

# Classificação de Condições Climáticas Usando Redes Neurais Convolucionais

Júlia Marques Boaventura, Vinícius Meireles Pereira Santos, João Fernando Mari

<sup>1</sup>Instituto de Ciências Exatas e Tecnológicas – Universidade Federal de Viçosa (UFV), Rio Paranaíba – MG, Brasil

{julia.m.boaventura, vinicius.m.santos, joaof.mari}@ufv.br

**Resumo**—A mudança climática é uma realidade que afeta o mundo inteiro, com impactos cada vez mais significativos sobre as populações e consequências financeiras crescentes. Analisar e identificar precisamente as condições climáticas de determinados locais ainda representa um desafio, devido à complexidade e variabilidade dos dados envolvidos. Esta pesquisa propõe o uso de Redes Neurais Convolucionais (CNNs) para a classificação de condições climáticas a partir de imagens, categorizando-as de acordo com o fenômeno meteorológico presente. O objetivo final é fornecer uma ferramenta que auxilie meteorologistas na análise precisa do clima em momentos registrados, contribuindo para uma maior eficiência e confiabilidade nas análises meteorológicas.

**Index Terms**—Clima e Meteorologia, Redes Neurais Convolucionais, Visão Computacional, Classificação de Imagens

## I. INTRODUÇÃO

A mudança climática é uma realidade que afeta o mundo inteiro. Seus impactos são amplos e abrangem desde desastres naturais, como inundações e enchentes que devastam infraestrutura construída pelo ser humano, até consequências no setor agrícola, essencial para a produção de alimentos e a manutenção da vida. Além das alterações nas condições climáticas médias, a Terra enfrenta “*feedbacks*” ambientais e impactos irreversíveis e catastróficos no sistema climático, como o colapso da circulação termohalina e o derretimento da camada de gelo da Groenlândia [7].

Diante dessa realidade, tornou-se necessário adotar medidas de adaptação. Essas adaptações podem ser reativas, desencadeadas por eventos climáticos passados ou em andamento, ou antecipatórias, baseando-se em avaliações de condições climáticas futuras, com o objetivo de mitigar possíveis danos [6].

Com o avanço das tecnologias de aprendizado de máquina, ferramentas como as Redes Neurais Convolucionais (CNNs) têm se mostrado extremamente eficazes na classificação de condições climáticas. Essas redes permitem a análise de padrões complexos em dados climáticos, os quais seriam difíceis de detectar por métodos tradicionais.

Esse avanço tecnológico pode ajudar a suprir uma lacuna existente na análise precisa das condições climáticas, uma área que enfrenta desafios devido à complexidade e variabilidade dos dados [6]. Por esse motivo, este trabalho propõe a identificação e classificação de condições climáticas utilizando Redes Neurais Convolucionais. O objetivo principal é auxiliar especialistas na previsão e análise de eventos climáticos extremos, como tempestades e secas, a partir de dados complexos.

Com isso, buscamos facilitar a tomada de decisões mais informadas e precisas, contribuindo para o desenvolvimento de estratégias mais eficazes na mitigação dos impactos das mudanças climáticas e no aprimoramento da capacidade de resposta e adaptação.

Todos os arquivos e códigos relacionados a este projeto estão disponíveis em um repositório no *GitHub*, que pode ser acessado em: [https://github.com/vinimeirelres/climaclassification]. O repositório contém toda a implementação do modelo proposto, os dados utilizados e as instruções necessárias para a reprodução dos resultados, incluindo um *README* detalhado para facilitar a navegação.

As seções seguintes deste artigo estão estruturadas da seguinte maneira: a Seção II apresenta uma revisão da literatura atual sobre o tema, com destaque para trabalhos semelhantes ao proposto. A Seção III descreve os materiais e métodos utilizados, incluindo o banco de dados, os modelos testados, os critérios de pré-processamento, os parâmetros de teste e as métricas empregadas para a avaliação dos modelos. Na Seção IV, são apresentados os resultados obtidos a partir dos testes executados, seguidos de uma análise detalhada. Por fim, na Seção V, concluímos o artigo com uma discussão sobre os resultados, suas implicações e possíveis trabalhos futuros.

## II. REVISÃO BIBLIOGRÁFICA

O trabalho [8] apresenta a criação de um método para reconhecimento automático de condições climáticas extremas (como tempestades, nevascas, neblina e dias ensolarados) utilizando redes neurais convolucionais (CNNs). A aplicação é voltada para rodovias e sistemas de transporte inteligente, utilizando o *WeatherDataset*, um conjunto de dados com 16.635 imagens, dividido em quatro categorias climáticas. Os autores empregaram o *GoogLeNet* [10], modelo pré-treinado no conjunto de dados *ILSVRC-2012* [11]. No *WeatherDataset*, as imagens foram redimensionadas para 224×224 pixels antes de serem utilizadas no treinamento. O modelo alcançou uma acurácia inicial de 90,1% sem ajuste fino e, após 40.000 iterações com ajuste fino, atingiu 94,5% de acurácia de treinamento, demonstrando uma melhoria significativa com o refinamento.

No estudo [9], os autores propõem uma abordagem baseada em CNNs para detectar eventos climáticos extremos, como ciclones tropicais, frentes meteorológicas e rios atmosféricos, utilizando grandes volumes de dados climáticos. Para garantir

a precisão na representação dos padrões climáticos, as imagens foram redimensionadas para diferentes tamanhos: 32x32 pixels para ciclones tropicais, 148x224 pixels para rios atmosféricos e 27x60 pixels para frentes climáticas, com caixas delimitadoras centralizadas nos eventos. A arquitetura de *CNN*, composta por 4 camadas (2 convolucionais e 2 totalmente conectadas), foi otimizada com técnicas *Bayesianas* e treinada no super-computador Edison (*NERSC*) utilizando a biblioteca *Neon*. Os resultados apresentaram acurácia de classificação variando entre 89% e 99%, com destaque para 99% nos ciclones tropicais e 89% nas frentes meteorológicas.

Em [12], os autores utilizaram duas arquiteturas de *CNNs*, *ResNet* e *AlexNet*, combinadas com um *SVM* multiclasse (*MCSVM*) para classificar imagens climáticas em dois experimentos. No primeiro, com um *dataset* de duas classes (ensolarado e nublado), o modelo *ResNet-MCSVM* obteve acurácia de 92% para imagens ensolaradas e 88% para imagens nubladas, enquanto o *AlexNet-MCSVM* alcançou 86% e 75%, respectivamente. No segundo experimento, utilizando a combinação dos *datasets* *Weather Database*, *DesnowNet* e *D-Hazy*, formou-se um conjunto com quatro classes (ensolarado, nublado, nevoeiro e nevado), cada uma com 1.000 imagens. A divisão dos dados seguiu a proporção de 70% para treinamento e 30% para teste. Os melhores resultados foram obtidos pelo modelo *CNN-MCSVM*, com acurácias de 97% para ensolarado, 100% para nublado, 96% para nevoeiro e 95% para nevado. Esses resultados demonstram a eficácia do uso combinado de *CNNs* e *SVM* multiclasse em cenários climáticos variados.

No trabalho [13], os autores abordaram a classificação de imagens de nuvens capturadas do solo como tempo limpo (*clear*) ou nublado (*cloudy*). Utilizou-se um *dataset* inicial com 1.500 imagens, expandido para 5.197 ao dividir cada imagem em quatro partes, removendo fragmentos com ruído. As imagens foram manualmente categorizadas e divididas em conjuntos de treino, validação e teste. Modelos pré-treinados, como *MobileNet V2*, *VGG-16*, *ResNet-152 V2* e *DenseNet-201*, foram ajustados com camadas específicas congeladas e treinamento refinado nas camadas superiores. O *VGG-16* destacou-se, alcançando 91,4% de acurácia, com equilíbrio entre precisão e *recall*, enquanto o *DenseNet-201* apresentou a maior *AUC* (97,77%) e a menor perda por entropia cruzada (0,2). Apesar disso, o *VGG-16* superou em acurácia geral. Já o *MobileNet V2* foi o mais rápido, mas menos preciso, com acurácia de 88,8%. Os autores sugerem que o uso de *datasets* maiores e a exploração de novos modelos podem melhorar ainda mais a robustez e a precisão dos resultados.

### III. MATERIAIS E MÉTODOS

#### A. Base de Dados

O conjunto de dados foi coletado na plataforma *Kaggle* [5] e contém 6.862 imagens representando diferentes tipos de clima. As imagens estão divididas em 11 classes: *dew* (orvalho), com 698 imagens; *fog/smog* (neblina ou poluição atmosférica), com 851 imagens; *frost* (geada), com 475 imagens; *glaze* (camada de gelo), com 639 imagens; *hail* (granizo), com 591 imagens; *lightning* (relâmpago), com 377 imagens; *rain* (chuva), com

526 imagens; *rainbow* (arco-íris), com 232 imagens; *rime* (geada branca), com 1.160 imagens; *sandstorm* (tempestade de areia), com 692 imagens; e *snow* (neve), com 621 imagens.

Nesta pesquisa, os modelos foram treinados utilizando as imagens dessas 11 classes, sendo que a saída prevista corresponde a uma das classes mencionadas. A Figura 1 apresenta amostras de imagens do conjunto de dados.

#### B. Arquiteturas

Neste trabalho, foram treinadas quatro arquiteturas de aprendizado profundo: as *CNNs* *AlexNet*, *MobileNetV3-Small*, *EfficientNet-B0* e *ResNet50*. Todos os modelos foram carregados e treinados utilizando a biblioteca *PyTorch*.

A *AlexNet* é um modelo proposto por *Krizhevsky*, *Sutskever* e *Hinton* [1], com cinco camadas convolucionais e três camadas totalmente conectadas. Este modelo inovou ao utilizar *ReLU* como função de ativação e ao empregar *GPUs* para acelerar o treinamento da rede.

A *MobileNet* é um modelo projetado para dispositivos móveis, ou seja, com menor poder de computação. Essa classe de Redes Neurais Convolucionais *open-source* utiliza convoluções separáveis em profundidade, o que reduz consideravelmente o número de parâmetros em comparação com convoluções regulares de mesma profundidade. O modelo foi proposto por *Howard et al.* [2]. A versão utilizada neste trabalho é a *MobileNetV3-Small*.

A *EfficientNet* é uma Rede Neural Convolutacional desenvolvida com foco em escalabilidade eficiente e melhor desempenho em tarefas de classificação de imagens. O modelo, proposto por *Tan e Le* [3], utiliza uma abordagem baseada em *compound scaling*, que ajusta de forma coordenada a profundidade, a largura e a resolução da rede para otimizar os recursos computacionais.

A *ResNet50* (*Residual Neural Network*) é uma Rede Neural Convolutacional com 50 camadas profundas. Introduzida pela primeira vez por *Kaiming He*, *Xiangyu Zhang*, *Shaoqing Ren* e *Jian Sun* em 2015 [4], a *ResNet* utiliza blocos residuais para facilitar o treinamento de redes muito profundas, evitando problemas como o gradiente desaparecendo.

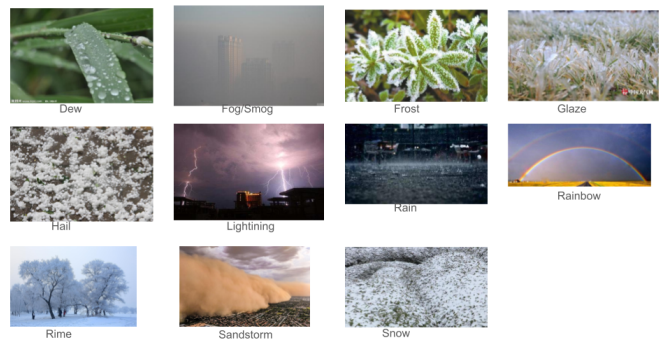


Figura 1. Exemplos de imagens presentes na base de dados

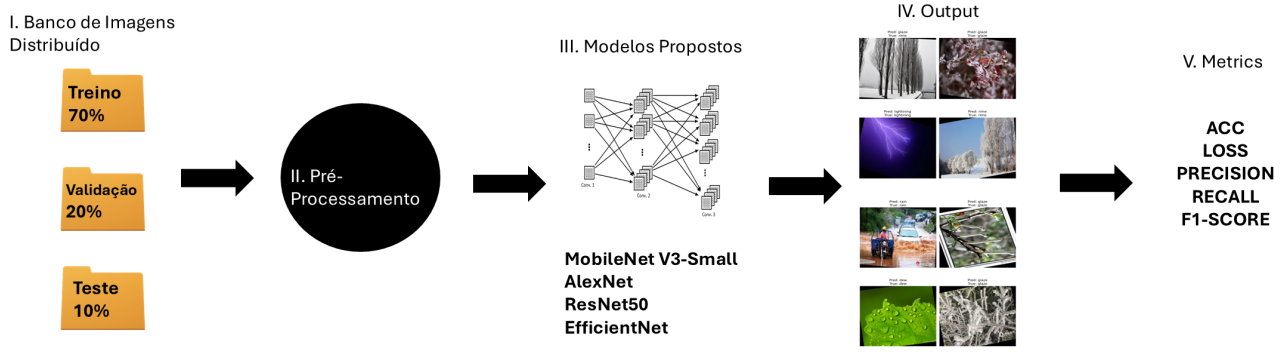


Figura 2. Diagrama Geral da Metodologia Aplicada

### C. Pré-Processamento

Todas as imagens foram redimensionadas para o tamanho de  $224 \times 224$  e normalizadas utilizando a média e o desvio padrão do *ImageNet*. Com o objetivo de aumentar a diversidade do conjunto de dados e reduzir o risco de *overfitting*, foram aplicadas as seguintes técnicas de *Data Augmentation*: espelhamento horizontal aleatório com probabilidade padrão de 50% e rotação da imagem aleatória dentro de um intervalo de  $\pm 15^\circ$ . As imagens foram, por fim, convertidas para tensores para serem processadas pelo *PyTorch*. Para essa etapa do pré-processamento, foi utilizada a biblioteca *torchvision.transforms*. O conjunto de dados foi dividido em três subconjuntos: treinamento (70%), validação (20%) e teste (10%). A Figura 2 apresenta um resumo da metodologia utilizada neste trabalho.

### D. Parâmetros de Treinamento e Avaliação

Definimos parâmetros básicos no código para a execução do treinamento, validação e testes, como o tamanho do lote (*batch size*) de 64, uma SEED de 42 e uma taxa de aprendizado (*learning rate*) de 0.001. Utilizamos como função de perda a *Cross-Entropy Loss* e o otimizador escolhido foi o *Adam*, carregado utilizando a biblioteca *PyTorch*. Os treinamentos e validações foram realizados em todos os modelos utilizando 50 e 100 épocas.

### E. Avaliação dos Modelos

Nós avaliamos o treinamento e a validação dos modelos utilizando acurácia (*accuracy*) e perda (*Cross-Entropy Loss*). No conjunto de teste, avaliamos precisão, *recall*, *F1-score* e a acurácia (*accuracy*).

Além disso, geramos gráficos de acurácia e perda, uma matriz de confusão, gráficos com métricas por classe (precisão, *recall* e *F1-score*) e gráficos com as curvas *ROC* para cada classe.

Ao analisar as métricas obtidas, é possível extrapolar a capacidade de classificação dos modelos para amostras não vistas.

A acurácia, perda, precisão, *recall* e *F1-score* estão definidas nas equações de 1 a 5, respectivamente:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{F1-Score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

Onde *TP* (*True Positive*) é o Verdadeiro Positivo, *TN* (*True Negative*) é o Verdadeiro Negativo, *FP* (*False Positive*) é o Falso Positivo, *FN* (*False Negative*) é o Falso Negativo, *N* é o número de amostras no lote (*batch size*),  $y_i$  é o rótulo da *i*-ésima amostra, e  $p_i$  é a probabilidade prevista para a classe positiva na *i*-ésima amostra.

### F. Recursos Computacionais

Os testes foram executados no ambiente *Google Colab*, utilizando uma *GPU NVIDIA Tesla T4* com 15 GB de memória dedicada e 12,7 GB de memória do sistema. Além disso, foram realizados testes em computadores com um processador *Intel Core i5-13450HX*, 16 GB de memória *DDR5* e uma *GPU NVIDIA GeForce RTX 4060 Ti* com 8 GB de memória dedicada.

Os ambientes de desenvolvimento foram todos baseados na linguagem de programação *Python*, utilizando o *framework PyTorch* com suporte a *CUDA* (no caso do segundo ambiente, nas versões 2.4.1 e 12.1, respectivamente). Foram também empregadas as bibliotecas *TorchVision*, *NumPy*, *Scikit-Learn*, *Matplotlib* e *Seaborn*.

#### IV. RESULTADOS E DISCUSSÃO

As Tabelas I e II apresentam os resultados obtidos para cada um dos conjuntos do *dataset*: treino, validação e teste, em cada uma das arquiteturas utilizadas, que incluem *ResNet50*, *MobileNet V2-Small*, *AlexNet* e *EfficientNet B0*. As tabelas mostram a acurácia (ACC) e a perda (LOSS) registradas após a execução de 50 e 100 épocas, respectivamente, para cada modelo. Em todos os experimentos, foi utilizado um *batch size* de 64, com o embaralhamento (*shuffle*) ativado para o conjunto de treinamento. Para a avaliação, foi considerado o modelo sem os pesos pré-treinados da *ImageNet*, utilizando os pesos da melhor época de validação, e a avaliação foi realizada com base no conjunto de teste.

As Tabelas III e IV apresentam as médias das métricas Precisão (*Precision*), Sensibilidade (*Recall*) e *F1-Score* para cada modelo após a execução de 50 e 100 épocas, respectivamente. A média foi utilizada para simplificar a apresentação dos resultados, considerando que a saída do modelo é composta por uma métrica separada por classe, o que dificultaria a exibição detalhada e clara de todas as informações.

Nas Figuras 3 e 4, são apresentados os gráficos das perdas e acurácias de treino e validação para os experimentos realizados. Nas Figuras 5 e 6, exibimos os gráficos das perdas de treino e validação após 50 e 100 épocas de execução, respectivamente. Nas Figuras 7 e 8, apresentamos as matrizes de confusão geradas com base na avaliação dos modelos, enquanto, nas Figuras 9 e 10, mostramos os gráficos das métricas por classe, por modelo, obtidos a partir do conjunto de teste.

Na avaliação com 50 épocas, a *EfficientNet B0* apresentou o melhor desempenho global, atingindo 98,50% de acurácia no conjunto de treino e 91,69% no conjunto de validação, com a menor perda observada (4,77%). No conjunto de teste, obteve 91,56%, confirmando sua robustez. Em contrapartida, a *AlexNet* registrou os piores resultados, com uma acurácia de 79,39% no conjunto de treino, 76,75% na validação e 76,27% no teste, além da maior perda geral (62,33% no treino e 74,27% na validação). A *ResNet50* e a *MobileNet V2-Small* apresentaram desempenhos intermediários, com a *MobileNet* destacando-se pela acurácia no treino (97,46%) e validação (89,87%).

Após 100 épocas, a *EfficientNet B0* manteve sua superioridade, alcançando 98,58% de acurácia no treino, 91,25% na validação e 96,51% no teste, com a menor perda registrada (4,15%). A *ResNet50* e a *MobileNet V2-Small* apresentaram melhorias significativas nos valores de acurácia e redução na perda em relação aos resultados de 50 épocas. A *AlexNet*, embora tenha mostrado progresso, ainda apresentou os piores resultados, com acurácia no treino de 81,36% e perda de 54,35%.

Nas Tabelas III e IV, que analisam as métricas de precisão (*Precision*), sensibilidade (*Recall*) e *F1-Score*, destaca-se novamente a *EfficientNet B0*, com os melhores valores em ambas as condições de 50 e 100 épocas. Após 100 épocas, a *EfficientNet* atingiu 96,88% de precisão, 96,19% de *recall* e 96,51% de

*F1-Score*, superando significativamente as demais arquiteturas. A *AlexNet*, embora tenha mostrado alguma melhora em relação à condição de 50 épocas, permaneceu com os piores valores, com *F1-Score* de 80,49%.

Os gráficos apresentados nas Figuras 3 a 10 corroboram os resultados observados nas tabelas, evidenciando a evolução do desempenho dos modelos ao longo do treinamento. A *EfficientNet B0* destacou-se com curvas consistentes de perda decrescente e acurácia crescente, enquanto a *AlexNet* apresentou maior oscilação, indicando dificuldade em generalizar os padrões do *dataset*.

Quando comparado a trabalhos prévios, os resultados obtidos neste estudo são competitivos, especialmente na *EfficientNet B0*, que alcançou acurácias superiores a 96% em todas as métricas avaliadas. Esses resultados ressaltam a importância da escolha da arquitetura no desempenho final, com a *EfficientNet* demonstrando ser a opção mais eficiente para este *dataset*. Por outro lado, o desempenho inferior da *AlexNet* evidencia suas limitações em cenários mais complexos, reforçando a relevância de arquiteturas mais modernas e otimizadas.

Tabela I  
DESEMPENHO DAS ARQUITETURAS COM ACC E LOSS PARA DIFERENTES CONJUNTOS DE 50 ÉPOCAS

Arquitetura	Conjunto	ACC (%)	LOSS (%)
ResNet50	Treino	95,04	13,22
	Validação	88,70	<b>33,83</b>
	Teste	88,50	-
MobileNet V2-Small	Treino	97,46	07,95
	Validação	89,87	46,80
	Teste	89,52	-
AlexNet	Treino	79,39	62,33
	Validação	76,75	74,27
	Teste	76,27	-
EfficientNet B0	Treino	<b>98,50</b>	<b>04,77</b>
	Validação	<b>91,69</b>	48,43
	Teste	<b>91,56</b>	-

Tabela II  
DESEMPENHO DAS ARQUITETURAS COM ACC E LOSS PARA DIFERENTES CONJUNTOS DE 100 ÉPOCAS

Arquitetura	Conjunto	ACC (%)	LOSS (%)
ResNet50	Treino	97,29	08,38
	Validação	90,45	<b>36,78</b>
	Teste	88,94	-
MobileNet	Treino	97,46	07,95
	Validação	89,87	46,80
	Teste	89,67	-
AlexNet	Treino	80,68	54,35
	Validação	80,47	70,70
	Teste	81,08	-
EfficientNet	Treino	<b>98,58</b>	<b>04,15</b>
	Validação	<b>91,25</b>	48,91
	Teste	<b>96,51</b>	-

#### V. CONCLUSÃO

Neste trabalho, realizamos diversas análises utilizando uma base de imagens de diferentes tipos de clima, com o objetivo de classificar corretamente cada tipo de clima a partir de entradas previamente desconhecidas. Para isso, utilizamos

Tabela III  
DESEMPENHO DAS ARQUITETURAS COM PRECISION, RECALL E F1 SCORE PARA 50 ÉPOCAS

Arquitetura	Precision (%)	Recall (%)	F1 Score (%)
ResNet50	88,52	88,12	88,23
MobileNet	89,41	89,54	89,44
AlexNet	72,33	75,43	75,66
EfficientNet	<b>91,92</b>	<b>91,25</b>	<b>91,49</b>

Tabela IV  
DESEMPENHO DAS ARQUITETURAS COM PRECISION, RECALL E F1 SCORE PARA 100 ÉPOCAS

Arquitetura	Precision (%)	Recall (%)	F1 Score (%)
ResNet50	89,16	88,41	88,64
MobileNet	89,81	89,85	89,75
AlexNet	81,36	80,14	80,49
EfficientNet	<b>96,88</b>	<b>96,19</b>	<b>96,51</b>

quatro Redes Neurais Convolucionais (CNNs) pré-treinadas: *ResNet50*, *MobileNet V2-Small*, *AlexNet* e *EfficientNet B0*. O pré-processamento das imagens consistiu na normalização do *dataset*, aplicação de técnicas de *Data Augmentation* e posterior divisão da base em conjuntos de treino, validação e teste.

Com essa abordagem e os parâmetros previamente definidos, incluindo treinamentos realizados com 50 e 100 épocas, obtivemos resultados que variaram entre 76,27% e 98,50% de acurácia com 50 épocas, e entre 80,47% e 98,58% com 100 épocas. Entretanto, o modelo *AlexNet* apresentou desempenho inferior em relação aos demais, registrando acurácias menores, perdas mais elevadas, e métricas como precisão (*precision*), sensibilidade (*recall*) e *F1-Score* consideravelmente abaixo do esperado, independentemente do conjunto avaliado.

Os resultados obtidos mostram que os modelos *MobileNet V2-Small*, *ResNet50* e, principalmente, *EfficientNet B0*, que apresentou os melhores resultados, são ferramentas promissoras para a classificação de diferentes tipos de clima por meio de imagens. Esses modelos demonstram potencial para serem aplicados em cenários reais, contribuindo para soluções que envolvem a análise de condições climáticas com base em imagens.

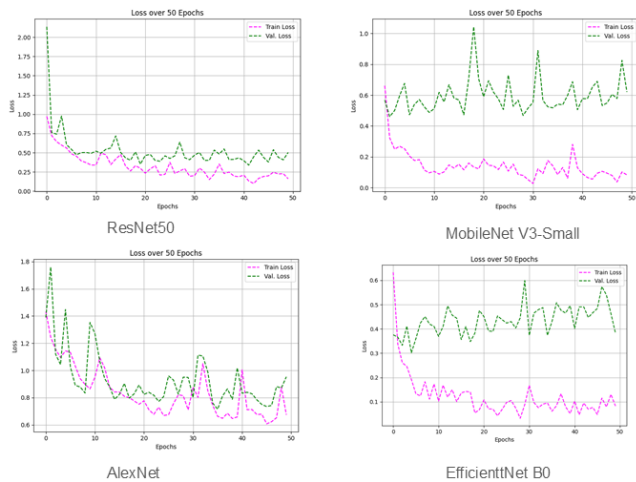


Figura 3. Perdas de Treino e Validação de todos os modelos com execução de 50 épocas

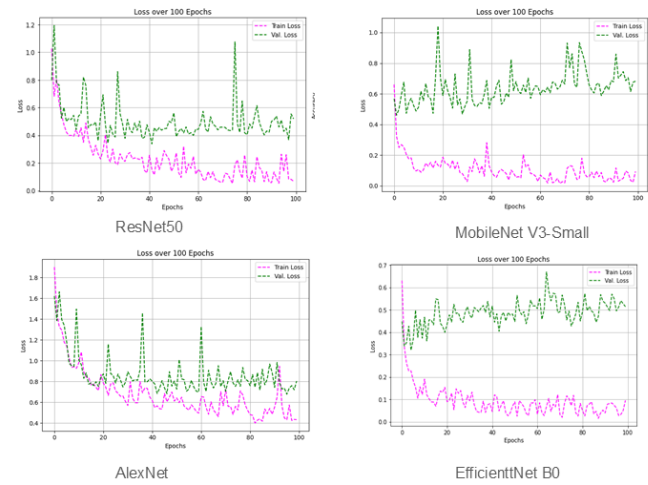


Figura 4. Perdas de Treino e Validação de todos os modelos com execução de 100 épocas

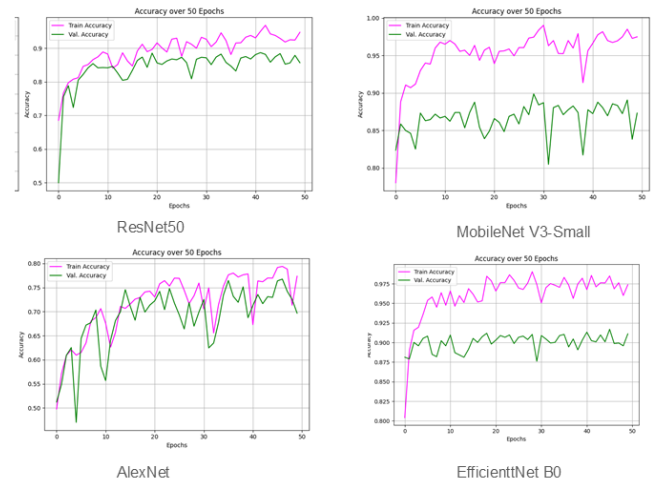


Figura 5. Acurácias de Treino e Validação de todos os modelos com execução de 50 épocas



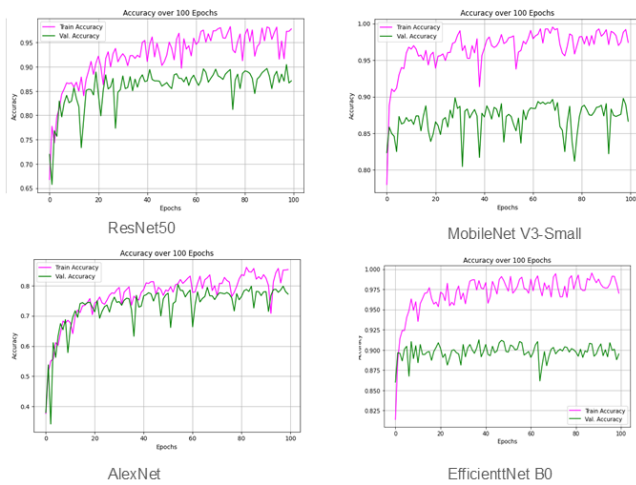


Figura 6. Acurácias de Treino e Validação de todos os modelos com execução de 100 épocas

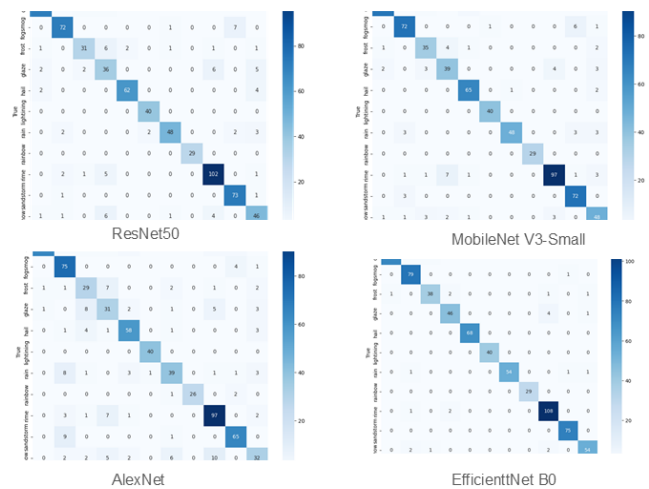


Figura 8. Matrizes de Confusão de todos os modelos geradas após a validação na base de testes com execução de 100 épocas

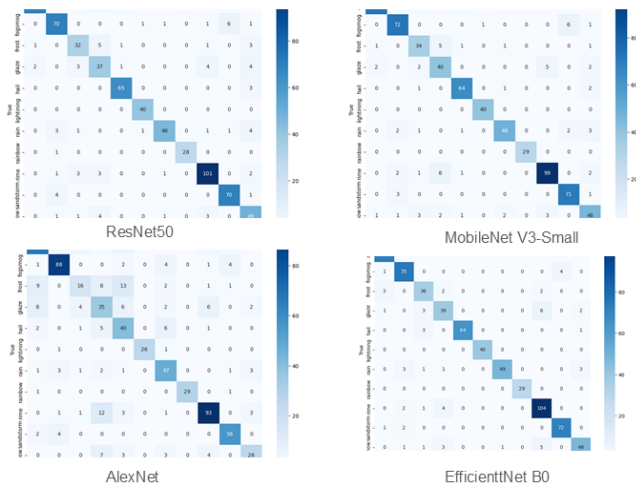


Figura 7. Matrizes de Confusão de todos os modelos geradas após a validação na base de testes com execução de 50 épocas

## REFERÊNCIAS

- [1] A. Krizhevsky, I. Sutskever e G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks,"\*Advances in Neural Information Processing Systems (NeurIPS)\*, vol. 25, pp. 1097–1105, 2012.
- [2] A. Howard, M. Sandler, G. Chu, *et al.*, "Searching for MobileNetV3,"em \*Anais da Conferência Internacional de Visão Computacional (ICCV)\*, 2019, pp. 1314–1324.
- [3] M. Tan e Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,"em \*Anais da 36ª Conferência Internacional de Aprendizado de Máquina (ICML)\*, 2019, pp. 6105–6114.
- [4] K. He, X. Zhang, S. Ren e J. Sun, "Deep Residual Learning for Image Recognition,"em \*Anais da Conferência de Visão Computacional e Reconhecimento de Padrões (CVPR)\*, 2016, pp. 770–778.
- [5] "Weather Image Recognition,"Kaggle, [Online]. Disponível em: <https://www.kaggle.com/datasets/jehanbathena/weather-dataset>. [Acesso em: 16 jan. 2025].
- [6] W. N. Adger, N. W. Arnell e E. L. Tompkins, "Successful adaptation to climate change across scales,"\*Global Environmental Change\*, vol. 15, n° 2, pp. 77–86, 2005.

- [7] H. Nakabuye, "O colapso do gelo da Groenlândia, artigo de José Eustáquio Diniz Alves,"[Online]. Disponível em: <https://www.ecodebate.com.br/2020/01/24/o-colapso-do-gelo-da-groenlandia-artigo-de-jose-eustaquio-diniz-alves/>. [Acesso em: 16 jan. 2025].
- [8] Z. Zhu, *et al.*, "Extreme Weather Recognition Using Convolutional Neural Networks,"em \*Simpósio Internacional de Multimídia da IEEE (ISM)\*, 2016, pp. 621–625, doi: 10.1109/ISM.2016.0133.
- [9] Y. Liu, *et al.*, "Application of Deep Convolutional Neural Networks for Detecting Extreme Weather in Climate Datasets,"\*arXiv preprint\*, arXiv:1605.01156, 2016.
- [10] C. Szegedy, *et al.*, "Going Deeper with Convolutions,"em \*Anais da Conferência de Visão Computacional e Reconhecimento de Padrões (CVPR)\*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [11] O. Russakovsky, *et al.*, "ImageNet Large Scale Visual Recognition Challenge,"\*International Journal of Computer Vision\*, vol. 115, pp. 211–252, 2015.
- [12] J. An, Y. Chen e H. Shin, "Weather Classification Using Convolutional Neural Networks,"em \*2018 International SoC Design Conference (ISOC)\*, Daegu, Coreia do Sul, 2018, pp. 245–246, doi: 10.1109/ISOC.2018.8649921.
- [13] M. Kalkan, G. E. Bostancı, M. S. Güzel, B. Kalkan, Ş. Özsarı, Ö. Soysal e G. Köse, "Cloudy/Clear Weather Classification Using Deep Learning Techniques with Cloud Images,"\*Computers and Electrical Engineering\*, vol. 102, 2022, Art. no. 108271, doi: 10.1016/j.compeleceng.2022.108271.