Classificação de Imagens em Cores Utilizando o Conjunto de Dados CIFAR-10

Steffie Gabriella Jean Gilles[1] and Abellard Christley Mainviel[2]

[1]Instituto Nacional de Telecomunicações, Brasil
[2]Instituto Nacional de Telecomunicações, Brasil

steffie@mtel.inatel.br, abellard.mainviel@mtel.inatel.br

Abstrato -

Neste artigo, propomos o uso do conjunto de dados CIFAR-10 para realizar a classificação de imagens coloridas. Realizamos uma análise abrangente do conjunto de dados CIFAR-10, abordando suas características e desafios. Ao longo deste projeto, aplicamos técnicas de processamento de imagens para realizar o pré-processamento de dados e a preparação para o treinamento de modelos de aprendizado de máquina. Implementamos e treinamos modelos de classificação de imagens usando o algoritmo de aprendizado de maquina, incluindo redes neurais convolucionais (CNNs), utilizando a biblioteca TensorFlow. Avaliamos o desempenho desses modelos usando a métricas da acurácial. Para melhorar o desempenho, fazemos ajustes nos modelos e no processo de treinamento, realizando testes de validação cruzada que garantiza a robustez dos resultados.

Palavras-chave -

CIFAR-10; Redes neurais convolucionais (CNNs); TensorFlow; acurácia;

1 Introdução

Na atualidade, a presença dos computadores transcende as tarefas rotineiras, estendendo-se à tomada de decisões complexas e análises abrangentes. Neste cenário, métodos de inteligência artificial, aprendizado de máquina e processamento de imagens tornam-se essenciais para impulsionar avanços significativos em diversas esferas da sociedade.

A aplicação do processamento e classificação de imagens para a tomada de decisões já permeia o cotidiano, desde a detecção de sorrisos em câmeras até sistemas que empregam reconhecimento facial. Contudo, desafios significativos surgem quando buscamos dotar os computadores da capacidade de interpretar imagens de maneira similar à mente humana, como a identificação da presença de um cachorro em uma imagem. Para atingir essa capacidade, faz-se necessário o desenvolvimento de modelos computacionais e técnicas de otimização do processamento e resultados.

A Classificação de Imagens, ao atribuir rótulos a imagens de entrada pertencentes a um conjunto fixo de categorias, torna-se uma tarefa desafiadora para as máquinas, embora seja algo natural para o ser humano. Diversos obstáculos, como variação de ponto de vista, escala e condições de iluminação, podem dificultar a interpretação algorítmica das imagens. Nesse contexto, as Redes Neurais Convolucionais (CNNs)[1] surgem como uma solução apropriada para enfrentar esses desafios, sendo capazes de capturar padrões complexos.

Este estudo tem como objetivo avaliar o impacto de cada hiperparâmetro nas redes neurais convolucionais, considerando o equilíbrio entre a acurácia dos testes e o tempo de execução do algoritmo. Para alcançar tal objetivo, as redes neurais precisarão classificar corretamente imagens provenientes dos conjuntos de dados Cifar-10.[2] Diversos hiperparâmetros serão testados,para aprimorar os resultados de classificação.

Os modelos desenvolvidos serão avaliados com base na acurácia, uma métrica que generaliza a performance entre todas as classes quando igualmente importantes e representadas. Este trabalho visa contribuir para o entendimento das complexidades inerentes à classificação de imagens coloridas e o refinamento das técnicas utilizadas, proporcionando uma base sólida para futuros avanços nesse campo em constante evolução.

2 Desenvolvimento

Ao iniciar a tarefa de classificar imagens em cores usando o conjunto de dados CIFAR-10, nos deparamos com desafios que exigiam inovação e precisão. Neste projeto, pretendemos abordar e superar esses desafios explorando estratégias eficientes, modelagem avançada e técnicas de pré-processamento para obter um modelo capaz de classificar imagens coloridas que não pertencem ao conjunto de dados CIFAR-10.

2.1 Importação de Bibliotecas

No âmbito do desenvolvimento de soluções em aprendizado de máquina, a importação de bibliotecas desempenha um papel crucial pois são ferramentas específicas

que facilitam o desenvolvimento de modelos de aprendizado de máquina. As bibliotecas em Python são conjuntos de código reutilizável que enriquecem a funcionalidade da linguagem, proporcionando acesso a ferramentas especializadas.[3] Diferentemente de linguagens como C++ ou C, as bibliotecas Python não estão vinculadas a contextos específicos, oferecendo flexibilidade para incorporação em uma variedade de projetos.

Para abordar o desafio específico da classificação de imagens em cores usando o conjunto de dados CIFAR-10, é imperativo selecionar e utilizar as bibliotecas certas. As ferramentas escolhidas não apenas simplificam tarefas complexas, mas também proporcionam um ambiente propício para o desenvolvimento eficiente de modelos de aprendizado automático. Abaixo, apresentam-se as principais bibliotecas que desempenharão um papel vital durante o desenvolvimento deste projeto:

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
!pip install keras-tuner --upgrade
import keras_tuner as kt
```

```
import matplotlib.pyplot as plt
import numpy as np
import os
import tensorflow as tf
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms, models
```

Essas bibliotecas fornecem funcionalidades especializadas essenciais para o processamento, análise e treinamento de modelos, constituindo a base sólida para a implementação bem-sucedida de soluções em aprendizado de máquina. Ao incorporar essas ferramentas, espera-se simplificar o desenvolvimento e maximizar a eficácia na tarefa de classificação de imagens em cores com o conjunto de dados CIFAR-10.

2.1.1 Tensorflow

Desenvolvida pelo Google Brain, TensorFlow emergiu como uma das bibliotecas de aprendizado de máquina mais amplamente utilizadas, capacitando pesquisadores e desenvolvedores a criar e treinar modelos complexos de forma eficiente. TensorFlow é uma biblioteca de código aberto que oferece suporte à criação e treinamento de modelos de aprendizado de máquina e deep learning.[4] Sua flexibilidade e escalabilidade tornam-na uma escolha popular para uma variedade de aplicações, desde reconhecimento de imagem até processamento de linguagem natural. Entre suas principais características, podemos citar:

1. **Grafos Computacionais:** TensorFlow representa modelos como grafos computacionais, permitindo uma visualização clara e eficiente das operações matemáticas subjacentes.

2. **Flexibilidade em Plataformas:** Com suporte para CPUs, GPUs e TPUs (Tensor Processing Units), TensorFlow é adaptável a diferentes ambientes de computação, garantindo desempenho otimizado.[5]

3. **Abstração de Alto Nível:** A interface de alto nível, Keras, integrada ao TensorFlow, simplifica o processo de construção e treinamento de modelos, tornando-o acessível a uma ampla gama de usuários.

2.1.2 KerasTurner

Dentro do ecossistema do TensorFlow, encontramos uma aliada valiosa: a biblioteca Keras. Desenvolvida para ser uma interface de alto nível, Keras facilita significativamente a criação, treinamento e avaliação de modelos de aprendizado de máquina e deep learning. Ao incorporar Keras em nosso projeto de classificação de imagens em cores usando o conjunto de dados CIFAR-10, buscamos simplificar ainda mais o processo de desenvolvimento.

Keras opera como uma camada de abstração sobre o TensorFlow, proporcionando uma experiência de programação mais amigável e acessível. Sua sintaxe intuitiva e estrutura modular permitem que os desenvolvedores construam e experimentem com modelos complexos sem a necessidade de lidar diretamente com as complexidades do TensorFlow de baixo nível.[6]

Ao integrar Keras no nosso projeto, aproveitamos o melhor de ambas bibliotecas: a potência do TensorFlow como backend e a simplicidade e praticidade oferecidas por Keras. A flexibilidade para criar arquiteturas de modelos personalizadas ou aproveitar modelos pré-treinados com poucas linhas de código é uma vantagem distintiva que acelera o desenvolvimento.

2.1.3 Torch e Torchvision

PyTorch e Torchvision são duas ferramentas, desenvolvidas pelo Facebook, que fornecem um ecossistema robusto para a investigação e implementação eficiente de modelos de aprendizagem profunda.

PyTorch, el marco de trabajo subyacente, se distingue por su enfoque dinámico y su filosofía ”Pythonic”. Su estructura intuitiva y flexible facilita la construcción y la experimentación con modelos complejos. Además, su capacidad para realizar cómputos dinámicos en tiempo de ejecución proporciona una mayor flexibilidad durante el desarrollo.[7]

O Torchvision, por outro lado, é uma extensão do PyTorch especializada em fornecer ferramentas e conjuntos

de dados específicos para tarefas de visão computacional. Facilita o carregamento e a transformação de conjuntos de dados, bem como a utilização de modelos pré-treinados. A sua integração perfeita com o PyTorch torna-o uma escolha natural para projectos que envolvam imagens.[8]

2.1.4 Matplotlib

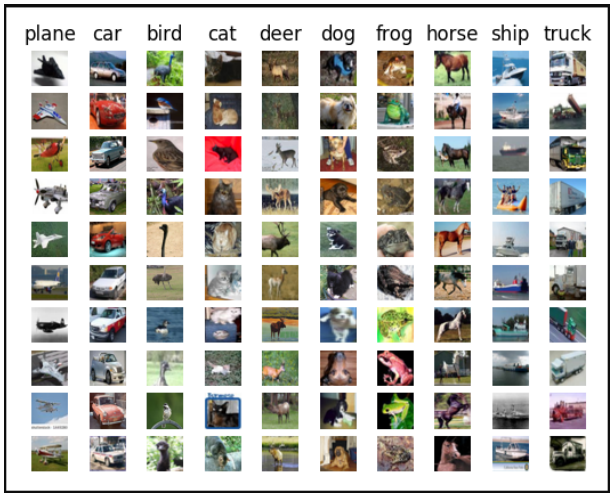
A Matplotlib fornece várias funções para criar visualizações claras e significativas. Esta biblioteca facilita a exploração das características do conjunto de dados, a identificação de padrões e a tomada de decisões informadas sobre estratégias de modelação.

Ao incorporar a Matplotlib no nosso fluxo de trabalho, podemos efetuar análises exploratórias de dados de forma eficiente. A visualização da distribuição das classes, a exploração das relações entre variáveis e a compreensão da complexidade do conjunto de dados tornam-se mais acessíveis, permitindo-nos tomar decisões informadas durante a fase de conceção do modelo.

A capacidade do Matplotlib para se integrar sem esforço noutras bibliotecas, como o NumPy e o Pandas, aumenta a sua utilidade. Ao combinar visualizações com análise estatística e manipulação de dados, criamos um ambiente holístico para uma compreensão profunda do nosso conjunto de dados.[9]

2.2 The Dataset

O CIFAR-10, sigla para "Canadian Institute For Advanced Research", é um conjunto de dados amplamente reconhecido no campo de visão computacional e aprendizado de máquina. Criado por pesquisadores do Instituto Canadense de Pesquisa Avançada, o conjunto contém 60.000 imagens coloridas distribuídas em 10 classes distintas. Cada classe representa um tipo específico de objeto ou cenário, fornecendo uma diversidade representativa para testes robustos de algoritmos de classificação.



As classes incluídas no CIFAR-10 são:

1. Avião
2. Carro
3. Pássaro
4. Gato
5. Cervo
6. Cachorro
7. Sapo
8. Cavalo
9. Navio
10. Caminhão

Cada imagem do CIFAR-10 tem dimensões de 32 por 32 pixels, com três canais de cor (vermelho, verde e azul). Essa resolução relativamente baixa, em conjunto com a presença de variados objetos e cenários, desafia os algoritmos a aprenderem representações significativas a partir de características aparentemente simples.

2.3 Redes neurais convolucionais (CNNs)

CNN (Convolutional Neural Network), ou Rede Neural Convolucional em português, é um tipo de arquitetura de rede neural profunda projetada para processar e analisar dados visuais. Ela é amplamente utilizada em tarefas de visão computacional, como reconhecimento de imagens, detecção de objetos, classificação de imagens, entre outras. As CNNs são especialmente eficazes na extração de padrões visuais hierárquicos e complexos a partir de dados de entrada. Elas consistem em camadas convolucionais, de pooling (agrupamento) e totalmente conectadas.[10] Aqui estão alguns conceitos fundamentais associados a uma CNN:

1. Camadas Convolucionais (Convolutional Layers): Essas camadas aplicam filtros convolucionais para extrair características locais da imagem. Os filtros são pequenas janelas que percorrem a imagem para detectar padrões específicos, como bordas, texturas e formas.

2. Camadas de Pooling (Pooling Layers): As camadas de pooling reduzem a dimensionalidade espacial da representação da imagem, preservando as características mais importantes. O pooling geralmente envolve a seleção do valor máximo (max pooling) em uma região específica.

3. Camadas Totalmente Conectadas (Fully Connected Layers): Essas camadas, no final da arquitetura, combinam as características extraídas para realizar a tarefa final, como classificação.[11] As CNNs são treinadas utilizando técnicas como retropropagação (backpropagation) para ajustar os pesos da rede de acordo com os erros cometidos durante a tarefa. Elas se destacam em problemas de visão computacional devido à sua capacidade de aprender representações hierárquicas e invariantes a translações, rotações e outras transformações.

2.4 Normalização de Dados no Conjunto de Dados CIFAR-10

Ao trabalhar com conjuntos de dados para aprendizado de máquina, no pré-processamento é importante a normalização dos dados. No caso específico do conjunto de dados CIFAR-10, a normalização desempenha um papel essencial para garantir que os modelos de aprendizado automático possam aprender de maneira eficaz a partir dos dados fornecidos.

O conjunto de dados CIFAR-10 apresenta 60.000 imagens coloridas distribuídas em 10 classes, com 6.000 imagens em cada classe. Esta coleção é subdividida em 50.000 imagens de treinamento e 10.000 imagens de teste. As classes são mutuamente exclusivas, não havendo sobreposição entre elas.

A normalização emerge como um procedimento fundamental para adequar os valores das características a uma escala comum, geralmente entre 0 e 1. Esta prática é especialmente crucial no contexto do CIFAR-10, onde os valores de pixel nas imagens variam de 0 a 255 para cada canal de cor (vermelho, verde e azul). Normalizar esses valores para uma escala mais restrita torna-se essencial para possibilitar que os algoritmos aprendam de maneira eficiente, evitando que características com magnitudes maiores predominem na contribuição durante o treinamento. Em termos simples, ao dividir os valores dos pixels por 255, ajustamos esses valores para o intervalo desejado, preservando a relação relativa entre os diferentes canais de cor.

Ao realizar a normalização nos conjuntos de dados do CIFAR-10, desfrutamos de diversas vantagens:

Estabilidade no Treino:: Evitamos problemas relacionados a gradientes muito grandes durante o treino, promovendo assim a estabilidade do modelo.

Convergência Mais Rápida: A normalização auxilia o modelo a convergir de forma mais ágil durante o treino, reduzindo o tempo necessário para alcançar resultados significativos.

Generalização Aprimorada: Facilita a capacidade do modelo de generalizar para novos dados, aprimorando o desempenho nos conjuntos de teste.

Ao incorporar a normalização dos dados como parte do pré-processamento no conjunto de dados CIFAR-10, estabelecemos um ambiente propício para o treinamento eficaz de modelos de classificação das imagens.

2.5 Otimização Hiperparamétrica

Dado que as imagens CIFAR-10 têm uma dimensão de 32x32, utilizamos a otimização de hiperparâmetros para ajustar os parâmetros do modelo de modo a melhorar o seu desempenho. Sendo a otimização hiperparamétrica um aspeto crucial no desenvolvimento de modelos de aprendizagem automática, existem várias abordagens à otimização hiperparamétrica, como a otimização bayesiana. A escolha do método depende da dimensão do espaço de pesquisa e dos recursos computacionais disponíveis.

No contexto do nosso projeto de classificação de imagens coloridas no CIFAR-10, optamos pela otimização bayesiana, uma técnica empregada na otimização de hiperparâmetros, especialmente na escolha dos melhores parâmetros para modelos de aprendizado de máquina. No âmbito do Keras Tuner, ela é uma estratégia de busca disponível para encontrar os hiperparâmetros mais eficazes.

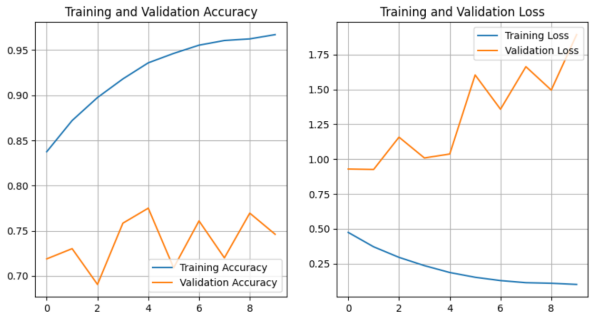
Essa abordagem opera de maneira eficiente no espaço de hiperparâmetros por meio de um processo probabilístico. Em vez de testar todas as combinações possíveis de hiperparâmetros, o método mantém uma distribuição de probabilidade das funções objetivas, como a precisão do modelo. Essa distribuição é então utilizada para determinar quais configurações de hiperparâmetros devem ser avaliadas a seguir.

A otimização bayesiana busca equilibrar a exploração (experimentação de diferentes configurações de hiperparâmetros) e a exploração (foco nas configurações com maior probabilidade de resultar em melhor desempenho). À medida que mais avaliações de modelos são realizadas, a distribuição de probabilidade é atualizada, refinando progressivamente a busca em direção às configurações de hiperparâmetros mais promissoras.

2.6 Avaliando o Modelo: Métricas e Análise de Desempenho

Após a otimização bayesiana, é importante avaliar o desempenho do modelo para garantir a sua eficiência e capacidade de generalização, uma vez que o conjunto de dados CIFAR-10 contém imagens de baixa resolução (32x32 pixels), o que pode limitar a qualidade do modelo durante o treinamento. Voltamos a compilar o nosso melhor modelo, ajustando-o para a classificação de imagens. O

treino é efectuado utilizando os conjuntos de treino e de teste, monitorizando a precisão e a perda ao longo de 10 épocas. Para compreender o comportamento do modelo, geramos gráficos de precisão e perda ao longo das épocas. Isto inclui curvas de treino e validação, fornecendo uma visão clara do desempenho do modelo.



2.7 Transfer learning

Embora tenhamos alcançado resultados notáveis utilizando a otimização bayesiana, tornou-se evidente que os resultados ainda não eram totalmente satisfatórios. A análise das curvas de treinamento revelou que o modelo não estava convergindo da maneira desejada ao lidar com imagens relativamente pequenas do conjunto de dados CIFAR-10, cada uma com apenas 32x32 pixels. Essa falta de convergência pode ser atribuída à sensibilidade do modelo a tais imagens.

Apesar dos esforços de otimização, obtivemos um modelo com desempenho considerado insatisfatório, com uma acurácia inferior a 75%. Diante dessa limitação, optamos por explorar uma abordagem alternativa, recorrendo à técnica de aprendizagem por transferência, mais especificamente utilizando o robusto modelo ResNet.

Ao adotar a técnica de transfer learning, alcançamos um modelo significativamente aprimorado, com uma acurácia superior a 90%. Esse resultado destaca claramente a eficácia e importância do transfer learning na melhoria do desempenho do modelo. Essa abordagem revelou-se crucial para lidar com as limitações anteriores e representa uma estratégia promissora para futuros desenvolvimentos.

A aprendizagem por transferência consiste em utilizar características aprendidas num problema e utilizá-las num problema novo e semelhante. Normalmente, é efectuada em tarefas em que o conjunto de dados é demasiado esparso para treinar um modelo completo a partir do zero. A encarnação mais comum da aprendizagem por transferência no contexto da aprendizagem profunda segue o seguinte fluxo de trabalho:

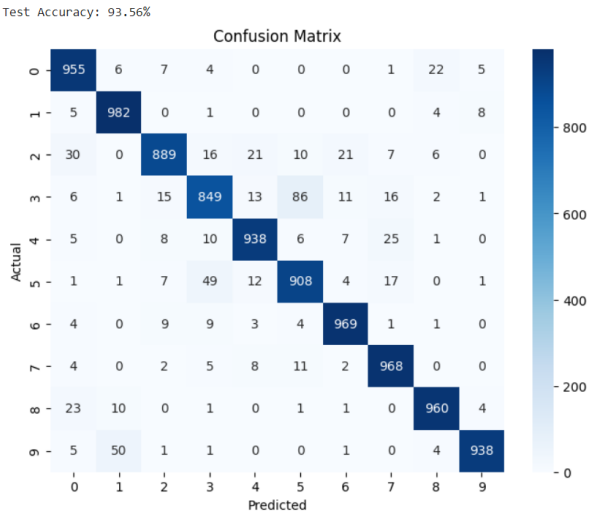
- Retirar camadas de um modelo previamente treinado.
- Congela-as para evitar a destruição de qualquer informação que contenham durante futuras rondas de formação.
- Adicionar novas camadas treináveis em cima das camadas congeladas. Estas aprenderão a transformar características antigas em previsões num novo conjunto de dados.
- Treine as novas camadas no seu conjunto de dados.

```
Files already downloaded and verified
Files already downloaded and verified
Epoch 1/5, Loss: 0.8980863980581878
Epoch 2/5, Loss: 0.6204992148140952
Epoch 3/5, Loss: 0.5599021633523287
Epoch 4/5, Loss: 0.5253469613467129
Epoch 5/5, Loss: 0.4895888911375463
Test Accuracy: 94.23%
```

Os resultados obtidos após a aplicação da aprendizagem por transferência são notáveis. O teste de precisão (test accuracy) atingiu um impressionante valor de 94,23%. Esse desempenho substancialmente superior destaca a eficácia da transferência de conhecimento de modelos pré-treinados para cenários específicos, como no nosso caso para o conjunto de dados CIFAR-10.

2.7.1 Matriz de Confusão

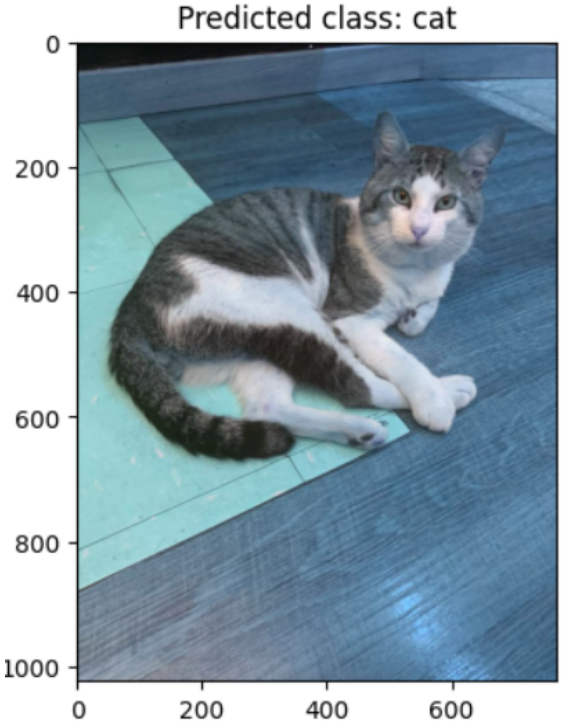
A matriz de confusão abaixo fornece uma representação clara de como o modelo classifica cada classe em comparação com os rótulos reais. Cada célula indica o número de casos em que uma classe foi prevista correcta ou incorretamente.

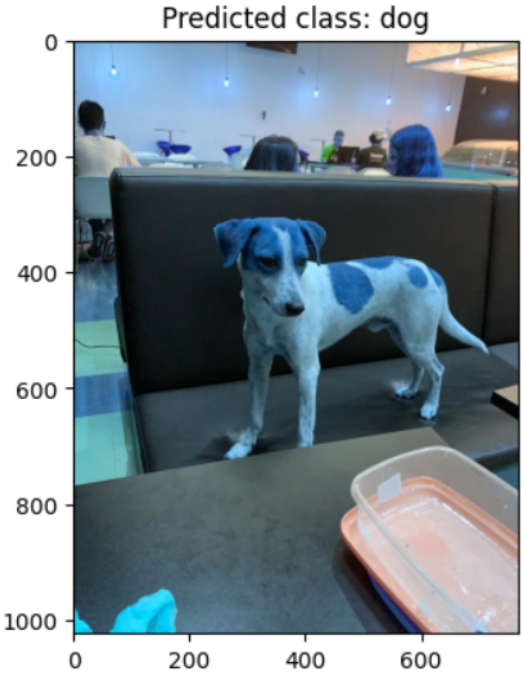


2.8 Testando o modelo

Una vez alcanzado un nivel de precisión bastante satisfactorio, pasamos a la fase de predicción, en la que elegimos una imagen de cada clase para comprobar la eficacia de nuestro modelo. Este test se realizó paso a

paso: Primero, se define una transformación que se aplicará a cada imagen de entrada. Luego, se cargan varias imágenes desde la carpeta /content utilizando la biblioteca PIL (Python Imaging Library). Se agrega una dimensión de lote a cada imagen para que coincida con las expectativas del modelo. Se envían las imágenes transformadas al mismo dispositivo donde se encuentra el modelo. El modelo se coloca en modo de evaluación (model.eval()) para desactivar la regularización por lotes y otras capas específicas de entrenamiento. Se realiza la predicción para cada imagen utilizando model(img) y se obtiene la clase predicha utilizando torch.max(outputs, 1). Se imprime la clase predicha para cada imagen utilizando las etiquetas de CIFAR-10. Se muestran las imágenes originales junto con el título que indica la clase predicha por el modelo. En las siguientes figuras se pueden visualizar las muestras con la clase predicha. Além disso Se puede verificar visualmente cómo el modelo está clasificando diferentes tipos de imágenes.





3 Conclusão

Neste artigo, abordamos a tarefa de classificação de imagens coloridas usando o conjunto de dados CIFAR-10. Utilizamos técnicas avançadas de processamento de imagens e modelos de aprendizado de máquina, com foco nas redes neurais convolucionais (CNNs) implementadas com a biblioteca TensorFlow.

Realizamos uma análise do conjunto de dados CIFAR-10, destacando suas características. Durante o projeto, aplicamos técnicas de pré-processamento, incluindo a normalização dos dados, para melhorar a eficiência do treinamento. A otimização hiperparamétrica, utilizando a otimização bayesiana, foi empregada para ajustar os parâmetros do modelo.

A avaliação do desempenho do modelo foi realizada utilizando a métrica acurácia, . Após identificar desafios na convergência do modelo, exploramos a técnica de aprendizado por transferência, incorporando o modelo pre-treinado ResNet18. Os resultados obtidos foram notáveis, alcançando uma acurácia no conjunto de teste de 94,23

Finalmente, apresentamos testes práticos do modelo, demonstrando sua eficácia na classificação de diversas classes, como gatos e cachorros, com base em imagens de entrada não pertencentes ao conjunto de treinamento CIFAR-10. Este estudo contribui para o entendimento das complexidades envolvidas na classificação de imagens coloridas e destaca a importância de abordagens avançadas, como a aprendizagem por transferência, para alcançar resultados significativos. O modelo desenvolvido apre-

sentou um desempenho promissor, abrindo portas para aplicações práticas em diversas áreas que requerem processamento eficiente de imagens.

Referências

[1] Prachi Juyal and Amit Kundaliya. Multilabel image classification using the cnn and dc-cnn model on pascal voc 2012 dataset. In *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pages 452–459, 2023. doi:10.1109/ICSCSS57650.2023.10169541.

[2] Raveen Doon, Tarun Kumar Rawat, and Shweta Gautam. Cifar-10 classification using deep convolutional neural network. In *2018 IEEE Punecon*, pages 1–5, 2018. doi:10.1109/PUNECON.2018.8745428.

[3] Luiz Eduardo Borges. *Python para desenvolvedores: aborda Python 3.3*. Novatec Editora, 2014.

[4] Chris Mattmann. *Machine Learning with TensorFlow, Second Edition*. 2020.

[5] Alexia Audevart, Konrad Banachewicz, and Luca Massaron. *Machine Learning Using TensorFlow Cookbook: Create powerful machine learning algorithms with TensorFlow*. 2021.

[6] Oliver Duerr and Beate Sick. 2020.

[7] Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. Pytorch-ood: A library for out-of-distribution detection based on pytorch. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4350–4359, 2022. doi:10.1109/CVPRW56347.2022.00481.

[8] Feras Albardi, H M Dipu Kabir, Md Mahbub Islam Bhuiyan, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi. A comprehensive study on torchvision pre-trained models for fine-grained inter-species classification. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2767–2774, 2021. doi:10.1109/SMC52423.2021.9659161.

[9] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science and Engineering*, 9(3): 90–95, 2007. doi:10.1109/MCSE.2007.55.

[10] Jia Shijie, Wang Ping, Jia Peiyi, and Hu Siping. Research on data augmentation for image classification based on convolution neural networks. In *2017 Chinese Automation Congress (CAC)*, pages 4165–4170, 2017. doi:10.1109/CAC.2017.8243510.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.