, Y. M.; RUEDA, L. M. A pictorial key to the species of the *Aedes* (*Zavortinkius*) in the Afrotropical Region (Diptera: Culicidae). **Zootaxa**, v. 4027, n. 4, p. 593–599, 2015.

HUANG, Y. M.; RUEDA, L. M. A pictorial key to the sections, groups, and species of the *Aedes* (*Diceromyia*) in the Afrotropical Region (Diptera: Culicidae). **Zootaxa**, v. 4079, n. 2, p. 281–290, 2016.

HUANG, Y. M.; RUEDA, L. M. Pictorial keys to the sections, groups, and species of the *Aedes* (*Finlaya*) in the Afrotropical Region (Diptera: Culicidae). **Zootaxa**, v. 4221, n. 1, p. 131–141, 2017.

HUBEL, D. H.; WIESEL, T. N. Receptive Fields and Functional Architecture of Monkey Striate Cortex. **The Journal of Physiology**, v. 195, p. 215–243, 1968.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Advances In Neural Information Processing Systems**, p. 1–9, 2012.

LARSSON, G.; MAIRE, M.; SHAKHNAROVICH, G. FractalNet: Ultra-Deep Neural Networks without residuals. **ICLR**, p. 1–11, 2017.

LECUN, Y. et al. Gradient-Based Learning Applied to Document Recognition. **Proceedings of the IEEE**, p. 46, 1998.

LIN, M.; CHEN, Q.; YAN, S. Network In Network. **arXiv:1312.4400v3**, p.

1–10, 2014.

MELO-SANTOS, M. A. V. DE M. *Aedes aegypti* (DIPTERA:CULICIDAE): Estudos populacionais e estratégias integradas para controle vetorial em municípios da região metropolitana do Recife, no período de 2001 a 2017. p. 218, 2008.

NEBBAK, A. et al. Standardization of sample homogenization for mosquito identification using an innovative proteomic tool based on protein profiling. **Proteomics**, v. 16, n. 24, p. 3148–3160, 2016.

NTAGEREKA, P. B. Mosquito Abundance and Molecular Detection of Arboviruses in Kyela District, Tanzania. n. June, 2015.

OBENAUER, P. J.; BUSS, L. J.; KAUFMAN, P. E. Utilizing Auto-Montage (Tm) Technology for Identifying Field-Collected Container-Inhabiting Mosquito Eggs. **Journal of the American Mosquito Control Association**, v. 25, n. 4, p. 517–520, 2009.

OUYANG, T. H. et al. Mosquito vector monitoring system based on optical wingbeat classification. **Computers and Electronics in Agriculture**, v. 118, p. 47–55, 2015.

PASZKE, A. et al. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. **arXiv: 1606.02147v1**, p. 1–10, 2016.

PECOR, J. et al. Confirmation of *Aedes koreicus* (Diptera: Culicidae) in Belgium and description of morphological ... **Zootaxa**, v. 3191, n. February 2012, p. 21–32, 2014.

REYES, A. M. M. D. L. et al. **Detection of Aedes Aegypti mosquito by digital image processing techniques and support vector machine** 2016 IEEE Region 10 Conference (TENCON), 2016.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, v. 115, n. 3, p. 211–252, 2015.

SANCHEZ-ORTIZ, A. et al. **Mosquito larva classification method based on convolutional neural networks**. 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP). **Anais...2017**

SANTOS, I. M. DOS; CALADO, D. Captura de mosquitos antropofílicos (Diptera, Culicidae) em uma área urbana da região oeste da Bahia, Brasil. **Iheringia. Série Zoologia**, v. 104, n. 1, p. 32–38, 2014.

SCHAPER, S.; HERNÁNDEZ-CHAVARRÍA, F. Scanning electron microscopy of the four larval instars of the Dengue fever vector *Aedes aegypti* (Diptera: Culicidae). **Rev. Biol. Trop. (Int. J. Trop. Biol)**, v. 54, n. 3, p. 847–852, 2006.

SCHERER, D.; MÜLLER, A.; BEHNKE, S. **Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition**. (K. Diamantaras, W. Duch, L. S. Iliadis, Eds.) International Conference on Artificial

Neural Networks (ICANN). **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2010

SILVA, D. F. et al. **Applying machine learning and audio analysis techniques to insect recognition in intelligent traps**. Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013. **Anais...**2013

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. p. 1–14, 2014.

SZEGEDY, C. et al. Going Deeper with Convolutions. **arXiv:1409.4842**, 2014.

SZEGEDY, C. et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. **arXiv:1602.07261v2**, 2016.

SZEGEDY, C.; VANHOUCHE, V.; SHLENS, J. Rethinking the Inception Architecture for Computer Vision. 2015.

VIDAL, P. O.; CARVALHO, E.; SUESDEK, L. Temporal variation of wing geometry in *Aedes albopictus*. **Memorias do Instituto Oswaldo Cruz**, v. 107, n. 8, p. 1030–1034, 2012.

WU, H.; GU, X. Towards dropout training for convolutional neural networks. **Neural Networks**, v. 71, p. 1–10, 2015.

ZEILER, M. D.; FERGUS, R. **Visualizing and Understanding Convolutional Networks BT - Computer Vision – ECCV 2014**. (D. Fleet et al., Eds.)Cham: Springer International Publishing, 2014.

3 CAPÍTULO DE LIVRO

Use of Artificial Intelligence on the Control of Vector-Borne Diseases.

Daniel da Silva Motta¹, Roberto Badaró², Alex Santos¹ and Frank Kirchner³

¹University Center SENAI/CIMATEC, Salvador, Bahia, Brazil, ²Technology Institute of Health – SENAI CIMATEC, Salvador, Bahia, Brazil, ³Research Centre for Artificial Intelligence – DFKI, Bremen, Germany.

Capítulo publicado online pela IntechOpen em 05/11/2018, como parte do livro *Vectors and Vector-Borne Zoonotic Diseases*. (MOTTA et al., 2018).

DOI: 10.5772/intechopen.81671, 2018

Este artigo é de propriedade da IntechOpen.

Copyright: © 2018 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Este capítulo avalia o cenário atual e o potencial de aplicação de técnicas de IA no combate às doenças transmitidas por vetores. Após uma introdução dos conceitos de IA, *machine learning* e *deep learning*, uma pesquisa em base de dados de artigos e patentes é realizada, correlacionando técnicas de IA com as arboviroses. Este levantamento evidencia o crescimento em aproximadamente seis vezes, entre 2000 e 2017, na quantidade de artigos publicados e de 10 vezes no número de patentes de 2017, comparado com a média dos anos anteriores. Também são apresentadas aplicações de como a entomologia pode se beneficiar com as técnicas de IA, bem como os principais desafios e limitações das técnicas de *machine learning* e *deep learning*. Este capítulo responde ao objetivo específico da tese de elaborar um estudo prospectivo e de revisão utilizando informações científicas e tecnológicas contidas nas bases de artigos e patentes sobre as diferentes técnicas utilizadas para classificação de insetos e sobre a aplicação de IA no controle de arboviroses.

Chapter

Use of Artificial Intelligence on the Control of Vector-Borne Diseases

Daniel da Silva Motta, Roberto Badaró, Alex Santos and Frank Kirchner

Abstract

Artificial intelligence has many fields of application with an increasing computational processing power, and the algorithms are reaching human performance on complex tasks. Entomological characterization of insects represents an essential activity to drive actions to control the vector-borne diseases. Identification of the species and sex of insects is essential to map and organize the control measurements by the public health system in most areas where transmission is actively occurring. In many places in the world, the methodology done for identification of the mosquitos is by visual examination from human trained researchers or technicians. This activity is time-consuming and requires several years of experience to have skills to do the job. This chapter addresses the application of artificial intelligence for identification of mosquitos associated with vector-borne diseases. Benefits, limitations, and challenges of the use of artificial intelligence on the control of vector-borne diseases are discussed in this review.

Keywords: artificial intelligence, machine learning, deep learning, mosquitoes classification, vector-borne diseases

1. Introduction

For those who are not familiar with artificial intelligence (AI), imagine that some tasks that are done by humans, such as object detection, visual interpretation, and speech recognition, can be done by computers without human interference. Why is that important? There are many benefits with the use of AI that we intend to discuss in this chapter.

AI is growing in fields that require algorithms (mathematical instructions for computers) and machines to solve problems that are intellectually difficult for human beings but relatively easy for programmable computers. Nevertheless, “the true challenge of AI is to solve tasks that are easy for people to perform, but hard to be described, once it requires intuition [1].” When we look at an image, our interpretation is instantaneous: Is there a car? Is there a person? Is there a house? Computers are able to interpret as well, but not in same way that humans do. Computers translate an image in numbers, as illustratively shown in **Figure 1**.

In the early years of artificial intelligence, a rapid growth has been experienced. “The AI index—2017 annual report, created at Stanford University, presents the volume of activities that involves AI. In this report, indicators help to understand the importance of artificial intelligence technologies for academia, industry, and

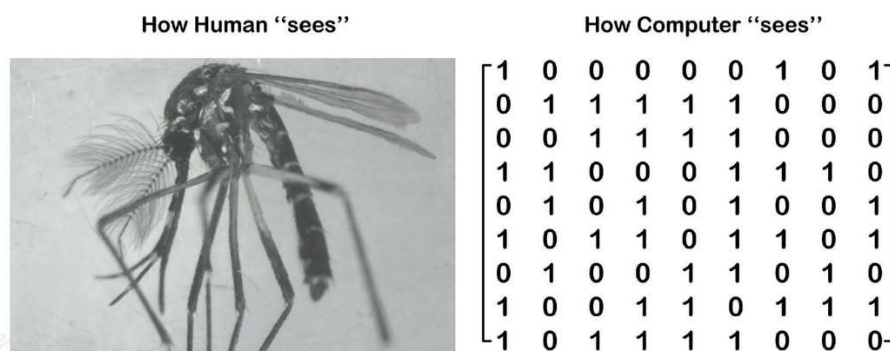


Figure 1.
Human vs. computer: how an image is recognized by each (illustrative).

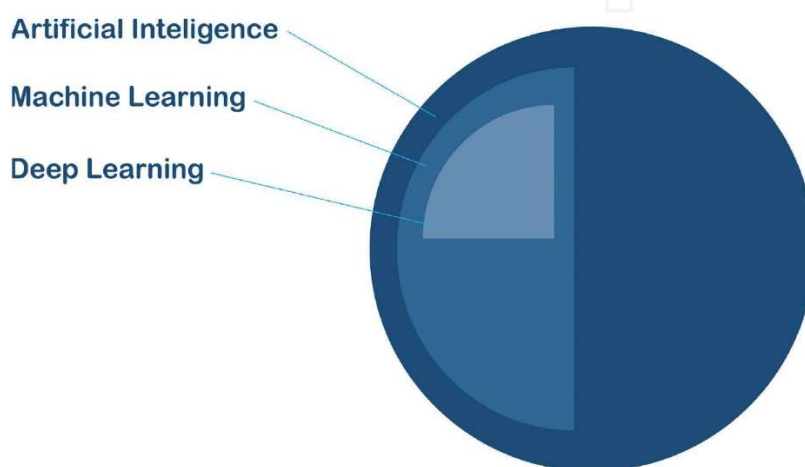


Figure 2.
Relationship between artificial intelligence, machine learning, and deep learning.

public sector. The number of AI published papers produced each year has increased by more than nine times since 1996. For industry, the number of active US startups developing AI systems has increased 14 times since 2000 [2].”

Machine learning (ML) is a subarea of artificial intelligence that is able to learn from previous experience. “ML algorithms are design to solve problems extracting features from existing data, learn from these features and predict the outcomes” [3]. For example, intelligent mosquito’s trap can be designed with the functionality to classify harmful from beneficial insects, release the nontarget insects, and kill the target ones. The classifying process can previously learn from wingbeat frequency data of different species of insects, and whenever a new insect approaches the trap, it will automatically classify and take the decision—release it or kill it. That was exactly what “De Souza and Silva proposed using machine learning techniques” [4, 5].

Recently, “Deep Learning (DL) methods—a subarea of Machine Learning—are considered essential for general object recognition” [6]. “Tasks that consist of mapping an input to an output and that are easy for a person to do rapidly, can be accomplished via Deep Learning, given sufficiently large models and dataset of labeled training examples” [1]. “In the largest contest for object recognition, ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a breakthrough for deep learning occurred in 2012 when a Deep Learning network won the competition, bringing the state-of-art top-5 error rate from 26.1% to 15.3%” [1]. **Figure 2** illustrates how artificial intelligence, machine learning, and deep learning are related.

An important field for application of artificial intelligence is health care. Based on the knowledge of medicine and historical data, AI can be used to support medical doctors to take better and faster decisions. For instance, AI can support medical doctors with robotics systems for some special tasks such as surgery, to increase the life expectancy of human beings, to increase the quality of life for people with some physical disability, and also to increase the community participation to improve the performance of a human care system.

In medicine, arboviruses have received a global attention, since “vector borne diseases are responsible for 17% of the estimated global burden of communicable diseases. It causes more than 700,000 deaths yearly and at least 80% of the global population lives in areas at risk” [7]. Entomology research is considered priority by the World Health Organization for the development of tools that can be applied to reduce incidence and mortality and prevent epidemics due to vector-borne diseases globally.

Identification of the species and sex of mosquitoes is essential to map and organize the control measurements by the public health system in most areas where transmission is actively occurring. In many places in the world, the methodology for identification of the mosquitos is done by visual examination from human trained technician. “This activity is time consuming and requires several years of experience to have skillful to do the job” [8].

This chapter addresses the application of artificial intelligence to help on the control of vector-borne diseases. Research trends and technologies connecting AI to vector-borne diseases are presented for a better understanding on how much researchers and institutions are becoming interested on both topics together. The use of machine learning and deep learning techniques, as a subarea of AI, is discussed for classification of mosquitos in their different life cycle—eggs, larval, pupal, and adult. Benefits and limitations are also presented to help the reader to understand the potential and challenges of artificial intelligence applied to entomology.

2. Research and technological trends on AI and vector-borne diseases

Since 2000, a continuous growth on published research (papers) and patents granted relating artificial intelligence with vector-borne diseases has been noticed. **Figure 3** represents the number of papers published yearly using the keywords: (insect OR mosquito OR culicid OR vector-borne OR zoonotic disease) AND (artificial intelligence OR machine learning OR deep learning). This review considered the following sources: IEEE, PlosOne, Capes, PubMed Web of Science, Current Contents Connect, Conference Proceedings, and Inspec.

It is interesting to notice that in 2017 the number of papers is almost six times it was in 2000. This result demonstrates that artificial intelligence has several possible applications on the control of vector-borne diseases as an important interest topic for many researchers around the world.

Figure 4 shows the number of patents granted in the world with the same keywords as the ones used for review papers. For the patents research, the platform Derwent Innovation was used.

In 2017, the number of patents granted is relevantly almost 10 times the average it was in the previous years. This result demonstrates that not only researchers but also companies have interest in intellectual property assets applying AI on the control of vector-borne diseases.

Figure 5 presents the top countries' and regions' intellectual properties' ownership. China, Korea, and Japan are the countries with more granted patents.

Vectors and Vector Borne Zoonotic Diseases

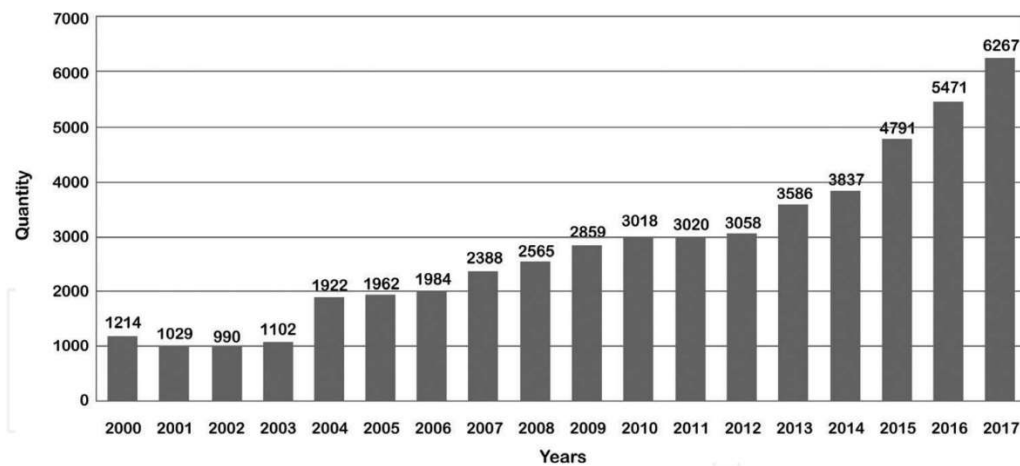


Figure 3.
The number of papers published from 2000 to 2017 relating AI to vector-borne diseases.

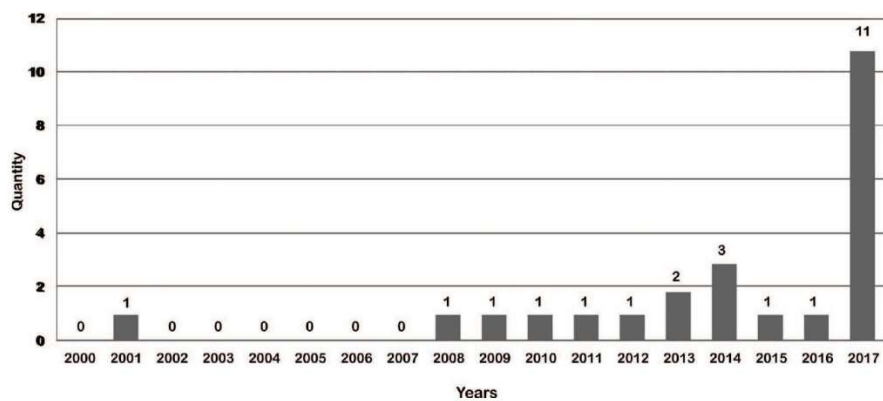


Figure 4.
The number of patents granted from 2000 to 2017 relating AI to vector-borne diseases.

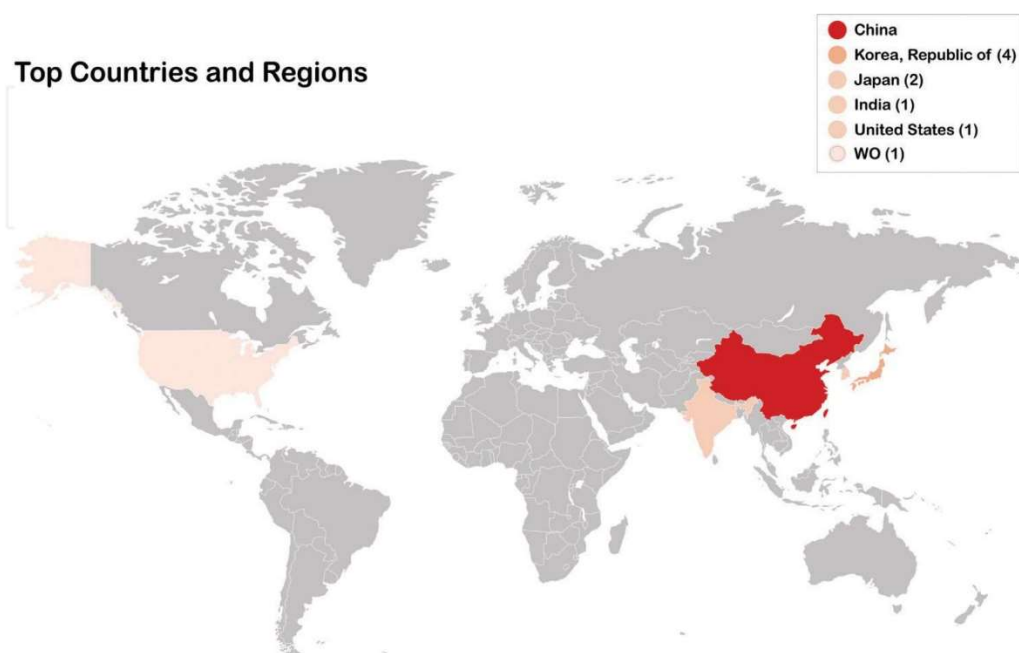


Figure 5.
Countries with more patents from 2000 to 2017 relating AI to vector-borne diseases.

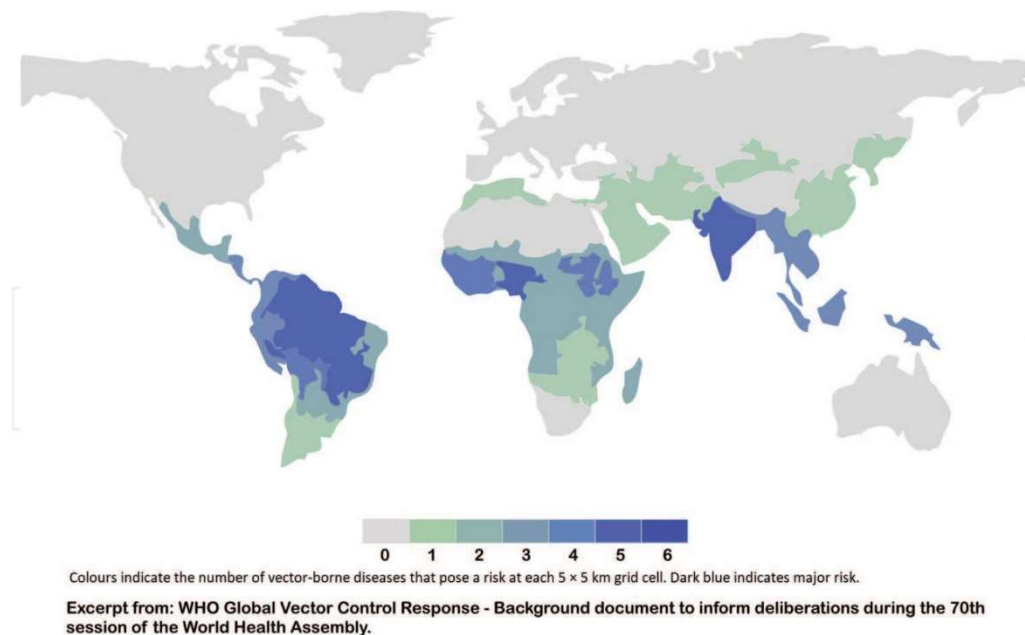


Figure 6.
 Combined global distribution of seven major vector-borne diseases: malaria, lymphatic filariasis, leishmaniasis, dengue, Japanese encephalitis, yellow fever, and Chagas disease transmission [7].

Figure 6 presents the combined global distribution of seven major vector-borne diseases. Correlating **Figures 5** and **6**, some countries that own IP relating AI to vector-borne diseases are not among the main ones that appear in “the global distribution of seven major vector-borne diseases for which integration of vector control programs may be beneficial—malaria, lymphatic filariasis, leishmaniasis, dengue, Japanese encephalitis, yellow fever and Chagas disease transmission—which evidences that vector borne is everyone’s problem [7].”

3. How AI can benefit entomology

In this section, we present some benefits on applying artificial intelligence techniques in areas that are of high importance on the control of vector-borne diseases. Over the last year, much attention is being dedicated to capture and kill harmful mosquitoes using different kinds of mosquito’s traps. Also, several methods have been developed to help mosquito’s species classification process.

A major benefit on the application of AI is to increase the community participation in the control of vector-borne diseases and therefore successfully decrease the burden of arboviruses' recurrent epidemics.

3.1 Mosquito’s trap

There are many mosquito’s traps available to capture and kill mosquitoes. Some of them are dedicated to attract females to deposit its eggs in the trap. Others are designed to capture and kill larva or adult mosquitoes.

Among the studies analyzed, some were dedicated to evaluate the performance of traps of capture of adult mosquitoes. “In [9], an approach is presented to remotely collect and identify field mosquitoes captured by two traps, “BG-trap” and “CDC light.” The motivation of the work is justified considering that the activity of capture and classification requires the presence of entomological specialists and, therefore, faces constraints of budget and logistic feasibility.”

Entomologists recognize that monitoring the traps is crucial to accomplishing its goal. Once the traps attract mosquito's female, if not periodically monitored, it might increase the density of mosquitos in the area the trap is located.

Another issue is the damage caused in the mosquito's body during the capture process. Some samples have its parts destroyed and also dried, what makes difficult the taxonomist's job to evaluate the morphological characteristics of the mosquito's species. **Figure 7** presents an image of *Culex quinquefasciatus* from Fiocruz—Oswaldo Cruz Foundation in Brazil. Some of the morphological characteristics are no longer presented in the sample.

Artificial intelligence can help the design of mosquito's traps by incorporating new important functions. For instance, it helps identify the targeted mosquitoes and separate from the nontargeted ones. Also, using AI, it is possible to acquire and store important information that can help to understand the mosquito's behavior and correlate data such as date and time of capture, species captured, and environmental data (humidity and temperature).

The application of machine learning techniques to design intelligent traps, using a laser sensor, and audio analysis techniques have been used to help insect recognition [5]. The device developed by the authors is able to attract and distinguish harmful from beneficial insects. Also let free the nontarget insects and kill the target ones, which can provide information to estimate the density of the target insect population. Different feature sets from audio analysis and machine learning algorithms achieved 98% accuracy in the insect classification.

Another example was the development of an automatic mosquito classification system consisted of an infrared recording device for profiling the wingbeat of the in-flight mosquito species. Also, a machine learning model was used for classifying the gender, genus, and species of the incoming mosquitoes by the signatures of their wingbeats [10]. To assess the performance of the system, the authors used living male and female *Aedes albopictus*, *Aedes aegypti*, and *Culex quinquefasciatus*. The results show that the accuracies of the proposed system are above 80% on identifying the gender and genus of the mosquitoes.

3.2 Mosquito classification

The correct identification of mosquito species is an essential step in the development of effective control strategies for vector-borne diseases. Ten years prior to



Figure 7.
Image of *Culex quinquefasciatus* from Fiocruz.

the occurrence of Zika virus, dengue, and chikungunya epidemic in Brazil, *Aedes aegypti* mosquito density increased almost 600 times.

Entomological characterization is fundamental to acquire information about mosquito's behavior. This activity requires trained and experienced personnel. "While the general interest in documenting species diversity has grown exponentially over the years, the number of taxonomists and other professionals trained in species identification has steadily declined [11, 12]."

According to Fiocruz, "the traditional method of classifying mosquitoes uses dichotomous keys [13]." These keys consist in analyzing morphological characteristics of the insect. "The dichotomous keys are mostly used to classify species beyond the 4th stage of larval phase" [14]. **Figure 8** represents the classification process using dichotomous keys for three different species—*Aedes aegypti*, *Aedes albopictus*, and *Culex quinquefasciatus*. The dichotomous keys are used to classify any species, not only the represented in **Figure 8** and uses images/figures/drawings to support the taxonomist during classification.

In order to use the dichotomous keys, the taxonomist needs to prepare the sample—if it is an adult, assemble the mosquito on entomological pin and observe the specimen under the microscope to evaluate the morphological characters. **Figure 9** represents the process of entomological characterization of an adult mosquito.

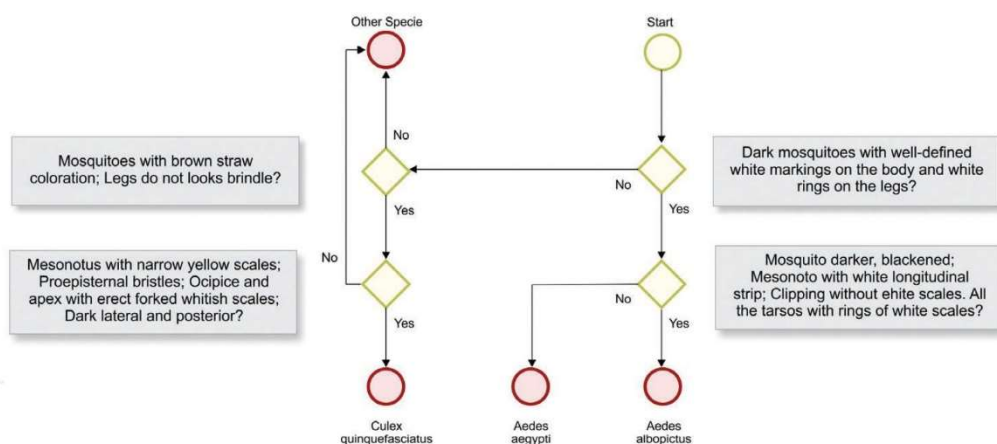


Figure 8. Representation of the process using dichotomous keys for the classification of mosquitoes.

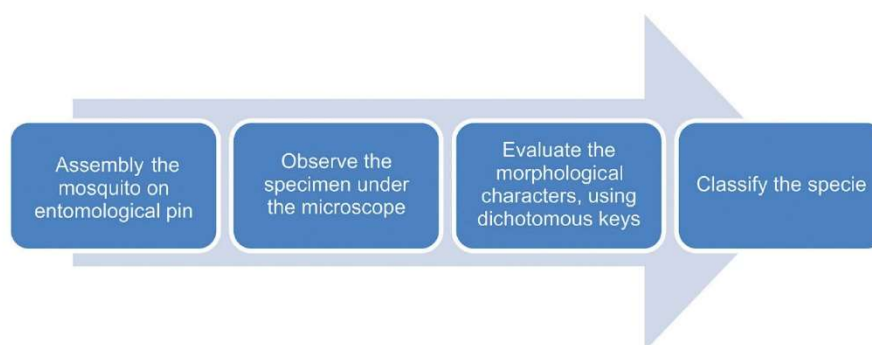


Figure 9. Process of entomological characterization of an adult mosquito.

As already mentioned, some of the mosquito's samples are damaged and lose morphological characteristics during the capture in the field and the transport to a laboratory. Besides that, the waiting time during capture and transportation is also an issue and might dry the mosquito's body, which affect some characteristics such as color.

Another possibility for the "identification of species can be made through the use of molecular techniques that have been shown in different studies such DNA barcodes" [15]. Furthermore, molecular identification of mosquito remains a slow and expensive process for most laboratories.

Artificial intelligence can be applied to automatize the mosquito's classification process. It can be used to classify in field by entomologists or even nontaxonomists and health workers. By doing that, AI can avoid the major issues presented previously, like the need of trained and experienced personnel and lose of the morphological characteristics. Artificial intelligence application also allows increasing the number of mosquito's data, obtaining online information of population density, and the correlation with cases of incidence and mortality of vector-borne diseases.

In one AI application, deep learning was used to recognize *Aedes*-utilized wings morphology. "In [16], 17 species of the genera *Anopheles*, *Aedes*, and *Culex* were classified based on wing shape characteristics to test the hypothesis that classification using Artificial Intelligence was better than traditional classification method by discriminant analysis. The results demonstrated the AI correctly classified species more efficiently with an accuracy of 86%–100%."

Some authors study support vector machine (SVM) techniques. "In [17], the authors use digital image processing and support vector machine (SVM) to detect *Aedes aegypti* mosquito. It is suggested for a method of identification as binary key of mosquitoes from the visual identification of their morphology. A camera is integrated with a circuit board, where images are fed to a support vector machine, corresponding to body characteristics of the insect. Photos of insects are taken and then delivered to the machine for data comparison, where photo properties are valued and then matched. By the construction of the equipment, the system only responds if the identified mosquito is *Aedes aegypti* or not, to which it has an accuracy of 90% in the data."

In other applications, mosquito's larva digital images were used in a machine learning algorithm for *Aedes* larva identification. "The authors proposed a method to identify larvae of *Aedes* mosquitoes using convolutional neural networks (CNN), a new method in multilayer neural network technology that has proven its performance especially in image analysis. Larva's images were captured by cell phones. The classification method is divided into the following steps: 1) acquisition of images; 2) preprocessing the images; 3) CNN training; 4) Real-time classification. The results shown a good performance with 100% accuracy for identification of *Aedes* larva, however, for other mosquitoes the misclassification rate was 30% [18]." Although the sample size in this study was very small, it shows that artificial intelligence can be used for the mosquito's species classification.

4. Limitations and challenges on the application of ML and DL

Applications of machine learning and deep learning techniques in many areas are rapidly growing, due to the flexibility of their algorithms and also because it is not required to model previously the scenario using a mathematical function. Prototypes and computer systems are being developed, but there are still some

bottlenecks to overcome. Although machine learning and deep learning algorithms are capable of capturing the complexity of several problems, in some cases the effective use of it depends on further research and development to increase the level of reliability before it can be used in the real world.

In this section, we present some limitation and challenges on the application of artificial intelligence, especially machine learning and deep learning techniques, which should be addressed in future researches.

4.1 Generic approach

An algorithm does not interpret a problem the same way that humans do. It needs a mathematical equation to build a scenario that represents the reality. The mathematical equation is a representation of the reality and usually simplifies the problem to be solved, due to that incorporates mistakes and has limitations to be generalized. Because of it, the application of machine learning and deep learning techniques to control vector-borne diseases must be designed and/or trained for this specific purpose. There is no such generic approach: each problem has its own specificity and therefore must be treated with exclusivity.

4.2 Robust dataset

Another important limitation of machine learning and deep learning is the need of historical data to be used for algorithm training, learn from these data, and predict a reliable outcome. The availability, disposal, and variability of these existing data are crucial for the computer learning process. “Objects in realistic settings exhibit considerable variability, so to learn to recognize them, it is necessary to use much larger training sets [6].”

Entomology, for instance, has small dataset size available open source, which turns to be difficult to adapt the model and solve the problem with proper accuracy and reliability. Researchers should be aware that the application of machine learning and deep learning for zoonotic diseases must consider the building of a robust dataset.

4.3 Underfitting and overfitting

Underfitting and overfitting also need to be addressed during the use of machine learning and deep learning techniques. The first one relies when small data are presented in the training or the training does not run a sufficient number of epochs (learning cycles). In this case, the mathematical model is unable to capture the features complexity of the input provided and present a high error level in the output—too many wrong predictions when new data are presented.

Overfitting relies when the data presented have small variability or the training learning cycles are too much, and instead of reducing the error after each epoch, it starts to increase. To clarify the understanding, imagine a student who, among the elementary arithmetic operations (addition, subtraction, division, and multiplication), only dominates multiplication. If a test is presented only with questions to multiply numbers, probably the student will have a good grade, but you cannot measure his/her knowledge with this test. That exactly what happens with the computer if the variability of data is low. The training result might present a high accuracy, but in the real world, it is not reliable.

Figure 10 graphically shows underfitting and overfitting—validation error represents the predicting error when new data are presented.

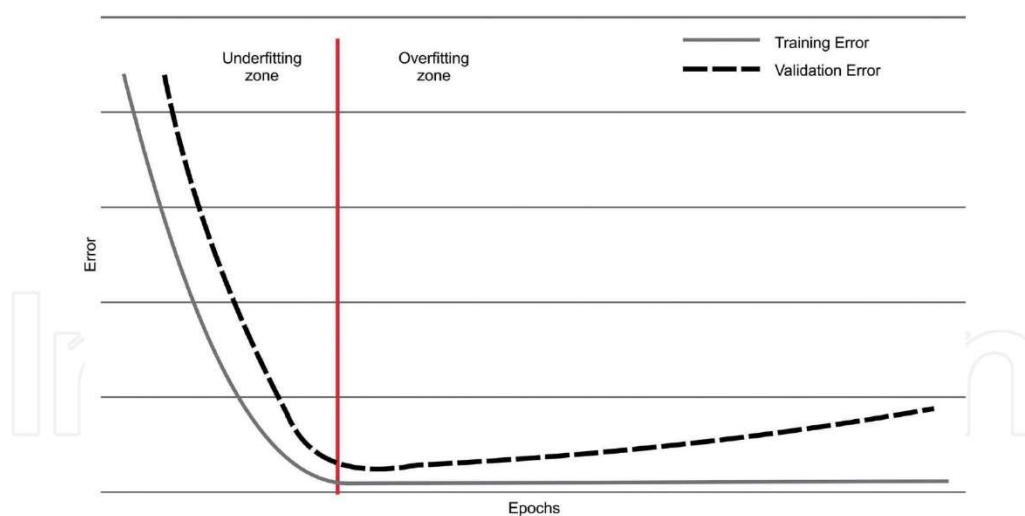


Figure 10. Representation of underfitting and overfitting. “Adapted from [1].”

There are some methods to reduce overfitting. “The easiest and most common method is to artificially enlarge the dataset [6].”

5. Conclusions

Novel and important applications are available with the development of data mining methods. Artificial intelligence techniques are an important field to be applied on the control of vector-borne diseases. A complete and accurate identification of the 5000 mosquito’s species that were already identified should be tested in this model as well as other species groups, such as complex or cryptic species, and in different populations of the same species.

Artificial intelligence could help to develop a system that anyone, who capture larvae or adult’s mosquitos in several regions, can identify the *Aedes* mosquito. In the near future, a complete identification of any insect or new nonclassified ones that exist in this world could be automatically classified by anyone using a smart-phone. AI will never replace mankind but will help to keep memories and activities that humans have discovered in our millenarian existence.

Acknowledgements

This project received financial support from Fundação de Amparo à Pesquisa do Estado da Bahia—FAPESB via scholarships. We are thankful to Daniel André Dias Imperial Pereira and Alexandre Morais Cavalcanti, students at University Center SENAI/CIMATEC. We also thank Eduardo Oyama, entomologist from the Technology Institute of Health—SENAI CIMATEC, for supporting the work and sharing his experience. We are also in debt with the Department of Culicidae collection from Fiocruz Rio de Janeiro, especially Maycon Neves and Monique Motta from their staff.

Conflict of interest

We have no “conflict of interest.”

Use of Artificial Intelligence on the Control of Vector-Borne Diseases
DOI: <http://dx.doi.org/10.5772/intechopen.81671>

Appendices and nomenclature

AI	artificial intelligence
ML	machine learning
DL	deep learning
ILSVRC	ImageNet large scale visual recognition challenge
WO	PCI patents (world)
CNN	convolutional neural networks
SVM	support vector machine

Author details

Daniel da Silva Motta^{1*}, Roberto Badaró², Alex Santos¹ and Frank Kirchner³

1 University Center SENAI/CIMATEC, Salvador, Bahia, Brazil

2 Technology Institute of Health—SENAI CIMATEC, Salvador, Bahia, Brazil

3 Research Centre for Artificial Intelligence—DFKI, Bremen, Germany

*Address all correspondence to: dmotta@fieb.org.br

IntechOpen

© 2018 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Goodfellow I, Bengio Y, Courville A. Deep Learning [Internet]. Cambridge, Massachusetts, USA: MIT Press; 2016. Available from: <http://www.deeplearningbook.org>
- [2] Yoav S, Perrault R, Brynjolfsson E, Jack C, Legassick C. Artificial Intelligence Index, 2017 Annual Report [Internet]. 2017. Available from: <http://aiindex.org/2017-report.pdf>
- [3] Kajaree D, Behera R. A survey on machine learning: Concept, algorithms and applications. International Journal of Electronics Communication and Computer Engineering. 2017;5: 1302-1309. DOI: 10.15680/IJIRCCE.2017
- [4] De Souza VMA, Silva DF, Batista GEAPA. Classification of data streams applied to insect recognition: Initial results. In: Proc—2013 Brazilian Conf Intell Syst, BRACIS 2013. 2013. pp. 76-81. DOI: 10.1109/BRACIS.2013.21
- [5] Silva DF, De Souza VMA, Batista GEAPA, Keogh E, Ellis DPW. Applying machine learning and audio analysis techniques to insect recognition in intelligent traps. In: Proceedings—2013 12th International Conference on Machine Learning and Applications, ICMLA 2013. 2013. pp. 99-104. DOI: 10.1109/ICMLA.2013.24
- [6] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012;25(2):1-9. DOI: 10.1016/j.protcy.2014.09.007
- [7] WHO. Global Vector Control Response 2017-2030—Background Document to Inform Deliberations during the 70th Session of the World Health Assembly. WHO. 2017. p. 47
- [8] Park SI, Bisgin H, Ding H, Semey HG, Langley DA, Tong W, et al. Species identification of food contaminating beetles by recognizing patterns in microscopic images of elytra fragments. PLoS One. 2016;11:1-22. DOI: 10.1371/journal.pone.0157940
- [9] Pombi M, Guelbeogo WM, Calzetta M, Sagnon N, Petrarca V, La Gioia V, et al. Evaluation of a protocol for remote identification of mosquito vector species reveals BG-sentinel trap as an efficient tool for *Anopheles gambiae* outdoor collection in Burkina Faso. Malaria Journal. 2015;14:161. DOI: 10.1186/s12936-015-0674-7
- [10] Ouyang TH, Yang EC, Jiang JA, Lin TT. Mosquito vector monitoring system based on optical wingbeat classification. Computers and Electronics in Agriculture. 2015;118:47-55. DOI: 10.1016/j.compag.2015.08.021
- [11] Utsugi J, Toshihide K, Motomi ITO. Current progress in DNA barcoding and future implications for entomology. Entomological Science. 2011;14:107-124. DOI: 10.1111/j.1479-8298.2011.00449.x
- [12] Karthika P, Vadivalagan C, Thirumurugan D, Kumar RR, Murugan K, Canale A, et al. DNA barcoding of five Japanese encephalitis mosquito vectors (*Culex fuscocephala*, *Culex gelidus*, *Culex tritaeniorhynchus*, *Culex pseudovishnui* and *Culex vishnui*). Acta Tropica. 2018;183:84-91. DOI: 10.1016/j.actatropica.2018.04.006
- [13] Consoli RAGB, de Oliveira RL. Principais mosquitos de importância sanitária no Brasil. Rio de Janeiro, Brasil: Fundação Oswaldo Cruz; 1998. DOI: 10.1590/S0102-311X1995000100027
- [14] Schaper S, Hernández-Chavarría F. Scanning electron microscopy of the four larval instars of the Dengue

Use of Artificial Intelligence on the Control of Vector-Borne Diseases
DOI: <http://dx.doi.org/10.5772/intechopen.81671>

fever vector *Aedes aegypti* (Diptera: Culicidae). *Revista de Biología Tropical*. 2006;54:847-852

[15] Kumar NP, Rajavel AR, Natarajan R, Jambulingam P. DNA barcodes can distinguish species of Indian mosquitoes (Diptera: Culicidae). *Journal of Medical Entomology*. 2007;44:1-7. DOI: 10.1603/0022-2585(2007)44[1:DBCDSO]2.0.CO;2

[16] Lorenz C, Sergio A, Suesdek L. Artificial neural network applied as a methodology of mosquito species identification. *Acta Tropica*. 2015;152:165-169. DOI: 10.1016/j.actatropica.2015.09.011

[17] Reyes AMMDL, Reyes ACA, Torres JL, Padilla DA, Villaverde J. Detection of *Aedes Aegypti* mosquito by digital image processing techniques and support vector machine. In: 2016 IEEE Region 10 Conference (TENCON). 2016. pp. 2342-2345. DOI: 10.1109/TENCON.2016.7848448

[18] Sanchez-Ortiz A, Fierro-Radilla A, Arista-Jalife A, Cedillo-Hernandez M, Nakano-Miyatake M, Robles-Camarillo D, et al. Mosquito larva classification method based on convolutional neural networks. In: 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP). 2017. pp. 1-6. DOI: 10.1109/CONIELECOMP.2017.7891835

4 ARTIGO I

Application of Convolutional Neural Networks for Classification of Adult Mosquitoes in the Field.

Daniel Motta¹, Alex Álisson Bandeira Santos¹, Ingrid Winkler¹, Bruna Aparecida Souza Machado^{1,2}, Daniel André Dias Imperial Pereira¹, Alexandre Morais Cavalcanti¹, Eduardo Oyama Lins Fonseca², Frank Kirchner³, Roberto Badaró^{1,2}

¹University Center SENAI/CIMATEC, National Service of Industrial Learning –SENAI, 41650-010, Salvador, Bahia, Brazil; ²Health Institute of Technologies (CIMATEC ITS), National Service of Industrial Learning – SENAI, 41650-010, Salvador, Bahia, Brazil, ³Research Centre for Artificial Intelligence, DFKI, 28359, Bremen, Germany.

Artigo publicado em ***PloS ONE*** 14(1): e0210829, 2019. (MOTTA et al., 2019).

DOI: 10.1371/journal.pone.0210829.

Este artigo é de propriedade da PloS ONE.

*Copyright: © 2019 Motta et al. This is an open access article distributed under the terms of the **Creative Commons Attribution License** (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

Este capítulo do trabalho avalia a viabilidade do uso de técnicas baseadas em CNN para classificação automática de mosquitos adultos a partir de imagens de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*. Com um banco de dados que incluiu 4.056 imagens de mosquitos, três redes neurais, LeNet, AlexNet e GoogleNet, foram usadas. Durante a fase de validação, a acurácia da classificação foi de 57,5% usando a rede LeNet, 74,7% usando a AlexNet e 83,9% usando a GoogleNet. Durante a fase de testes, o melhor resultado (76,2%) foi obtido usando a GoogleNet; os

resultados de 52,4% e 51,2% foram obtidos usando LeNet e AlexNet, respectivamente.

Os resultados obtidos neste artigo evidenciaram a viabilidade de aplicação e sugeriram um estudo mais aprofundado para aumento da acurácia com o uso de arquiteturas mais complexas de CNN. Este capítulo responde ao objetivo específico da tese de avaliar a viabilidade na aplicação de CNN para classificação de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*, a partir de imagens.

RESEARCH ARTICLE

Application of convolutional neural networks for classification of adult mosquitoes in the field

Daniel Motta¹✉, Alex Álisson Bandeira Santos^{1‡}, Ingrid Winkler^{1‡}, Bruna Aparecida Souza Machado^{1,2}✉*, Daniel André Dias Imperial Pereira¹✉, Alexandre Morais Cavalcanti¹✉, Eduardo Oyama Lins Fonseca^{2‡}, Frank Kirchner^{3‡}, Roberto Badaró^{1,2}✉

1 University Center SENAI CIMATEC, National Service of Industrial Learning–SENAI, Salvador, Bahia, Brazil, **2** Health Institute of Technologies (CIMATEC ITS), National Service of Industrial Learning–SENAI, Salvador, Bahia, Brazil, **3** Research Centre for Artificial Intelligence, DFKI, Bremen, Germany

✉ These authors contributed equally to this work.

‡ These authors also contributed equally to this work.

* brunam@fieb.org.br



OPEN ACCESS

Citation: Motta D, Santos AÁB, Winkler I, Machado BAS, Pereira DADI, Cavalcanti AM, et al. (2019) Application of convolutional neural networks for classification of adult mosquitoes in the field. *PLoS ONE* 14(1): e0210829. <https://doi.org/10.1371/journal.pone.0210829>

Editor: Humberto Lanz-Mendoza, Instituto Nacional de Salud Pública, MEXICO

Received: September 3, 2018

Accepted: January 2, 2019

Published: January 14, 2019

Copyright: © 2019 Motta et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript.

Funding: The study was unfunded. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Dengue, chikungunya and Zika are arboviruses transmitted by mosquitos of the genus *Aedes* and have caused several outbreaks in world over the past ten years. Morphological identification of mosquitos is currently restricted due to the small number of adequately trained professionals. We implemented a computational model based on a convolutional neural network (CNN) to extract features from mosquito images to identify adult mosquitoes from the species *Aedes aegypti*, *Aedes albopictus* and *Culex quinquefasciatus*. To train the CNN to perform automatic morphological classification of mosquitoes, we used a dataset that included 4,056 mosquito images. Three neural networks, including LeNet, AlexNet and GoogleNet, were used. During the validation phase, the accuracy of the mosquito classification was 57.5% using LeNet, 74.7% using AlexNet and 83.9% using GoogleNet. During the testing phase, the best result (76.2%) was obtained using GoogleNet; results of 52.4% and 51.2% were obtained using LeNet and AlexNet, respectively. Significantly, accuracies of 100% and 90% were achieved for the classification of *Aedes* and *Culex*, respectively. A classification accuracy of 82% was achieved for *Aedes* females. Our results provide information that is fundamental for the automatic morphological classification of adult mosquito species in field. The use of CNN's is an important method for autonomous identification and is a valuable and accessible resource for health workers and taxonomists for the identification of some insects that can transmit infectious agents to humans.

Introduction

Arthropod-borne viruses are responsible for more than 100 of the diseases that comprise the estimated global burden of communicable human diseases[1]. Vector-borne diseases cause

more than one billion infections and more than one million deaths in humans every year[2]. Dengue, chikungunya and Zika are the most common arboviruses and have caused several major epidemics in France Polynesia and Latin America during the past 10 years[3–6]. These three viral diseases are among the greatest public health challenges in the world[7]. Entomology research is considered to be a priority by the World Health Organization to develop tools that can be applied to reduce incidence and mortality and prevent epidemics caused by vector-borne diseases worldwide[2].

Aedes aegypti and *Aedes albopictus* have received worldwide attention since both species are efficient vectors for the transmission of human arboviral diseases, such as Zika, dengue, chikungunya, and yellow fever[3,8,9]. Additionally, controversial studies have also indicated that the genus *Culex* could be a possible vector of the Zika virus[10–12]. *Ae. aegypti*, *Ae. albopictus* and *C. quinquefasciatus* are very common domiciliary mosquito vectors that are present in almost all urban areas in Brazil[7]. *Ae. aegypti*, *Ae. Albopictus* are, among all *Aedes* species, the ones that circulate in endemic areas. Furthermore, *Aedes* mosquitoes have been demonstrated to be capable of transmitting Zika and other viruses in many tropical areas in the Americas, Africa, and Asia[13].

Entomological characterization is fundamental for acquiring information about mosquito behaviour. In general, the current procedure used to identify the species of an insect requires an individual visual examination of the insect, which is time consuming and requires several years of experience[14,15]. While the general interest in documenting the diversity of insect species has grown exponentially over the years, the number of taxonomists and other professionals trained in species identification has steadily declined[16–18]. Important factors that can impede the correct identification of mosquito species include the method of preservation used during the transport of samples and the use of the appropriate equipment to capture mosquitoes without damaging them[19]. Due to that, developing a tool to allow the classification of adult mosquitoes in the field, considering the environmental issues, would be very important[20].

Another possibility for species identification is the use of molecular techniques that have been validated by various studies, such DNA barcoding, environmental DNA testing and real-time PCR (qPCR)[21–23]. However, molecular identification of mosquitoes is a slow and expensive process for most laboratories.

Recently, new models that facilitate the automatic classification of mosquitoes have been developed. Some studies have attempted to classify mosquito species based on the frequency and harmonics of their wingbeats[24,25]. Techniques based on image feature analysis have also been used as a classification method [26–29]. In addition, Machine Learning and Deep Learning techniques have been used for mosquito classification[30–32]. Most of these studies, however, aimed to develop methods and tools that requires a laboratory environment. The development of a tool to support taxonomist and health workers in the field would be very helpful in accelerating the knowledge of which mosquito is circulating in the community and supporting health authorities in controlling harmful mosquitoes, once it will reduce the lag between the time the trap is placed and the taxonomic inspection occurs[33].

Wingbeat vibration techniques have been reported to demonstrate a high accuracy in the classification of mosquitoes[32]. However, the sensors used have a limited amount of memory and are unable to store the entire data stream for later processing. Indeed, the sensors must process the data stream in real time to identify events of interest and filter out background noise[34].

Within the field of image feature analysis, studies have been conducted to identify mosquitoes according to their life cycle stage (egg, larval phase, pupal stage and adult). Some studies have attempted to automatize the process of egg counting to assess the level of fecundity and

thereby estimate the mosquito population size and conduct morphometric analysis[35–38]. Studies conducted to classify mosquito-based image in adult phase still focus on parts of the body of the mosquito. The outlines of body parts, such as wings, are stable and diverse but have not been frequently used in conventional taxonomy due to difficulties with lexical descriptions[15]. A classification method capable of analysing the features of the whole body is essential, since in the field there is no equipment that allow the health worker to separate parts of the body and analyse it.

Deep learning methods are essential for the processes underlying general object recognition [39]. In 2012, during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which was the largest contest conducted in the object recognition field, a breakthrough in deep learning occurred when a convolutional network won the competition by reducing the state-of-the-art top-5 error rate from 26.1% to 15.3%[40]. Convolutional neural networks (CNN) constitute a class of models that utilize prior knowledge to compensate for data that is not available[39]. A CNN method has been developed to classify *Aedes* mosquito larva using a small dataset; when 200 epochs were used, the network achieved 96.8% accuracy[31].

It should be emphasized that correct identification of adult *Aedes* mosquito species is essential for both the recognition of the vectors involved in disease transmission and the development of fast and efficient control strategies[15]. In this study, we compared three CNN's, LeNet, AlexNet and GoogLeNet to evaluate the potential application of a Deep Learning technique to classify adult mosquito species in the field. The objective is to develop an epidemiological tool to be used in the real-world environment to facilitate the work of entomologists and health workers in the classification of adult mosquitoes of the species *Ae. aegypti*, *Ae. albopictus* and *C. quinquefasciatus*. As showed, this tool would be very useful to reduce the time, to allow its use by less experienced expert and to capture preserved features of the body characteristics of the mosquito. Another objective of this work is to initiate a preliminary scientific study that would allow the community be part of the control of vector-borne diseases.

Considering that the selection of algorithm training parameters is essential to improve the accuracy and reliability of the method, this study also applied the statistical analysis prior to the selection to determine the parameters set to be used to classify the mosquitoes.

Materials and methods

Ethics statement

No permits were required for sampling for this study. The field sampling did not involve any endangered or protected species.

Sample collection

The mosquito samples used for image capture were obtained from the Parasitology Laboratory of the Federal University of Bahia–UFBA (Salvador, Brazil) and were also collected in the field. The samples obtained from UFBA were *Ae. aegypti* mosquitos (10 females and 6 males). The capture of adult insects in the field resulted in the collection of 73 specimens of *Ae. aegypti*, 94 specimens of *Ae. albopictus* and 110 specimens of *C. quinquefasciatus*.

The field sampling took place between September and October 2017 in the city of Salvador in two collection areas (Bahia, Brazil). CDC light traps and suction tubes were used for the collection of adult insects.

The captured specimens were euthanized with ethyl acetate and stored in entomological collection tubes until identification was performed by an entomologist.

Construction of the dataset: Structure, acquisition and distribution

A robust, correctly structured dataset was created to allow the differentiation of the *Ae. aegypti*, *Ae. albopictus* and *C. quinquefasciatus* species. In addition, differentiation of males and females was conducted as described in a previous study[41]. A model was developed to classify the mosquitoes into six different classes according to gender and species. At this stage, our work does not intend to classify other species and non-mosquitoes images.

The images in the dataset were extracted from the ImageNet platform and were also photographed with various cameras, including a Leica DMC2900 (Leica Microsystems, Heerbrugg, Switzerland) coupled to a stereoscopic Leica M205C at the Oswaldo Cruz Institute of Entomology at FIOCRUZ (Rio de Janeiro–Brazil), a Canon Power Shot D30 (Canon, Tokyo, Japan) coupled to a Wild M3C stereomicroscope (Leica Microsystems, Heerbrugg, Switzerland) at SENAI CIMATEC (Salvador–Brazil) and Samsung J5 (Samsung, Seoul, South Korea) and Apple iPhone 7 (Apple, Cupertino, California, USA) mobile phone cameras. The images were collected at different resolutions and levels of quality to develop a classification method that utilized a wide range of images to prevent possible overfitting of the trained model. An average of 10 photographs of each specimen were taken at different angles and proximities.

Once the dataset was structured and the images acquired, the number of images to be used for training, validating and testing the model was determined. Fig 1 summarizes how the dataset was used in the computational model and shows the three different phases of the development of the model: training (phase 1), validation (phase 2) and testing (phase 3).

Determination of the application images

The application images were those that we intended use for the classification of mosquitoes after the model was developed. This was done to allow the model to be used by any person, not only a specialist using a microscope, who is able to photograph an insect and then run the model to classify mosquitoes.

A total of 823 out of the 4,056 images used in this study were manually selected according to the following characteristics: the image should show the entire insect body (to visualize all relevant morphological features) and should have been taken with a mobile phone. Fig 2 shows some captured images of male and female *Ae. aegypti* photographed using a digital camera (Fig 2A, 2B, 2C and 2D) and mobile phone (Fig 2E and 2F).

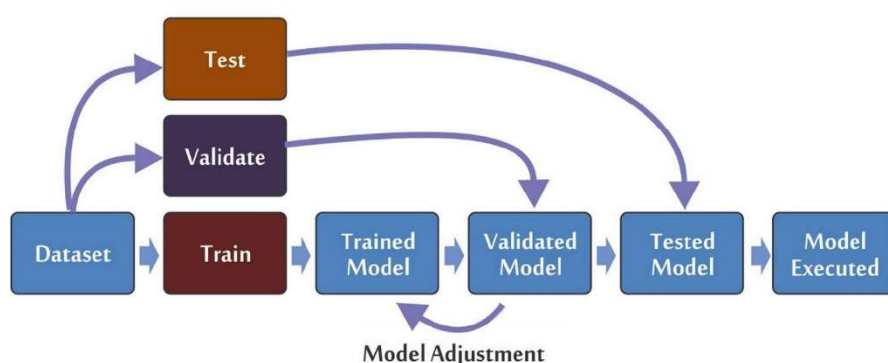


Fig 1. Representation of the training (phase 1), validating (phase 2) and testing (phase 3) processes of the mosquito classification model.

<https://doi.org/10.1371/journal.pone.0210829.g001>

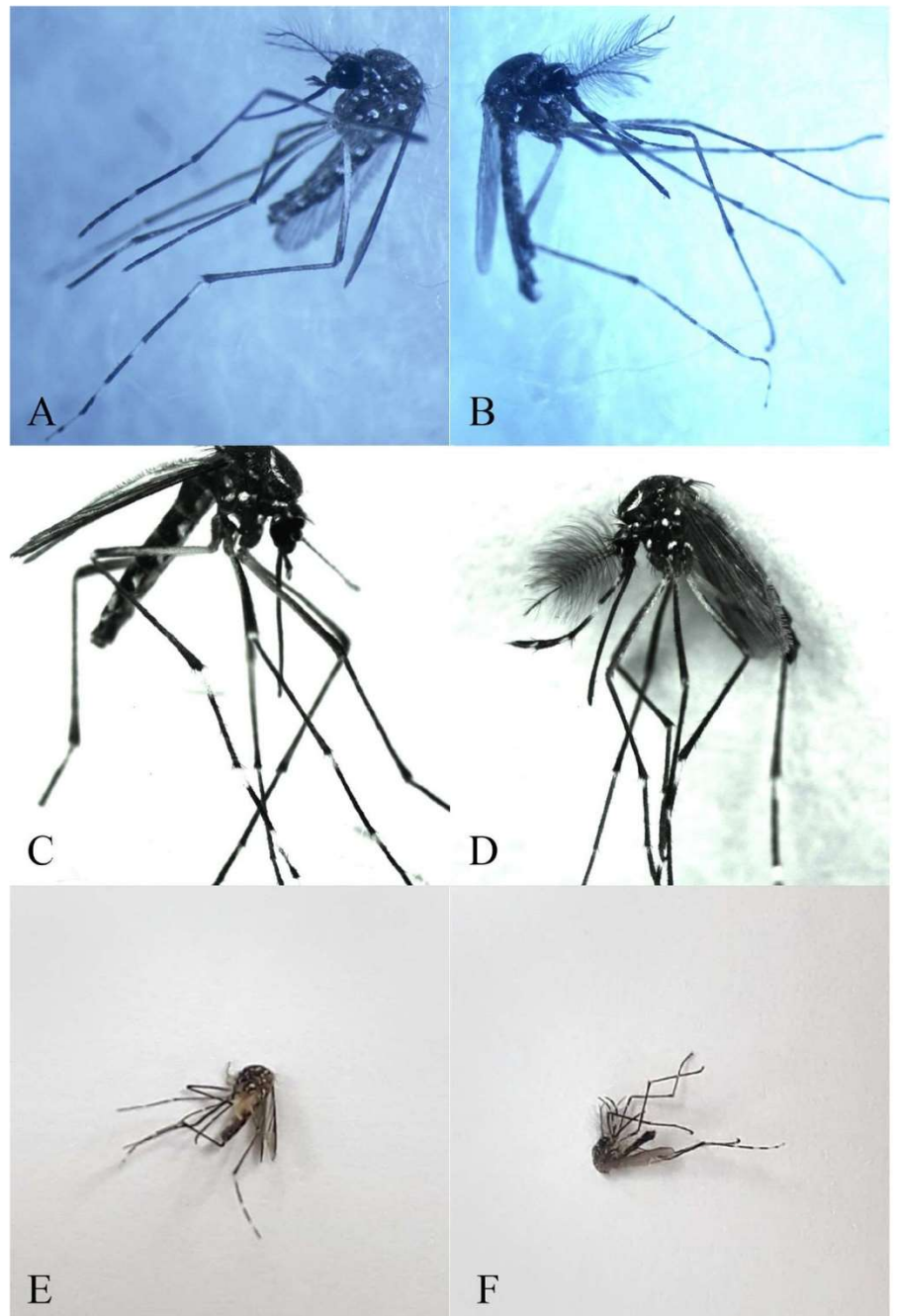


Fig 2. Images used for the validation phase during the development of the model used in this study. (A and C) *Ae. aegypti* females photographed using a digital camera; (B and D) *Ae. aegypti* males photographed using a digital camera; (E) an *Ae. aegypti* female photographed using a mobile phone; (F) an *Ae. aegypti* male photographed using a mobile phone.

<https://doi.org/10.1371/journal.pone.0210829.g002>

Determination of the application images to be used for testing

For the testing phase (phase 3), images were extracted from the dataset and used only after the model was validated. Additionally, the images used for testing were not used to train or validate the model. This was done to avoid research bias and minimize the chance of overfitting.

To standardize the analysis of the results, the test dataset contained the same number of images from each of the six classes of mosquitoes. It was decided that the number of images corresponding to 20% of the number of images in the smallest class would be used for testing for all classes. As the *C. quinquefasciatus* female class was the class with the smallest number of application images (70 images), fourteen images (20%) from each of the other classes were randomly removed from the dataset. Python scripts were employed to randomize the chosen images.

Distribution of the application images used for the training and validation phases

Within the references used in this work, no consensus was found concerning the ideal dataset distribution between the training (phase 1), validation (phase 2) and testing (phase 3) phases.

To determine the best image distribution for training and validation, once the test sample size was already defined, the percentage of application images used in the training phase was defined as a factor, varying in four levels (either 30%, 40%, 50% or 60%), in a full factorial experiment. The objective was to evaluate statistically the best distribution of application images between the training and validation phases that would present the best classification accuracy for the intended application in entomology.

Selection of neural networks

We used nVidia DIGITS software (nVidia, Santa Clara, California, USA) with a NVIDIA GeForce GTX TITAN GPU for processing. We used the software on the Linux Ubuntu operating system LTS Distribution 16.04 (Linux, San Francisco, California, USA). The Caffe framework was used for the LeNet[42,43], AlexNet[31] and GoogLeNet convolutional neural networks [42,43].

For a long period, LeNet was considered the state of the art of Artificial Neural Networks (ANN's). LeNet was one of the first ANN's used to improve the original backpropagation algorithm; the remarkable development of this well-known network was the first step towards the application of the Deep Learning method.

In 2012, Alex Krizhevsky released AlexNet, which was a deeper and much wider version of the LeNet that won, by a large margin, the difficult ImageNet competition[39]. AlexNet has a very similar architecture to that of LeNet, but it can better describe images. This network relies on eight layers, including convolutional, local responses, max-pooling and fully connected layers.

In 2014, Christian Szegedy, from Google, began to investigate methods to reduce the computational burden of deep neural networks[43]. In doing so, he focused on the efficiency of the architecture of the deep neural network (codenamed *Inception*, later named *GoogLeNet*). This neural network was the winner of ILSVRC 2014, surpassing AlexNet, and was proclaimed the new paradigm for convolutional neural networks.

Classification of the model parameters

The CNN algorithms requires the definition of several parameters prior to the training phase. Depending on the application, these parameters might have more or less influence on the classification results. Prior random tests were performed to understand which set parameters have

more influence and should be statistically analysed as a factor in the full factorial experiment in order to establish the best value. Other parameters with less influence on the results were set as default in the algorithm.

Number of epochs and Seed were parameters defined as a constant value. The prior random tests showed that 200 epochs are enough to stabilize the training and validation phases. In case it generates overfitting it would be easily identified. Define the seed as a constant value was important to standardize the neural network weights initiation for all CNN and also, for each time we run the algorithm. The parameters evaluated in the full factorial experiment were: percentage of the application image used in the training phase; solver algorithm; the learning rate (LR); and learning rate decay function. The other parameters, such as the batch size, batch accumulation, learning rate step size, learning rate power, learning rate gamma, crop size and mean subtraction, were set according to pre-defined default for each network.

Full factorial experiments

To understand the influence of each parameter and their correlations with one another, full factorial experiments were conducted using the Minitab 17 software (Minitab, State College, Pennsylvania, USA). The parameters used as the inputs for the full factorial experiments are shown in Table 1.

Using the Minitab software, we created an experiment with four factors: the percentage of application images used at training, solver algorithm, learning rate, and LR decay function. The number of epochs and the seed was fixed. For the experiments, we defined the number of levels as shown in Table 1. During the next step, the formats of the selected parameters were defined as either numerical or text; the percentage of application images used at training and the LR were defined as numerical, and the solver algorithm and LR decay function were defined as text. The values for each level was defined based on the random experiments performed before.

Based on the result of the factorial experiments, it was calculated that 192 experiments should be conducted on one of the three convolutional neural networks. The output variables to be evaluated were the percent accuracy during the validation phase, loss function and percent accuracy during the testing phase.

Full factorial experiments analysis

Based on the full factorial experiments, GoogLeNet was the network selected for conducting the 192 planned experiments.

The first step in the analysis was to interpret the statistical significance of the parameters that were used (percent accuracy during the validation phase, loss function and percent accuracy during the testing phase) and their relationship with the results. The 'P value' was

Table 1. Parameters used as inputs for the full factorial experiments in this study.

Parameters	Levels	Level			
		1	2	3	4
Number of epochs	1	200			
Seed	1	10			
% application images used at training	4	30	40	50	60
Solver algorithm	4	SGD	NAG	Adam	AdaGrad
Learning rate	3	0.001	0.01	0.1	
LR decay function	4	Step Down	Exponential	Sigmoid	Polynomial

<https://doi.org/10.1371/journal.pone.0210829.t001>

calculated to evaluate the statistical significance of the parameters and the effects of their interactions on the outcome of the experiments.

The best performance obtained for each outcome was then defined as the target value for optimization analysis and was used to determine the best combination of parameters. These combinations were used to train the other two convolutional neural networks, LeNet and AlexNet.

Results

The species and gender of each of the dataset classes used to classify the 4,056 images are shown in Table 2.

Fig 3 shows the effects of each parameter (1st level analysis) on the accuracy of the testing phase (phase 3). In this analysis, the use of the solver algorithm '3' (Adam) and a learning rate of '3' (0.1) significantly reduced the accuracy of the mosquito image classification. However, a learning rate of '1' (0.001) was an important parameter that was used ('p value' < 0.00001). Almost every parameter had a significant influence on the results of the performance of the trained neural network ('p value' < 0.05), with the exception of the interaction between '% application images (train) * Solver' and '% application images (train) * LR Decay Function' ('p value' > 0.05).

Fig 4 shows the effects of the parameter combination (2nd level analysis) on the interaction of the parameters with one another. It is clear that a learning rate of '3' (0.1) reduced the accuracy of the testing phase when combined with any of the other parameters. Consequently, this learning rate should not be used subsequent analysis. It was also confirmed that the solver algorithm '3' (Adam) reduced the accuracy of the testing phase. However, the combination of a learning rate of '2' (0.01) with solver algorithm '4' (AdaGrad) obtained better results than the combination of a learning rate of '1' (0.001) with the same solver algorithm ('p value' < 0.00001). Overall, the learning rate of '1' (0.001) produced the best results.

To define the best set of parameters, an optimization function was calculated in Minitab that set the target values for each outcome. Fig 5 shows the variation of the data obtained for each outcome. The best performance in terms of the percentage accuracy during validation was 83.9%, with a loss function of 0.67 and percentage accuracy during testing of 86.9%.

The best parameter combination was obtained in the 4-2-1-3 set, which resulted in the use of 60% of the application images during training (the NAG Solver algorithm; 0.001 LR and Sigmoid LR decay function). Therefore, this parameter set was used for the subsequent training of all three convolutional neural networks.

Fig 6 shows the results obtained during the training and validation phases for the convolutional neural networks LeNet, AlexNet and GoogLeNet.

Table 2. Dataset structure and sample sizes of the mosquito images used in this study.

Class name	Total number of images	Non-application images (used in training)	Application images	Application images used for training and validation	Application images used for testing
<i>Ae. aegypti</i> female	947	723	224	210	14
<i>Ae. aegypti</i> male	282	201	81	67	14
<i>Ae. albopictus</i> female	1050	821	229	215	14
<i>Ae. albopictus</i> male	436	327	109	95	14
<i>C. quinquefasciatus</i> female	965	895	70	56	14
<i>C. quinquefasciatus</i> male	376	266	110	96	14

<https://doi.org/10.1371/journal.pone.0210829.t002>

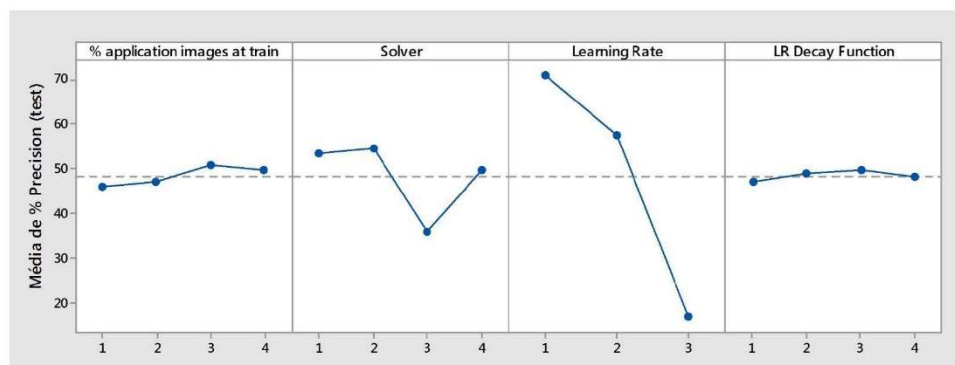


Fig 3. Effects of the first level parameters on the accuracy of the testing phase (average accuracy percentage).

<https://doi.org/10.1371/journal.pone.0210829.g003>

The LeNet neural network achieved an accuracy of 57.5% for the classification of mosquitoes (Fig 6A). Interestingly, the loss function (2.07) significantly increased during the training of the network, indicating possible overfitting. Table 3 presents the accuracy achieved by this network for each mosquito species. The overall performance of the LeNet neural network during the testing phase (phase 3) was 52.4%.

Fig 6B shows the results obtained during the training and validation phases for the AlexNet neural network. Using the previously selected parameters (200 epochs; seed of 10; 60% of application images used for training; NAG solver algorithm; 0.001 LR; sigmoid LR decay function), the network achieved an accuracy of 74.7%, and the validation and training losses were reduced by approximately 0.83 and 0.11, respectively. Table 4 presents the overall accuracy

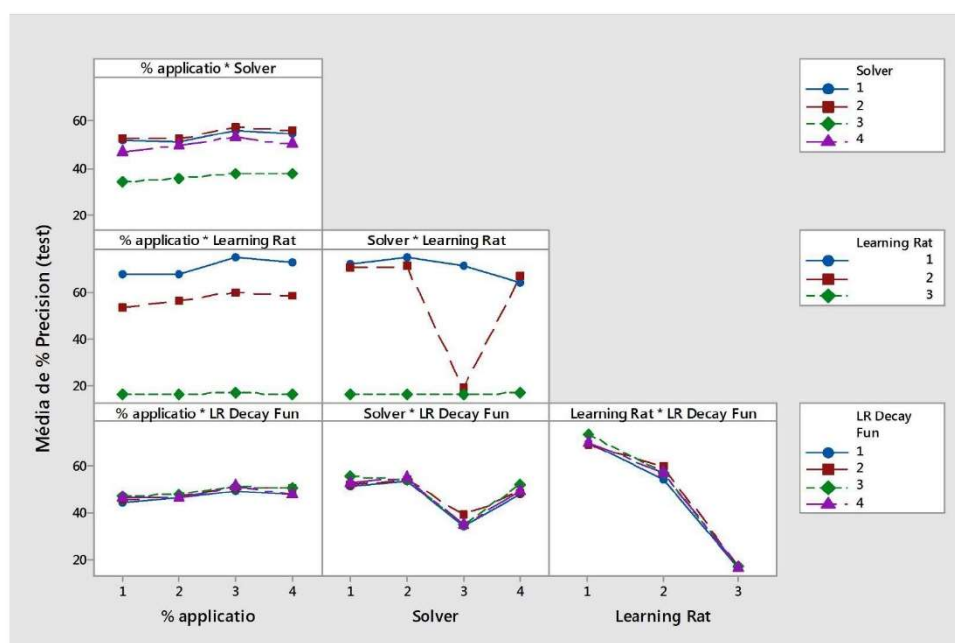


Fig 4. Effects of the second level parameters on the accuracy of the testing phase (average accuracy percentage).

<https://doi.org/10.1371/journal.pone.0210829.g004>

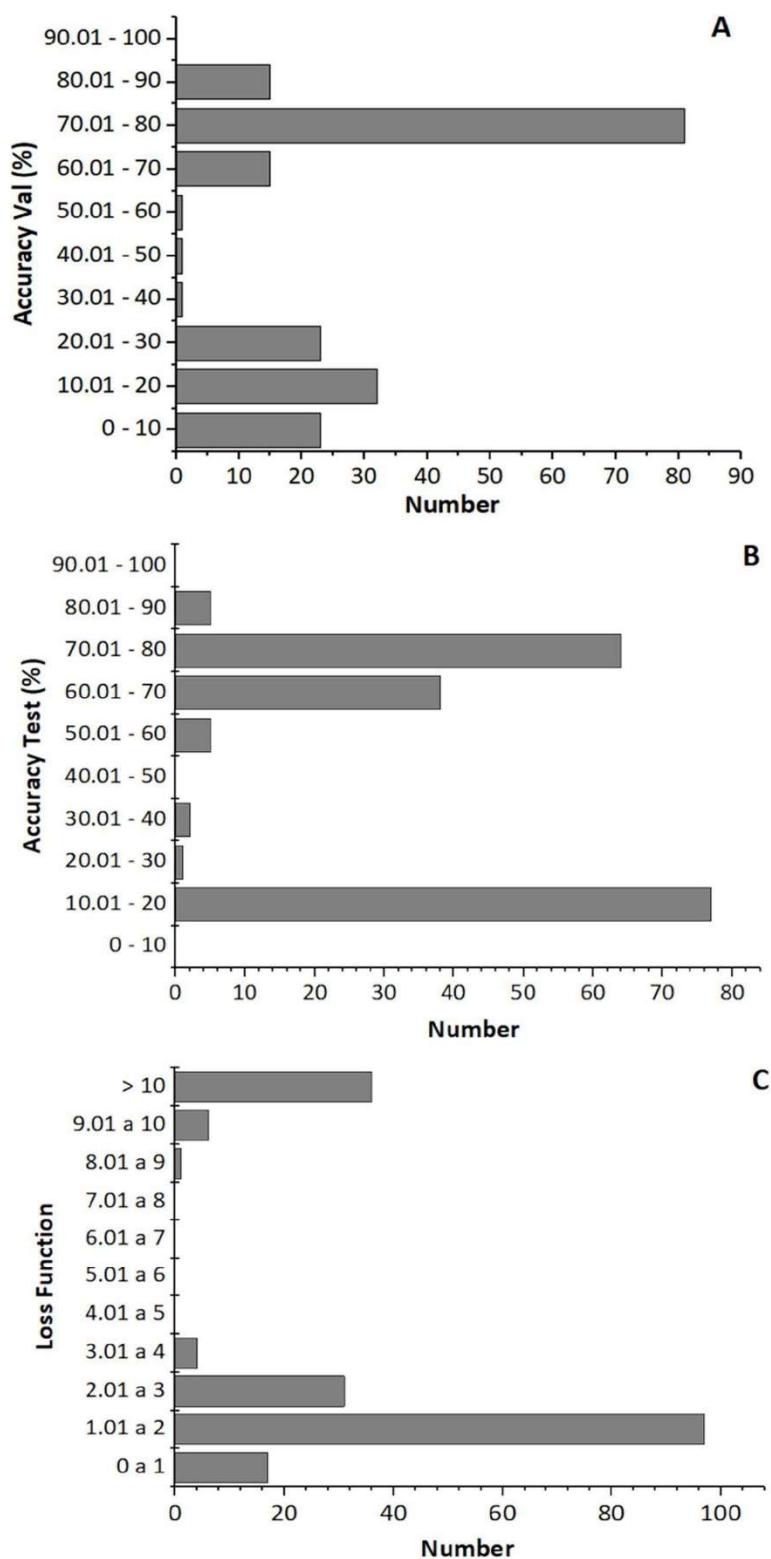


Fig 5. Histograms representing the outcome data: (A) percent accuracy–validation, (B) percent accuracy–testing, (C) loss function.

<https://doi.org/10.1371/journal.pone.0210829.g005>

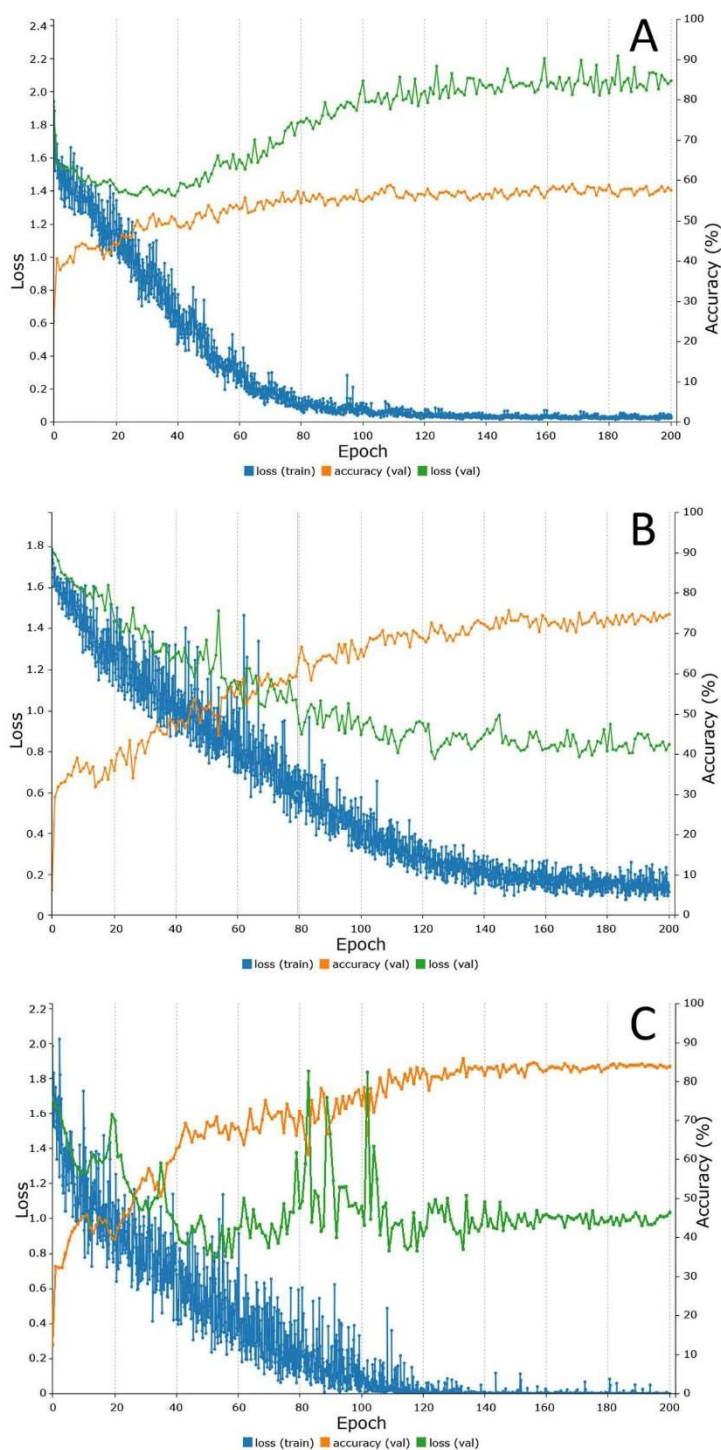


Fig 6. Representation of the training process of each CNN: (A) LeNet, (B) AlexNet and (C) GoogLeNet (orange line: accuracy validation; green line: loss validation; blue line: loss training). A—LeNet (after 200 epochs): percent accuracy (validation) 57.50%, loss (validation) 2.07, loss (training) 0.03; B—AlexNet (after 200 epochs): percent accuracy (validation) 74.69%, loss (validation) 0.83, loss (training) 0.11; C—GoogLeNet (after 200 epochs): percent accuracy (validation) 83.88%, loss (validation) 1.03, loss (training) 0.00.

<https://doi.org/10.1371/journal.pone.0210829.g006>

Table 3. Confusion matrix showing the results of the testing phase for the LeNet neural network.

Class	<i>Ae. aegypti</i> female	<i>Ae. aegypti</i> male	<i>Ae. albopictus</i> female	<i>Ae. albopictus</i> male	<i>C. quinquefasciatus</i> female	<i>C. quinquefasciatus</i> male	Accuracy (%)
<i>Ae. aegypti</i> female	7	1	2	4	0	0	50.00
<i>Ae. aegypti</i> male	5	6	3	0	0	0	42.86
<i>Ae. albopictus</i> female	1	2	10	0	1	0	71.43
<i>Ae. albopictus</i> male	2	0	0	7	3	2	50.00
<i>C. quinquefasciatus</i> female	1	0	1	1	9	2	64.29
<i>C. quinquefasciatus</i> male	2	3	2	0	2	5	35.71

<https://doi.org/10.1371/journal.pone.0210829.t003>

obtained during the testing phase (phase 3) for each of the mosquito species for the AlexNet neural network. The overall result of the testing phase was 51.2%.

Fig 6C shows the results obtained during training and validation phases for the GoogLeNet convolutional neural network. Using the previously selected parameters, the network achieved an accuracy of 83.9%, and the validation and training losses were reduced by approximately 1.03 and 0, respectively. Table 5 presents the accuracy of the results obtained by the GoogLeNet network during the testing phase for each of the mosquito species (phase 3). Overall, the performance of the GoogLeNet neural network during the testing phase was 76.2%.

With the exception of its performance during the classification of *C. quinquefasciatus* in the validation phase in terms of the loss function after 200 epochs, GoogLeNet demonstrated the best performance for the classification of adult mosquitoes.

It is also relevant to evaluate the classification performance for the different genera of mosquitoes. For the classification of the *Aedes* and *Culex* genera, an accuracy of 96.4% was obtained. Another important outcome was the performance in the classification of *Aedes* females, during which an overall accuracy of 82.1% was achieved.

Discussion

To the best of our knowledge, this is the first study that uses convolutional neural networks (CNN's) to extract features from images of adult mosquitoes to identify *Ae. aegypti* and *Ae. albopictus* species in the field. Members of the genus *Aedes* are well-known carriers and disseminators of arboviruses[1,13,44,45]. Also, we included the *C. quinquefasciatus* mosquito in our study because its medical importance as a vector for some arboviruses associated with the recent explosive epidemic occurred in Brazil[10,12]. However, the competence of *Culex* sp. to transmit Zika virus has not been confirmed by recent studies[11,46].

Table 4. Confusion matrix showing the results of the testing phase for the AlexNet neural network.

Class	<i>Ae. aegypti</i> female	<i>Ae. aegypti</i> male	<i>Ae. albopictus</i> female	<i>Ae. albopictus</i> male	<i>C. quinquefasciatus</i> female	<i>C. quinquefasciatus</i> male	Accuracy (%)
<i>Ae. aegypti</i> female	5	1	6	0	1	1	35.71
<i>Ae. aegypti</i> male	2	7	2	1	0	2	50.00
<i>Ae. albopictus</i> female	1	1	11	1	0	0	78.57
<i>Ae. albopictus</i> male	0	2	3	9	0	0	64.29
<i>C. quinquefasciatus</i> female	2	4	0	0	2	6	14.29
<i>C. quinquefasciatus</i> male	2	1	0	0	2	9	64.29

<https://doi.org/10.1371/journal.pone.0210829.t004>

Table 5. Confusion matrix showing the results of the testing phase for the GoogLeNet neural network.

Class	<i>Ae. aegypti</i> female	<i>Ae. aegypti</i> male	<i>Ae. albopictus</i> female	<i>Ae. albopictus</i> male	<i>C. quinquefasciatus</i> female	<i>C. quinquefasciatus</i> male	Accuracy (%)
<i>Ae. aegypti</i> female	9	3	2	0	0	0	64.29
<i>Ae. aegypti</i> male	1	12	1	0	0	0	85.71
<i>Ae. albopictus</i> female	0	1	12	1	0	0	85.71
<i>Ae. albopictus</i> male	0	1	0	13	0	0	92.86
<i>C. quinquefasciatus</i> female	1	1	0	0	8	4	57.14
<i>C. quinquefasciatus</i> male	0	0	0	1	3	10	71.43

<https://doi.org/10.1371/journal.pone.0210829.t005>

Sanchez-Ortiz et al.[31] used a very similar CNN-like method as our study, but focused on the use of digital images of larva as input for a machine learning algorithm for *Aedes* larva identification. The algorithm showed excellent performance, with 100% accuracy, in the identification of *Aedes* larva; however, for other mosquito species, the misclassification rate was 30%. Additionally, the sample size used in this study was tiny. Although their review have a significant scientific contribution, it requires the use of laboratory equipment and do not address issues such as time and logistics required in the traditional classification method for capturing and analysing the samples.

In our work, we used over 4,000 mosquito images in this preliminary study of the use of CNN's for the identification of mosquitoes; with such a small sample size, we achieved an accuracy of 76.2% in the automatic recognition of species and gender for six classes of mosquito. However, it is very promising that we were able to distinguish *Aedes* mosquitoes from other mosquitoes correctly. Our method was able to achieve 100% accuracy in classifying between *Aedes* and *Culex*, although *Culex* was misclassified as *Aedes* 10% of the time during the test phase. In the recognition of the *Aedes* female, which is the most important vector of arboviruses, we achieved 82% accuracy. It is essential to understand that it is a training process and every time we add a new species further training of the network will be necessary.

The importance of insects to transmit human diseases have motivated researchers to develop an arsenal of mechanical, chemical, biological and educational tools to help mitigate the harmful effects of insects. Batista et al.[33] reinforce that the efficiency of such systems depends on the knowledge of the spatiotemporal dynamics of insects. In that work, the authors developed a sensor that can measure the wingbeat frequency of flying insects at a distance. However, the study collected data under laboratory conditions. Thus, future works would be required to obtain additional features for real-world deployment.

Taxonomists have been searching for efficient methods to meet real world insect identification requirements[47]. The correct identification of insects is critically important in entomological research, pest control, and insect-resource utilization for the development of effective control strategies for diseases transmitted by these vectors and can aid in preventing arbovirus infections in humans[2,15]. Also, identifying the vectors associated with pathogens is fundamental to understand the transmission of arthropod-borne viruses[48]. This problem is recognized since the end of the last century, and computer-based recognition systems have been developed to identify insect species automatically, not only for mosquitoes, but also for wasps, moths, spiders, bees and others[15].

Unfortunately, as mentioned previously, taxonomic identification of mosquitoes is a time-consuming and challenging process that requires trained specialists. The high variability also compromises this method of identification of the morphological and molecular characteristics

found in members of the *Culicidae* subfamily, which was the object of our study. Furthermore, molecular identification, although generally accurate, is costly and often impractical. Other disadvantages of molecular techniques are the high cost and time applied in the laboratory. In this way, developing countries, which are often the main endemic areas, do not have sufficient resources for investments in laboratories with high technologies.

Although there is progress in autonomous identification systems for insects, two inadequacies remain. First, some systems obtain high accuracy of identification for limited insect species but have very narrow functionality, because the source code for those computer programs must be rewritten when these types of system are tested with new morphological insect data. Second, some systems, can be used to identify different taxonomic groups of insects, but the accuracy is not sufficient due to high variations during processes of digitizing morphological features and insufficient powerful classifiers which only can search local optimizing solutions rather than the global optimizing solutions during computation[15].

Autonomous identification method could be valuable and more accessible to health workers and other non-taxonomists for use in the identification of insects that can transmit infectious agents to humans. Herein, we demonstrated that it is possible to use a CNN to analyse photographic images to identify and classify mosquitoes by species and gender.

When adequate data are available, many methods, such as the Support Vector Machine (SVM) and Tangent Distance Classifiers (TDCs), can achieve good accuracy in automatic classification[42]. In a report by LeCun et al. on handwritten character recognition, more than 60,000 handwritten characters were used, resulting in a significantly reduced error rate[42]. Additionally, the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) uses approximately 1.2 million images for training, 50,000 for validation, and 150,000 images for testing [39].

In another application of a convolutional neural network for the identification of *Aedes* wings and larva, Lorenz et al.[26] used 17 species from the *Anopheles*, *Aedes* and *Culex* genera to test the hypothesis that classification based on wing shape characteristics using an Artificial Neural Network (ANN) was more accurate than traditional classification using discriminant analysis. The ANN was able to correctly classify species with greater accuracy than the conventional method using multivariate discriminant analysis, with an accuracy varying from 86% to 100%. However, this study focus on parts of the body of the mosquito and requires the use of laboratory equipment.

Another challenge in the use of convolutional neural networks for image recognition with a limited number of images is the risk of overfitting. Application of different sets of images during the training, validation and testing phases may reduce the risk of overfitting[39]. Additionally, the use of different, randomly selected images during the testing phase, as demonstrated in our study, is a promising method to avoid overfitting. Limitations due to dataset size and its balance between the classes may explain the low accuracy achieved for some species classified individually in our study. We firmly believe that increasing the dataset for all three phases (training, validation and testing) and the use of different neural network architectures will improve the precision of automatic classification.

Another possible way to increase the reliability of our neural network for the identification of mosquito images is to re-design the layers of the neural network to include additional morphological characteristics, such as wing shapes, palps and other features.

This preliminary study is an opportunity to construct a dense dataset of images of mosquitoes that can transmit diseases to humans, including *Anopheles* and other species among the 3,556 species of *Culicidae* that are currently recognized.

The final goal is to embed the model in a mobile APP that will allow for community participation and thereby facilitate efforts to control vector borne diseases.

Conclusion

This work demonstrates that CNN's can be applied to the autonomous classification of mosquitoes to aid in the screening of possible vectors of arboviruses by health workers and other non-taxonomists. High accuracy of classification of mosquitoes using CNN's can be achieved using different neural networks. LeNet and AlexNet had inferior performances to that of GoogleLeNet, which indicates that the use of more complex networks with more layers is necessary to improve accuracy. Taxonomic classification of insects can be performed automatically via the acquisition of image's features that differentiate one insect from another. Only by using both the genus and sex we achieved high accuracy in the identification of *Aedes* females.

Our results provide information that is fundamental to the automatic morphological classification of a species of interest applying CNN to classify the adult mosquito in the environment it lives. In the critical discussion, the published works develop mainly a laboratory tool. In the taxonomy of adult mosquitoes, an epidemiological classification has a different implication. The application method we propose will be a robust epidemiological instrument for the rapid identification of focus of mosquitoes allowing the community to be part of the control of vector-borne diseases.

Another essential contribution relies on the construction of a mosquito dataset. Much data is required to increase the reliability of any autonomous method to recognize objects. This study already provides more than 4,000 mosquito images, which is vital in creating a significant and robust dataset for later applications.

We hope that later, the model can be embedded in a mobile APP to allow for community participation and thereby facilitate efforts to control vector borne diseases. Indeed, this model can be used to improve vector control operations that are linked with fast and reliable identification of targeted species and to provide knowledge of their biology and ecology.

Acknowledgments

We are thankful to the department that facilitated the collection of *Culicidae* at Fiocruz Rio de Janeiro. We give special thanks to Maycon Neves and Monique Motta, who are members of their staff. Additionally, we would like to thank Otavio Ribeiro and João Marcelo from the High Performance Computing Center of SENAI CIMATEC—Bahia. Tiago Trocoli and Nelson Alves from the Brazilian Institute of Robotics, SENAI CIMATEC—Bahia, provided valuable support.

Author Contributions

Conceptualization: Daniel Motta, Alex Álisson Bandeira Santos, Bruna Aparecida Souza Machado, Eduardo Oyama Lins Fonseca, Frank Kirchner, Roberto Badaró.

Data curation: Daniel Motta, Ingrid Winkler, Bruna Aparecida Souza Machado, Daniel André Dias Imperial Pereira, Alexandre Morais Cavalcanti, Eduardo Oyama Lins Fonseca, Roberto Badaró.

Formal analysis: Daniel Motta, Bruna Aparecida Souza Machado, Daniel André Dias Imperial Pereira, Alexandre Morais Cavalcanti, Eduardo Oyama Lins Fonseca.

Investigation: Daniel Motta, Alex Álisson Bandeira Santos, Ingrid Winkler, Eduardo Oyama Lins Fonseca, Roberto Badaró.

Methodology: Daniel Motta, Ingrid Winkler, Bruna Aparecida Souza Machado, Daniel André Dias Imperial Pereira, Alexandre Morais Cavalcanti, Eduardo Oyama Lins Fonseca, Frank Kirchner, Roberto Badaró.

Project administration: Alex Álisson Bandeira Santos.

Software: Daniel Motta, Daniel André Dias Imperial Pereira, Alexandre Morais Cavalcanti, Frank Kirchner.

Supervision: Alex Álisson Bandeira Santos, Frank Kirchner, Roberto Badaró.

Validation: Daniel Motta, Bruna Aparecida Souza Machado, Frank Kirchner, Roberto Badaró.

Visualization: Daniel Motta, Bruna Aparecida Souza Machado, Roberto Badaró.

Writing – original draft: Daniel Motta, Alex Álisson Bandeira Santos, Ingrid Winkler, Bruna Aparecida Souza Machado, Frank Kirchner, Roberto Badaró.

References

- Musso D, Rodriguez-Morales AJ, Levi JE, Cao-Lormeau V-M, Gubler DJ. Unexpected outbreaks of arbovirus infections: lessons learned from the Pacific and tropical America. *Lancet Infect Dis*. Elsevier; 2018; [https://doi.org/10.1016/S1473-3099\(18\)30269-X](https://doi.org/10.1016/S1473-3099(18)30269-X)
- WHO. Global Vector Control Response—Background document to inform deliberations during the 70th session of the World Health Assembly. WHO. 2017;2030: 47.
- Leta S, Beyene TJ, De Clercq EM, Amenu K, Kraemer MUG, Revie CW. Global risk mapping for major diseases transmitted by *Aedes aegypti* and *Aedes albopictus*. *Int J Infect Dis*. Elsevier; 2018; 67: 25–35. <https://doi.org/10.1016/j.ijid.2017.11.026> PMID: 29196275
- Cauchemez S, Besnard M, Bompard P, Dub T, Guillemette-Artur P, Eyrolle-Guignot D, et al. Association between Zika virus and microcephaly in French Polynesia, 2013–2015: a retrospective study. *Lancet (London, England)*. 2016; 387: 2125–2132. [https://doi.org/10.1016/S0140-6736\(16\)00651-6](https://doi.org/10.1016/S0140-6736(16)00651-6) PMID: 26993883
- del Carpio-Orantes L, González-Clemente M del C, Lamothe-Aguilar T. Zika and its vector mosquitoes in Mexico. *J Asia-Pacific Biodivers*. 2018; 11: 317–319. <https://doi.org/10.1016/j.japb.2018.01.002>
- Ding F, Fu J, Jiang D, Hao M, Lin G. Mapping the spatial distribution of *Aedes aegypti* and *Aedes albopictus*. *Acta Trop*. 2018; 178: 155–162. <https://doi.org/10.1016/j.actatropica.2017.11.020> PMID: 29191515
- PAHO. Tool for the diagnosis and care of patients with suspected arboviral diseases [Internet]. Us1.1. 2017. Available: <http://iris.paho.org/xmlui/handle/123456789/33895>
- Thangamani S, Huang J, Hart CE, Guzman H, Tesh RB. Vertical Transmission of Zika Virus in *Aedes aegypti* Mosquitoes. *Am J Trop Med Hyg*. The American Society of Tropical Medicine and Hygiene; 2016; 95: 1169–1173. <https://doi.org/10.4269/ajtmh.16-0448> PMID: 27573623
- Cevallos V, Ponce P, Waggoner JJ, Pinsky BA, Coloma J, Quiroga C, et al. Zika and Chikungunya virus detection in naturally infected *Aedes aegypti* in Ecuador. *Acta Trop*. 2018; 177: 74–80. <https://doi.org/10.1016/j.actatropica.2017.09.029> PMID: 28982578
- Guedes DR, Paiva MH, Donato MM, Barbosa PP, Krokovsky L, Rocha SW dos S, et al. Zika virus replication in the mosquito *Culex quinquefasciatus* in Brazil. *Emerg Microbes Infect*. 2017; 6. <https://doi.org/10.1038/emi.2017.59> PMID: 28790458
- Main BJ, Nicholson J, Winokur OC, Steiner C, Riemersma KK, Stuart J, et al. Vector competence of *Aedes aegypti*, *Culex tarsalis*, and *Culex quinquefasciatus* from California for Zika virus. *PLoS Negl Trop Dis*. Public Library of Science; 2018; 12: e0006524. Available: <https://doi.org/10.1371/journal.pntd.0006524> PMID: 29927940
- Guo X, Li C, Deng Y, Xing D, Liu Q, Wu Q, et al. *Culex pipiens quinquefasciatus*: a potential vector to transmit Zika virus. *Emerg Microbes Infect*. Nature Publishing Group; 2016; 5: e102. <https://doi.org/10.1038/emi.2016.102> PMID: 27599470
- Boyer S, Calvez E, Chouin-Carneiro T, Diallo D, Failloux A-B. An overview of mosquito vectors of Zika virus. *Microbes Infect*. 2018; <https://doi.org/10.1016/j.micinf.2018.01.006>
- Park SI, Bisgin H, Ding H, Semey HG, Langley DA, Tong W, et al. Species identification of food contaminating beetles by recognizing patterns in microscopic images of elytra fragments. *PLoS One*. 2016; 11: 1–22. <https://doi.org/10.1371/journal.pone.0157940> PMID: 27341524
- Yang HP, Ma C Sen, Wen H, Zhan Q Bin, Wang XL. A tool for developing an automatic insect identification system based on wing outlines. *Sci Rep*. Nature Publishing Group; 2015; 5: 1–11. <https://doi.org/10.1038/srep12786> PMID: 26251292
- Sifferlin A. Fewer Scientists Are Studying Insects. Here's Why That's So Dangerous. *Time*. 2018.

17. Utsugi J, Toshihide K, Motomi ITO. Current progress in DNA barcoding and future implications for entomology. *Entomol Sci*. Wiley/Blackwell (10.1111); 2011; 14: 107–124. <https://doi.org/10.1111/j.1479-8298.2011.00449.x>
18. Karthika P, Vadivalagan C, Thirumurugan D, Kumar RR, Murugan K, Canale A, et al. DNA barcoding of five Japanese encephalitis mosquito vectors (*Culex fuscocephala*, *Culex gelidus*, *Culex tritaeniorhynchus*, *Culex pseudovishnui* and *Culex vishnui*). *Acta Trop*. 2018; 183: 84–91. <https://doi.org/10.1016/j.actatropica.2018.04.006> PMID: 29625090
19. Mewara A, Sharma M, Kaura T, Zaman K, Yadav R, Sehgal R. Rapid identification of medically important mosquitoes by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *Parasit Vectors*. BioMed Central; 2018; 11: 281. <https://doi.org/10.1186/s13071-018-2854-0> PMID: 29720246
20. Zhong Y, Gao J, Lei Q, Zhou Y. A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors (Switzerland)*. 2018; 18. <https://doi.org/10.3390/s18051489> PMID: 29747429
21. Vossenberg BTLH Van De, Ibáñez-Justicia A, Metz-Verschure E, Veen EJ Van, Bruil-Dieters ML, Scholte EJ. Real-Time PCR Tests in Dutch Exotic Mosquito Surveys; Implementation of *Aedes aegypti* and *Aedes albopictus* Identification Tests, and the Development of Tests for the Identification of *Aedes atropalpus* and *Aedes japonicus japonicus* (Diptera: Culicidae). *J Med Entomol*. Entomological Society of America; 2015; 52: 336–350. <https://doi.org/10.1093/jme/tjv020> PMID: 26334807
22. Schneider J, Valentini A, Dejean T, Montarsi F, Taberlet P, Glaizot O, et al. Detection of Invasive Mosquito Vectors Using Environmental DNA (eDNA) from Water Samples. *PLoS One*. Public Library of Science; 2016; 11: e0162493. Available: <https://doi.org/10.1371/journal.pone.0162493> PMID: 27626642
23. Kumar NP, Rajavel AR, Natarajan R, Jambulingam P. DNA Barcodes Can Distinguish Species of Indian Mosquitoes (Diptera: Culicidae). *J Med Entomol*. Entomological Society of America; 2007; 44: 1–7. [https://doi.org/10.1603/0022-2585\(2007\)44\[1:DBCDSO\]2.0.CO;2](https://doi.org/10.1603/0022-2585(2007)44[1:DBCDSO]2.0.CO;2) PMID: 17294914
24. Ouyang TH, Yang EC, Jiang JA, Lin T Te. Mosquito vector monitoring system based on optical wing-beat classification. *Comput Electron Agric*. Elsevier B.V.; 2015; 118: 47–55. <https://doi.org/10.1016/j.compag.2015.08.021>
25. Arthur BJ, Emr KS, Wytttenbach RA, Hoy RR. Mosquito (*Aedes aegypti*) flight tones: Frequency, harmonicity, spherical spreading, and phase relationships. *J Acoust Soc Am*. 2014; 135: 933–941. <https://doi.org/10.1121/1.4861233> PMID: 25234901
26. Lorenz C, Sergio A, Suesdek L. Artificial Neural Network applied as a methodology of mosquito species identification. *Acta Trop*. 2015; 152: 165–169. <https://doi.org/10.1016/j.actatropica.2015.09.011> PMID: 26394186
27. Kesavaraju B, Dickson S. New Technique to Count Mosquito Adults: Using ImageJ Software to Estimate Number of Mosquito Adults in a Trap. *J Am Mosq Control Assoc*. 2012; 28: 330–333. <https://doi.org/10.2987/12-6254R.1> PMID: 23393760
28. Yao Q, Lv J, Liu Q jie, Diao G qiang, Yang B jun, Chen H ming, et al. An Insect Imaging System to Automate Rice Light-Trap Pest Identification. *J Integr Agric*. Chinese Academy of Agricultural Sciences; 2012; 11: 978–985. [https://doi.org/10.1016/S2095-3119\(12\)60089-6](https://doi.org/10.1016/S2095-3119(12)60089-6)
29. Mundada RG, Gohokar V V. Detection and Classification of Pests in Greenhouse Using Image Processing. 2013; 5: 57–63.
30. Schmidhuber J. Deep Learning in neural networks: An overview. *Neural Networks*. Elsevier Ltd; 2015; 61: 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003> PMID: 25462637
31. Sanchez-Ortiz A, Fierro-Radilla A, Arista-Jalife A, Cedillo-Hernandez M, Nakano-Miyatake M, Robles-Camarillo D, et al. Mosquito larva classification method based on convolutional neural networks. 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP). 2017. pp. 1–6. <https://doi.org/10.1109/CONIELECOMP.2017.7891835>
32. Silva DF, De Souza VMA, Batista GEAPA, Keogh E, Ellis DPW. Applying machine learning and audio analysis techniques to insect recognition in intelligent traps. *Proceedings—2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*. 2013. pp. 99–104. <https://doi.org/10.1109/ICMLA.2013.24>
33. Batista GEAPA, Hao Y, Keogh E, Mafra-Neto A. Towards automatic classification on flying insects using inexpensive sensors. *Proc - 10th Int Conf Mach Learn Appl ICMLA 2011*. 2011; 1: 364–369. <https://doi.org/10.1109/ICMLA.2011.145>
34. De Souza VMA, Silva DF, Batista GEAPA. Classification of data streams applied to insect recognition: Initial results. *Proc—2013 Brazilian Conf Intell Syst BRACIS 2013*. 2013; 76–81. <https://doi.org/10.1109/BRACIS.2013.21>
35. Gaburro J, Duchemin J-B, Paradkar PN, Nahavandi S, Bhatti A. Assessment of ICount software, a precise and fast egg counting tool for the mosquito vector *Aedes aegypti*. *Parasit Vectors*. Parasites & Vectors; 2016; 9: 590. <https://doi.org/10.1186/s13071-016-1870-1> PMID: 27863526

36. Henry A, Thongsripong P, Fonseca-Gonzalez I, Jaramillo-Ocampo N, Dujardin JP. Wing shape of dengue vectors from around the world. *Infect Genet Evol.* 2010; 10: 207–214. <https://doi.org/10.1016/j.meegid.2009.12.001> PMID: 20026429
37. Bova JE, Paulson SL, Paulson G. Morphological differentiation of the eggs of North American container-inhabiting *Aedes* mosquitoes. *J Am Mosq Control Assoc.* 2016; 32: 244–246. <https://doi.org/10.2987/15-6535.1> PMID: 27802396
38. Schaper S, Hernández-Chavarría F. Scanning electron microscopy of the four larval instars of the Dengue fever vector *Aedes aegypti* (Diptera: Culicidae). *Rev Biol Trop (Int J Trop Biol.* 2006; 54: 847–852.
39. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. *Adv Neural Inf Process Syst.* 2012; 1–9. <http://doi.org/10.1016/j.protcy.2014.09.007>
40. Goodfellow I, Bengio Y, Courville A. *Deep Learning* [Internet]. MIT Press; 2016. Available: <http://www.deeplearningbook.org>
41. CDC C for DC and P. Surveillance and Control of *Aedes aegypti* and *Aedes albopictus* in the United States. *Centers Dis Control Prev—CDC.* 2016; 1–16. <https://doi.org/10.1371/journal.pntd.0004043>
42. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition. *Proc IEEE.* 1998; 46. <https://doi.org/10.1109/5.726791>
43. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going Deeper with Convolutions. *arXiv:14094842.* 2014; <https://doi.org/10.1109/CVPR.2015.7298594>
44. Rezza G, Nicoletti L, Angelini R, Romi R, Finarelli AC, Panning M, et al. Infection with chikungunya virus in Italy: an outbreak in a temperate region. *Lancet.* 2007; 370: 1840–1846. [https://doi.org/10.1016/S0140-6736\(07\)61779-6](https://doi.org/10.1016/S0140-6736(07)61779-6) PMID: 18061059
45. Kraemer MUG, Sinka ME, Duda KA, Mylne AQN, Shearer FM, Barker CM, et al. The global distribution of the arbovirus vectors *Aedes aegypti* and *Ae. albopictus*. *Jit M, editor. Elife. eLife Sciences Publications, Ltd;* 2015; 4: e08347. <https://doi.org/10.7554/eLife.08347> PMID: 26126267
46. Epelboin Y, Talaga S, Epelboin L, Dusfour I. Zika virus: An updated review of competent or naturally infected mosquitoes. *PLoS Negl Trop Dis. Public Library of Science;* 2017; 11: e0005933. Available: <https://doi.org/10.1371/journal.pntd.0005933> PMID: 29145400
47. Wang J, Lin C, Ji L, Liang A. A new automatic identification system of insect images at the order level. *Knowledge-Based Syst. Elsevier B.V.;* 2012; 33: 102–110. <https://doi.org/10.1016/j.knosys.2012.03.014>
48. Santos J, Meneses BM. An integrated approach for the assessment of the *Aedes aegypti* and *Aedes albopictus* global spatial distribution, and determination of the zones susceptible to the development of Zika virus. *Acta Trop.* 2017; 168: 80–90. <https://doi.org/10.1016/j.actatropica.2017.01.015> PMID: 28111132

5 ARTIGO II

Optimization of CNN hyperparameters for automatic classification of adult mosquitoes

Daniel Motta¹, Alex Álisson Bandeira Santos¹, Bruna Aparecida Souza Machado^{1,2}, Otávio Ribeiro³, Luis Octavio Arriaga Camargo⁴, Matias Valdenegro-Toro⁴, Frank Kirchner⁴, Roberto Badaró^{1,2}

¹University Center SENAI/CIMATEC, Salvador, Bahia, Brazil; ²Technology Institute of Health – SENAI CIMATEC, Salvador, Bahia, Brazil, ³High Performance Computing Center – SENAI CIMATEC, Salvador, Bahia, Brazil, ⁴Research Centre for Artificial Intelligence, DFKI, Bremen, Germany.

Artigo ainda não submetido para publicação. Será submetido à ***PloS ONE***.

Este capítulo do trabalho consiste no desenvolvimento de modelos computacionais, baseados em redes neurais convolucionais (CNN) e na otimização dos hiperparâmetros do modelo para melhoria no desempenho de classificação automática de mosquitos adultos a partir de imagens das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*. Os modelos computacionais foram elaborados na linguagem *python*, com a utilização da biblioteca do *framework Keras*. O processo de otimização passou por etapas de investigação aleatória e de pesquisa exaustiva para definição dos melhores hiperparâmetros do modelo. Com um banco de dados de 7.561 imagens dos mosquitos-alvo do estudo e 1.187 imagens de outros insetos, arquiteturas mais complexas de CNN, incluindo a Xception e a DenseNet, foram utilizadas para treinamento da rede. O modelo foi testado para a atividade de classificação de espécies e para detecção do gênero *Aedes*, obtendo resultados de acurácia balanceada superiores a 90%. Uma investigação sobre o ângulo do mosquito na imagem também foi conduzida para avaliar a influência da forma de aquisição da imagem na acurácia do modelo. Os resultados obtidos neste artigo evidenciaram que a aplicação de modelos baseados em CNN pode se tornar uma ferramenta entomológica importante para uso por especialistas e para melhoria da eficácia dos planos de

combate às arboviroses, a exemplo da dengue, chikungunya e Zika. Este capítulo do estudo responde aos objetivos específicos de desenvolver modelos computacionais, utilizando CNN's, para extração de atributos de imagens e para classificação das mesmas, em uma abordagem supervisionada; realizar a otimização dos hiperparâmetros do modelo para classificação de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus* ; avaliar o desempenho do modelo otimizado na classificação e na detecção de mosquitos adultos e; avaliar a influência do ângulo da imagem no desempenho do modelo para classificação de mosquitos adultos.

Otimização de hiperparâmetros de CNN para classificação automática de mosquitos adultos

5.1 RESUMO

O impacto econômico e social causado pelas doenças transmitidas por vetores no mundo nos últimos anos é significativo. Atualmente não há tratamento específico ou vacina comercial para o controle e prevenção das arboviroses, o que torna a caracterização entomológica fundamental no combate a doenças como dengue, chikungunya e Zika. Atualmente, a identificação morfológica de mosquitos inclui um exame visual individual do inseto, o que demanda tempo elevado para a sua classificação, e, além disso, requer expertise na área (vários anos de experiência do profissional). Dentro desta problemática, o desenvolvimento de um novo método automatizado de percepção e classificação de mosquitos torna-se cada vez mais essencial. Com base nesse contexto, neste trabalho, foi desenvolvido um modelo computacional baseado em redes neurais convolucionais (CNN's) para extrair características de imagens de mosquitos com o objetivo de classificar as espécies de *Aedes aegypti*, *Aedes albopictus* e de *Culex quinquefasciatus*, bem como para detectar mosquitos do gênero *Aedes*. Dessa forma, para treinar as CNN's, com o intuito de realizar a classificação morfológica automática dos mosquitos, foi usado um banco de imagens que incluiu 7.561 imagens das três espécies-alvo do estudo e 1.187 imagens de outros insetos. Diferentes redes neurais, incluindo Xception e DenseNet, foram utilizadas para o desenvolvimento do modelo de classificação automática a partir de imagens. Um processo estruturado de pesquisa aleatória e pesquisa exaustiva foi desenvolvido para otimização dos hiperparâmetros na busca do aumento da acurácia do modelo e uma investigação sobre a influência do ângulo do mosquito na imagem foi conduzida. Estratégias de eliminação de *overfitting* foram empregadas computacionalmente, visando a maior capacidade de generalização do modelo proposto. O modelo otimizado foi testado para as atividades de classificação dos mosquitos-alvo do estudo e para detecção do gênero *Aedes*, obtendo o resultado de acurácia balanceada global de 93,5% para classificação e de 97,3% para detecção. Os resultados fornecem

informações fundamentais para a classificação morfológica automática de espécies de mosquitos. A incorporação do modelo em uma poderosa ferramenta entomológica é um recurso valioso e acessível para profissionais de saúde e outros não especialistas para a identificação de insetos que podem transmitir agentes infecciosos para seres humanos.

Palavras-chave: Aedes; Culex; Classificação; Aprendizagem de Máquina; Aprendizagem Profunda; Redes Neurais Convolucionais.

5.2 INTRODUÇÃO

O impacto econômico causado pelas doenças transmitidas por vetores à sociedade é bastante significativo. Para governos em países endêmicos, isso inclui os custos das atividades de controle de vetores e de gerenciamento de casos (WHO, 2017). Além disso, muitos casos de arboviroses não são reportados, o que dificulta estimar a verdadeira extensão econômica dessas doenças (SHEPARD et al., 2016).

Dengue, chikungunya e Zika são as doenças virais mais comuns transmitidas por vetores (PAHO, 2017) e atualmente causam o maior impacto negativo na saúde pública e danos econômicos em todo o mundo (MAYER; B. TESH; VASILAKIS, 2016).

Shepard et al. (2016), estimaram que em 2013 houve um total de 58,4 milhões de infecções sintomáticas por vírus da dengue, incluindo 13.586 casos fatais. Desta forma, o custo global da dengue foi em torno de 9 bilhões de dólares.

Em relação à Zika, o Programa de Desenvolvimento das Nações Unidas (2017), em parceria com a Federação Internacional da Cruz Vermelha, avaliaram os impactos socioeconômicos do vírus da Zika em países da América Latina e no Caribe, em especial no Brasil, Colômbia e Suriname. A Zika foi responsável por perdas tangíveis no produto interno bruto, estimado entre sete e oito bilhões de dólares somente no período de 2015 a 2017, impondo um ônus imediato aos sistemas de saúde e assistências sociais.

É importante destacar que o impacto econômico vai além dos custos relacionados à saúde pública e ao produto interno bruto dos países, pois afeta diretamente a economia doméstica. Em estudo realizado por Gopalan & Das (2017), no estado de Orissa na Índia, 10% da renda mensal familiar foi destinada à despesas com cuidados de saúde por conta da chikungunya. Entre as pessoas entrevistadas, em média, o trabalhador perdeu 35 dias de trabalho por consequência da doença.

O mosquito da espécie *Aedes. aegypti*, consiste no principal vetor de transmissão das arboviroses e é responsável pela infecção de milhões de pessoas globalmente. Além disso, a espécie *Aedes. albopictus* é também um vetor transmissor dessas doenças e nas últimas três décadas experimentou uma rápida expansão geográfica de sua população no mundo (MAUCOURANT et al., 2019). O *Culex quinquefasciatus*, além de também ser um vetor transmissor de doenças, é, juntamente com o *Ae. aegypti*, um dos mosquitos urbanos, mais comuns em ambientes tropicais e subtropicais, causando incômodo para os seres humanos (NG et al., 2018).

Atualmente não há tratamento específico ou vacina comercial para o controle e prevenção das arboviroses, portanto, o controle populacional de mosquitos é a única alternativa de prevenção (PARRA-AMAYA et al., 2016). A caracterização entomológica é fundamental para adquirir informações sobre o comportamento dos mosquitos, contudo, a prática atualmente utilizada para identificar espécies de insetos é manual, demanda tempo elevado e requer vários anos de experiência dos profissionais que a executam (PARK et al., 2016; WANG et al., 2012; YANG et al., 2015). O desenvolvimento de um novo método automatizado de percepção e classificação de mosquitos torna-se cada vez mais essencial (FUCHIDA et al., 2017).

Dentre as técnicas estudadas para classificação automática, as técnicas de inteligência artificial (IA), baseadas em *machine learning* e *deep learning*, têm-se destacado (LORENZ; SERGIO; SUESDEK, 2015; OUYANG et al., 2015; SANCHEZ-ORTIZ et al., 2017; SILVA et al., 2013). Entretanto, conforme discutido por Batista et al. (2011), a eficiência dessas ferramentas depende do conhecimento da dinâmica espaço temporal dos vetores transmissores e o

tempo que se leva entre a coleta dos mosquitos e a análise em laboratório afeta negativamente essa eficiência.

Em estudo recente, Motta et al. (2019), evidenciaram a viabilidade da aplicação de CNN's para classificação automática de mosquitos adultos em campo. Esse estudo apresentou resultados importantes e que fomentam a investigação com arquiteturas de redes neurais mais complexas, com o objetivo de aumentar a acurácia do modelo.

Adicionalmente à utilização de arquiteturas mais complexas, os algoritmos de *deep learning* envolvem otimizações em diversos contextos e de todos os problemas de otimização de *deep learning*, os mais difíceis são os de redes neurais (GOODFELLOW; BENGIO; COURVILLE, 2016). As CNN's exigem a definição de diversos parâmetros de treinamento que não são ajustados automaticamente durante o processo de aprendizagem. É bastante comum investir meses para otimizar um limitado número de parâmetros de treinamento e, por se tratar de um problema tão importante e caro, existem técnicas de otimização desenvolvidas para resolvê-lo (GOODFELLOW; BENGIO; COURVILLE, 2016). As técnicas de pesquisa aleatória (*random search*) e pesquisa de grade (*grid search*) são as estratégias mais usadas para otimização dos hiperparâmetros (BERGSTRA; BENGIO, 2012).

Dentro deste contexto, o objetivo deste trabalho foi o de desenvolver uma ferramenta entomológica, baseada em CNN, para uso em campo por especialistas e não especialistas, visando automatizar a classificação das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*, bem como permitir a detecção do gênero *Aedes*. Ressalta-se que arquiteturas mais complexas de CNN, incluindo a Xception e a DenseNet, foram utilizadas para treinamento da rede e os hiperparâmetros do modelo foram otimizados, com uma abordagem em etapas aleatórias e exaustivas de investigação para o aumento da sua acurácia.

5.3 MATERIAIS E MÉTODOS

5.3.1 Declaração de ética

Não foram necessárias autorizações para amostragem usada neste estudo. A amostragem de campo não envolveu nenhuma espécie ameaçada ou protegida.

5.3.2 Coleta de amostras

As amostras de mosquitos utilizadas para captura de imagens foram obtidas no Laboratório de Parasitologia da Universidade Federal da Bahia – UFBA (Salvador, Brasil), no Instituto de Entomologia Oswaldo Cruz da FIOCRUZ (Rio de Janeiro, Brasil) e também foram coletadas em campo. As amostras obtidas da UFBA foram dos mosquitos *Ae. aegypti* (10 fêmeas e 6 machos) e *Ae. albopictus* (5 fêmeas e 5 machos). As amostras obtidas da FIOCRUZ foram dos mosquitos *Ae. aegypti* (3 fêmeas), *Ae. albopictus* (3 fêmeas) e *C. quinquefasciatus* (3 fêmeas).

A captura de insetos adultos no campo foi realizada por um entomologista e resultou na coleta de 120 exemplares de *Ae. aegypti*, sendo 94 fêmeas e 26 machos, 94 exemplares de *Ae. albopictus*, 71 fêmeas e 23 machos, além de 110 exemplares de *C. quinquefasciatus*, 97 fêmeas e 13 machos. A amostragem de campo ocorreu em dois períodos, entre setembro e outubro de 2017 e entre março e abril de 2018, ambos na cidade de Salvador (Bahia, Brasil) e em duas áreas de coleta localizadas nos Bairros de Piatã e Águas Claras. Utilizaram-se armadilhas do tipo CDC luminosa, tubos de sucção e puçá para a coleta de insetos adultos. Os espécimes capturados foram mortos com acetato de etila, armazenados em tubos entomológicos e separados por espécie em laboratório. A identificação foi realizada por entomologista utilizando microscópio estereoscópio.

5.3.3 Aquisição das imagens

A abordagem de *deep learning* utilizada neste trabalho foi a de aprendizado supervisionado e, desta forma, a correta aquisição e rotulação das imagens é um passo fundamental para treinamento e avaliação do modelo desenvolvido.

Das 7.561 imagens de *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus* utilizadas neste estudo, aproximadamente 95% (7.180) foi fruto de aquisição própria, a partir das amostras coletadas. As espécies foram fotografadas com diferentes câmeras, incluindo uma Leica DMC2900 (Leica Microsystems, Heerbrugg, Suíça) acoplada a uma Leica M205C estereoscópica do Instituto de Entomologia Oswaldo Cruz da FIOCRUZ (Rio de Janeiro, Brasil), uma Canon Power Shot D30 (Canon, Tóquio, Japão) acoplada a um estereomicroscópio Wild M3C (Leica Microsystems, Heerbrugg, Suíça) do SENAI CIMATEC (Salvador, Brasil), um celular Samsung J5 (Samsung, Seul, Coreia do Sul) e um iPhone Apple 7 (Apple, Cupertino, Califórnia, EUA). As imagens foram coletadas em diferentes resoluções e níveis de qualidade, bem como a posição (ângulo e zoom) do mosquito na imagem foi variada. O objetivo foi ampliar a variabilidade dos dados de entrada do modelo para reduzir o efeito de *overfitting*, melhorando o desempenho do método de classificação. A Figura 1 apresenta uma amostra das imagens adquiridas e utilizadas neste trabalho.

As outras 381 imagens, aproximadamente 5% do total, foram fornecidas pelo Instituto de Entomologia Oswaldo Cruz da FIOCRUZ (Rio de Janeiro, Brasil) ou extraídas da *Internet* (banco de dados da *Imagenet*). Também foram adquiridas 1.187 imagens de aranhas, besouros e abelhas da *internet* (site de pesquisa *Google*) e rotuladas no trabalho como 'outros'.

Todas as imagens utilizadas neste estudo foram avaliadas por um entomologista para validar a sua classificação. A Tabela 1 apresenta o número de imagens, por classe, utilizadas neste trabalho.

Tabela 1. Número de imagens por classe.

Nome da classe	Número de imagens
<i>Ae. aegypti</i> fêmea	1.193
<i>Ae. aegypti</i> macho	1.562
<i>Ae. albopictus</i> fêmea	1.448
<i>Ae. albopictus</i> macho	1.360
<i>C. quinquefasciatus</i> fêmea	1.025
<i>C. quinquefasciatus</i> macho	973
Outros	1.187
Total	8.748

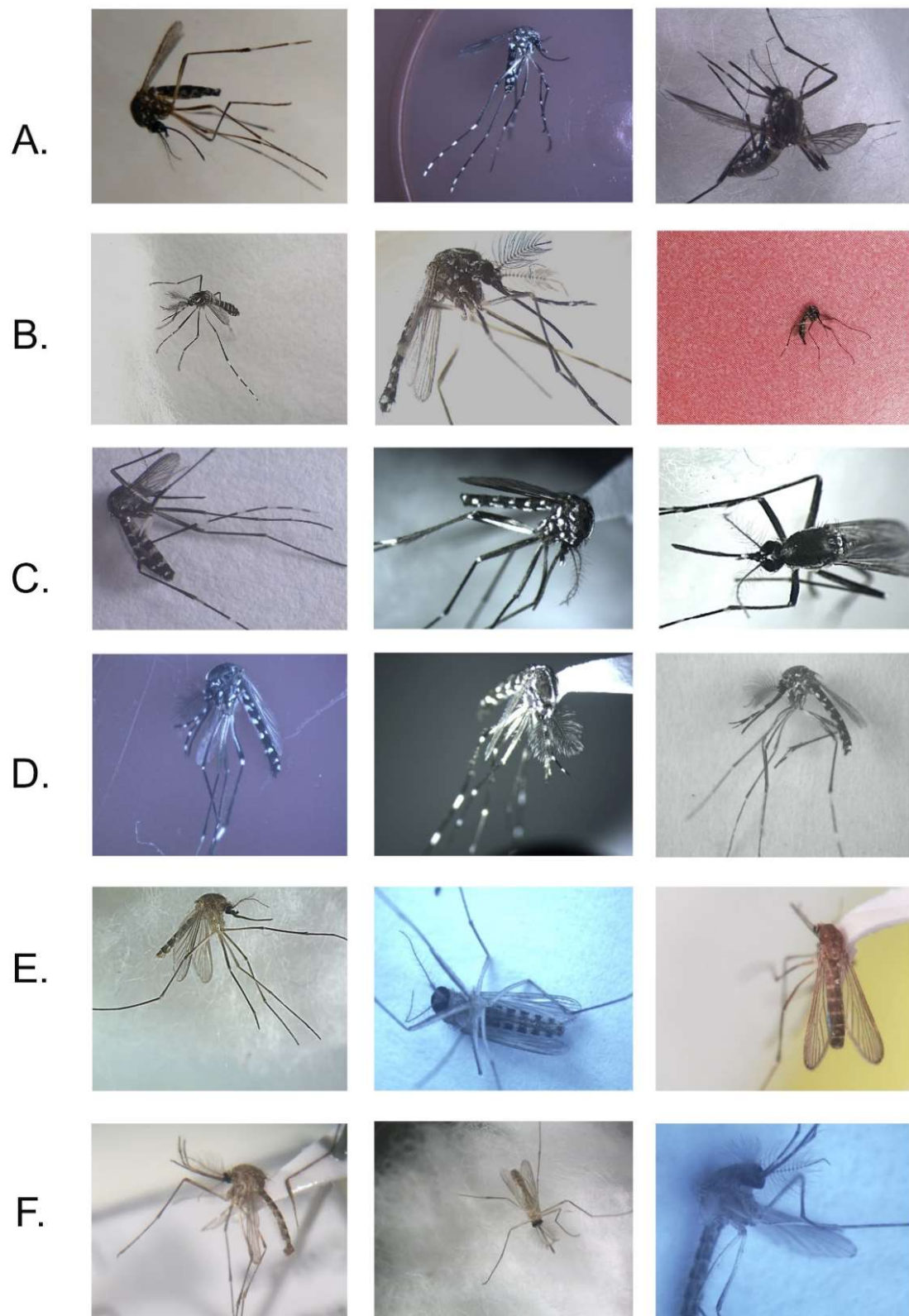


Figura 1. Amostra de Imagens utilizadas para treinamento, validação e teste do modelo: (A) *Ae. aegypti* fêmea; (B) *Ae. aegypti* macho; (C) *Ae. albopictus* fêmea; (D) *Ae. albopictus* macho; (E) *C. quinquefasciatus* fêmea; (F) *C. quinquefasciatus* macho.

5.3.4 Divisão das imagens em treinamento, validação e teste

A metodologia utilizada para desenvolvimento do modelo de classificação automática dos mosquitos seguiu a metodologia de treinamento da rede proposta por Motta et al (2019).

O banco de dados foi dividido para ser utilizado em três fases distintas: treinamento, validação e teste; (a) treinamento: nesta fase o conjunto de dados é apresentado ao algoritmo com seus rótulos, desta forma a máquina é treinada com os atributos de cada classe; (b) validação: é realizada ao mesmo tempo que a fase de treinamento, servindo para medir a qualidade do treinamento com dados nunca vistos. Nesta fase, a classificação é avaliada e os pesos da rede são ajustados com o objetivo de aumentar a acurácia e diminuir a perda; (c) teste: um novo conjunto de dados é apresentado ao modelo já treinado e validado. Não há ajustes de pesos nessa fase, mas apenas a avaliação da acurácia da classificação.

Neste trabalho foi adotada a divisão entre as três fases da seguinte forma: 60% das imagens utilizadas para a fase de treinamento; 20% para validação e 20% para teste. As imagens foram divididas em cada fase aleatoriamente, de acordo com os percentuais definidos.

5.3.5 Indicadores de avaliação de desempenho do modelo

5.3.5.1 Fases de treinamento e validação

Durante as etapas de treinamento e validação, os indicadores utilizados para avaliação de desempenho do modelo foram: função perda e acurácia. A função perda utilizada neste trabalho foi a função de custo de entropia cruzada (*cross entropy*). O cálculo desta função perda, tanto na fase de treinamento, como na fase de validação foi realizado computacionalmente como uma função do *Keras* e sua forma de cálculo está apresentada na equação 1 (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$H(P, Q) = -E_{X \sim P}[\log Q(x)] \quad (1)$$

Onde, H é a entropia cruzada, P é o valor real, Q é o valor previsto, x é o sinal de entrada e $E_{X \sim P}[\log Q(x)]$ corresponde à expectativa de $\log Q(x)$ em relação à $P(x)$. O cálculo de acurácia, assim como a função perda, foi realizado computacionalmente como uma função do *Keras*.

5.3.5.2 Fase de teste

Na fase de teste, o desempenho do modelo foi avaliado em uma matriz de confusão. Destaca-se que o banco de dados utilizado neste trabalho foi desbalanceado, ou seja, cada uma das classes teve quantidades diferentes no banco de dados e, portanto, um peso diferente no resultado global.

Desta forma, optou-se por aplicar uma métrica que ponderasse o efeito desse desbalanceamento e, definiu-se por utilizar a métrica de acurácia balanceada (*Balanced Accuracy*), conforme proposto por Straube & Krell (2014).

O Quadro 1 apresenta uma matriz de confusão 2x2. As equações 2, 3, 4 e 5 apresentam as formas de cálculo dos indicadores desta matriz.

Quadro 1: Matriz de Confusão 2x2.

Fonte: (STRAUBE; KRELL, 2014).

		Valor Previsto	
		Positivo	Negativo
Valor Real	Positivo	VP	FN
	Negativo	FP	VN

$$\text{Precisão: } PR = \frac{VP}{VP+FP} \quad (2)$$

$$\text{Taxa Verdadeiro-Positivo: } TVP = \frac{VP}{VP+FN} \quad (3)$$

$$\text{Taxa Verdadeiro-Negativo: } TVN = \frac{VN}{VN+FP} \quad (4)$$

$$\text{Acurácia Balanceada: } BA = 0,5 \times (TVP + TVN) \quad (5)$$

Adicionalmente, neste trabalho, o processo de classificação também considerou matrizes acima de duas classes. A forma de cálculo dos indicadores, nestes

casos, deve ser alterada em relação à uma matriz 2x2. Para a obtenção de um indicador único de avaliação de desempenho, utilizou-se primeiramente à análise de desempenho para cada classe, conforme as equações apresentadas no quadro 2. Para os casos acima de duas classes, o resultado global de acurácia do modelo foi considerado como sendo a média das acurácias balanceadas de cada classe.

Quadro 2: Matriz de Confusão acima de duas classes.

		Valor Previsto			TVP	TVN	BA
		Classe 1	Classe 2	Classe 3			
Valor Real	Classe 1	a	b	c	$\frac{a}{a+b+c}$	$\frac{e+f+h+i}{d+e+f+g+h+i}$	$BA^1 = \frac{TVP + TVN}{2}$
	Classe 2	d	e	f	$\frac{e}{d+e+f}$	$\frac{a+c+g+i}{a+b+c+g+h+i}$	$BA^2 = \frac{TVP + TVN}{2}$
	Classe 3	g	h	i	$\frac{i}{g+h+i}$	$\frac{a+b+d+e}{a+b+d+e+g+h}$	$BA^3 = \frac{TVP + TVN}{2}$
Precisão		$\frac{a}{a+d+g}$	$\frac{e}{b+e+h}$	$\frac{i}{c+f+i}$			$\frac{BA^1 + BA^2 + BA^3}{3}$

5.3.6 Aumento dos dados (*data augmentation*)

Utilizou-se a estratégia de ampliar computacionalmente o banco de dados das fases de treinamento e validação, a partir das imagens originais obtidas. O banco de dados de teste não foi ampliado.

Diferentes parâmetros (rotação, deslocamento de largura, deslocamento de altura, brilho, ângulo de corte e zoom) foram testados aleatoriamente e após uma avaliação do desempenho do modelo, definiu-se, por realizar uma pesquisa de grade, em duas etapas.

Na primeira etapa, definiu-se o melhor conjunto de hiperparâmetros para geração de novas imagens, conforme experimentos da Tabela 2. Inicialmente foi determinada variação das imagens em relação à rotação, deslocamento de largura e deslocamento de altura. Posteriormente houve a definição da variação da rotação em dois valores (45° e 90°) e dos deslocamentos de largura e altura em três valores (5%, 15% e 25%). Por fim, foi realizado o treinamento da rede, variando os hiperparâmetros em seis diferentes conjuntos e comparando os resultados com o treinamento da rede sem ampliação do banco de dados. A

partir dos resultados, o conjunto de hiperparâmetros com melhor desempenho em relação à acurácia e à função perda, ambas na fase de validação, foi definido como padrão a ser usado na segunda etapa.

Tabela 2. Pesquisa de grade – hiperparâmetros para ampliação das imagens no banco de dados.

Hiperparâmetros	Experimentos						
	1°	2°	3°	4°	5°	6°	7°
Rotação da imagem	Sem aumento	45°	45°	45°	90°	90°	90°
Deslocamento de largura da imagem		5%	15%	25%	5%	15%	25%
Deslocamento de altura da imagem		5%	15%	25%	5%	15%	25%

Cabe ressaltar que todos os outros hiperparâmetros do modelo foram mantidos fixos nesta etapa. A arquitetura de CNN utilizada foi a DenseNet201, o tamanho de lote (*batch size*) foi 32, o número de épocas foi 200, a taxa de aprendizagem inicial foi 2E-5 e o otimizador foi o Adam.

Na segunda etapa, foi avaliado em quantas vezes os dados deveriam ser aumentados. Desta forma, utilizando o conjunto de valores selecionado na primeira etapa, o banco de dados foi aumentado em 5, 10, 15 e 20 vezes. O multiplicador com o melhor desempenho, considerando os mesmos critérios da etapa anterior, foi selecionado para o desenvolvimento do modelo otimizado.

5.3.7 Definição da arquitetura de CNN e otimização dos hiperparâmetros das camadas de classificação

A biblioteca do *Keras* (“Keras Documentation”, 2019) foi utilizada com arquiteturas de rede pré-treinadas. A utilização de uma CNN pré-treinada, também conhecido como *transfer learning*, foi definido para melhorar o desempenho de classificação de imagens, uma vez que seus pesos já foram otimizados com atributos que são importantes na maioria dos problemas de visão computacional.

O processo de seleção da CNN e otimização dos hiperparâmetros foi desenvolvido em cinco etapas. Primeiramente, de forma aleatória, avaliou-se o desempenho de algumas arquiteturas de rede e de alguns hiperparâmetros no

processo de classificação automática de mosquitos, em relação à acurácia e à perda nas etapas de treinamento e validação.

Posteriormente, 16 experimentos foram conduzidos, conforme apresentado na Tabela 3, para avaliar o desempenho das arquiteturas Xception e DenseNet201, bem como a variação dos hiperparâmetros das camadas de classificação do modelo (número de épocas e taxa de aprendizagem). Considerando o comportamento das redes na etapa de pesquisa aleatória, definiu-se que nessa etapa de otimização as redes Xception e DenseNet201 seriam as únicas a serem utilizadas.

Tabela 3. Pesquisa de grade – CNN para extração dos atributos das imagens e hiperparâmetros para as camadas de classificação.

CNN e Hiperp.	Experimentos															
	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°	11°	12°	13°	14°	15°	16°
CNN	Xcp	Xcp	Xcp	Xcp	Xcp	Den	Den	Den	Den	Den	Den	Den	Den	Den	Den	Den
Épocas	200	200	200	200	300	200	200	200	200	200	200	200	200	200	200	300
LR	4e-7	4e-6	2e-5	1e-4	4e-6	4e-6	1e-5	2e-5	3e-5	6e-5	1e-4	2e-4	3e-4	6e-4	1e-3	4E-6

Épocas: número de épocas; LR: taxa de aprendizagem inicial; Xcp: rede Xception; Den: rede DenseNet201.

A partir dos resultados de desempenho dos hiperparâmetros da Tabela 3, os cinco melhores desempenhos em relação à acurácia na fase de validação e os cinco melhores em relação à função perda na fase de validação foram selecionados para avaliação da classificação com os bancos de dados de testes.

Considerando que os valores dos pesos iniciais atribuídos em um treinamento de rede neural iniciam-se de forma aleatória e, que, essa aleatoriedade pode alterar o resultado final de desempenho, definiu-se por submeter os dois melhores modelos, em relação à acurácia balanceada na fase de teste, à dois treinamentos adicionais. Desta forma, foi possível avaliar essa variação e o seu impacto na seleção do conjunto de hiperparâmetros.

Por fim, avaliou-se o desempenho do modelo otimizado, com melhor desempenho na fase de teste, com a alteração do tamanho de lote, variando-o com os valores 32, 64, 128 e 256.

5.3.8 Avaliação do modelo otimizado

Com o intuito de avaliar a aplicabilidade do modelo desenvolvido para utilização por pessoas não especialistas, e, tendo em vista que a classificação da espécie pode não ser o requisito mais importante para a classe de não especialistas, o modelo também foi treinado para a detecção de gênero e, desta forma, permitir a classificação adequada do gênero *Aedes*. Além disso, destaca-se que para os programas de saúde pública envolvidos na prevenção das arboviroses, detectar e georeferenciar o gênero *Aedes*, independente da espécie, já se configura em um importante resultado. Outro ponto importante para avaliação do modelo é a sua capacidade em reconhecer outros insetos e distingui-los dos mosquitos-alvo utilizados neste estudo.

Desta forma, o modelo otimizado foi utilizado para classificação de imagens em três cenários distintos: (a) seis classes – neste caso a classificação das três espécies, bem como a diferenciação entre machos e fêmeas foi avaliada; (b) duas classes – neste caso houve a diferenciação entre *Aedes* e *Culex*; (c) sete classes – neste caso incluiu-se uma categoria de outros para avaliar o comportamento da rede quando uma imagem de outro inseto (não mosquito) é apresentada.

Para avaliar o desempenho da rede na distinção de seis classes, a classe ‘outros’ (Tabela 1) foi excluída do treinamento do modelo. Cabe ressaltar que esta foi a configuração utilizada no processo de otimização. Para avaliar o desempenho da rede na distinção de gênero, todos os mosquitos das classes *Ae. aegypti* fêmea, *Ae. aegypti* macho, *Ae. albopictus* fêmea, *Ae. albopictus* macho foram rotulados como *Aedes*, e todos os mosquitos das classes *C. quinquefasciatus* fêmea e *C. quinquefasciatus* macho foram rotulados como *Culex*. Para avaliar o desempenho da rede na distinção de sete classes, a classe ‘outros’ (Tabela 1) foi incluída no treinamento do modelo.

5.3.9 Avaliação da influência do ângulo da imagem

A influência do ângulo do mosquito na imagem sobre o desempenho na predição do modelo também foi avaliada neste estudo. Desta forma, um especialista em

fotografia separou o banco de dados de teste das seis classes estudadas (*Ae. aegypti* fêmea, *Ae. aegypti* macho, *Ae. albopictus* fêmea, *Ae. albopictus* macho, *C. quinquefasciatus* fêmea e *C. quinquefasciatus* macho) em quatro diferentes conjuntos: (A) ângulo frontal e de baixo, que considera as fotos tiradas de frente da região da cabeça e abaixo do tórax do mosquito; (B) ângulo de costas, que consiste na região das asas; (C) ângulo direito e (D) ângulo esquerdo.

Na Figura 2 está apresentada uma amostra de cada um dos quatro conjuntos utilizados durante a etapa de avaliação da influência do ângulo de imagens. O banco de dados de testes apresentou 344 imagens com ângulo frontal e de baixo, 287 imagens com ângulo de costas, 460 com ângulo direito e 419 com ângulo esquerdo.

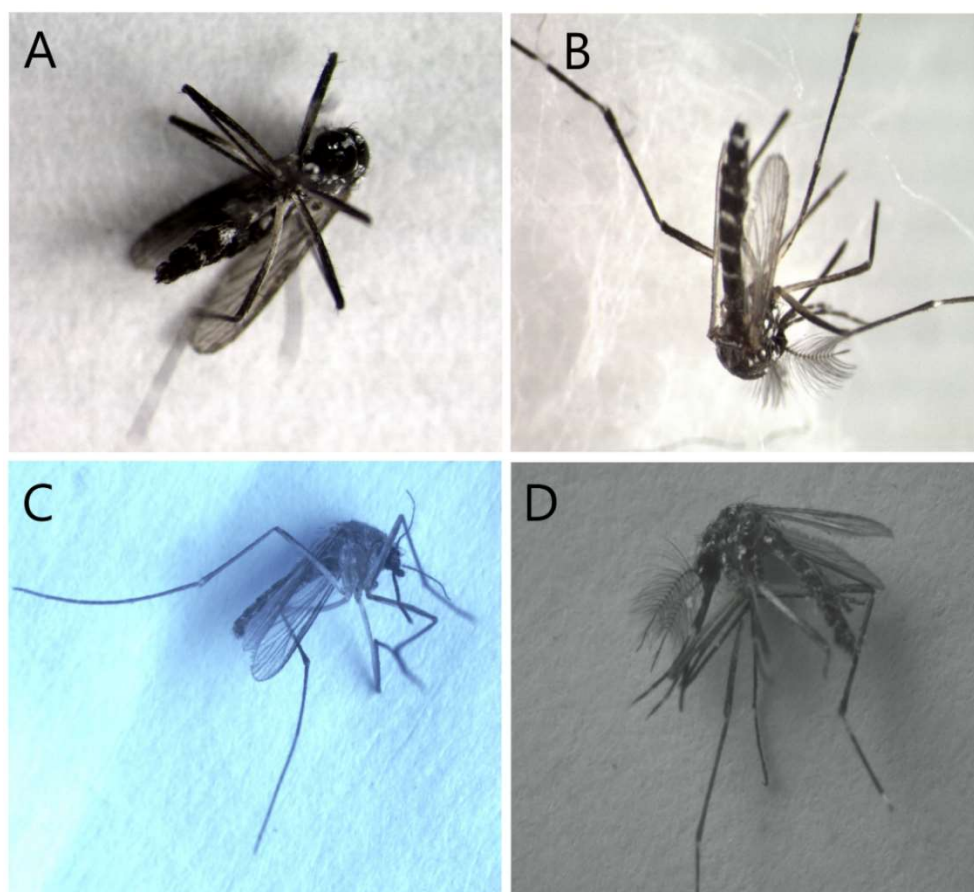


Figura 2. Amostras de imagens com diferentes ângulos dos mosquitos:(A): ângulo de baixo; (B): ângulo de costas; (C): ângulo direito; (D): ângulo esquerdo.

O modelo otimizado foi utilizado para a classificação das imagens do banco de dados de teste de cada um dos quatro diferentes conjuntos.

5.4 MODELO COMPUTACIONAL

5.4.1 Programas

Foram elaborados cinco códigos de programas, conforme apresentado na Figura 3. A linguagem computacional escolhida para a programação foi python e a interface de programação de aplicativos (*framework*) foi o *Keras*.

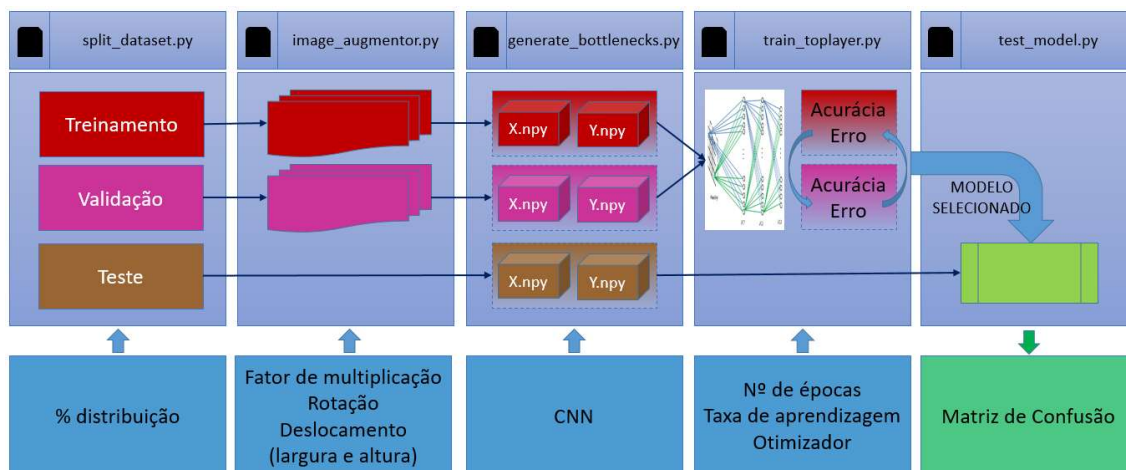


Figura 3. Programas elaborados e utilizados para treinamento e avaliação do modelo de classificação.

Foram utilizados, para execução dos programas, os nós de GPU (*Graphics Processing Unit*) do supercomputador Ogun (Bull Atos, Sequana X410, França), disponível no Centro de Supercomputação para a Inovação Industrial (CS2I) do Centro Universitário SENAI CIMATEC (Salvador, Brasil).

5.4.1.1 SPLIT_DATASET.PY

O `split_dataset.py` foi utilizado para dividir aleatoriamente as classes de imagens em três blocos: treinamento, validação e teste. As linhas de código deste programa encontram-se detalhadas no Anexo I.

5.4.1.2 IMAGE_AUGMENTOR.PY

O `image_augmentor.py` foi utilizado para realizar a ampliação do banco de dados, entretanto, neste estudo, somente as imagens de treinamento e

validação foram ampliadas. As linhas de código deste programa encontram-se detalhadas no Anexo II.

A geração de novas imagens levou em média 5 horas para ampliação do banco de imagens de treinamento e 2 horas para ampliação do banco de imagens de validação, considerando a ampliação de imagens para seis classes em 20 vezes.

5.4.1.3 GENERATE_BOTTLENECKS.PY

O `generate_bottlenecks.py` foi utilizado para extrair os atributos das imagens. Para cada uma das três fases do desenvolvimento do modelo (treinamento, validação e teste), foram geradas dois vetores (X e Y), sendo X o vetor de atributos das imagens que caracterizam o sinal de saída presente no vetor Y , de acordo com o seu rótulo. Os *bottlenecks* de atributos X são os vetores dos atributos das imagens processados pela rede e os *bottlenecks* de atributos Y são os vetores que representam as imagens com seus rótulos.

Nesta etapa, arquitetura de CNN foi selecionada. O programa, detalhado no Anexo III, foi desenvolvido para permitir a utilização de quaisquer dos modelos pré-treinados disponibilizados atualmente pela biblioteca do *Keras* (“Keras Documentation”, 2019).

O tempo médio de geração dos *bottlenecks* na fase de treinamento foi de 30 horas, na fase de validação foi de 2,5 horas e na fase de teste foi de 10 minutos. Este tempo representa uma média para cada vez que se utiliza o programa e considerando a geração de *bottlenecks* para seis classes.

5.4.1.4 TRAIN_TOPLAYER.PY

O `train_toplayer.py` representa as camadas de classificação. Nesta fase, os mapas de atributo (*feature maps*) 2D foram convertidos em um vetor de atributo (*feature vector*) 1D e realizam o processo de classificação ou reconhecimento do objeto. Além de utilizarem os *bottlenecks* de atributos gerados pelo programa `generate_bottlenecks.py`, nesta etapa selecionou-se os valores de

hiperparâmetros que impactam diretamente no processo de aprendizado da rede, como número de épocas, taxa de aprendizagem inicial e otimizador.

O programa também contou com estratégias para redução do risco de *overfitting* e melhoria de desempenho no processo de classificação. A função *dropout* foi utilizada para desativar aleatoriamente alguns neurônios e, desta forma, prevenir *overfitting* e a função *Reduce Learning Rate on Plateau* foi utilizada para reduzir automaticamente a taxa de aprendizagem, caso o desempenho da rede não melhorasse após um determinado número de épocas. Para o uso desta função foi necessário definir a variável de desempenho a ser monitorada (*monitor*), o fator de redução da taxa de aprendizagem (*factor*), o número máximo de ciclos sem melhoria do desempenho da variável monitorada, antes de se aplicar o fator de redução da taxa de aprendizagem (*patience*) e o valor mínimo limite para a taxa de aprendizagem (*min_lr*).

Ao rodar este programa foi possível salvar o modelo após as fases de treinamento e validação. Isso foi importante para permitir o uso do modelo com os *bottlenecks* de atributos do banco de dados de teste e, assim, avaliar o desempenho da rede para classificação de imagens totalmente novas ao modelo. As linhas de código deste programa encontram-se detalhada no Anexo IV. O tempo médio de treinamento da rede, para 200 épocas, foi de 2 horas.

5.4.1.5 TEST_MODEL.PY

O `test_model.py`, detalhado no Anexo V, utilizou o modelo salvo com os pesos treinados para predição do banco de dados da fase de teste. Nesta fase um novo conjunto de dados foi apresentado e uma matriz de confusão foi gerada, permitindo a avaliação do desempenho e sua capacidade de generalização para cada classe.

5.5 RESULTADOS E DISCUSSÕES

5.5.1 Pesquisa aleatória (*random search*)

A definição do plano de pesquisa de grade para otimização dos hiperparâmetros de aprendizagem do modelo foi precedida de uma pesquisa aleatória. Esta pesquisa teve a função de sustentar e direcionar a definição dos hiperparâmetros que foram variados e o alcance para essa variação. O banco de dados com seis classes (7.561 imagens) foi o utilizado para otimização do modelo desenvolvido. A Tabela 4 apresenta os resultados após a última época de aprendizagem da rede para os 24 experimentos realizados durante a pesquisa aleatória.

Para a interpretação dos resultados, a avaliação dos dados da Tabela 4 foi complementada com a análise do comportamento temporal de aprendizagem de cada rede. As Figuras 4, 5, 6, 7 e 8 são apresentadas como um resumo dos experimentos realizados durante a pesquisa aleatória e exemplificam os critérios usados para realização da pesquisa de grade. São discutidos os resultados considerados mais relevantes.

Tabela 4. Pesquisa aleatória para definição dos hiperparâmetros da pesquisa de grade.

#	M	CNN	Épocas	Otimiz.	LR	Resultados			
						L_t	Acc_t (%)	L_v	Acc_v (%)
1	10*	ResNet50	200	SGD	1e-3	1,77	21,0	1,77	21,0
2	0	ResNet50	200	SGD	1e-3	1,65	28,8	1,64	28,3
3	0	ResNet50	200	RMSprop	1e-3	1,48	39,3	1,50	36,8
4	0	ResNet50	200	Adagrad	1e-3	1,54	36,6	1,55	35,7
5	0	Xception	200	RMSprop	1e-3	0,06	98,3	1,41	79,9
6	0	VGG16	200	RMSprop	1e-3	0,42	83,4	0,81	75,1
7	0	DenseNet201	200	RMSprop	1e-3	0,01	99,6	1,01	87,2
8	0	DenseNet201	120	RMSprop	1e-3	0,02	99,4	0,96	87,2
9	2*	VGG16	200	RMSprop	1e-3	1,69	26,1	1,77	24,7
10	0	VGG16	300	RMSprop	1e-3	0,38	86,1	0,86	75,1
11	0	VGG16	200	RMSprop	1e-2	1,77	21,3	1,77	21,3
12	0	VGG16	200	RMSprop	1e-4	0,62	77,6	0,80	71,1
13	0	InceptionV3	200	RMSprop	1e-3	0,06	97,6	1,40	79,7
14	0	Xception	200	Adam	1e-3	0,04	98,8	1,34	80,3
15	0	Xception	200	Adam	1e-4	0,11	97,1	0,72	79,8
16	0	Xception	200	Adam	1e-5	0,48	82,8	0,63	76,0
17	2	Xception	200	Adam	5e-5	0,24	91,7	0,65	78,4
18	2	DenseNet201	200	Adam	5e-5	0,24	91,3	0,62	79,2
19	2	DenseNet201	200	Adam	1e-5	0,44	83,6	0,61	77,6
20	2	DenseNet201	200	Adam	2e-5	0,33	87,8	0,59	78,8
21	2	VGG16	200	Adam	1e-3	0,54	78,8	0,99	66,5
22	2	VGG16	200	Adam	1e-4	0,76	71,1	0,95	63,1

23	10	DenseNet201	200	Adam	2e-5	0,21	92,3	0,42	85,9
24	15	DenseNet201	200	Adam	2e-5	0,19	92,9	0,41	86,9

* Ampliação do *dataset* com alteração indiscriminada de hiperparâmetros.

#: Experimentos; M: fator de multiplicação da ampliação do dataset; CNN: arquitetura de rede; Épocas: nº de épocas; Otimiz.: Otimizador para classificação; LR: taxa de aprendizagem; L_t: perda (cross entropy) na fase de treinamento; Acc_t: acurácia na fase de treinamento; L_v: perda (cross entropy) na fase de validação; Acc_v: acurácia na fase de validação.

A Figura 4 apresenta o comportamento da perda e da acurácia na fase de validação do experimento 3 (Tabela 4), com a utilização da ResNet50.

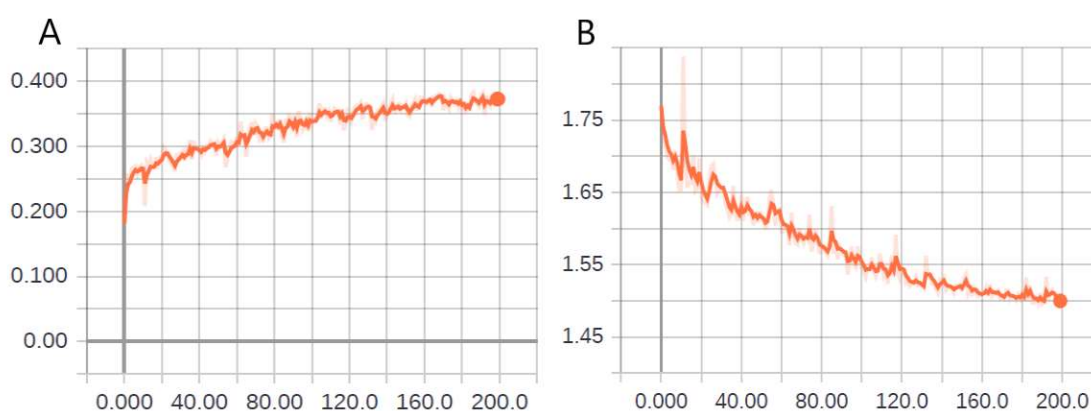


Figura 4. Evolução da aprendizagem da rede ResNet50: (A): acurácia na fase de validação; (B): perda (*cross entropy*) na fase de validação.

Apesar da ResNet (*Residual Network*) ter sido considerada a melhor rede no desafio ILSVRC-2015 na categoria de classificação de objetos com a margem de erro-top-5 de 3,57% (ALOM et al., 2018; HE et al., 2015; RUSSAKOVSKY et al., 2015), para a classificação de mosquitos adultos, utilizados no desenvolvimento do modelo, esta rede não apresentou bom desempenho (Tabela 4). Entretanto, cabe ressaltar, que não houve uma otimização para o uso da ResNet, o que pode justificar o desempenho inadequado para o modelo proposto. Desta forma, por conta do desempenho apresentado nos experimentos 2, 3 e 4 (Tabela 4), usando diferentes otimizadores (SGD, RMSprop e Adagrad respectivamente), esta rede não foi incluída na pesquisa de grade.

Importante destacar que nas Figuras 4A e 4B, há um indicativo de *underfitting*, ou seja, com um maior número de épocas, esta rede tende a melhorar o seu desempenho.

Uma outra avaliação importante nesta etapa de pesquisa aleatória foi com relação ao impacto da ampliação do banco de dados. A Figura 5 apresenta o comportamento da perda e da acurácia na fase de validação dos experimentos 6 e 9 (Tabela 4), utilizando a VGG16. A diferença entre os experimentos 6 e 9 está relacionada à ampliação do banco de imagens.

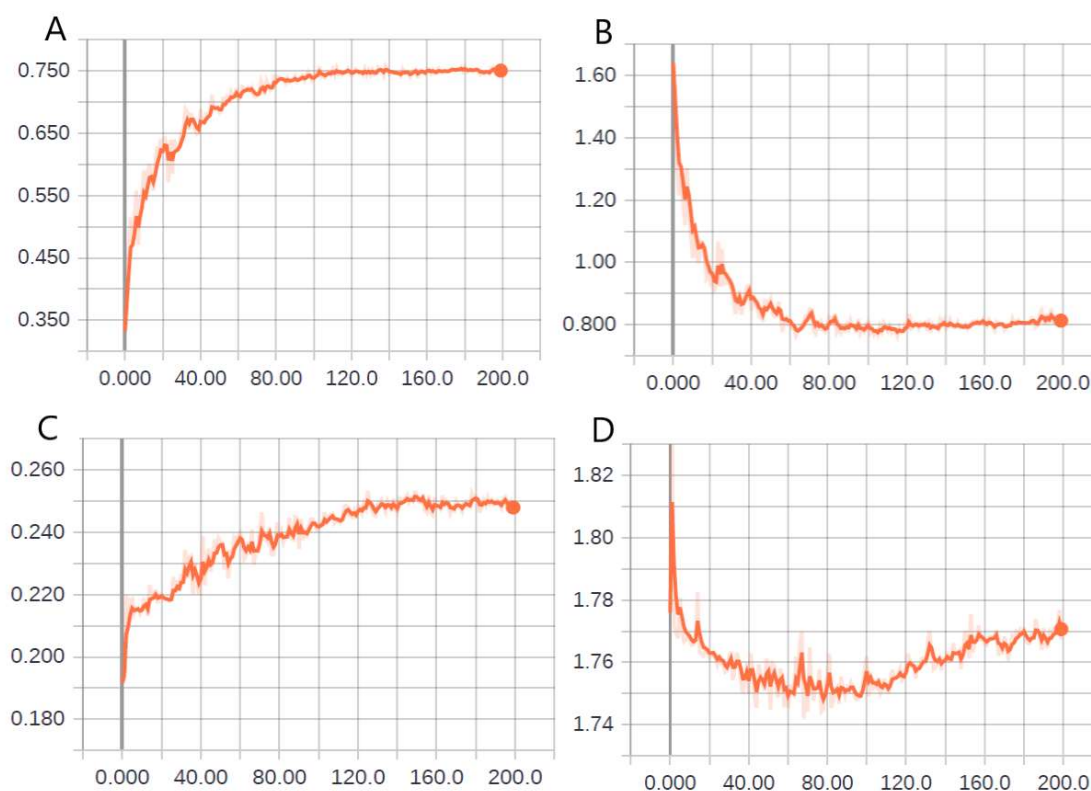


Figura 5. Evolução da aprendizagem da rede com o uso da VGG16: (A): acurácia na fase de validação – experimento 6; (B): perda (*cross entropy*) na fase de validação – experimento 6; (C): acurácia na fase de validação – experimento 9; (D): perda (*cross entropy*) na fase de validação – experimento 9.

As Figuras 5A e 5B não apresentam ampliação do banco de dados, enquanto que nas Figuras 5C e 5D, o banco de imagens foi aumentado em 2 vezes. Desta forma, foi evidenciado que o desempenho da rede diminuiu após o aumento do banco de dados. Os benefícios da ampliação de dados no desempenho de tarefas de classificação já foram demonstrados em trabalhos recentes. Wong et

al. (2016), avaliaram experimentalmente os benefícios de aumento de dados para uma rede neural convolucional, para uma máquina suporte a vetor convolucional e para um classificador de máquina de aprendizagem extrema convolucional, usando o conjunto de dados de dígitos manuscritos padrão do MNIST. A ampliação de dados resultou em aumento de desempenho e redução de *overfitting*.

Portanto, o resultado obtido com o experimento 9 (Tabela 4) passou por uma avaliação criteriosa das imagens geradas computacionalmente. Verificou-se que a maioria das imagens não apresentavam mosquitos, e, a partir desta análise, definiu-se por revisar o número de hiperparâmetros e sua amplitude para geração de novas imagens para o banco de dados, além de validar as imagens geradas ao final da ampliação. A ampliação das imagens a partir do experimento 17 seguiu uma nova configuração, e, o resultado desta ampliação, ainda na pesquisa aleatória, demonstrou que a ampliação do banco de dados pode melhorar o desempenho da rede na classificação de imagens de mosquitos adultos.

A Figura 6 apresenta o comportamento da perda e da acurácia na fase de validação dos experimentos 5 e 7 (Tabela 4), com o uso das redes Xception e DenseNet201 respectivamente. Após um número reduzido de épocas o valor mínimo de perda (*cross entropy*) é alcançado e a partir desse ponto (em torno de 20 épocas) o erro na fase de validação começa a aumentar a cada interação de aprendizagem, sem, contudo, impactar negativamente na acurácia do modelo.

Normalmente espera-se uma correlação inversa entre acurácia e perda. Quanto maior a acurácia, menor a perda. Contudo um efeito que pode acontecer durante o processo de aprendizado de uma rede neural é o aumento da função perda e, ainda assim, a manutenção ou aumento do valor da acurácia (Figura 6). O correto entendimento deste fenômeno é importante para a tomada de decisão sobre qual o “melhor” conjunto de parâmetros para uma determinada aplicação.

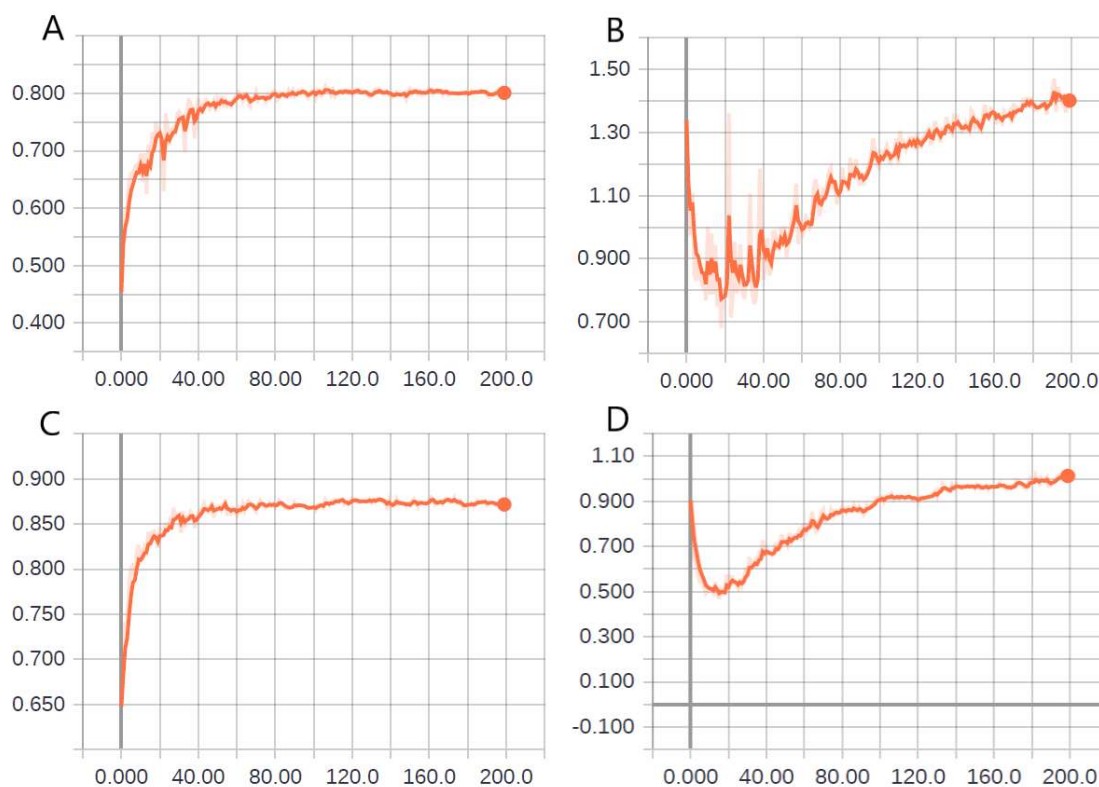


Figura 6. Evolução da aprendizagem das redes Xception e DenseNet201: (A): acurácia na fase de validação – experimento 5; (B): perda (*cross entropy*) na fase de validação – experimento 5; (C): acurácia na fase de validação – experimento 7; (D): perda (*cross entropy*) na fase de validação – experimento 7.

A função perda mede a distância entre o valor real e o valor previsto. O valor previsto é probabilístico, já o real é determinístico. Considerando uma situação hipotética, tem-se uma imagem de um mosquito *Ae. aegypti* macho. O valor real considera que a imagem apresentada é 100% de um *Ae. aegypti* macho. Quando esta imagem é apresentada ao algoritmo de predição, esta avaliação pode variar entre 0% e 100%. A função perda mede justamente essa variação entre o valor real e o valor previsto.

A acurácia, por sua vez, mede o valor previsto como sendo determinístico. Ao avaliar a mesma imagem anterior, se o algoritmo classifica a imagem como sendo 80% de probabilidade de ser um *Ae. aegypti* macho e 20% de ser um *Ae. albopictus* fêmea, para o valor de acurácia, o modelo teve 100% de acerto.

Portanto, se em uma determinada época de treinamento da rede, o modelo prevê que uma imagem é de 80% de ser *Ae. aegypti* macho e na época subsequente, o mesmo modelo prevê que uma imagem é de 60% de ser *Ae. aegypti* macho, a função perda aumenta (piora), mas a acurácia permanece inalterada.

Esse efeito não deve ser confundido com *overfitting*. *Overfitting* está relacionado à incapacidade de generalização do modelo para aplicação com novos dados, não usados no treinamento da rede (GOODFELLOW; BENGIO; COURVILLE, 2016). Para detectar *overfitting* é importante avaliar o desempenho na classificação das imagens ao se utilizar o banco de dados de teste.

Uma alternativa para minimizar o efeito de aumento da função perda é reduzir a taxa de aprendizagem inicial. O desempenho de método baseado em *deep learning* depende muito da seleção e evolução da taxa de aprendizagem e pode aumentar significativamente o processo de treinamento (CHIN et al., 2015). A Figura 7 apresenta graficamente o efeito da redução da taxa de aprendizagem inicial no uso da rede Xception.

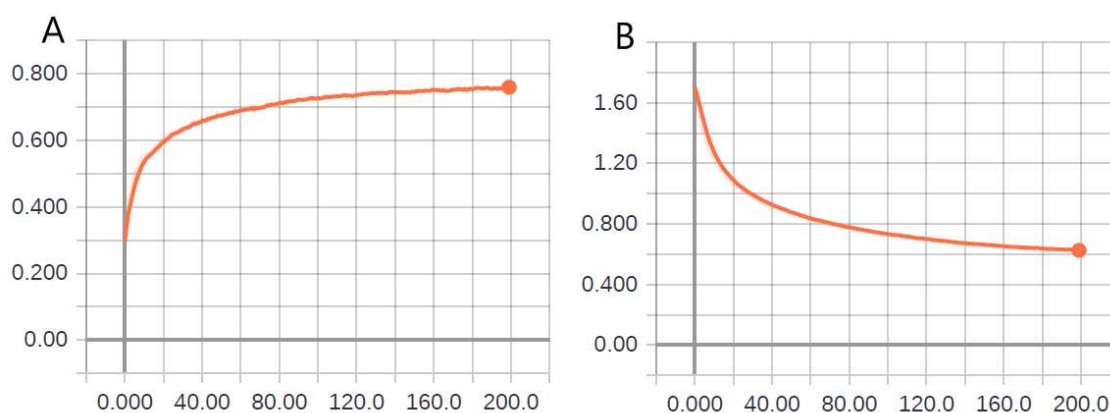


Figura 7. Evolução da aprendizagem da rede Xception, após a redução da taxa de aprendizagem inicial para 1E-5 (experimento 16): (A): acurácia na fase de validação; (B): perda (*cross entropy*) na fase de validação.

Contudo, a redução da taxa inicial altera a velocidade de aprendizagem da rede e, por consequência, o seu desempenho final. Portanto, a investigação dos efeitos de alteração de taxa de aprendizagem inicial foi realizada na etapa de pesquisa de grade. Também pode ser observado na Figura 7 o efeito de

underfitting, reforçando a importância de avaliar, além da taxa de aprendizagem inicial, o número de épocas de treinamento da rede.

A Figura 8 apresenta o comportamento da perda e da acurácia na fase de validação dos experimentos 20 e 24 (Tabela 4), com a utilização da rede DenseNet201. Constatou-se que a ampliação do banco de dados melhora o desempenho da rede no processo de classificação, e além disso, acelera o processo de aprendizado, reforçando os benefícios da ampliação de dados no desempenho de tarefas de classificação (WONG et al., 2016). Nas Figuras 8A e 8B, observa-se a presença do efeito de *underfitting*, enquanto que nas Figuras 8C e 8D este efeito não está presente.

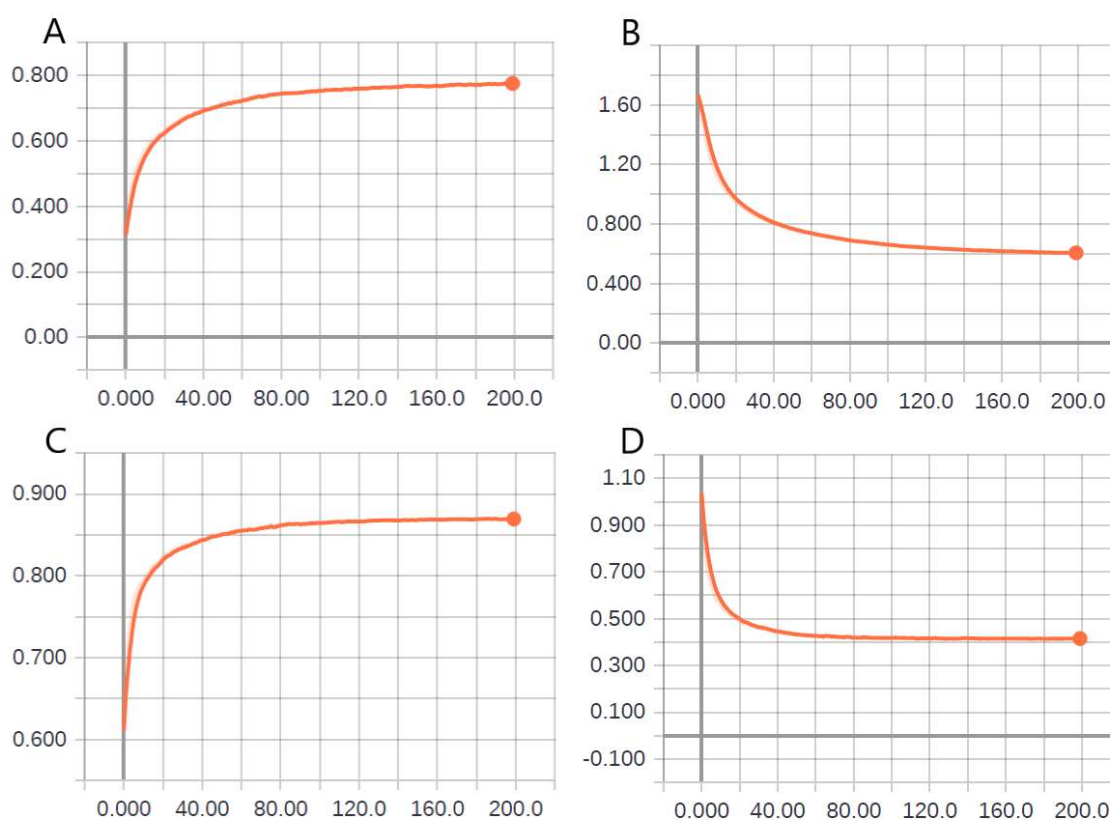


Figura 8. Evolução da aprendizagem da rede DenseNet, após a ampliação do *dataset*: (A): acurácia na fase de validação – experimento 20; (B): perda (*cross entropy*) na fase de validação – experimento 20; (C): acurácia na fase de validação – experimento 24; (D): perda (*cross entropy*) na fase de validação – experimento 24.

5.5.2 Otimização dos hiperparâmetros para ampliação dos dados (*data augmentation*)

A otimização dos hiperparâmetros de ampliação dos dados foi realizada em duas etapas. Na primeira etapa, definiu-se o melhor conjunto de hiperparâmetros para geração de novas imagens e na segunda foi avaliado em quantas vezes os dados deveriam ser aumentados.

As Figuras 9 e 10 apresentam respectivamente o comportamento da acurácia na classificação e da função perda (*cross entropy*) nas fases de validação dos experimentos nas duas etapas de otimização dos hiperparâmetros para geração de novas imagens.

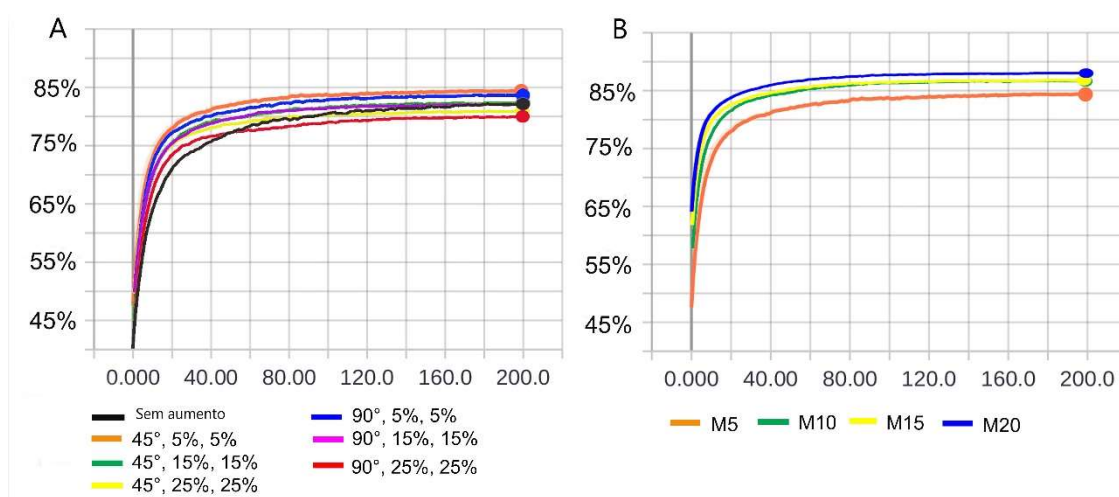


Figura 9. Impacto da variação dos hiperparâmetros de ampliação de dados na acurácia na fase de validação: (A): etapa 1 da otimização; (B): etapa 2 da otimização.

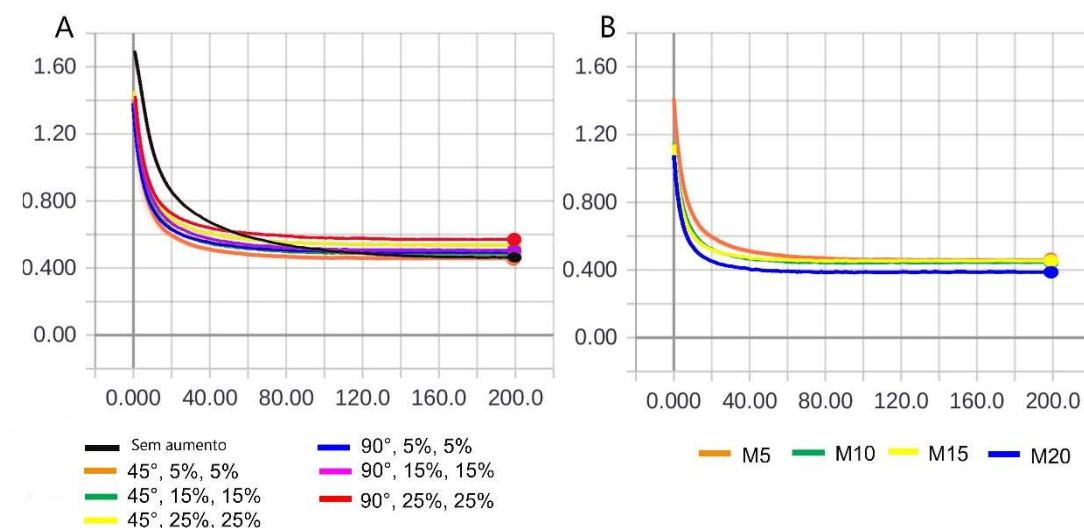


Figura 10. Impacto da variação dos hiperparâmetros de ampliação de dados na função perda (*cross entropy*) na fase de validação: (A): etapa 1 da otimização; (B): etapa 2 da otimização.

Na primeira etapa da otimização (Figuras 9A e 10A), em relação ao desempenho dos conjuntos avaliados, os melhores resultados para acurácia e função perda foi determinado para 45° de rotação das imagens, 5% de deslocamento de largura das imagens e 5% de deslocamento de altura das imagens. Por conta deste desempenho, este foi o conjunto de valores escolhido para ampliação das imagens na segunda etapa no treinamento do modelo otimizado. Cabe ressaltar que a aplicação de cada um dos valores é feita de forma aleatória no banco de dados, o que significa que nem todas as imagens foram ampliadas da mesma forma.

Na segunda etapa da otimização (Figuras 9B e 10B), o melhor desempenho em relação à acurácia da classificação e à função perda (*cross entropy*) foi obtido com o fator de multiplicação 20, reforçando, uma vez mais, que a ampliação de dados traz benefícios ao desempenho de modelos nas tarefas de classificação (WONG et al., 2016). Para o objetivo proposto neste trabalho, definiu-se por não utilizar um fator de multiplicação acima de 20.

Nesta etapa da pesquisa de grade, outro ponto de destaque foi o comportamento das curvas da taxa de aprendizagem (Figura 11). Conforme mencionado, a função *Reduce Learning Rate on Plateau* foi utilizada para reduzir automaticamente a taxa de aprendizagem, caso o desempenho da rede não melhorasse após um determinado número de épocas.

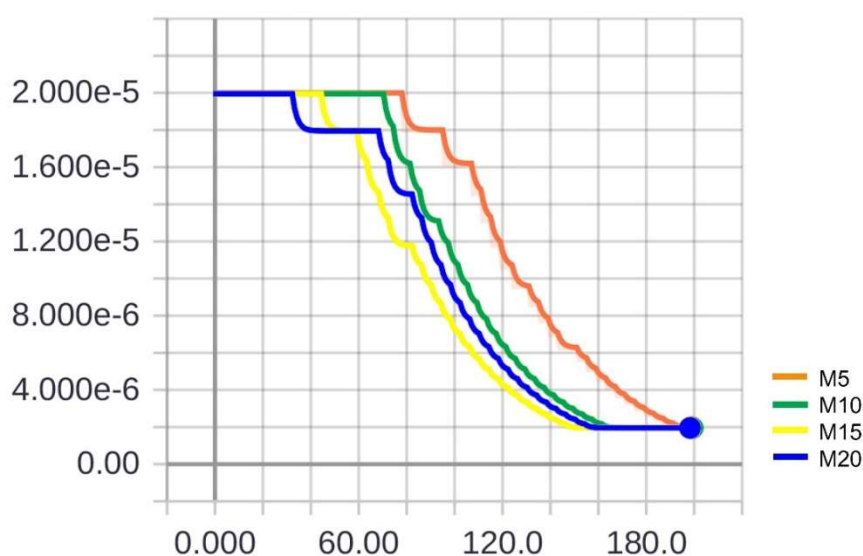


Figura 11. Função da taxa de aprendizagem (*learning rate*) ao longo das épocas de treinamento.

Um dos hiperparâmetros dessa função, *min_lr*, estabelece o valor mínimo limite para a taxa de aprendizagem. Para a aprendizagem do modelo deste estudo, o valor definido para este hiperparâmetro foi de '0,1*lr', sendo 'lr' a taxa de aprendizagem inicial que foi, nesta etapa do processo de otimização, de 2E-5. Conforme demonstrado na Figura 11, verificou-se que o valor mínimo (2E-6) para os modelos M10, M15 e M20 foi atingido quando o processo de aprendizagem estava em torno de 160 épocas. A partir desta constatação, definiu-se por revisar o hiperparâmetro em questão, alterando-o para o valor zero, ou seja, sem limite para redução da taxa de aprendizagem.

Após o processo de otimização dos hiperparâmetros para ampliação dos dados, a rede DenseNet201 obteve 87,7% de acurácia na fase de validação, após 200 épocas (Figura 9B, M20). Conforme mencionado, a avaliação final de desempenho do modelo foi realizada na fase de teste, com o intuito de verificar a sua capacidade de generalização. A Tabela 5 apresenta o resultado, em uma matriz de confusão, da predição do modelo na fase de teste, após a segunda etapa do processo de otimização.

Tabela 5. Matriz de confusão da rede DenseNet201 após o processo de otimização dos hiperparâmetros de ampliação do banco de dados.

Classe	Ae. aegypti fêmea	Ae. aegypti macho	Ae. albop. fêmea	Ae. albop. macho	C. quinq. fêmea	C. quinq. macho	Total	TVP (%)	TVN (%)	BA (%)
Ae. aegypti fêmea	201	9	22	3	3	0	238	84,5	95,7	90,1
Ae. aegypti macho	16	240	20	32	4	0	312	76,9	96,7	86,8
Ae. albop. fêmea	24	4	251	9	0	1	289	86,9	94,3	90,6
Ae. albop. macho	8	26	23	214	0	1	272	78,7	96,0	87,3
C. quinq. fêmea	7	1	3	3	190	1	205	92,7	98,0	95,3
C. quinq. macho	0	0	1	3	19	171	194	88,1	99,8	94,0
Total	256	280	320	264	216	174	1510			90,7
Precisão (%)	78,5	85,7	78,4	81,1	88,0	98,3				

TVP: Taxa Verdadeiro-Positivo; TVN: Taxa Verdadeiro-Negativo; BA: Acurácia Balanceada

A acurácia balanceada na classificação geral, até esta fase de otimização dos hiperparâmetros, foi de 90,7%. O desempenho do modelo para cada uma das seis classes também é apresentado na Tabela 5.

Nesta etapa da otimização dos hiperparâmetros, apesar do resultado de acurácia balanceada global já superar o valor de 90%, estabeleceu-se como meta que cada classe deveria ter uma taxa verdadeiro-positivo acima de 80%. Essa meta foi estabelecida, considerando o resultado alcançado anteriormente com a CNN GoogLeNet, quando se obteve uma acurácia geral de 76,2%, contudo, das seis classes avaliadas, três apresentaram acurácia consideradas não satisfatórias (64,3% para classificação da espécie *Ae. aegypti* fêmea, 57,1% para classificação da espécie *C. quinquefasciatus* fêmea e 71,4% para classificação da espécie *C. quinquefasciatus* macho) (MOTTA et al., 2019). Desta forma, verifica-se na Tabela 5 que para as classes *Ae. aegypti* macho e *Ae. albopictus* macho o modelo precisaria ser melhorado.

5.5.3 Arquitetura de CNN e otimização dos hiperparâmetros das camadas de classificação

A partir da otimização dos hiperparâmetros de ampliação do banco de dados, iniciou-se a investigação das arquiteturas de CNN, Xception e DenseNet201, para geração dos vetores de atributos (*bottlenecks*) e dos hiperparâmetros das camadas de classificação das imagens (número de épocas, taxa de aprendizagem inicial e tamanho do lote). A Tabela 6 apresenta os resultados dos desempenhos da pesquisa de grade dos hiperparâmetros para extração de atributos e classificação (Tabela 3), ao final das épocas de treinamento.

Tabela 6. Resultado da pesquisa de grade (*grid search*) dos hiperparâmetros para extração de atributos e classificação.

#	CNN	Épocas	Otimiz.	LR	Resultados			
					L_t	Acc_t (%)	L_v	Acc_v (%)
1	Xception	200	Adam	4e-7	0,76	71,3	0,75	71,9
2	Xception	200	Adam	4e-6	0,31	88,8	0,55	80,4
3	Xception	200	Adam	2e-5	0,19	93,3	0,58	81,6
4	Xception	200	Adam	1e-4	0,08	97,2	0,70	83,5
5	Xception	300	Adam	4e-6	0,29	89,6	0,55	80,9

6	DenseNet201	200	Adam	4e-6	0,26	90,5	0,42	85,6
7	DenseNet201	200	Adam	1e-5	0,18	93,6	0,41	86,9
8	DenseNet201	200	Adam	2e-5	0,13	95,3	0,41	88,0
9	DenseNet201	200	Adam	3e-5	0,12	95,6	0,41	88,1
10	DenseNet201	200	Adam	6e-5	0,08	97,0	0,41	89,4
11	DenseNet201	200	Adam	1e-4	0,05	98,1	0,43	90,0
12	DenseNet201	200	Adam	2e-4	0,03	99,0	0,47	90,4
13	DenseNet201	200	Adam	3e-4	0,02	99,4	0,51	90,7
14	DenseNet201	200	Adam	6e-4	0,01	99,7	0,56	90,8
15	DenseNet201	200	Adam	1e-3	0,01	99,7	0,59	90,7
16	DenseNet201	300	Adam	4e-6	0,23	91,8	0,41	86,3

#: Experimentos; M: fator de multiplicação da ampliação do dataset; CNN: arquitetura de rede; Épocas: nº de épocas; Otimiz.: Otimizador para classificação; LR: taxa de aprendizagem; L_t: perda (cross entropy) na fase de treinamento; Acc_t: acurácia na fase de treinamento; L_v: perda (cross entropy) na fase de validação; Acc_v: acurácia na fase de validação.

A partir dos resultados apresentados na Tabela 6, os cinco melhores desempenhos em relação à acurácia na fase de validação foram os experimentos 11, 12, 13, 14 e 15, enquanto que os cinco melhores em relação à função perda na fase de validação foram os experimentos 7, 8, 9, 10 e 16. Desta forma, os modelos desses experimentos foram selecionados para avaliação da classificação com os bancos de dados de testes.

A Figura 12 apresenta os resultados de acurácia balanceada (BA) global na classificação das espécies *Ae. aegypti* fêmea, *Ae. aegypti* macho, *Ae. albopictus* fêmea, *Ae. albopictus* macho, *C. quinquefasciatus* fêmea e *C. quinquefasciatus* macho.

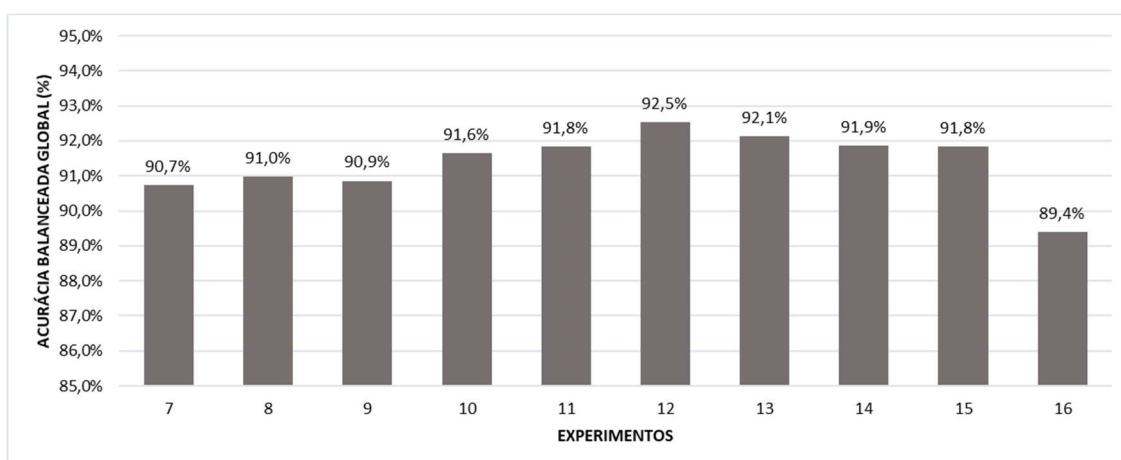


Figura 12. Acurácia balanceada global para os experimentos 7 ao 16 (Tabela 6).

Ressalta-se que os valores dos pesos iniciais atribuídos em um treinamento de rede neural iniciam-se de forma aleatória. Na medida em que o aprendizado evolui, esses pesos são ajustados, contudo, a aleatoriedade inicial desses valores pode alterar o resultado final de desempenho. Os fatores que determinam o mínimo local final são o peso inicial e o algoritmo de treinamento. Além disso, a inicialização do peso influencia a velocidade de convergência, a probabilidade de convergência e generalização do modelo (HERNÁNDEZ-ESPINOSA, 2001).

Desta forma, é provável que os resultados de acurácia apresentados na Figura 12 apresentem uma variação, caso as redes sejam treinadas novamente. Para avaliar a influência dessa variação e o impacto na seleção do conjunto de hiperparâmetros deste estudo, os dois experimentos que apresentaram melhor desempenho global de acurácia (experimentos 12 e 13) foram retreinados mais duas vezes e seus novos modelos foram resubmetidos à avaliação do banco de dados de teste.

A Figura 13 apresenta o desempenho dos experimentos 12 e 13 (Tabela 6) e a variação dos resultados de acurácia global para três treinamentos independentes.

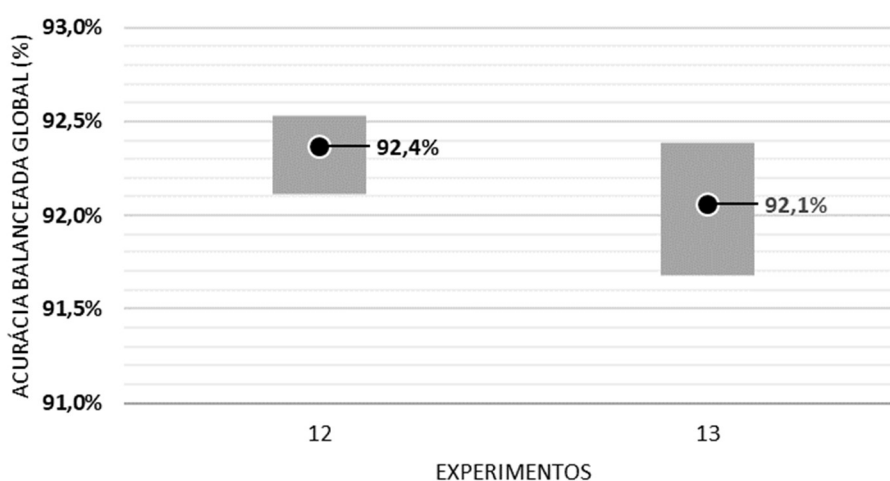


Figura 13. Variação da acurácia balanceada global dos experimentos 12 e 13.

Com base na avaliação da variação de desempenho, conclui-se que, apesar do modelo com os parâmetros do experimento 13 conseguir alcançar desempenho

semelhante e eventualmente superior ao modelo com os parâmetros do experimento 12, este último apresentou, na média, desempenho superior em relação à acurácia balanceada global e, por isso, foi o modelo selecionado como o modelo final após o processo de otimização.

Ainda nesta etapa da otimização, a variação de outros hiperparâmetros também foi testada, entretanto, os resultados não apresentaram melhoria no desempenho da rede. Dessa forma, introduziu-se a função *earlystopping*, que é uma forma de regularização com o objetivo de interromper o processo de aprendizagem da rede, quando não se observa redução da função perda (GOODFELLOW; BENGIO; COURVILLE, 2016). Como na maioria dos experimentos deste trabalho, o valor mínimo da função perda não correspondeu ao valor máximo da acurácia, o desempenho dos modelos treinados com a função *earlystopping* na fase de teste, apresentaram menor acurácia balanceada do que os modelos sem a inclusão desta função. Outro hiperparâmetro avaliado foi o fator de redução da taxa de aprendizagem da função *Reduce Learning Rate on Plateau*. Este valor inicialmente configurado para um fator de 0,90 foi alterado para 0,50, sem, contudo, representar ganho no desempenho do modelo.

A partir dos resultados encontrados, e, considerando a arquitetura de CNN e os hiperparâmetros do experimento 12 (Tabela 6), como a melhor configuração até esta etapa do processo de otimização, iniciou-se a investigação da variação do tamanho de lote (*batch size*). A Figura 14 apresenta os resultados de acurácia balanceada (BA) global com o banco de dados de teste.

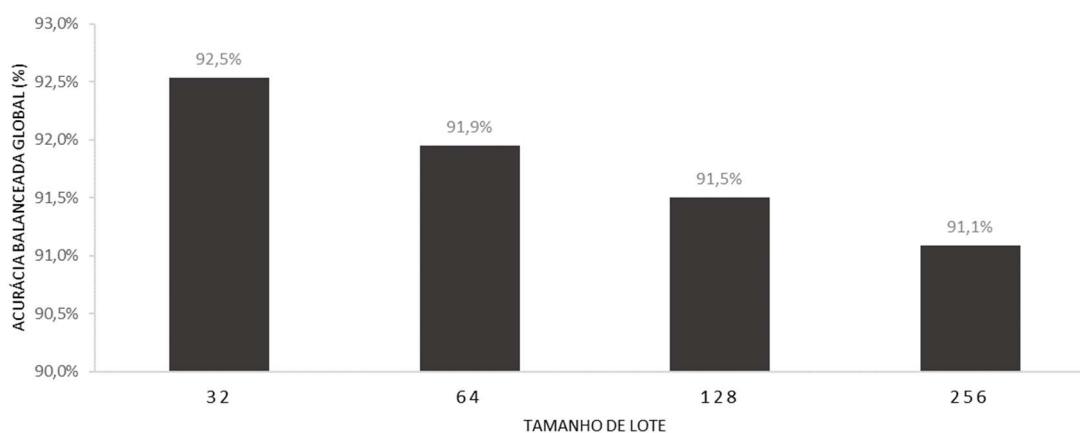


Figura 14. Acurácia balanceada global com a variação do tamanho de lote.

No resultado global, o tamanho do lote 32 apresentou o melhor desempenho (Figura 14). Desta forma, após a conclusão de todo o processo de otimização, a configuração de hiperparâmetros obtidos com o melhor desempenho para classificação de mosquitos adultos, em seis diferentes classes, ficou conforme apresentado no Quadro 3.

Quadro 3: Configuração dos hiperparâmetros obtidos após a otimização para classificação de mosquitos adultos.

Hiperparâmetro	Configuração
CNN	DenseNet201 (pre-treinada)
Otimizador	Adam
Número de épocas	200
Taxa de aprendizagem inicial	0,0002
Tamanho do lote	32
Ampliação do banco de dados	
Rotação	45%
Deslocamento de largura	5%
Deslocamento de altura	5%
Fator de multiplicação	20
Reduce Learning Rate on Plateau	
Monitor	Perda (<i>cross entropy</i>) na validação
Fator	0,9
Paciência (<i>patience</i>)	4
Valor mínimo da taxa de aprendizagem (<i>min_lr</i>)	0
Camadas de classificação	
<pre> pool0 = GlobalAveragePooling2D()(inputs) dense0 = Dense(512, activation="tanh")(pool0) dpo0 = Dropout(0.45)(dense0) dense1 = Dense(64, activation="relu")(dpo0) dpo1 = Dropout(0.35)(dense1) outputs = Dense(6, activation="softmax")(dpo1) </pre>	

Na Tabela 7 apresenta-se a matriz de confusão obtida para a rede DenseNet201 para as seis classes com o banco de dados de teste, após realizado o

treinamento com a configuração de hiperparâmetros (Quadro 3) otimizada para classificação automática de mosquitos adultos.

Tabela 7. Matriz de confusão da rede DenseNet201 após o processo de otimização do modelo.

Classe	<i>Ae. aegypti</i> fêmea	<i>Ae. aegypti</i> macho	<i>Ae. albop.</i> fêmea	<i>Ae. albop.</i> macho	<i>C. quinq.</i> fêmea	<i>C. quinq.</i> macho	Total	TVP (%)	TVN (%)	BA (%)
<i>Ae. aegypti</i> fêmea	216	6	12	3	1	0	238	90,8	96,4	93,6
<i>Ae. aegypti</i> macho	11	252	18	28	3	0	312	80,8	97,4	89,1
<i>Ae. albop.</i> fêmea	19	4	258	8	0	0	289	89,3	96,1	92,7
<i>Ae. albop.</i> macho	8	21	15	227	0	1	272	83,5	96,4	90,0
<i>C. quinq.</i> fêmea	7	0	2	1	194	1	205	94,6	98,2	96,4
<i>C. quinq.</i> macho	1	0	1	4	19	169	194	87,1	99,8	93,5
Total	262	283	306	271	217	171	1510			92,5
Precisão (%)	82,4	89,0	84,3	83,8	89,4	98,8				

TVP: Taxa Verdadeiro-Positivo; TVN: Taxa Verdadeiro-Negativo; BA: Acurácia Balanceada

Comparando-se os resultados apresentados na Tabela 7, com os resultados da Tabela 5, observa-se que a melhoria da acurácia global foi de 1,8%, chegando ao valor de 92,5%. Com exceção da acurácia para classificação do *C. quinquefasciatus* macho, todos os demais indicadores evidenciam a melhoria no desempenho de aprendizado da rede após o processo de otimização.

A aplicação do modelo, para seis classes, pode ser comparada com os resultados obtidos por Ouyang et al. (2015) e por Motta et al. (2019). Ouyang et al. (2015) aplicaram técnicas de perfil da frequência das batidas de asas e avaliaram a classificação das mesmas espécies-alvo deste trabalho. Como resultado, obtiveram uma acurácia média, com a aplicação de redes neurais, de 79,5%. Motta et al. (2019) aplicaram o método baseado em CNN para classificação a partir de imagens e obtiveram uma acurácia com a rede GoogLeNet de 76,2%.

Na comparação dos resultados por cada classe, no trabalho de Motta et al. (2019) as classes *Ae. aegypti* fêmea, *C. quinquefasciatus* fêmea e *C. quinquefasciatus* macho apresentaram acurácias de 64,3%, 57,1% e 71,4% respectivamente, enquanto que neste estudo essas mesmas classes apresentaram acurácia balanceada de 93,6%, 93,5% e 96,4%.

Cabe ressaltar, contudo, que nos estudos realizados por Ouyang et al. (2015) e por Motta et al. (2019), o banco de dados de teste foi bem pequeno e o cálculo de acurácia não considerou o desbalanceamento do banco de dados.

Ressalta-se também que para todas as classes estudadas, o valor da taxa verdadeiro-positivo foi superior à 80%, meta estabelecida neste trabalho.

5.5.4 Desempenho do modelo para duas classes

Conforme apresentado anteriormente, o desempenho do modelo também foi treinado para diferenciação entre *Aedes* e *Culex*, com o objetivo de avaliar o seu desempenho na detecção do gênero *Aedes*.

A Figura 15 apresenta o resultado da evolução do aprendizado do modelo otimizado com de hiperparâmetros (Quadro 3) otimizada para classificação automática de mosquitos adultos para a diferenciação entre duas classes.

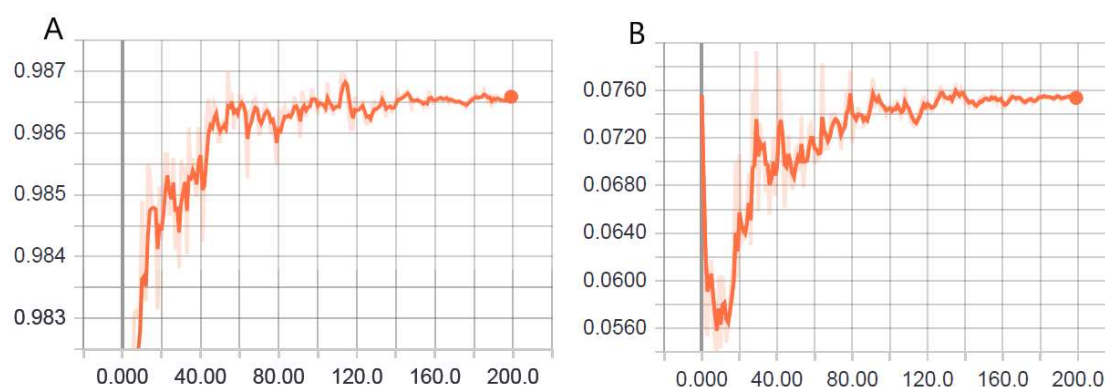


Figura 15. Evolução da aprendizagem do modelo otimizado para duas classes: (A): acurácia na fase de validação; (B): perda (*cross entropy*) na fase de validação.

Reforça-se que, conforme mencionado anteriormente, o efeito de aumento da função perda e, ainda assim, o aumento e posterior estabilização da acurácia

não deve ser confundido com *overfitting*, uma vez que este está relacionado à incapacidade de generalização do modelo para aplicação com novos dados, não usados no treinamento da rede (GOODFELLOW; BENGIO; COURVILLE, 2016). Torna-se relevante, também, levar em consideração a escala dos gráficos (Figuras 15A e 15B). Tanto a amplitude de variação da acurácia e da função perda são bem pequenas.

A Tabela 8 apresenta a matriz de confusão para duas classes com o banco de dados de teste, após o treinamento do modelo e, no Quadro 4 são apresentados os cálculos dos indicadores de desempenho da matriz de confusão 2x2.

Tabela 8. Matriz de confusão do modelo otimizado para 2 classes.

Classe	<i>Aedes</i>	<i>Culex</i>	TOTAL
<i>Aedes</i>	1104	7	1111
<i>Culex</i>	19	380	399
TOTAL	1123	387	1510

Quadro 4: Cálculos dos indicadores de desempenho da matriz de confusão 2x2.

Indicador	Cálculo
Precisão	$PR = \frac{VP}{VP + FP} = \frac{1104}{1104 + 19} = 98,3\%$
Taxa Verdadeiro-Positivo	$TVP = \frac{VP}{VP + FN} = \frac{1104}{1104 + 7} = 99,4\%$
Taxa Verdadeiro-Negativo	$TVN = \frac{VN}{VN + FP} = \frac{380}{380 + 19} = 95,2\%$
Acurácia Balanceada	$BA = 0,5 \times (TVP + TVN) = 0,5 \times (99,4 + 95,2) = 97,3\%$

Nesta etapa, os resultados obtidos demonstraram uma acurácia balanceada de 97,3% na detecção de mosquitos do gênero *Aedes*, quando comparados com os mosquitos do gênero *Culex*. A aplicação do modelo, para as duas classes, pode ser comparada com os resultados obtidos por Sanches-Ortiz et al. (2017), Reyes et al. (2016) e Motta et al. (2019) para detecção do mosquito *Aedes*. Sanches-Ortiz et al. (2017), apesar de avaliarem a classificação na fase larvária,

encontraram uma taxa verdadeiro-positivo de 85%. Por sua vez, Reyes et al. (2016) obtiveram um resultado de 92,5% para a taxa verdadeiro-positivo quando aplicado a técnica SVM. No trabalho desenvolvido por Motta et al. (2019), não houve uma avaliação direta para duas classes, mas nenhuma espécie de *Culex* foi classificada como *Aedes*. Contudo, deve-se levar em consideração que nesses três trabalhos, os autores utilizaram um número bastante reduzido de imagens banco de dados de teste, quando comparado com este estudo. Sanches-Ortiz et al. (2017) utilizaram dez imagens de *Aedes* e dez imagens de não-*Aedes*, Reyes et al. (2016) utilizaram 20 imagens de *Aedes* e 20 imagens de não-*Aedes* e Motta et al. (2019) utilizaram 56 imagens de *Aedes* e 28 imagens de não-*Aedes*.

Considerando o resultado verdadeiro-positivo, deste trabalho, de 99,4% (para duas classes) para detecção de mosquitos do gênero *Aedes* e um banco de dados de teste contendo 1.111 imagens de *Aedes* e 399 imagens de *Culex*, o modelo proposto apresentou resultado bastante satisfatório.

5.5.5 Desempenho do modelo para sete classes

O desempenho do modelo também foi avaliado para diferenciação em sete classes, com a inclusão de outros insetos (aranhas, besouros e abelhas), com o objetivo de avaliar a capacidade da diferenciação dos mosquitos de outros insetos.

A Figura 16 apresenta os resultados obtidos para a evolução do aprendizado do modelo otimizado com os hiperparâmetros (Quadro 3) para a diferenciação entre sete classes.

A acurácia na fase de validação ao final de 200 épocas de treinamento para sete classes (Figura 16) apresentou valor superior a acurácia na validação para seis classes (Tabela 6, experimento 12), representado por 91,1 % e 90,4% respectivamente. Tais resultados reforçam a capacidade do modelo em reconhecer um “não mosquito”.

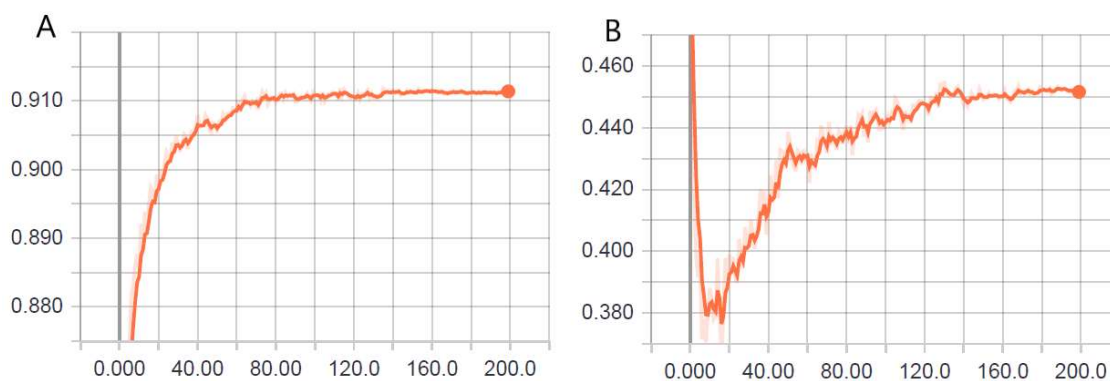


Figura 16. Evolução da aprendizagem do modelo otimizado para sete classes: (A): acurácia na fase de validação; (B): perda (*cross entropy*) na fase de validação.

A Tabela 9 apresenta a matriz de confusão para sete classes com o banco de dados de teste, após treinamento do modelo. Desta forma, observou-se a capacidade do modelo desenvolvido para realizar a distinção de mosquitos de outros insetos. Com uma acurácia balanceada de 98,9%, a classe outros é a que apresenta melhor desempenho em comparação com as demais classes. A acurácia global nesta etapa foi de 93,5% e, conforme esperado, a taxa verdadeiro-positivo foi superior à 80% em todas as classes avaliadas (Tabela 9).

Tabela 9. Matriz de confusão do modelo otimizado para 7 classes.

Classe	<i>Ae. aegypti</i> fêmea	<i>Ae. aegypti</i> macho	<i>Ae. albop.</i> fêmea	<i>Ae. albop.</i> macho	<i>C. quinq.</i> fêmea	<i>C. quinq.</i> macho	Outros	Total	TVP (%)	TVN (%)	BA (%)
<i>Ae. aegypti</i> fêmea	203	9	18	5	3	0	0	238	85,3	97,2	91,3
<i>Ae. aegypti</i> macho	12	253	9	32	5	0	1	312	81,1	98,1	89,6
<i>Ae. albop.</i> fêmea	14	1	260	11	0	0	3	289	90,0	96,5	93,2
<i>Ae. albop.</i> macho	6	17	19	229	0	1	0	272	84,2	96,7	90,4
<i>C. quinq.</i> fêmea	7	0	3	0	192	2	1	205	93,7	98,5	96,1
<i>C. quinq.</i> macho	0	0	1	1	15	175	2	194	90,2	99,8	95,0
Outros	3	0	1	0	0	0	233	237	98,3	99,5	98,9
Total	245	280	311	278	215	178	240	1747			93,5
Precisão (%)	82,9	90,4	83,6	82,4	89,3	98,3	97,1				

A aplicação do modelo, para sete classes, também pode ser comparada com os resultados obtidos por Ouyang et al. (2015), que avaliaram a classificação das mesmas espécies-alvo deste trabalho e obtiveram uma acurácia média, com a aplicação de redes neurais, de 79,5% e por Motta et al. (2019), que aplicaram o método baseado em CNN para classificação a partir de imagens e obtiveram uma acurácia com a rede GoogLeNet de 76,2%.

5.5.6 Avaliação da influência do ângulo da imagem

O modelo otimizado foi submetido à avaliação e classificação das imagens do banco de dados de teste para avaliação da influência do ângulo da imagem nos resultados de acurácia balanceada (Figura 17).

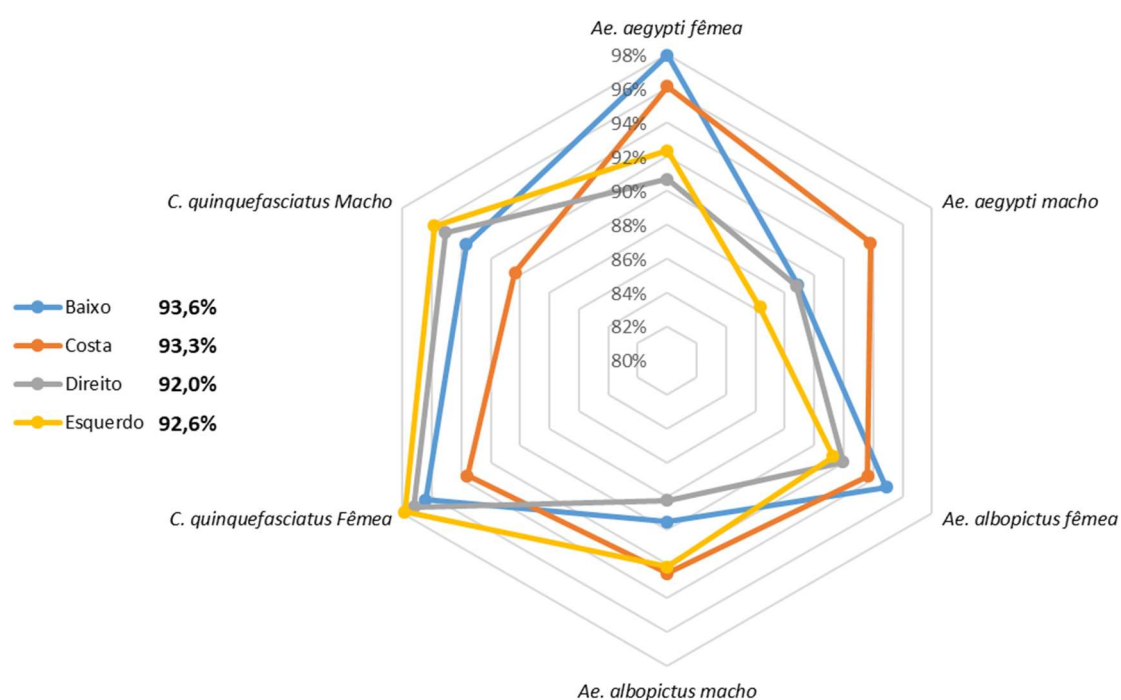


Figura 17. Desempenho da acurácia balanceada global e de cada uma das seis classes com diferentes ângulos dos mosquitos: (Baixo): ângulo frontal e de baixo; (Costa): ângulo de costas; (Direito): ângulo direito; (Esquerdo): ângulo esquerdo.

As técnicas de CNN têm sido utilizadas com o objetivo de corrigir ou estimar a orientação dos objetos em uma imagem. Fischer, Dosovitskiy & Brox (2015) focaram em estimar e corrigir a orientação exata de imagens e demonstraram que uma rede convolucional pode aprender recursos para prever a orientação canônica das imagens. Saxena, Driemeyer & Ng (2009) propuseram um algoritmo de aprendizagem para estimar a orientação 3-D dos objetos, considerando situações com objetos simétricos e assimétricos.

Contudo, não se encontrou trabalhos que tivessem como propósito orientar a forma de adquirir a imagem, em relação ao ângulo do objeto a ser fotografado, para classificação automática utilizando CNN.

Atualmente existe uma ferramenta via aplicativo móvel, desenvolvida na Espanha, disponível para o controle da espécie *Aedes albopictus* (também conhecido como *tiger mosquito*). Essa ferramenta, contudo, demanda uma avaliação das imagens capturadas por entomologistas (RODRIGUEZ ANTONIO, 2016) e dependendo da forma que o ângulo do mosquito está na imagem, o entomologista não consegue classificar com precisão (MOSQUITO ALERT, 2016). Desta forma, esta análise é extrema importância para uma possível orientação de como a imagem deve ser capturada antes de submeter à avaliação do modelo proposto, visando aumentar a acurácia do modelo de classificação automática.

A análise do desempenho global de acurácia (Figura 17) sugere que os ângulos frontal, de baixo e de costas apresentam um melhor desempenho, no entanto, observa-se que, dependendo da espécie de mosquito, a orientação de captura da imagem pode variar. No caso apresentado, para a espécie *C. quinquefasciatus*, as imagens capturadas dos ângulos direito e esquerdo apresentaram maior acurácia balanceada.

Considerando que a acurácia na detecção e na classificação dos mosquitos do gênero *Aedes* apresenta maior relevância entomológica, pode-se sugerir, a partir desses resultados, que os usuários de uma ferramenta entomológica baseada em CNN capturem as imagens nos ângulos frontal, de baixo e de costas.

5.6 CONCLUSÕES

Este trabalho evidencia que o uso de modelos baseados em CNN, com arquiteturas mais complexas, pode representar um importante avanço para o aumento da acurácia na automação dos processos de detecção e classificação de mosquitos adultos.

Reforça a necessidade da otimização de hiperparâmetros dos modelos, como uma etapa específica e não automatizada, para a aplicação no campo da entomologia. Apesar da disponibilização de modelos pré-treinados para classificação a partir de imagens, a utilização de cada um desses modelos necessita de otimização e avaliação de seu desempenho, antes da sua aplicação no mundo real.

Considerando que a aplicação de técnicas de *deep learning* precisam de um grande número de dados (centenas de milhares), a aquisição de novas imagens é um passo fundamental para o desenvolvimento de uma ferramenta entomológica para aplicação em campo.

A influência do ângulo do mosquito na imagem, sugere que captura das imagens seja realizada com os mosquitos na posição frontal, de baixo e de costas. Importante salientar que para o treinamento do modelo, a validação das imagens por um especialista, para a sua correta rotulação, é imprescindível.

Por fim, destaca-se que o modelo com a arquitetura de CNN DenseNet201, otimizador Adam, taxa inicial de aprendizagem de 0,0002 e com 200 épocas de treinamento apresentou os melhores resultados para classificação automática de mosquitos adultos das espécies *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus*, bem como para detecção dos mosquitos do gênero *Aedes*.

Cabe ressaltar, que continuamente surgem novas arquiteturas de redes convolucionais e, portanto, é fundamental que o modelo para classificação automática de mosquitos tenha flexibilidade para incorporar novas arquiteturas de redes ou camadas específicas que possam melhorar o seu desempenho.

5.7 REFERÊNCIAS

ALERT, M. **Macrophotography of mosquitos and citizen science**. [s.l: s.n.].

ALOM, M. Z. et al. **The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches**. [s.l: s.n.].

BATISTA, G. E. A. P. A. et al. Towards automatic classification on flying insects using inexpensive sensors. **Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011**, v. 1, n. December, p. 364–369, 2011.

BERGSTRA, J.; BENGIO, Y. Random Search for Hyper-Parameter Optimization. **Journal of Machine Learning Research**, v. 13, p. 281–305, 2012.

CHIN, W.-S. et al. **A Learning-Rate Schedule for Stochastic Gradient Methods to Matrix Factorization BT - Advances in Knowledge Discovery and Data Mining**. (T. Cao et al., Eds.) Cham: Springer International Publishing, 2015

FISCHER, P.; DOSOVITSKIY, A.; BROX, T. Image orientation estimation with convolutional networks. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 9358, p. 368–378, 2015.

FUCHIDA, M. et al. Vision-Based Perception and Classification of Mosquitoes Using Support Vector Machine. **Applied Sciences**, v. 7, n. 1, p. 51, 5 jan. 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [s.l.] MIT Press, 2016.

GOPALAN, S. S.; DAS, A. Household economic impact of an emerging disease in terms of catastrophic out-of-pocket health care expenditure ... Household economic impact of an emerging disease in terms of catastrophic out-of-pocket health care expenditure and loss of productivity : . n. January, 2017.

HE, K. et al. Deep Residual Learning for Image Recognition. **arXiv:1512.03385v1**, 2015.

HERNÁNDEZ-ESPINOSA, M. F. C. Weight Initialization Methods for Multilayer. **Neural Networks**, n. April, p. 119–124, 2001.

Keras Documentation. Disponível em: <<https://keras.io/applications/#applications>>.

LORENZ, C.; SERGIO, A.; SUESDEK, L. Artificial Neural Network applied as a methodology of mosquito species identification. **Acta Tropica**, v. 152, p. 165–169, 2015.

MAUCOURANT, C. et al. Control of Acute Arboviral Infection by Natural Killer Cells. **Viruses**, v. 11, n. 2, p. 131, 31 jan. 2019.

MAYER, S.; B. TESH, R.; VASILAKIS, N. **The emergence of arthropod-borne viral diseases: A global prospective on dengue, chikungunya and zika fevers.** [s.l: s.n.]. v. 166

MOTTA, D. et al. Application of convolutional neural networks for classification of adult mosquitoes in the field. **PLOS ONE**, v. 14, n. 1, p. e0210829, 14 jan. 2019.

NG, K.-C. et al. Increased Adult *Aedes aegypti* and *Culex quinquefasciatus* (Diptera: Culicidae) Abundance in a Dengue Transmission Hotspot, Compared to a Coldspot, within Kaohsiung City, Taiwan. **Insects**, v. 9, n. 3, p. 98, 13 ago. 2018.

OUYANG, T. H. et al. Mosquito vector monitoring system based on optical wingbeat classification. **Computers and Electronics in Agriculture**, v. 118, p. 47–55, 2015.

PAHO. **Tool for the diagnosis and care of patients with suspected arboviral diseases.** [s.l: s.n.].

PARK, S. I. et al. Species identification of food contaminating beetles by recognizing patterns in microscopic images of elytra fragments. **PLoS ONE**, v. 11, n. 6, p. 1–22, 2016.

PARRA-AMAYA, M. et al. Early Detection for Dengue Using Local Indicator of Spatial Association (LISA) Analysis. **Diseases**, v. 4, n. 4, p. 16, 29 mar. 2016.

REYES, A. M. M. D. L. et al. **Detection of *Aedes Aegypti* mosquito by digital image processing techniques and support vector machine** 2016 IEEE Region 10 Conference (TENCON), 2016.

RODRIGUEZ ANTONIO. Data classification system for the citizens science platform Mosquito Alert. p. 1–64, 2016.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, v. 115, n. 3, p. 211–252, 2015.

SANCHEZ-ORTIZ, A. et al. **Mosquito larva classification method based on convolutional neural networks.** 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP). **Anais...**2017

SAXENA, A.; DRIEMEYER, J.; NG, A. Y. Learning 3-D object orientation from images. **Proceedings - IEEE International Conference on Robotics and Automation**, p. 794–800, 2009.

SHEPARD, D. S. et al. The global economic burden of dengue: a systematic analysis. **The Lancet Infectious Diseases**, v. 16, n. 8, p. 935–941, 1 ago. 2016.

SILVA, D. F. et al. **Applying machine learning and audio analysis techniques to insect recognition in intelligent traps.** Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013.

Anais...2013

STRAUBE, S.; KRELL, M. M. How to evaluate an agent's behavior to infrequent events?—Reliable performance estimation insensitive to class distribution. **Frontiers in Computational Neuroscience**, v. 8, n. April, p. 1–6, 2014.

UNITED NATIONS DEVELOPMENT PROGRAMME. **A Socio-economic Impact Assessment of the Zika Virus in Latin America and the Caribbean: with a focus on Brazil, Colombia and Suriname**Undp. [s.l: s.n.]. Disponível em: <<http://www.undp.org/content/undp/en/home/librarypage/hiv-aids/a-socio-economic-impact-assessment-of-the-zika-virus-in-latin-am.html>>.

WANG, J. et al. A new automatic identification system of insect images at the order level. **Knowledge-Based Systems**, v. 33, p. 102–110, 2012.

WHO. Global Vector Control Response - Background document to inform deliberations during the 70th session of the World Health Assembly. **WHO**, v. 2030, p. 47, 2017.

WONG, S. C. et al. Understanding Data Augmentation for Classification: When to Warp? **2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016**, 2016.

YANG, H. P. et al. A tool for developing an automatic insect identification system based on wing outlines. **Scientific Reports**, v. 5, p. 1–11, 2015.

6 CONSIDERAÇÕES FINAIS

Este trabalho demonstra o potencial de uso das CNNs na atividade de classificação automática de mosquitos, a partir de imagens. O desempenho, tanto na classificação de espécies, como na detecção de mosquitos do gênero *Aedes*, apresentou resultados que reforçam a crença da viabilidade do uso das técnicas de *deep learning* para o desenvolvimento de uma poderosa ferramenta entomológica para aplicação em campo.

Esta ferramenta pode ter como objetivo a sua utilização por laboratórios especializados para classificação de espécies ou para sua utilização por não-especialistas, permitindo assim o envolvimento da comunidade na detecção de espécies nocivas à saúde e no alerta às autoridades para um controle mais efetivo dos vetores de transmissão de doenças. Pode ser um instrumento de identificação rápida e confiável de espécies-alvo de transmissão de agentes infecciosos causadores de epidemias em mamíferos, ampliando o conhecimento de sua biologia e permitindo o seu mapeamento ecológico.

O banco de imagens gerado neste trabalho constitui em um passo importante para um conjunto denso de imagens de mosquitos transmissores de doenças para humanos, incluindo *Anopheles* e outras espécies de *Culicídeos*. Destaca-se também o estudo investigativo dos hiperparâmetros e a obtenção de pesos otimizados para o processo de extração de atributos e classificação de imagens de mosquitos adultos.

O tamanho do banco de dados utilizado neste trabalho ainda se constitui na principal limitação para o desenvolvimento dessa ferramenta e sua utilização por laboratórios e pela comunidade.

Cabe ressaltar ainda que uma outra abordagem adotada comumente para treinamento de redes neurais é a validação cruzada (*cross validation*), onde os bancos de dados de validação e teste, da forma apresentada neste estudo, deixam de existir e o banco de dados de treinamento é dividido em um determinado número de folders (k) e cada modelo é treinado com um número de $k - 1$ folders.

Como sugestão para trabalhos futuros, destaca-se:

- Ampliar o banco de imagens dos mosquitos *Ae. aegypti*, *Ae. albopictus* e *C. quinquefasciatus* e incorporação nova outras espécies de mosquitos transmissores de doenças.
- Desenvolver aplicativo móvel protótipo para avaliação do modelo em campo, bem como para viabilizar a aquisição de novas imagens.
- Aplicação da estratégia de validação cruzada (*cross validation*) para treinamento do modelo de classificação e identificação de mosquitos.
- Estender o estudo, baseado em técnicas de deep learning, para outras aplicações na área de saúde e/ou industriais.

ANEXOS

ANEXO I: *SPLIT_DATASET.PY*

```

import os
from glob import glob
import numpy as np
from tqdm import tqdm
import argparse

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Split the data into
validation and train")

    parser.add_argument("-i", metavar="path", type=str, required=True,
        help="Path to the dataset")

    parser.add_argument("-o", metavar="path", type=str, required=True,
        help="Output path of the dataset")

    parser.add_argument("--split", metavar="path", type=float,
default=0.2,
        help="Percentage of the data to be used as validation and
test")

    args = parser.parse_args()

    input_path = args.i
    output_path = args.o
    split = min(1.0, args.split)

    os.system("mkdir -p {}".format(output_path))

    ds_classes_paths = dict()
    tgt_paths = dict()

    for root, dirs, files in os.walk(input_path):
        for name in dirs:
            train_path = os.path.join(output_path, "Train", name)
            valid_path = os.path.join(output_path, "Val", name)
            test_path = os.path.join(output_path, "Test", name)
            os.system("mkdir -p {}".format(train_path))
            os.system("mkdir -p {}".format(valid_path))
            os.system("mkdir -p {}".format(test_path))
            tgt_paths[name] = {
                "train": train_path,
                "valid": valid_path,
                "test": test_path
            }
            ds_classes_paths[name] = os.path.join(root, name)

    for ds_class in ds_classes_paths.keys():
        fls = os.listdir(ds_classes_paths[ds_class])

        split_size = int(split*len(fls))

        fls_valid = np.random.choice(fls, split_size, replace=False)
        for f in fls_valid:
            fls.remove(f)

```

```

fls_test = np.random.choice(fl, split_size, replace=False)
for f in fls_test:
    fl.remove(f)

print("Splitting dataset for class", ds_class)

for f in tqdm(fl, desc="Train"):
    src = os.path.join(ds_classes_paths[ds_class], f)
    tgt = os.path.join(tgt_paths[ds_class]["train"], f)
    cmd = "cp \"{0}\" \"{1}\"".format(src, tgt)
    os.system(cmd)

for f in tqdm(fl_valid, desc="Valid"):
    src = os.path.join(ds_classes_paths[ds_class], f)
    tgt = os.path.join(tgt_paths[ds_class]["valid"], f)
    cmd = "cp \"{0}\" \"{1}\"".format(src, tgt)
    os.system(cmd)

for f in tqdm(fl_test, desc="Test"):
    src = os.path.join(ds_classes_paths[ds_class], f)
    tgt = os.path.join(tgt_paths[ds_class]["test"], f)
    cmd = "cp \"{0}\" \"{1}\"".format(src, tgt)
    os.system(cmd)
print()

```

ANEXO II: *IMAGE_AUGMENTOR.PY*

```

import keras.backend as K
from keras.applications import *
from keras.preprocessing.image import *
from keras.optimizers import *
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
from tqdm import tqdm
import os
import argparse
from glob import glob
from datetime import date

def image_augmentor(src_path, tgt_path, mul_factor, rotation_range,
width_shift_range,height_shift_range, brightness_range, shear_range,
zoom_range,horizontal_flip, vertical_flip):
    """
    Função para aumentar imagens

    Parametros:
    - src_path: caminho de onde as imagens vão ser retiradas
    - tgt_path: caminho para onde as imagens devem ser gravadas
    - mul_factor: fator de multiplicação das imagens
    """

    # Limpa o diretorio caso ele exista ou cria caso contrário
    if os.path.exists(tgt_path):
        for p in os.listdir(tgt_path):
            try:
                shutil.rmtree(os.path.join(tgt_path, p))
            except:

```

```

        os.remove(os.path.join(tgt_path, p))
else:
    os.system("mkdir -p {0}".format(tgt_path))

classes = set()
for root, dirs, files in os.walk(src_path):
    for f in files:
        classes.add(root[root.rfind("/") + 1:])

classes = sorted(list(classes))

# data_generator = ImageDataGenerator(
#     rotation_range=90,
#     horizontal_flip=True,
#     vertical_flip=True,
#     zoom_range=0.5
# )

data_generator = ImageDataGenerator(
    rotation_range=rotation_range,
    width_shift_range=width_shift_range,
    height_shift_range=height_shift_range,
    brightness_range=brightness_range,
    shear_range=shear_range,
    zoom_range=zoom_range,
    horizontal_flip=horizontal_flip,
    vertical_flip=vertical_flip
)

generator = data_generator.flow_from_directory(
    src_path, # this is the target directory
    interpolation="lanczos",
    classes=classes,
    shuffle=False
)

save_prefix = str(date.today()).replace("-", "")
indices_class = dict()
class_counters = dict()
for k, v in generator.class_indices.items():
    indices_class[v] = k
    class_counters[v] = 0
    os.mkdir(os.path.join(tgt_path, k))

print("Building train batches...")
print("{0} augmented images will be created...".format(mul_factor
* generator.samples))
with tqdm(total = mul_factor * int(np.ceil(generator.samples /
generator.batch_size))) as pbar:
    for _ in range(mul_factor):
        for _ in range(int(np.ceil(generator.samples /
generator.batch_size))):
            bX, bY = generator.next()
            bX = np.round(bX, decimals=0).astype(np.uint8)
            bY = np.argmax(bY, axis=-1)
            for x, y in zip(bX, bY):
                file_name = save_prefix +
                "_{0}.png".format(class_counters[y])
                class_counters[y] += 1
                img = Image.fromarray(x)

```

```

        img.save(os.path.join(tgt_path, indices_class[y],
file_name))
        pbar.update()

if __name__ == "__main__":
    """
    https://keras.io/preprocessing/image/

    - rotation_range: Int. Degree range for random rotations.
    - width_shift_range: Float, 1-D array-like or int
      - float: fraction of total width, if < 1, or pixels if >=
1.
      - 1-D array-like: random elements from the array.
      - int: integer number of pixels from interval (-
width_shift_range, +width_shift_range)
      - With width_shift_range=2 possible values are integers [-
1, 0, +1], same as with width_shift_range=[-1, 0, +1], while with
width_shift_range=1.0 possible values are floats in the interval [-
1.0, +1.0).
      - height_shift_range: Float, 1-D array-like or int
      - float: fraction of total height, if < 1, or pixels if >=
1.
      - 1-D array-like: random elements from the array.
      - int: integer number of pixels from interval (-
height_shift_range, +height_shift_range)
      - With height_shift_range=2 possible values are integers
[-1, 0, +1], same as with height_shift_range=[-1, 0, +1], while with
height_shift_range=1.0 possible values are floats in the interval [-
1.0, +1.0).
      - brightness_range: Tuple or list of two floats. Range for
picking a brightness shift value from.
      - shear_range: Float. Shear Intensity (Shear angle in counter-
clockwise direction in degrees)
      - zoom_range: Float or [lower, upper]. Range for random zoom.
If a float, [lower, upper] = [1-zoom_range, 1+zoom_range].
    """
    parser = argparse.ArgumentParser(description="Aumgment images,
multiplying the size of a dataset by a chosen factor")

    parser.add_argument("-i", metavar="path", type=str, required=True,
help="Path to the images that will be used to produce the
bottlenecks")

    parser.add_argument("-o", metavar="path", type=str, required=True,
help="Path to the files where the bottlenecks should be
stored")

    parser.add_argument("-m", metavar="multiplier", type=int,
default=2,
help="Multiplier factor to the size of the dataset. Default:
2")

    parser.add_argument("-rr", metavar="x degrees", type=int,
default=90,
help="Rotation range : Int. Degree range for random rotations.
Default: 90")

    parser.add_argument("-wr", metavar="% of img width", type=float,
default=0.05,

```

```

        help="Width shift range: float representing the width shift
percentage of the image.Default: 0.05")

    parser.add_argument("-hr", metavar="% of img height", type=float,
default=0.05,
        help="Height shift range: float representing the height shift
percentage of the image. Default: 0.05")

    parser.add_argument("-br", metavar="[min, max]", type=float,
nargs=2, default=None,
        help="Brightness range: tuple or list of two floats. Range for
picking a brightness shift value from.")

    parser.add_argument("-sr", metavar="x dregrees", type=float,
default=0,
        help="Shear range: float representing the shear angle in
counter-clockwise direction in degrees. Default: 15.0")

    parser.add_argument("-zr", metavar="[min, max]", type=float,
nargs=2, default=0,
        help="Zoom range: Float or [lower, upper]. Range for random
zoom, as float [lower, upper].")

    parser.add_argument("-hf", metavar="True|False", type=bool,
default=False,
        help="Randomly flip inputs horizontally. Default: False.")

    parser.add_argument("-vf", metavar="True|False", type=bool,
default=False,
        help="Randomly flip inputs vertically. Default: False")

    args = parser.parse_args()

    mul_factor = args.m
    rotation_range = args.rr
    width_shift_range = args.wr
    height_shift_range = args.hr
    brightness_range = args.br
    shear_range = args.sr
    zoom_range = args.zr
    horizontal_flip = args.hf
    vertical_flip = args.vf

    src_path = args.i
    tgt_path = args.o + "_" + str(date.today()).replace("-", "")

    # image_augmentor(src_path, tgt_path, 2)
    image_augmentor(src_path, tgt_path, mul_factor, rotation_range,
width_shift_range,height_shift_range, brightness_range, shear_range,
zoom_range,horizontal_flip, vertical_flip)

```

ANEXO III: *GENERATE_BOTTLENECKS.PY*

```

import keras.backend as K
from keras.models import Model
from keras.layers import (Input, Dense, Conv2D, Activation,
GlobalAveragePooling2D, GlobalMaxPooling2D, MaxPooling2D,
AveragePooling2D, Flatten, BatchNormalization, Dropout)
from keras.applications import *

```



```

from keras.preprocessing.image import *
from keras.optimizers import *
from keras.callbacks import ModelCheckpoint
from PIL import Image
import numpy as np
from glob import glob
import argparse
import os
import shutil
from tqdm import tqdm
import matplotlib.pyplot as plt
from datetime import date
import pickle

def save_classes(classes, classes_path):
    with open(classes_path + '.pkl', 'wb') as f:
        pickle.dump(classes, f, pickle.HIGHEST_PROTOCOL)

def load_classes(classes_path):
    with open(classes_path, 'rb') as f:
        return pickle.load(f)

def generate_bottlenecks(src_path, tgt_path, pre_trained_nw,
nw_input_shape, cls_path):
    """
    Função para cálculo dos bottlenecks

    Parametros:
        - src_path: caminho onde as imagens devem ser encontradas
        - tgt_path: local onde os bottlenecks devem ser salvos
    """

    # model = InceptionV3(include_top=False, weights='imagenet')
    model = pre_trained_nw(include_top = False, weights = "imagenet")

    print(model.summary())

    print()

    data_generator = ImageDataGenerator(
        rescale=1.0/255.0
    )

    classes = None
    if cls_path is not None:
        classes = load_classes(cls_path)

    generator = data_generator.flow_from_directory(
        src_path, # this is the target directory
        target_size=nw_input_shape, # all images will be resized to
299x299
        classes = classes
    )

    save_classes(generator.class_indices, tgt_path + "_classes")

    bottleneck_batches_X = None
    bottleneck_batches_Y = None

    print()
    print("Building bottlenecks...")

```

```

    for _ in tqdm(range(int(np.ceil(generator.samples /
generator.batch_size)))):
        bX, bY = generator.next()
        if bottleneck_batches_X is None:
            bottleneck_batches_X = model.predict(bX)
            bottleneck_batches_Y = bY
        else:
            bottleneck_batches_X =
np.concatenate((bottleneck_batches_X, model.predict(bX)))
            bottleneck_batches_Y =
np.concatenate((bottleneck_batches_Y, bY))
        print()
        print("Bottlenecks X shape:", bottleneck_batches_X.shape)
        print("Bottlenecks Y shape:", bottleneck_batches_Y.shape)
        print()
        fts_path = tgt_path + "_X.npy"
        lbl_path = tgt_path + "_Y.npy"
        cls_path = tgt_path + "_classes"
        print("Saving bottlenecks X at", fts_path)
        print("Saving bottlenecks Y at", lbl_path)
        print("Saving classes dictionary at", cls_path)
        np.save(fts_path, bottleneck_batches_X)
        np.save(lbl_path, bottleneck_batches_Y)
        save_classes(generator.class_indices, cls_path)

if __name__ == "__main__":
    """
    Default arguments are:

        - nw: The network to be used
          - Xception: input size (299, 299)
          - VGG16: input size (224, 224)
          - VGG19: input size (224, 224)
          - ResNet50: input size (224, 224)
          - InceptionV3: input size (299, 299)
          - InceptionResNetV2: input size (299, 299)
          - MobileNet: input size (224, 224)
          - DenseNet121: input size (224, 224)
          - DenseNet169: input size (224, 224)
          - DenseNet201: input size (224, 224)
          - NASNetMobile: input size (224, 224)
          - NASNetLarge: input size (331, 331)
          - MobileNetV2: input size (224, 224)
    """

pre_trained_networks = {
    "Xception": {
        "input_shape": (299, 299),
        "network": Xception,
    },
    "VGG16": {
        "input_shape": (224, 224),
        "network": VGG16,
    },
    "VGG19": {
        "input_shape": (224, 224),
        "network": VGG19,
    },
    "ResNet50": {
        "input_shape": (224, 224),
        "network": ResNet50,
    },
}

```

```

    },
    "InceptionV3": {
        "input_shape": (299, 299),
        "network": InceptionV3,
    },
    "InceptionResNetV2": {
        "input_shape": (299, 299),
        "network": InceptionResNetV2,
    },
    "MobileNet": {
        "input_shape": (224, 224),
        "network": MobileNet,
    },
    "DenseNet121": {
        "input_shape": (224, 224),
        "network": DenseNet121,
    },
    "DenseNet169": {
        "input_shape": (224, 224),
        "network": DenseNet169,
    },
    "DenseNet201": {
        "input_shape": (224, 224),
        "network": DenseNet201,
    },
    "NASNetMobile": {
        "input_shape": (224, 224),
        "network": NASNetMobile,
    },
    "NASNetLarge": {
        "input_shape": (331, 331),
        "network": NASNetMobile,
    },
    "MobileNetV2": {
        "input_shape": (224, 224),
        "network": MobileNetV2,
    }
}
}

```

```

parser = argparse.ArgumentParser(description = "Generates
bottlenecks for the selected classes")

```

```

parser.add_argument("-i", metavar="path", type=str, required=True,
help="Path to the images that will be used to produce the
bottlenecks")

```

```

parser.add_argument("-o", metavar="path", type=str, required=True,
help="Path to the files where the bottlenecks should be stored")

```

```

parser.add_argument("--network", type=str, required=False,
default="InceptionV3",
help="Pre-trained network that will be used to generate the
bottlenecks. Default: InceptionV3. Possible values:[Xception, VGG16,
VGG19, ResNet50, InceptionV3, InceptionResNetV2, MobileNet,
DenseNet121, DenseNet169, DenseNet201, NASNetMobile, NASNetLarge,
MobileNetV2]")

```

```

parser.add_argument("--classespath", metavar="classes path",
type=str, default=None, help="Path to the classes dictionary
[optional]. Default: None")

```

```

args = parser.parse_args()

src_path = args.i
tgt_path = args.o + "_" + str(date.today()).replace("-", "")
cls_path = args.classspath

pre_trained_nw = pre_trained_networks[args.network]["network"]
nw_input_shape = pre_trained_networks[args.network]["input_shape"]
generate_bottlenecks(src_path, tgt_path, pre_trained_nw,
nw_input_shape, cls_path)

# src_path = "/scratch/drddl/Dataset/Val"
# tgt_path = "/scratch/drddl/bottlenecks_validation" + "_" +
str(date.today()).replace("-", "")
# generate_bottlenecks(src_path=src_path, tgt_path=tgt_path)

```

ANEXO IV: TRAIN_TOPLAYER.PY

```

import keras.backend as K
from keras.models import Model
from keras.layers import (Input, Dense, Conv2D, Activation,
GlobalAveragePooling2D, GlobalMaxPooling2D, MaxPooling2D,
AveragePooling2D, Flatten, BatchNormalization, Dropout)
from keras.applications import *
from keras.preprocessing.image import *
from keras.optimizers import *
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau,
TensorBoard
from keras.losses import *
from datetime import date
import numpy as np
import os
from tqdm import tqdm
import argparse
from datetime import date
import random

def build_top_layer(input_shape):
    """
    Função utiliza somente para construir o top layer
    """

    inputs = Input(shape=input_shape)

    # Este tipo de pooling pode ser alterado
    pool0 = GlobalAveragePooling2D()(inputs)
    dense0 = Dense(512, activation="tanh")(pool0)
    dpo0 = Dropout(0.45)(dense0)
    dense1 = Dense(64, activation="relu")(dpo0)
    dpo1 = Dropout(0.35)(dense1)
    # Pra multiplas classes a função de ativação deve ser
    modificada para softmax
    # e a quantidade de unidades deve corresponder a quantidade de
    classes
    outputs = Dense(6, activation="softmax")(dpo1) # -> camada de
    saída para 15 classes
    #outputs = Dense(1, activation="sigmoid")(dpo1)

    top_layer = Model(inputs=inputs, outputs=outputs)

```

```

return top_layer

def train_toplayer(train_bottlenecks_X_path, train_bottlenecks_Y_path,
                  valid_bottlenecks_X_path=None,
                  valid_bottlenecks_Y_path=None,
                  epochs = 50, lr = 0.001,
                  optimizer = Adam, loss = categorical_crossentropy,
                  model_path = "top_layer.h5", log_dir = 'logs'):
    """
    Funcao responsã;vel por treinar o top layer do classificador

    Parametros:
        - bottlenecks_X: vetor com as features bottlenecks
        - bottlenecks_Y: vetor com os labels dos bottlenecks
        - epochs: quantidade de épocas em que a rede deve ser
treinada
        - lr: learning rate
        - optimizer: tipo de otimizador que deve ser utilizado
        - weights_path: caminho onde os pesos devem ser salvos
    """

    train_bottlenecks_X = np.load(train_bottlenecks_X_path)
    train_bottlenecks_Y = np.load(train_bottlenecks_Y_path)

    input_shape = train_bottlenecks_X.shape[1:]

    top_layer = build_top_layer(input_shape)
    top_layer.summary()

    print()
    print("train BNX shape:", train_bottlenecks_X.shape)
    print("train BNY shape:", train_bottlenecks_Y.shape)

    validation_data = None
    if valid_bottlenecks_X_path is not None and
valid_bottlenecks_Y_path is not None:
        valid_bottlenecks_X = np.load(valid_bottlenecks_X_path)
        valid_bottlenecks_Y = np.load(valid_bottlenecks_Y_path)
        validation_data = (valid_bottlenecks_X, valid_bottlenecks_Y)
        print("valid BNX shape:", valid_bottlenecks_X.shape)
        print("valid BNY shape:", valid_bottlenecks_Y.shape)

    print()

    if log_dir is not None:
        if os.path.exists(log_dir):
            if os.listdir(log_dir):
                os.system("rm -rf" + log_dir + "/*")
            else:
                os.mkdir(log_dir)

    best_weights_path =
"best_weights_{0:08x}_{1}.h5".format(random.SystemRandom().randrange(2*
*32),

str(date.today()).replace("-", ""))

    callbacks = [
        ModelCheckpoint(best_weights_path, verbose=1,
save_best_only=True),

```

```

        ReduceLROnPlateau(monitor='val_loss', factor=0.90, patience=4,
min_lr=lr*0.1)
    ]
    if log_dir is not None:
        callbacks.append(TensorBoard(log_dir=log_dir))

    if(optimizer == Adam):
        optimizer = optimizer(lr, amsgrad = True)
    else:
        optimizer = optimizer(lr)

    top_layer.compile(optimizer=optimizer, loss=loss,
metrics=["accuracy"])

    batch_size=128

    top_layer.fit(x=train_bottlenecks_X, y=train_bottlenecks_Y,
epochs=epochs,
                validation_data=validation_data,
callbacks=callbacks)

    top_layer.save(model_path)

if __name__ == "__main__":
    losses = {
        "mean_squared_error": mean_squared_error,
        "mean_absolute_error": mean_absolute_error,
        "mean_absolute_percentage_error":
mean_absolute_percentage_error,
        "mean_squared_logarithmic_error":
mean_squared_logarithmic_error,
        "categorical_hinge": categorical_hinge,
        "logcosh": logcosh,
        "categorical_crossentropy": categorical_crossentropy,
        "sparse_categorical_crossentropy":
sparse_categorical_crossentropy,
        "kullback_leibler_divergence": kullback_leibler_divergence,
        "poisson": poisson,
        "cosine_proximity": cosine_proximity
    }

    optimizers = {
        "SGD": SGD,
        "RMSprop": RMSprop,
        "Adagrad": Adagrad,
        "Adadelta": Adadelta,
        "Adam": Adam,
        "Adamax": Adamax,
        "Nadam": Nadam
    }

    parser = argparse.ArgumentParser(description="Train a top layer
network")

    parser.add_argument("--trainBNX", metavar="train BN X", type=str,
required=True,
                        help="Path to the bottlenecks features used for
training")

    parser.add_argument("--trainBNY", metavar="train BN Y", type=str,
required=True,

```

```

        help="Path to the labels for the bottlenecks
features for training")

    parser.add_argument("--validBNX", metavar="valid BN X", type=str,
default=None,
        help="Path to the bottlenecks features used for
validation")

    parser.add_argument("--validBNY", metavar="valid BN X", type=str,
default=None,
        help="Path to the labels for the bottlenecks
features for validation")

    parser.add_argument("--epochs", metavar="N epochs", type=int,
default=50)

    parser.add_argument("--lr", metavar="LR", type=float,
default=0.001,
        help="Learning rate. Default: 0.001")

    parser.add_argument("--optimizer", metavar="optimizer", type=str,
default="SGD",
        help="Optimizer user to train the top-layer.
Default: SGD. Possible values: [SGD, RMSprop, Adagrad, Adadelta, Adam,
Adamax, Nadam] ")

    parser.add_argument("--loss", metavar="loss", type=str,
default="categorical_crossentropy",
        help="Type of loss to be used during training.
Default: categorical_crossentropy. Possible values:
[mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, mean_squared_logarithmic_error,
categorical_hinge, logcosh, categorical_crossentropy,
sparse_categorical_crossentropy, kullback_leibler_divergence, poisson,
cosine_proximity]")

    parser.add_argument("--modelpath", metavar="model path", type=str,
default="top_layer",
        help="Path to where the top layer model should
be saved. Default: top_layer_[date]")

    parser.add_argument("--logdir", metavar="logs path", type=str,
default=None,
        help="Path to where the logs used for
tensorboard should be saves. If no path is specified, no logs will be
saved")

    args = parser.parse_args()

    trainBNX = args.trainBNX
    trainBNY = args.trainBNY
    validBNX = args.validBNX
    validBNY = args.validBNY

    epochs = args.epochs
    lr = args.lr
    optimizer = args.optimizer
    loss = args.loss
    model_path = args.modelpath + "_" + str(date.today()).replace("-",
"") + "_model.h5"
    log_dir = args.logdir

```



```

print("trainBNX:", trainBNX)
print("trainBNY:", trainBNY)
print("validBNX:", validBNX)
print("validBNY:", validBNY)

print("epochs:", epochs)
print("lr:", lr)
print("optimizer:", optimizer)
print("loss:", loss)
print("model_path:", model_path)
print("log_dir:", log_dir)

train_toplayer(train_bottlenecks_X_path = trainBNX,
train_bottlenecks_Y_path = trainBNY, valid_bottlenecks_X_path =
validBNX, valid_bottlenecks_Y_path = validBNY, epochs = epochs, lr =
lr, optimizer = optimizers[optimizer], loss = losses[loss], model_path
= model_path, log_dir = log_dir)

```

ANEXO V: TEST_MODEL.PY

```

import argparse
import pickle
import numpy as np
import sklearn.metrics as metrics
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
import keras.backend as K

def load_classes(classes_path):
    with open(classes_path, 'rb') as f:
        return pickle.load(f)

def test_top_layer(model_path, test_bottlenecks_X_path,
test_bottlenecks_Y_path, classes_path):
    """
    """
    # Loading saved model
    top_layer = load_model(model_path)
    print(top_layer.summary())
    print()

    classes = None
    if classes_path is not None:
        classes = load_classes(classes_path)
        id2Classes = dict(zip(classes.values(), classes.keys()))
        # v_id2Classes = np.vectorize(lambda x: id2Classes[x])

    test_bottlenecks_X = np.load(test_bottlenecks_X_path)
    test_bottlenecks_Y = np.load(test_bottlenecks_Y_path)
    # test_bottlenecks_Y labels =
    v_id2Classes(np.argmax(test_bottlenecks_Y, axis=-1))

    eval_loss, eval_accu = top_layer.evaluate(x=test_bottlenecks_X,
y=test_bottlenecks_Y)
    print("Loss:", eval_loss)
    print("Accuracy:", eval_accu)

```

```

preds = top_layer.predict(test_bottlenecks_X)

print()
print("Classes:", id2Classes)
print()

conf_mat = metrics.confusion_matrix(np.argmax(test_bottlenecks_Y,
axis=-1), np.argmax(preds, axis=-1))

print("Confusion matrix:")
print(conf_mat)
print()

if __name__ == "__main__":
    '''
    '''
    parser = argparse.ArgumentParser(description="Test a top layer
network")

    parser.add_argument("--modelpath", metavar="weights path",
type=str, help="Path to where the top layer model have been saved.")

    parser.add_argument("--testBNX", metavar="test BN X", type=str,
required=True,
                        help="Path to the bottlenecks features used for
testing")

    parser.add_argument("--testBNY", metavar="test BN Y", type=str,
required=True,
                        help="Path to the labels for the bottlenecks
features for testing")

    parser.add_argument("--classespath", metavar="classes path",
type=str, default=None, help="Path to the classes dictionary.
[Optional]")

    args = parser.parse_args()

    model_path = args.modelpath
    testBNX = args.testBNX
    testBNY = args.testBNY
    classes_path = args.classespath

    print("model_path:", model_path)
    print("classes path:", classes_path)
    print()

    test_top_layer(model_path, testBNX, testBNY, classes_path)

```