Classificação de imagens dos datasets CIFAR-10 e Hymenoptera com uso das redes AlexNet e VGG-16

Beatriz A. Benedicto Heleno Victor Souza Salles

2023

Resumo

O presente trabalho tem como objetivo analisar a performance das redes neurais convolucionais AlexNet e VGG-16 para classifcação de imagens em dois tipos de dataset, um multicategoria (CIFAR-10) e um binário (Hymenoptera) a fim de avaliar o impacto de parâmetros como taxa de aprendizado, número de épocas, tamanho de bactch, além de alterações mínimas na arquitetura das redes citadas visando aumento da acurácia.

1 Introdução

Este trabalho se propõe a analisar e comparar os resultados obtidos pelos modelos de redes neurais convolucionais AlexNet e VGG-16, utilizando dois conjuntos de dados distintos: o CIFAR-10 e Hymenoptera. Para alcançar esse objetivo, uma série de experimentos foi conduzida, envolvendo a variação de hiperparâmetros essenciais e a manipulação da arquitetura das redes. Essas análises detalhadas permitirão uma compreensão mais profunda das nuances de desempenho desses modelos em contextos específicos.

2 Modelos implementados

A AlexNet é composta por 8 camadas, incluindo 5 convolucionais e 3 conectadas, a rede introduziu a utilização de camadas de normalização local e dropout, demonstrando a eficácia das CNNs em tarefas de reconhecimento de imagens em grande escala. A VGG-16 possui 16 camadas, incluindo 13 camadas convolucionais e 3 totalmente conectadas, sua abordagem consistente de convoluções 3x3 em cascata e a profunda pilha de camadas tornaram a VGG16 uma referência em arquiteturas de redes neurais para classificação de imagens

Neste trabalho as redes foram implementadas utilizando as bibliotecas TensorFlow e o Keras para a composição das camadas, compilção e treinamento dos modelos. A referência para construção das arquiteturas utilizadas se encontram nas referências do trabalho. A implementação do modelo pode ser encontrada no repositório neste link.

3 Datasets

3.1 CIFAR-10

O dataset CIFAR-10 é um conjunto de dados padrão na área de visão computacional, composto por 60.000 imagens coloridas de 32x32 pixels, divididas em 10 classes, com 6.000 imagens por classe. As classes incluem objetos do cotidiano como 'avião', 'automóvel', 'pássaro', 'gato', 'veado', 'cachorro', 'sapo', 'cavalo', 'navio' e 'caminhão'. O conjunto de dados está dividido em 50.000 imagens para treinamento e 10.000 para teste, garantindo uma distribuição balanceada das classes em ambos os conjuntos. Para o processamento do CIFAR-10, utilizamos funções da biblioteca TensorFlow/Keras, especificamente a função `cifar10.load_data()`, que facilita o carregamento e a separação dos conjuntos de treino e teste. A uniformidade nas dimensões das imagens (32x32 pixels) elimina a necessidade de redimensionamento ou padronização do tamanho das imagens, permitindo que nos concentremos diretamente na construção e no treinamento das redes neurais convolucionais. Ao trabalhar com o CIFAR-10, optamos por categorizar os rótulos das imagens no modo 'categorical', o que é adequado para problemas de classificação multiclasse, como é o caso deste dataset. Essa escolha implica na utilização da função de perda *CategoricalCrossEntropy* durante o treinamento, pois esta função é especialmente projetada para lidar com problemas onde cada entrada pode pertencer a uma entre várias classes. O CIFAR-10 oferece um desafio equilibrado e realista para algoritmos de visão computacional e aprendizado de máquina, sendo amplamente utilizado para avaliar e comparar diferentes arquiteturas de redes neurais em tarefas de classificação de imagens.

3.2 Hymenoptera

O dataset Hymenoptera é composto de imagens divididas em apenas duas classes: formigas (ants) e abelhas (bees). O conjunto de treinamento totaliza 245 imagens, sendo 124 de formigas e 121 de abelhas. O conjunto de teste totaliza 153 imagens sendo 70 de formigas e 83 de abelhas. As dimensões das entradas variam de e 200x100 a 1488x1984 pixels. O dataset foi importado a partir de uma pasta por meio da função `image_dataset_from_directory` da biblioteca Keras, os parâmetros essenciais são: o tamanho da imagem, os rótulos que categorizam as entradas e modelo de rótulo. Optou-se pela realização dos experimentos finais com o dataset processado no modo categorical, porque essa configuração está diretamente ligada à arquitetura original das redes neurais propostas para problemas multiclasse e além disso o modo binário não apresentou diferenças signficativas na acurácia dos modelos a partir de testes iniciais para a escolha do modo. Para esse modo a função de erro a ser utilizada no treinamento também é a *CategoricalCrossEntropy*.

Figura 1 – Exemplos de amostras do datset Hymenoptera

4 Experimentos com o dataset Hymenoptera

Foram realizados na plataforma Google Colab com ambiente de execução contando com as seguintes configurações: RAM 12.GB , GPU RAM 15GB e 78.2GB de espaço em disco. Para o dataset binário há foco especial nas métricas de precisão, recall e F1 Score cujas definições são apresentadas abaixo:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad F1\ Score = 2 * \frac{Precisão * Recall}{Precisão + Recall}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$

Figura 2 – Fórmulas para o cálculo de precisão, recall e F1 score

Para experimentos com aumento de dados o mesmo foi feito a partir da adição, para cada imagem do conjunto de treinamento, a sua rotação em 90º, uma inversão de 180º e a imagem com escala aumentada em 4 vezes.

4.1 AlexNet

O primeiro experimento se trata da implementação da arquitetura original do modelo alterando-se apenas a quantidade de nós da saída para a quantidade de classes do dataset, que são duas.

- **Parâmetros do dataset:** Tamanho da imagem 224x224 normalizadas pelo fator 1/255, modo categórico, sem aumento de dados.
- **Parâmetros do modelo:** Entrada (224,224,3) saída 1 de 2 classes e função de ativação da saída softmax
- **Hiperparâmetros**: Batch tamanho 64, 10 épocas, taxa de aprendizado 0.001

Os resultados obtidos foram os seguintes:

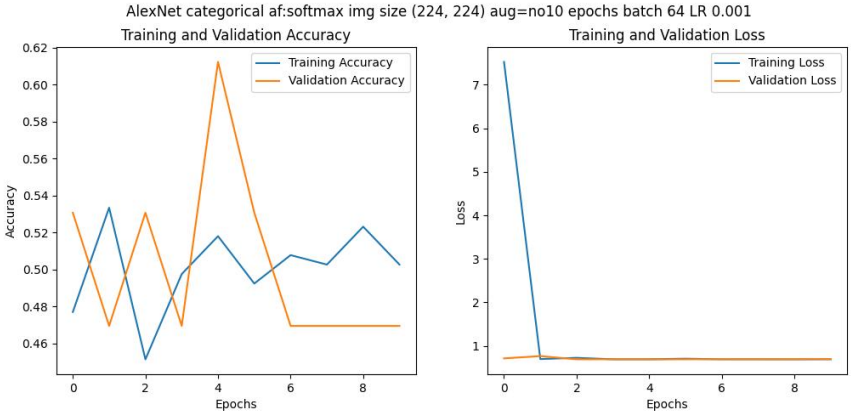

Figura 3 – Acurácia e perda AlexNet para 10 épocas batch tamanho 64 taxa de aprendizado 1e-3 e sem uso de aumento de dados

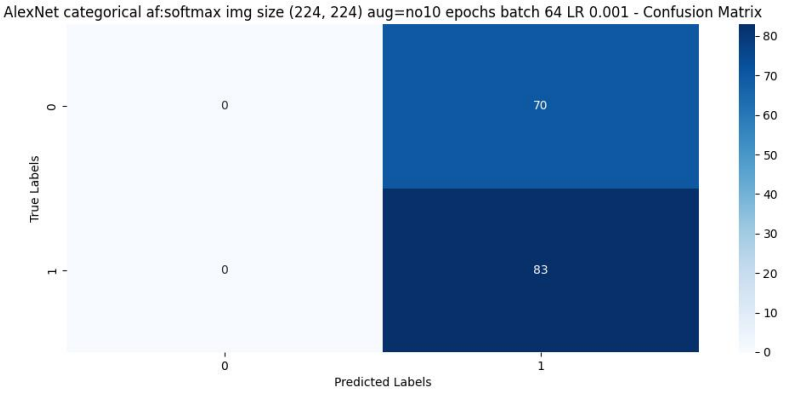

Figura 4 – Matriz de confusão AlexNet para 10 épocas batch tamanho 64 taxa de aprendizado 1e-3 e sem uso de aumento de dados

Na primeira experimentação a acurácia do conjunto de treinamento apresenta picos e então uma estagnação em torno de 47%, isso poderia indicar inicialmente problemas com a taxa de aprendizado, mas também poderia ser uma questão de quantidade de épocas utilizadas. A avaliação de qual parâmetro teria maior impacto no desempenho motivou a primeira mudança de parâmetros para o segundo experimento. A acurácia tanto do subconjunto de validação durante o treinamento quanto do conjunto de teste chega a 54%. Por meio da matriz de confusão é observado que o modelo apenas realiza predições para uma única classe, a de abelhas (representada pelo valor 1), com precisão de 0.5425 , recall 1 e F1 Score 0.70.

Para o segundo experimento a taxa de aprendizado e tamanho de batch foram mantidos e a quantidade de épocas foi alterada para 30. A acurácia do conjunto de treinamento ainda se mostrou instável e com pontos de estagnação, além de não se ajustar bem à acurácia do subconjunto de validação. A acurácia do conjunto de teste se manteve praticamente a mesma, em 54%, com isso é possível observar que a taxa de aprendizado precisava ser melhorada e era o parâmetro de maior impacto nesses resultados. Apesar de não haver ganho na acurácia, a matriz de confusão abaixo mostra evolução na distribuição das predições. A precisão aumentou para 0.618 e o recall caiu para 0.409, o F1 Score obteve o valor de 0.492 indicando que o modelo ficou mais equilibrado. Contudo, é possível notar que agora o modelo realiza mais predições para a classe de formigas.

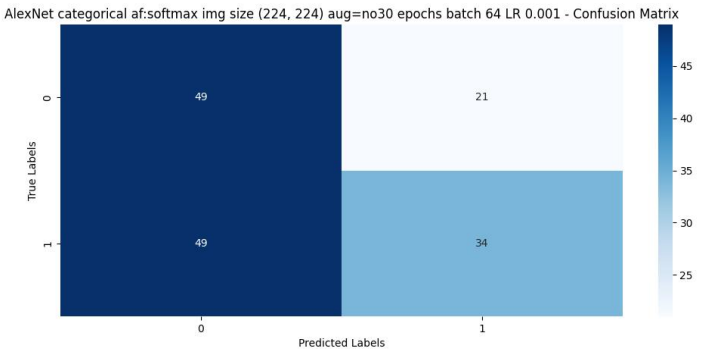

Figura 5 – Matriz de confusão AlexNet para 30 épocas batch tamanho 64 taxa de aprendizado 1e-3 e sem uso de aumento de dados

No terceiro experimento foram testadas diferentes taxas de aprendizado para diferentes quantidades de época. Partindo do aumento da taxa de aprendizado para 0.0001 (1e-4) e mantendo a quantidade de épocas em 30 e tamanho de batch 64, a acurácia do conjunto de treinamento aumentou para 70%. O conjunto de teste teve acurácia de 67%. As demais métricas foram precisão 0.7679, recall 0.5180 e F1 Score 0.6187. O modelo apresentou melhora nas métricas apesar de apresentar leve overfitting. Com 40 épocas e taxa de aprendizado agora sendo 0.00001 (1e-5) a acurácia do treinamento chega a 67% e do conjunto de teste 62%, não há ganho significativo de acurácia porém o modelo passa a se ajustar um pouco melhor ao conjunto de dados de teste. O F1 score nessa configuração é de 0.6184

No quarto experimento um novo tamanho de batch foi testado partindo da melhor configuração do experimento anterior, de 30 épocas, taxa de aprendizado 1e-4, e alterando agora o tamanho do batch de 64 para 128, o modelo teve a acurácia do treinamento chegando a 90% e de quase 73% (0.7255) no conjunto de teste, foi a melhor acurácia obtida até então apesar do overfitting. Com precisão 0.7204, recall 0.6286, e F1 Score 0.7614 também foi a melhor pontuação para essas métricas, no entanto, agora o modelo volta a fazer mais predições para a classe de abelhas.

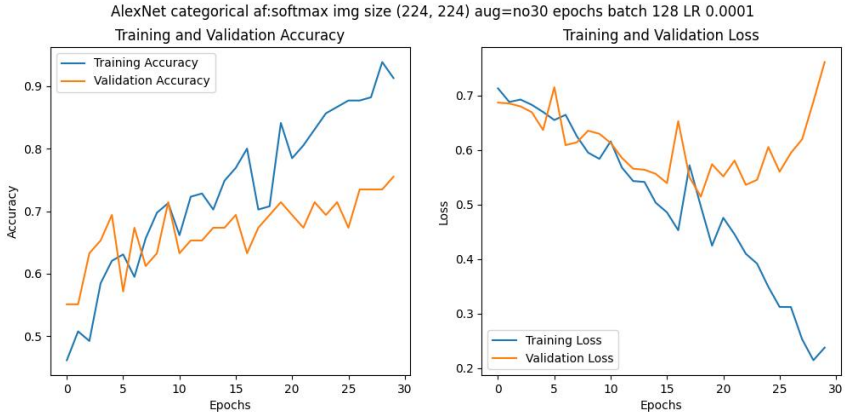

Figura 6 – Acurácia e perda AlexNet para 30 épocas batch tamanho 128 taxa de aprendizado 1e-4 e sem uso de aumento de dados

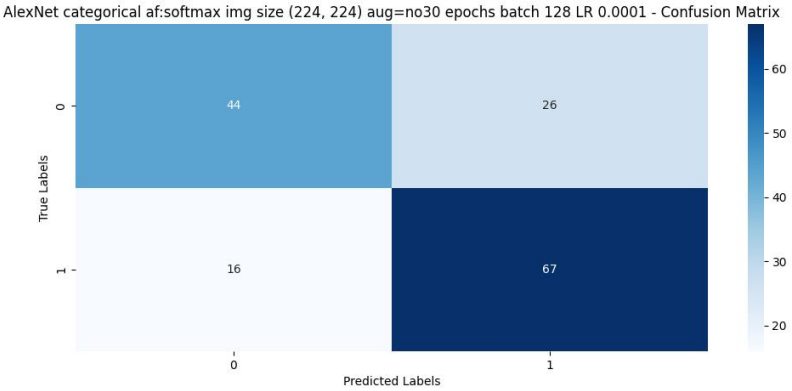

Figura 7 – Matriz de confusão AlexNet para 30 épocas batch tamanho 128 taxa de aprendizado 1e-4 e sem uso de aumento de dados

Mantendo o batch com tamanho 128 e aumentando a quantidade de épocas para 40 a acurácia do conjunto de teste cai para 63% apesar do conjunto de treinamento chegar a 99%.Nesta configuração o modelo passa a se ajustar muito aos dados de treinamento de forma que o overfitting passa a ser prejudicial. No quinto experimento modelo é testado com aumento de dados. Com 30 épocas e batch de 128 a acurácia se manteve em 72%, porém dessa vez o modelo foi capaz de distinguir melhor entre as duas classes como mostra a matriz de confusão:

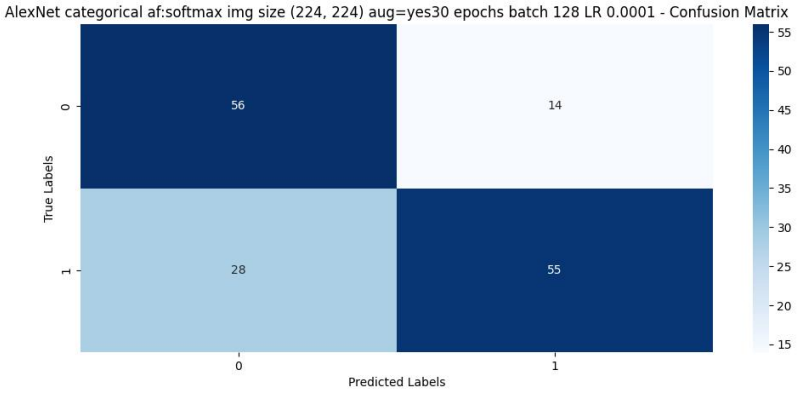

Figura 8 – Matriz de confusão AlexNet para 30 épocas batch tamanho 128 taxa de aprendizado 1e-4 e com uso de aumento de dados

4.2 VGG-16

O primeiro experimento se trata da implementação da arquitetura original da rede alterando a quantidade de nós da saída para dois e também o tamanho da imagem de entrada de 224 para 100. A escolha para esse tamanho de imagem foi devido à ocorrência de estouro de RAM ao usar imagens a partir de 200x200 pixels, uma limitação da plataforma Google Colab.

- **Parâmetros do dataset:** Tamanho da imagem 100x100 normalizadas pelo fator 1/255, modo categórico, sem aumento de dados.
- **Parâmetros do modelo:** Entrada (100,100,3) saída 1 de 2 classes e função de ativação da saída softmax
- **Hiperparâmetros**: Batch tamanho 64, 10 épocas, taxa de aprendizado 0.00001

A primeira opção de taxa de aprendizado escolhida foi 1e-5 por conta de ser uma rede mais profunda. Os valores obtidos foram os seguintes:

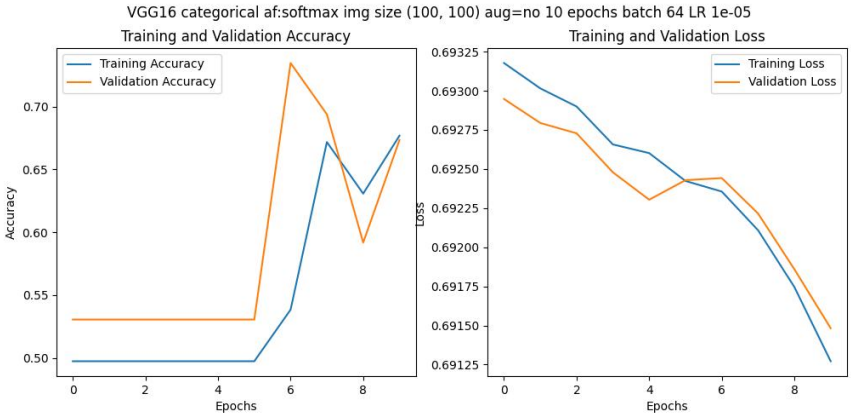

Figura 9 – Acurácia e perda VGG-16 para 10 épocas batch tamanho 64 taxa de aprendizado 1e-5 sem uso de aumento de dados

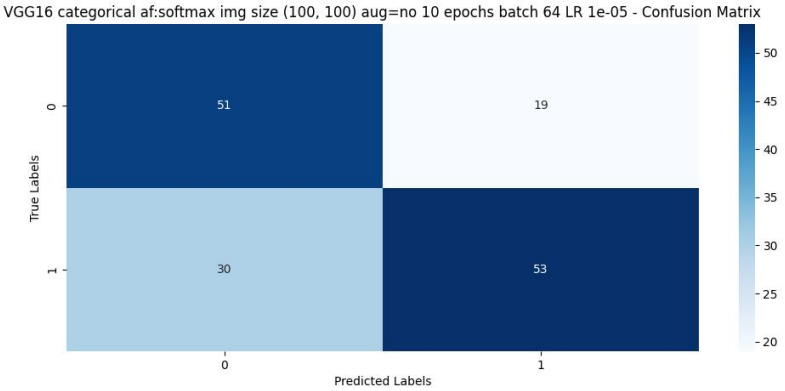

Figura 10 – Matriz de confusão VGG-16 para 10 épocas batch tamanho 64 taxa de aprendizado 1e-5 sem uso de aumento de dados

O modelo já apresentou acurácia significativa de 67% tanto no conjunto de treinamento quanto de teste, as curvas tanto de acurácia quanto de perda mostram como o modelo foi capaz de se ajustar a novas entradas, e indica que essa taxa de aprendizado foi adequada. Pela matriz de confusão se observa que na primeira configuração o modelo também já faz melhor diferenciação entre as classes com precisão de 0.7361, recall 0.6386 e F1 Score 0.6839. No segundo experimento a quantidade de épocas foi aumentada para 30, com isso o ajuste entre dados de treino e teste é ainda maior, apesar da acurácia ficar em 64% , e o modelo passa a fazer mais predições corretas para a classe de formigas

No terceiro experimento foi testado um novo tamanho de batch, 128, para diferentes quantidades de épocas. Visto que a quantidade de execuções por época diminuiu com o aumento do batch, partiu-se da premissa que uma quantidade maior de épocas aumentaria a capacidade de diferenciação. Com 30 épocas o modelo já volta a classificar de forma mais equilibrada as abelhas e formigas e a acurácia volta a 67%. Neste ponto foi observado que de fato o aumento da quantidade de épocas melhorava não somente a acurácia como também a distribuição das predições. Foram testadas 60, 70, 90 e 100 épocas. Com 90 épocas, o modelo atinge 75% de acurácia no conjunto de teste e com uma boa distribuição de predições. O modelo com essa combinação de parâmetros obteve a melhor acurácia

entre todos os experimentos, tanto dessa rede quanto da AlexNet. Com 100 épocas, no entanto, o modelo já começa a apresentar overfitting e a acurácia cai para 73%.

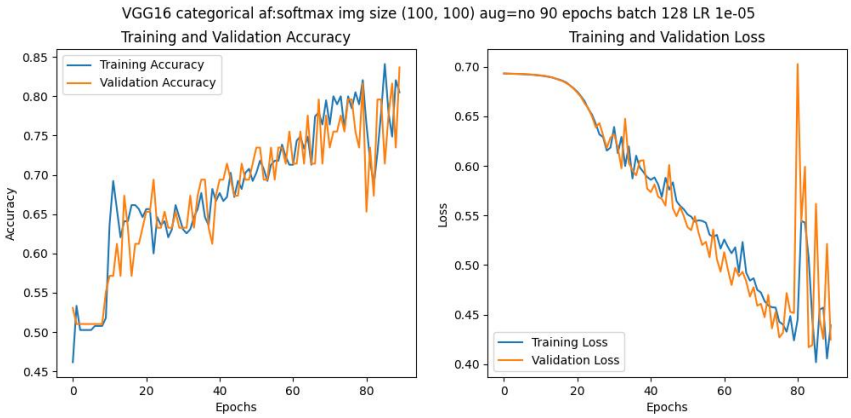

Figura 11 – Acurácia e perda VGG-16 para 90 épocas batch tamanho 128, taxa de aprendizado 1e-5 sem uso de aumento de dados

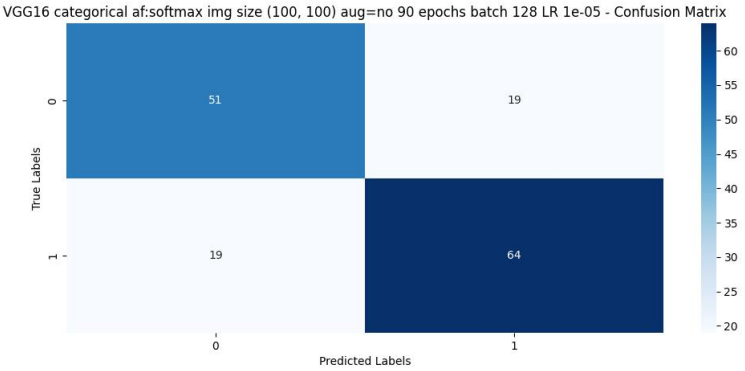

Figura 12 – Matriz de confusão VGG-16 para 90 épocas batch tamanho 128, taxa de aprendizado 1e-5 sem uso de aumento de dados

No quarto experimento houve uso de aumento de dados, usando a melhor configuraão no experimento anterior, não houve melhora na acurácia que foi de 75% as distribuições das predições também se mantiveram praticamente a mesma.

4.3 Análises dos experimentos

O modelo VGG16 performa melhor que a AlexNet mesmo consumindo menos informação da entrada, isso pode ser devido a quantidade maior de camadas que extraem características em diferentes escalas. Por esse motivo também é possível pensar que se não houvesse a limitação do tamanho das imagens a acurácia da VGG-16 seria significativamente maior.

No geral a acurácia do dataset não foi capaz de chegar a 80% e o aumento de dados não impactou essa métrica. Além disso, as predições corretas para a classe de abelhas se sobressaem, isso pode ser devido a natureza dos dados, que são imagens que apresentam muito ruído como flores, folhas, solo e etc, que se apresentam ocupando maior espaço

na imagem desviando o foco do objeto de interesse,além de, também, tornar as entradas semelhantes quanto a sua ambientação no contexto de cores, formatos, e afins, gerando ambiguidade. Além disso a quantidade de imagens de abelhas no conjunto de teste é maior que a de formigas em 13 unidades.

5 Experimentos com o dataset CIFAR-10

Os experimentos foram conduzidos na plataforma Google Colab, utilizando um ambiente de execução equipado com uma GPU Tesla T4, 12 GB de RAM e 78.2 GB de espaço em disco. O foco principal estava nas métricas de precisão, recall e F1 Score, aplicadas ao contexto do dataset CIFAR-10, que consiste em 10 classes de imagens.

5.1 AlexNet

Na experimentação inicial com a AlexNet, adaptada para o dataset CIFAR-10, conseguimos alcançar resultados notáveis, representando os melhores obtidos nesta fase do projeto. As adaptações e os parâmetros foram cuidadosamente selecionados para se adequarem às características específicas do CIFAR-10.

- **Parâmetros do dataset:** As imagens foram redimensionadas para 32x32 pixels, normalizadas pelo fator 1/255.
- **Parâmetros do modelo:** Configurada com entrada (32,32,3) e saída de 10 classes, utilizando a função de ativação softmax.
- **Hiperparâmetros:** Optamos por um batch size de 64, 10 épocas e uma taxa de aprendizado inicial de 0.001.

Os resultados desta etapa inicial revelaram um desempenho diferenciado nas várias classes do CIFAR-10:

- **Recall e F1 Score:** O modelo alcançou um recall de 0.75 e F1 Score de 0.77 para 'airplane', enquanto para 'automobile' o recall foi de 0.88 com um F1 Score de 0.87. Em categorias mais desafiadoras como 'cat' e 'dog', o recall foi de 0.54 e 0.58, respectivamente, com F1 Scores correspondentes de 0.53 e 0.61.
- **Performance Geral:** A acurácia geral foi de 73%, com uma média macro e ponderada de recall e F1 Score de 73%.

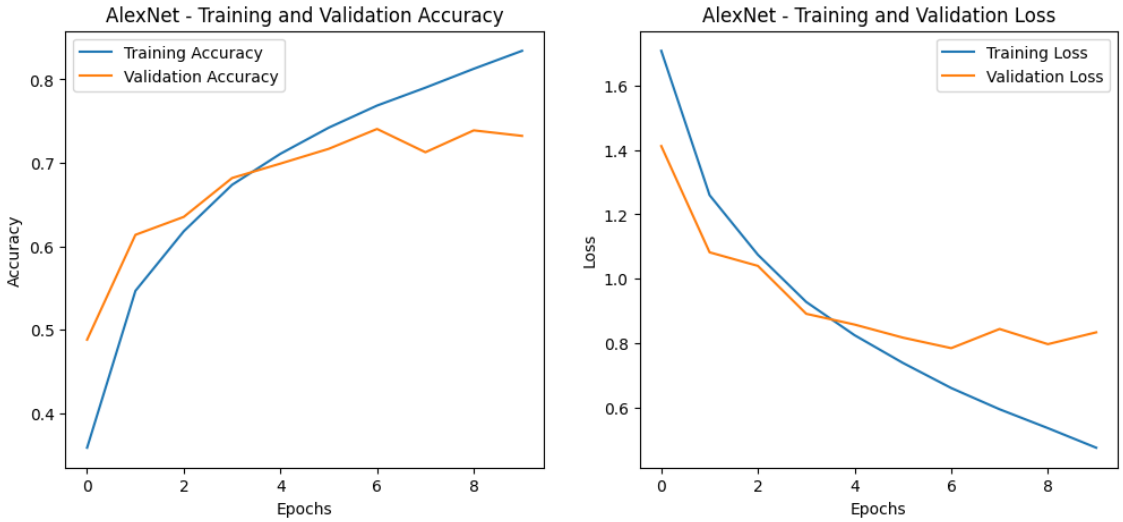

Figura 13 – Gráfico de Acurácia e Perda da AlexNet no CIFAR-10

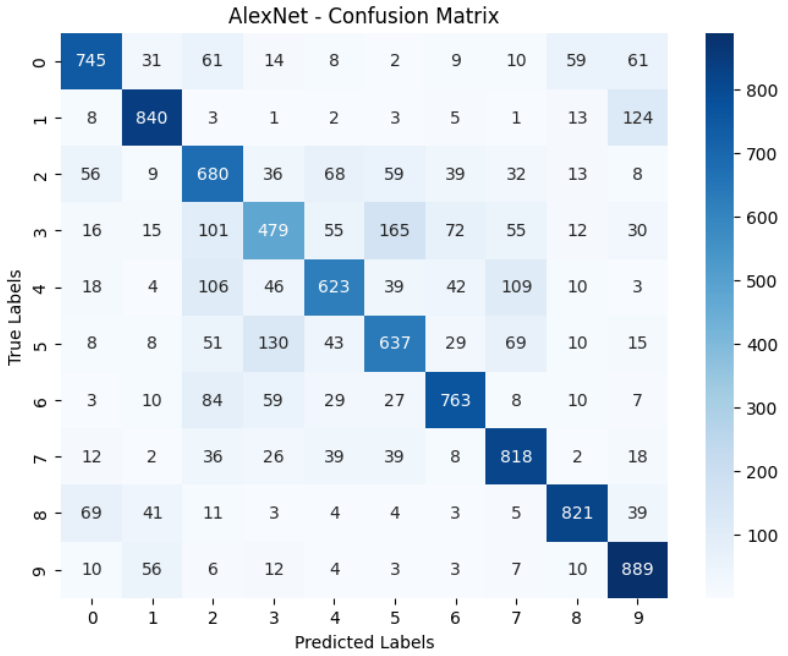

Figura 14 – Matriz de Confusão da AlexNet no CIFAR-10

Esses resultados iniciais foram fundamentais para estabelecer um padrão de desempenho e orientar os ajustes necessários em experimentos futuros, realçando o potencial da AlexNet em contextos de classificação de imagens complexos.

5.2 VGG16

A implementação da VGG16 no CIFAR-10 foi ajustada para maximizar o desempenho do modelo, considerando as restrições do ambiente de execução no Google Colab e as características específicas do dataset.

- **Parâmetros do dataset:** Imagens de 32x32 pixels, normalizadas pelo fator 1/255.
- **Parâmetros do modelo:** Configuração com entrada (32,32,3) e saída de 10 classes, utilizando a função de ativação softmax.
- **Hiperparâmetros iniciais:** Batch size de 64, 10 épocas, taxa de aprendizado inicial de 0.001.

Após a primeira rodada de experimentos com a taxa de aprendizado de 0.001, a VGG16 apresentou baixa acurácia e alta perda. Reduzindo a taxa para 0.0001, observamos melhorias significativas, como refletido nos seguintes valores de Recall e F1 Score:

- **Airplane:** Recall de 0.77 e F1 Score de 0.78.
- **Automobile:** Recall de 0.85 e F1 Score de 0.88.
- **Cat:** Recall de 0.55 e F1 Score de 0.52.
- **Dog:** Recall de 0.60 e F1 Score de 0.62.
- **Frog:** Recall de 0.85 e F1 Score de 0.79.
- **Accuracy geral:** 74%.
- **Média macro e ponderada:** Ambas 74% para Recall e F1 Score.

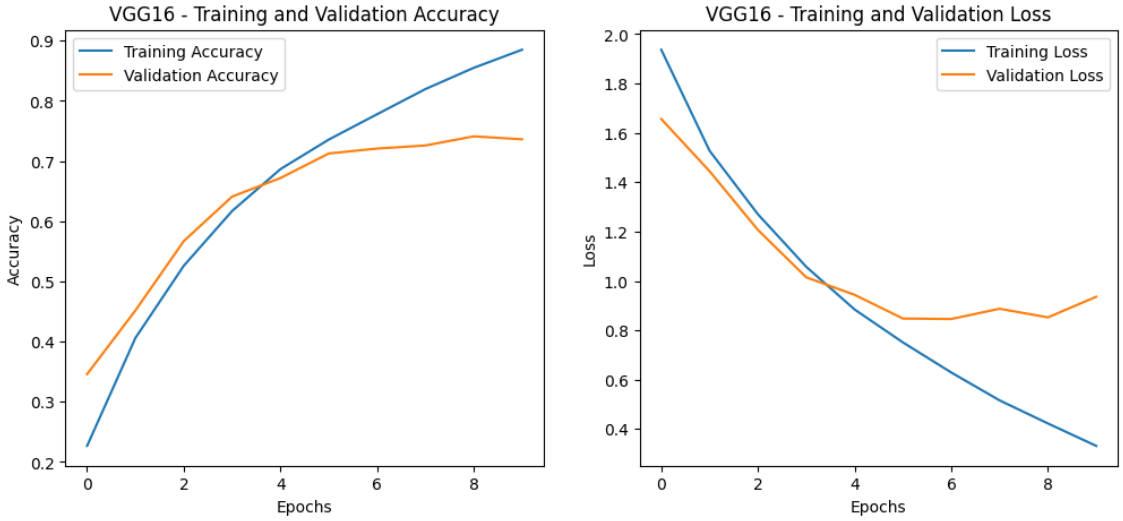

Figura 15 – Gráfico de Acurácia e Perda da VGG16 no CIFAR-10

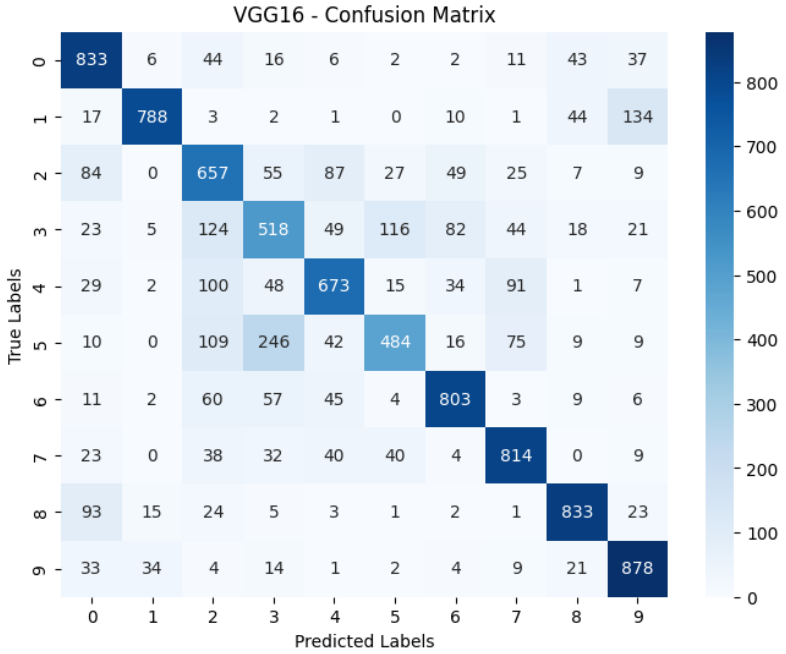

Figura 16 – Matriz de Confusão da VGG16 no CIFAR-10

Estes resultados demonstram a eficácia do ajuste da taxa de aprendizado, evidenciando uma melhora na capacidade do modelo de distinguir entre as diversas classes do CIFAR-10. A precisão e o recall variaram entre as classes, com destaque para categorias como 'Automobile' e 'Frog', onde o modelo apresentou um desempenho particularmente forte.

5.3 Análises dos experimentos

Através dos experimentos realizados com as arquiteturas AlexNet e VGG16 no dataset CIFAR-10, observamos diferenças significativas no desempenho de cada modelo. A VGG16, com sua estrutura mais profunda e complexa, demonstrou uma capacidade superior de extrair e aprender características das imagens, apesar de trabalhar com entradas de menor resolução. Este resultado sugere que a VGG16 pode se beneficiar ainda mais de imagens de maior resolução, caso não houvesse restrições de processamento.

Por outro lado, a AlexNet, apesar de ter uma arquitetura menos complexa, mostrou-se eficaz em várias classes do CIFAR-10. No entanto, os experimentos indicaram que aumentos significativos no volume de dados não resultaram em melhorias notáveis na acurácia da AlexNet, ao contrário da VGG16, que se beneficiou do maior detalhamento proporcionado pelo aumento de dados. Isso pode ser atribuído à maior capacidade da VGG16 de capturar detalhes finos e sutilezas nas imagens, um aspecto que pode ter sido enfatizado com entradas mais detalhadas.

Em termos gerais, ambos os modelos não alcançaram a marca de 80% de acurácia. Essa limitação pode ser parcialmente atribuída à natureza do dataset CIFAR-10, que contém imagens com uma variedade considerável de ruídos e características ambíguas. Esse fator, combinado com a resolução relativamente baixa das imagens, pode ter contribuído para a dificuldade dos modelos em distinguir consistentemente entre as diferentes classes.

6 Conclusões

6.1 Retomada do Trabalho

Neste trabalho, analisamos a performance das redes neurais convolucionais AlexNet e VGG-16 em dois datasets: o multicategoria CIFAR-10 e o binário Hymenoptera. Por meio dos experimentos foi possível analisar e comprovar a relação e impacto entre taxa de aprendizdo, quantidade de épocas e tamanho de batch no treinamento dos modelos.

6.2 Conclusões dos Experimentos

- A VGG-16, apesar de trabalhar com entradas de menor resolução, superou a AlexNet em termos de aprendizado e extração de características, sugerindo que sua arquitetura mais profunda é mais eficaz em cenários complexos de classificação.
- Nos dois datasets, a acurácia não ultrapassou 80%, uma limitação que pode ser atribuída à natureza dos dados, especialmente no CIFAR-10, onde a baixa resolução das imagens e a presença de ruídos representaram desafios adicionais.
- Apesar disso o dataset CIFAR-10 obteve melhores resultados de forma mais rápida que o dataset binário isso tem forte relação com a quantidade de dados para treinamento que é 200 vezes maior que do Hymenoptera.
- Apenas o valor da acurácia não diz sobre a eficiência do modelo, a AlexNet com o dataset Hymenoptera alcançou o mesmo valor de acurácia de 72% para diferentes distribuições de predições, por isso a análise de matriz de confusão se faz importanate. Uma acurácia alta apenas para predições de determinada classe não é algo desejável pois caracteriza o modelo como enviesado.
- Para o dataset Hymenoptera apesar da quantidade de dados de treinamento ser igualmente distribuida os modelos apresentaram viés para a classe de abelhas. O F1 Score alto apesar de desejável também não implica em bom desempenho nas predições, visto que seu valor alto é proporcional a uma precisão alta, porém a precisão está atrelada a quantidade de acertos, que está relacionada a qualidade dos dados de teste, como mencionado no experimento a maior quantidade de dados de teste sendo da classe abelha impactou o resultado.

6.3 Dificuldades Encontradas

- A escolha e ajuste de hiperparâmetros adequados, como taxa de aprendizado e número de épocas, que se mostraram críticos para o desempenho dos modelos.
- Limitações do ambiente de execução, como restrições de processamento no Google Colab, que impactaram o tamanho das imagens e, consequentemente, a eficácia do treinamento.
- A natureza dos datasets, com características diversas e, em alguns casos, ambíguas, que impuseram desafios na classificação precisa das imagens.

6.4 Reflexão Final

Este trabalho reforçou a importância da escolha cuidadosa de arquiteturas de redes neurais e hiperparâmetros em função das características específicas dos datasets.

A prática envolvida nesse trabalho permitiu aos autores estarem em contato com um tipo de atividade muito realizada na área de visão computacional e aprendizado de máquina atualmente. Por meio da observação e análise das nuances acerca desse problema, foi possível além de aprender, contribuir com o mesmo. Dssa forma o trabalho serve como uma base sólida sobre o assunto e beneficia quem deseja engajar com a esfera acadêmica ou com a área de atuação.

Referências

[1] **AlexNet Wikipedia**. Disponível em: https://en.wikipedia.org/wiki/AlexNet

[2] **Transfer Learning — Part — 4.0!! VGG-16 and VGG-19**. Disponível em: https://becominghuman.ai/transfer-learning-part-4-0-vgg-16-and-vgg-19-d7f0045032de

[3] **Implementing AlexNet CNN Architecture Using TensorFlow 2.0+ and Keras**. Disponível em: https://towardsdatascience.com/implementing-alexnet-cnn-architecture-using-tensorflow-2-0-and-keras-2113e090ad98

[4] **How to Develop a CNN From Scratch for CIFAR-10 Photo Classification**. Disponível em: https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/