UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PEDRO RAIMUNDO DOS SANTOS NETO

ANÁLISE DO DESEMPENHO DE PERCEPTRONS MULTICAMADA MODERNOS NA CLASSIFICAÇÃO DE OBJETOS DO COTIDIANO

CAMPINA GRANDE - PB

2024

PEDRO RAIMUNDO DOS SANTOS NETO

ANÁLISE DO DESEMPENHO DE PERCEPTRONS MULTICAMADA MODERNOS NA CLASSIFICAÇÃO DE OBJETOS DO COTIDIANO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Herman Martins Gomes

CAMPINA GRANDE - PB

2024

PEDRO RAIMUNDO DOS SANTOS NETO

ANÁLISE DO DESEMPENHO DE PERCEPTRONS MULTICAMADA MODERNOS NA CLASSIFICAÇÃO DE OBJETOS DO COTIDIANO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Herman Martins Gomes

Orientador – UASC/CEEI/UFCG

Professor Dr. Eanes Torres Pereira

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 15 de Maio de 2024.

CAMPINA GRANDE - PB

RESUMO

Dispositivos eletrônicos que auxiliam em tarefas domésticas vem ganhando popularidade nos últimos anos e ajudam as pessoas a ganhar tempo nas suas rotinas. Indivíduos com limitações físicas necessitam ainda mais de suporte nas suas atividades cotidianas. Contudo, não existem muitas soluções na atualidade que consigam desempenhar tarefas normalmente executadas por humanos, devido a limitações de hardware, por exemplo. Uma habilidade importante é o reconhecimento de objetos em uma cena visual. Dessa forma, esta pesquisa tem por objetivo avaliar o desempenho de alguns modelos baseados em perceptron multicamada (MLP) em classificar imagens de objetos comumente encontrados em ambientes domésticos, para verificar a eficácia dessas soluções nesse contexto de aplicação. Foram conduzidos experimentos de classificação com os modelos, observando as métricas obtidas, como acurácia, tempos de treinamento e teste, para qualificar o desempenho. A análise dos modelos confirmou a capacidade de classificar os objetos com uma boa taxa de acerto. Os resultados obtidos indicam que é possível aplicar MLPs em soluções de auxílio a atividades domésticas, reduzindo o custo computacional de implementação em relação a modelos mais complexos.

PERFORMANCE ANALYSIS OF MODERN MULTILAYER PERCEPTRONS IN THE CLASSIFICATION OF EVERYDAY OBJECTS

ABSTRACT

Electronic devices that assist with household chores have gained popularity in recent years and help people save time in their routines. Individuals with physical limitations need even more support in their daily activities. However, there are not many solutions currently available that can perform tasks normally performed by humans, due to hardware limitations, for example. An important skill is the recognition of objects in a visual scene. Thus, this research aims to evaluate the performance of some models based on multilayer perceptron (MLP) in classifying images of objects commonly found in domestic environments, to verify the effectiveness of these solutions in this application context. Classification experiments were conducted with the models, observing the metrics obtained, such as accuracy, training and testing times, to qualify the performance. The analysis of the models confirmed the ability to classify the objects with a good accuracy rate. The results obtained indicate that it is possible to apply MLPs in solutions of household activities assistance, reducing the computational cost of implementation in relation to more complex models.

Análise do Desempenho de Perceptrons Multicamada Modernos na Classificação de Objetos do Cotidiano

Pedro Raimundo dos Santos Neto
Unidade Acadêmica de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil
pedro.santos.neto@ccc.ufcg.edu.br

Herman Martins Gomes
Unidade Acadêmica de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil
hmg@computacao.ufcg.edu.br

RESUMO

Dispositivos eletrônicos que auxiliam em tarefas domésticas vem ganhando popularidade nos últimos anos e ajudam as pessoas a ganhar tempo nas suas rotinas. Indivíduos com limitações físicas necessitam ainda mais de suporte nas suas atividades cotidianas. Contudo, não existem muitas soluções na atualidade que consigam desempenhar tarefas normalmente executadas por humanos, devido a limitações de hardware, por exemplo. Uma habilidade importante é o reconhecimento de objetos em uma cena visual. Dessa forma, esta pesquisa tem por objetivo avaliar o desempenho de alguns modelos baseados em perceptron multicamada (MLP) em classificar imagens de objetos comumente encontrados em ambientes domésticos, para verificar a eficácia dessas soluções nesse contexto de aplicação. Foram conduzidos experimentos de classificação com os modelos, observando as métricas obtidas, como acurácia, tempos de treinamento e teste, para qualificar o desempenho. A análise dos modelos confirmou a capacidade de classificar os objetos com uma boa taxa de acerto. Os resultados obtidos indicam que é possível aplicar MLPs em soluções de auxílio a atividades domésticas, reduzindo o custo computacional de implementação em relação a modelos mais complexos.

Palavras-chave

Aprendizagem de Máquina, Redes Neurais, *Multilayer Perceptron*, Classificação de objetos em imagens.

1. INTRODUÇÃO

Soluções computacionais que utilizam Aprendizagem de Máquina são capazes de resolver problemas de alta complexidade como a classificação de um conjunto de dados, sendo um exemplo o reconhecimento de objetos em uma cena visual. Usualmente são utilizadas redes neurais convolucionais (RNCs) nesse contexto, dada a sua capacidade em resolver com eficiência problemas que envolvem dados de entrada na forma de imagens [1]. Contudo, elas possuem um alto custo computacional associado, pois demandam um tempo de treinamento grande e exigem muitos recursos de processamento nessa etapa. Perceptrons multicamadas (Multilayer Perceptrons, MLPs) se apresentam como uma alternativa para esse tipo de aplicação, oferecendo uma menor demanda de recursos computacionais.

Entretanto, MLPs clássicos não são tão capazes quanto RNCs com esse propósito, possuem, de fato, uma limitação. Nesse contexto, alguns modelos baseados em MLPs que não aplicam técnicas como a auto-atenção ampliam as potencialidades da implementação clássica, utilizando ideias como o *patch embedding*, que é a combinação de valores próximos localmente em uma matriz que representa a imagem, servindo como um agregador de informação espacial [1]. Neste trabalho iremos nos concentrar em estudar os modelos MLP-Mixer [1] e o gMLP [2], que são baseados em MLP, porém aplicam técnicas complementares que modificam suas capacidades devido a forma distinta que processam as entradas.

Dessa forma, testar a capacidade desses modelos no contexto das aplicações em ambientes internos de habitação mostra-se como um problema relevante, tanto pela sua utilidade, que é permitir a dispositivos eletrônicos reconhecerem o seu redor, como pelo seu potencial de reduzir custos associados à construção de soluções desse tipo. Colocar os modelos à prova frente a um banco de imagens contendo objetos de interesse é uma maneira interessante de observar o comportamento deles nessa situação e dependendo dos resultados obtidos, questionar a aplicabilidade real dessas ferramentas.

Alguns dos exemplos de aplicação seriam soluções de apoio em atividades domésticas e dispositivos de suporte a pessoas com limitações físicas. A importância dessas aplicações é discutida por Mol [4], em que ela destaca o objetivo da chamada tecnologia assistiva, que é melhorar a qualidade de vida de pessoas que precisam de apoio para realizar suas atividades cotidianas e ao cumprir isso, dá autonomia a elas. Assim, modelos com melhor capacidade de resolução em contextos que os recursos disponíveis são limitados se apresentam como uma opção viável, ampliando a gama de possibilidades de aplicações para soluções que utilizam aprendizado de máquina.

2. FUNDAMENTAÇÃO TEÓRICA

Peña et al. [5] investigaram a performance de quatro diferentes métodos de aprendizagem de máquina na classificação de imagens de satélite de diferentes tipos de plantio para agricultura e observaram bons resultados para a estratégia que utilizou MLP, essa equiparada com a alternativa de máquina de vetores de suporte (SVM), ligeiramente melhor que regressão logística e ainda melhor que árvore de decisão C4.5. É importante salientar que nesse trabalho foram utilizadas versões tradicionais de MLP e os métodos com melhor desempenho demandaram mais recursos computacionais que as alternativas menos assertivas.

Resultados interessantes foram alcançados por Meng et al. [6] ao classificar imagens hiperespectrais (imagens compostas por múltiplos canais de entrada) utilizando um modelo desenvolvido por eles chamado SS-MLP, que aproveita as informações espectrais e espaciais para extrair características com mais detalhes, além de correlacionar essas informações para capturar dependências de mais alto nível.

Zhang et al. [7] discutem como o modelo MLP-Mixer, que é um dos objetos de estudo deste trabalho, é competitivo frente a opções alternativas. Eles reforçam que, quando treinado em um grande volume de dados ($\geq 10^6$ imagens) o modelo atinge um balanço entre acurácia e custo que performa muito próximo a soluções como redes neurais convolucionais e *Transformers*. Outros trabalhos procuraram aprimorar a arquitetura do MLP-Mixer, como o ResMLP e o gMLP (este último também objeto de estudo deste trabalho), incluindo o trabalho desenvolvido por Zhang et al., que foi o desenvolvimento do Multi-Scale MLP-Mixer. Essa é uma solução que considera o tamanho do *patch* escolhido para treinar cada imagem de entrada, pois diferentes imagens possuem diferentes características que treinadas com *patches* de tamanho mais apropriado, resultam em uma melhor performance geral do modelo. A escala de análise é selecionada de forma adaptativa a cada imagem, o que melhora a eficiência computacional do modelo.

O intuito deste trabalho é colocar os modelos MLP-Mixer e gMLP à prova frente à tarefa de classificar imagens do conjunto selecionado para o experimento, que está melhor descrito na seção de metodologia. Os artigos destacados até agora mostram que esse ferramental de aprendizagem de máquina é bastante capaz de realizar tarefas de classificação de imagens com diferentes propósitos de utilização. Assim, o objetivo central desta pesquisa é avaliar a performance desses modelos diante da nova base de dados escolhida e compará-la com a de outras soluções que realizam a mesma tarefa por estratégias internas diferentes. A ideia é observar o desempenho para identificar se essas soluções alternativas (MLP-Mixer e gMLP) poderiam ser aproveitadas em contextos que os recursos disponíveis de *hardware* são limitados.

2.1 Inteligência Artificial

IA é um conceito que se transformou diversas vezes desde as primeiras discussões sobre o tema, relacionado também com as mudanças na direção da pesquisa da área, que passou de computadores comparáveis com a mente humana a resolvedores de tarefas ou problemas práticos e imitadores de funções cognitivas básicas. Em resumo, inteligência artificial é uma grande área do conhecimento que engloba outras tantas subáreas, que compartilham origens históricas em comum, mas não necessariamente estão dentro do mesmo escopo teórico [8].

Para o escopo dessa pesquisa, IA pode ser definida como a capacidade de uma dada solução computacional resolver problemas de ordem abstrata (partindo da perspectiva humana), como a tarefa de identificar diferenças e características visuais de um dado alvo de observação que permitam categorizar um conjunto de informações. Dobrev produziu uma definição mais geral nesse sentido: “IA deve ser um programa que em um mundo qualquer irá lidar com um problema de forma não pior que um humano.” [9]

2.2 Aprendizagem de Máquina

Aprender é um processo que envolve esforço, repetição e coleta de informações, além das experiências que ajudam no entendimento de um dado objeto de estudo. Aprendizagem ou Aprendizado de Máquina é uma técnica que melhora a performance de um sistema inspirada nesse processo humano, ao passo que utiliza da experiência como guia do aprendizado, através de métodos computacionais. A experiência é obtida através de dados que são usados pelos algoritmos de aprendizagem que constroem modelos com base nesses dados. Os modelos construídos devem conseguir realizar novas previsões diante de novas observações, com base na experiência obtida por meio dos dados que recebeu [10].

2.3 Redes Neurais

Haykin [11] oferece uma definição de rede neural completa e generalista:

> Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:
>
> 1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
>
> 2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Os modelos computacionais avaliados neste trabalho são tipos específicos de redes neurais, implementando estratégias complementares que potencializam suas capacidades. Elas são eficazes para esse tipo de tarefa, pois conseguem extrair informações a partir de dados de entrada e processá-los, produzindo novas informações úteis para um dado objetivo, sem a necessidade de conhecimento de uma solução prévia, pois a rede aprende sobre os dados através da definição de conexões entre características da entrada.

2.4 Multilayer Perceptron

Almeida [12] define perceptron multicamada como um tipo de rede neural constituída por unidades que calculam uma soma ponderada dos valores que recebem na entrada acrescentada de um valor constante. Essa soma resultante passa por uma função de ativação, que produz um sinal de saída ao ser atingido um dado limiar base. O treinamento desse tipo de rede é do tipo supervisionado, o que significa que os dados do conjunto alvo de treino são rotulados com o valor esperado. O conjunto de treino é composto pelos padrões de entrada e os padrões de saída esperados correspondentes. O processo de treinamento é baseado na minimização de alguma medida de erro, calculado entre as saídas produzidas pela rede e as saídas esperadas. O erro é então retropropagado através das camadas da rede, a partir da camada de saída até a camada de entrada, ao passo que os pesos das conexões entre as unidades que compõem as camadas internas são ajustados com base no erro.

2.4.1 MLP-Mixer

A primeira arquitetura de modelo escolhida para análise de desempenho foi a MLP-Mixer, desenvolvida por Tolstikhin et al. no Google Research, Brain Team (2021). A ideia por trás da arquitetura Mixer é separar de forma clara dois tipos de operações realizadas por arquiteturas profundas de redes de aprendizado

visual: aquelas aplicadas em uma região específica da imagem e aquelas aplicadas entre diferentes regiões da imagem. Ambas operações são realizadas por MLPs nessa arquitetura [1].

Como entrada recebe uma sequência de S *patches* não sobrepostos da imagem de mesmo tamanho, ou seja, a imagem é dividida como uma grade. Cada um dos *patches* é projetado em uma *hidden dimension* C de escolha, o que produz uma tabela X bidimensional com números reais, de tamanho S×C. O Mixer é constituído por múltiplas camadas de mesmo tamanho e cada camada composta por dois blocos MLP. O primeiro é o *token-mixing* MLP, que realiza as operações ao longo dos *patches*, nas colunas da tabela transposta de X, e o segundo é o *channel-mixing* MLP, que realiza as operações nas linhas da tabela transposta de X. Cada bloco MLP contém duas camadas totalmente conectadas e uma função de ativação GELU (*Gaussian Error Linear Unit*).

Alguns resultados dessa arquitetura são que a complexidade computacional da rede é linear com relação ao número de *patches* de entrada e que a complexidade com relação ao número de *pixels* da imagem também é linear. Além das suas particularidades, o MLP-Mixer utiliza alguns componentes arquiteturais padrão, como *skip-connections* e *layer normalization*. Nas camadas de saída, utiliza um *head* de classificação padrão com a camada de *global average pooling*, seguido por um classificador linear.

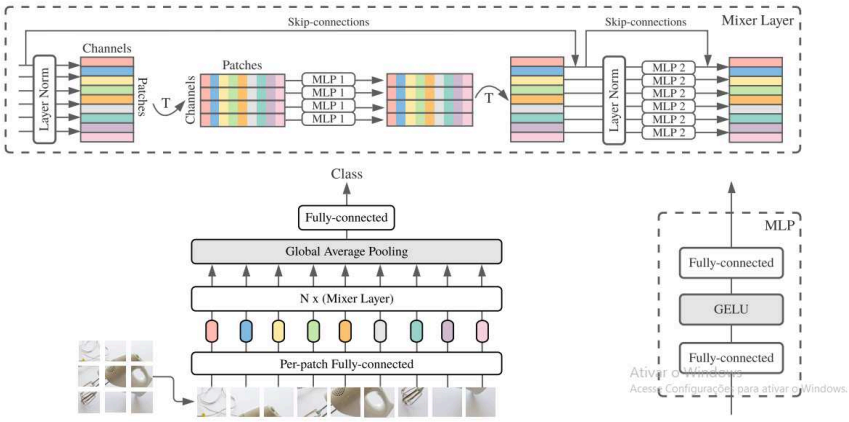
2.4.2 gMLP

A outra arquitetura escolhida foi a gMLP, desenvolvida por Liu et al. (2021). O artigo [3] essencialmente estuda a necessidade dos módulos de *self-attention* em aplicações de processamento de imagem e texto baseados em *Transformers*. Como alternativa propõe uma arquitetura baseada em MLP que não faz uso de *self-attention*, que consiste de projeções de canal e de informação espacial com parametrização estática.

Os autores testaram diferentes designs para a arquitetura e observaram que as projeções espaciais funcionam bem quando são lineares e combinadas com *gating* multiplicativo, esse último que é uma operação que controla o fluxo da informação com relação à sua dimensionalidade e ao aspecto temporal do aprendizado [2]. O desempenho do gMLP na classificação de imagens foi comparável ao do Vision Transformer (ViT) e com acurácia 3% maior que o MLP-Mixer, mas com 66% menos parâmetros. Com esses resultados e os de outros autores citados no artigo, Liu et al. questionam a necessidade das camadas de *self-attention* em Vision Transformers.

Eles também constatam que o *tradeoff* entre a acurácia, o número de parâmetros e a taxa de FLOPs para o gMLP é superior ao de todas as outras arquiteturas baseadas em MLP analisadas no artigo, que estão referenciadas no texto [2]. Tal desempenho é atribuído pelos autores à efetividade da Spatial Gating Unit (SGU), que é um módulo constituinte do gMLP que realiza o processamento *cross-token*, ou seja, que calcula as interações entre os blocos de informação. Apesar do desempenho dos gMLP ser competitivo com o de Transformers, ainda é abaixo do que os melhores modelos de CNN e híbridos em termos de acurácia.

Figura 1: Macroestrutura da arquitetura MLP-Mixer.



Fonte: Figura extraída de [1].

3. METODOLOGIA

A pesquisa deste trabalho é do tipo exploratória, na forma de uma pesquisa experimental, realizando análise quantitativa dos dados. A base de dados usada foi a MYNursingHome, publicada juntamente com o artigo produzido por Ismail et al. [3] e disponível publicamente para *download*, que contém imagens de objetos usualmente presentes em ambientes domésticos. Como ambiente de desenvolvimento e treinamento dos modelos utilizou-se a plataforma *Google Colab* e as bibliotecas *TensorFlow* e *Keras*. O *link*[1] do *notebook* desenvolvido está disponível para consulta.

Para ter acesso ao modelo de referência do experimento, foi utilizado o módulo *applications* de *Keras*, que provê modelos de *deep learning* pré-construídos. Para avaliar o desempenho dos modelos utilizou-se o módulo *metrics* de *Keras*, computando as métricas de acurácia, precisão e *recall*. Para calcular os tempos de treinamento e inferência foi utilizado o módulo *time* de *Python*. Para gerar os gráficos de acurácia e perda dos modelos foi utilizada a biblioteca *matplotlib*.

3.1 Conjunto de Dados

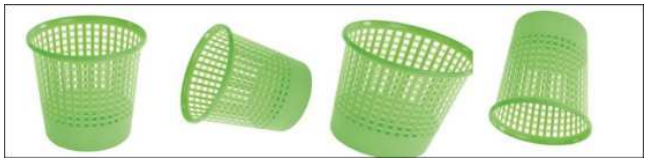
O conjunto de imagens MYNursingHome utilizado no experimento contém fotos variadas de objetos do cotidiano capturadas num sensor de câmera principal (traseira) do iPhone XS Max. A base de dados é composta por 37.500 imagens de 25 diferentes tipos de objetos de interesse comumente encontrados em casas de abrigo de idosos, que é o cenário específico em que as fotos foram tiradas, em diversas dessas casas localizadas na Malásia. As categorias de objetos são: lixeira, cama, banco (assento), armário, campainha, bengala, cadeira, porta, tomada, ventilador, extintor de incêndio, corrimão, ser humano, prateleira, geladeira, chuveiro, pia, sofá, mesa, televisão, vaso sanitário, andador, guarda-roupa, bebedouro e cadeira de rodas. O *dataset* está exemplificado na Figura 2.

Figura 2. Algumas das imagens e categorias presentes no *dataset* MYNursingHome.

Fonte: Figura extraída de [3].

As imagens foram capturadas em diferentes posições e ângulos, e para aumentar o volume de dados, as imagens originais foram randomicamente rotacionadas e modificadas utilizando o processo de *data augmentation* através de transformações geométricas simples. Esse processo produz variações das imagens como mostrado na Figura 3.

Figura 3. Exemplo de um objeto no *dataset* MYNursingHome em diferentes orientações.



Fonte: Figura extraída de [3].

Para inserir o *dataset* no ambiente *Keras*, foi utilizada a função *keras.utils.image_dataset_from_directory*, definindo um valor fixo para o parâmetro *seed* (*seed*: 1337), de forma a permitir repetições do experimento. Além disso, a função recebe um parâmetro *image_size* que define o tamanho final de cada uma das imagens processadas por ela, para que a entrada de dados esteja padronizada, pois as redes utilizadas no treinamento recebem tensores com as mesmas dimensões (largura da imagem, altura da imagem, quantidade de canais de cor).

Para particionar os dados foi definida a proporção do particionamento no parâmetro *validation_split* dessa mesma função, de forma a obter dois *datasets* que contêm o conjunto original. O valor escolhido para *validation_split* foi 0.2, o que significou reservar 80% dos dados para treino. Os 20% dos dados restantes foram separados em 10% para o conjunto de validação e 10% para o de teste.

3.2 Pré-Processamento

Para melhorar o tempo de treinamento e o desempenho dos modelos foi aplicada uma técnica de normalização sobre as imagens, que é transformar os valores de intensidade *RGB* da escala padrão, 0 a 255, para uma escala com valores entre -1 e 1. Além disso foram feitas novas etapas de *data augmentation*, aplicando *random flip* e *random zoom* nas imagens, ou seja, inversões em torno do eixo vertical e *zooms* efetuados de forma aleatória, para aumentar a variedade de apresentações dos objetos.

3.3 Treinamento dos Modelos

Para construir os modelos de classificação, foram escolhidas as arquiteturas MLP-Mixer e gMLP, mencionadas na seção anterior, como as representativas dos MLPs modernos. Como modelo de referência foi utilizada a arquitetura MobileNetV2 [13], que é uma alternativa de rede neural com melhor aproveitamento dos recursos computacionais. Os detalhes sobre a arquitetura de referência podem ser consultados no seu artigo de apresentação e os detalhes de implementação e funcionamento na respectiva documentação de *Keras* [14].

Para fazer as rodadas de experimentos, foi utilizado o algoritmo de otimização *AdamW*, provido por *Keras*, que faz o ajuste da taxa de aprendizado do modelo e da penalização sobre a função de perda. O ajuste desses parâmetros auxilia o modelo a alcançar mais rapidamente os seus pesos ótimos. A função de perda definida foi a *categorical_crossentropy*, que é preparada para lidar com problemas de multiclassificação.

[1] https://colab.research.google.com/drive/1-jxX2E6UvQtnmKybXYscoEd1cRzRKj56?usp=sharing

O treinamento foi realizado por 120 épocas, utilizando parada antecipada, de forma que todos os modelos tivessem condições o mais justas possível, alcançando sua melhor performance. A parada antecipada foi definida observando a perda de validação durante o treinamento, em que após 10 épocas sucessivas sem queda nesse valor o treinamento seria finalizado.

Foi utilizada a classe *ReduceLROnPlateau* de *Keras*, que controla a taxa de aprendizado do algoritmo, observando uma métrica de referência. A métrica escolhida foi a perda de validação, de tal forma que caso ela não reduzisse após 5 épocas, ela sofreria uma redução pela metade. Isso ajuda o modelo a fazer o ajuste fino dos parâmetros, para que o aprendizado não se estagne e ele se aproxime dos pesos ótimos da rede.

Com relação aos hiperparâmetros, foram utilizados os valores de melhor performance para os *datasets* de menor tamanho usados nos experimentos dos artigos das arquiteturas baseadas em MLP. Os valores estão descritos na Tabela 1 para cada um dos modelos.

Tabela 1: Hiperparâmetros

	MLP-Mixer	**gMLP**	**gMLP***
Learning rate	0.001	0.001	0.001
Weight decay	0.1	0.05	0.05
Dropout	0.0	0.0	0.2
Hidden Units	192	192	256

3.4 Teste dos Modelos

Após o treinamento dos modelos, foi realizado o teste desses frente a um conjunto de imagens ainda não observado por eles e coletadas métricas de desempenho. Do módulo *metrics*, foram utilizadas as classes *CategoricalAccuracy*, *Precision* e *Recall*.

4. RESULTADOS E DISCUSSÕES

Após executados os experimentos, foram registrados os valores para as métricas de análise escolhidas. Eles estão apresentados na Tabela 2. As três primeiras métricas se referem ao desempenho para o conjunto de teste.

Quanto ao número de parâmetros, o interessante é que os modelos baseados em MLP possuem bem menos parâmetros que o MobileNetV2 ou outros modelos de redes convolucionais, mas conseguem alcançar taxas de acurácia próximas desses, sendo no pior caso 15,9% inferior a esses.

Os tempos de treinamento dos modelos gMLP são menores que os dos outros, atingindo um balanço entre desempenho e custo.

A utilização de *dropout* e de mais *hidden units* melhora o desempenho do gMLP, o que indica que ainda há margem para melhoria das métricas, visto que é possível fazer um ajuste fino desses parâmetros.

Outro parâmetro que pode ser ajustado para testar o impacto no desempenho dos modelos é o número de blocos MLP utilizados na arquitetura para o experimento. Não realizamos esse tipo de análise porque poderia afetar os tempos de treinamento, e isso esteve limitado durante o trabalho por conta da utilização da versão gratuita do *Google Colab*.

É notório observar o sucesso do MobileNetV2 na tarefa de classificação de imagens, mostrando que CNNs são extremamente capazes de lidar com esse tipo de processamento, apesar de um maior custo computacional.

O modelo gMLP performou melhor que o MLP-Mixer, alcançando resultados competitivos com o MobileNetV2. Contudo, esse último está um degrau acima do primeiro em termos das suas taxas de acerto.

Um comportamento curioso de se observar é o das curvas de aprendizado, referente aos valores de acurácia e perda nos conjuntos de treino e validação. Para os modelos baseados em MLP, a evolução das curvas é mais gradual e suave, o que significa que qualquer instância do modelo em qualquer uma das épocas é a melhor versão até o momento. Já para o MobileNetV2, a evolução tem alguns altos e baixos, alcançando sua melhor versão definitiva apenas quando as curvas encontram o ponto de estabilidade do algoritmo de aprendizado.

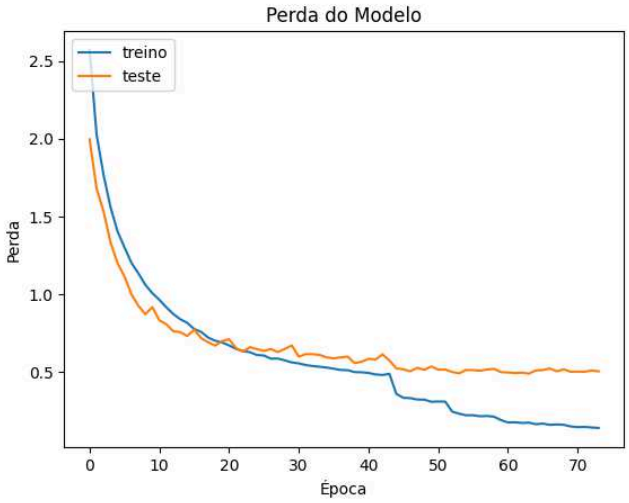
Quanto aos tempos de inferência dos modelos, os resultados são equiparáveis, o que não resulta em diferenças significativas no seu uso. O conjunto de teste possui 3.750 imagens que são pré-processadas e classificadas.

É importante salientar que o *dataset* utilizado possui 37.500 imagens, o que não é uma quantidade tão grande de dados para o treinamento de modelos de *machine learning* na tarefa de classificação de imagens. Então, para *datasets* maiores e com boa diversidade de dados, é possível que os resultados dos experimentos sejam ainda melhores.

Tabela 2: Métricas observadas

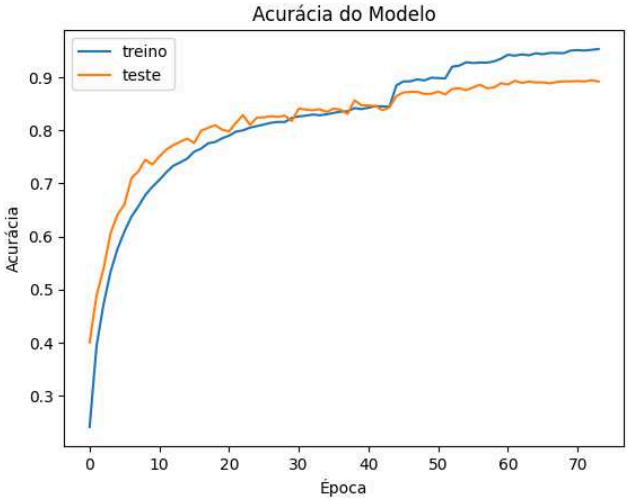
	MLP-Mixer	**gMLP**	**gMLP***	**MobileNetV2**
Acurácia	80.45%	85.53%	89.63%	95.62%
Precision	86.41%	88.16%	91.45%	95.88%
Recall	76.13%	84.37%	88.69%	95.38%
Tempo de Treinamento (minutos)	164.6	88.53	121.6	170.4
Tempo de Inferência (segundos)	16.395	20.488	20.489	20.496
Número de parâmetros	264,537	684,249	684,249	2,290,009

Figura 3. Gráfico de perda para o modelo gMLP*.



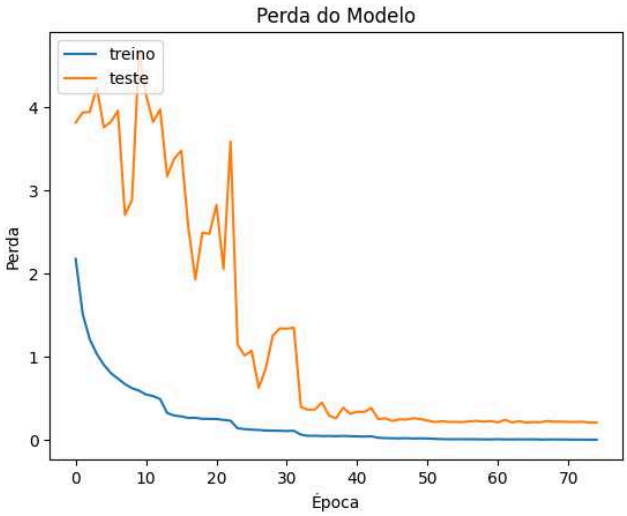
Fonte: Autoria própria.

Figura 3. Gráfico de acurácia para o modelo gMLP*.



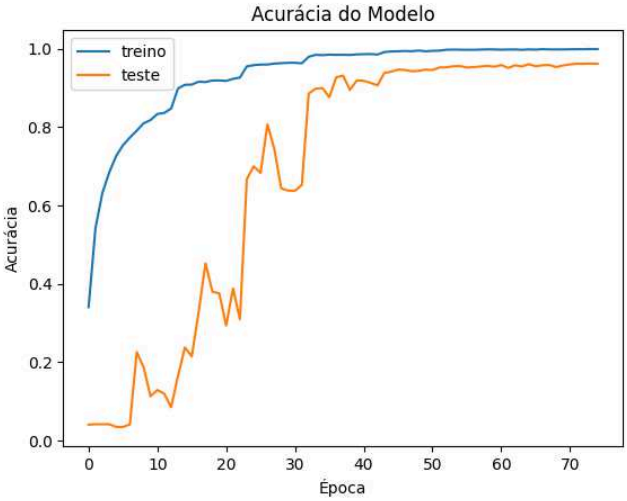
Fonte: Autoria própria.

Figura 3. Gráfico de perda para o modelo MobileNetV2.



Fonte: Autoria própria.

Figura 3. Gráfico de acurácia para o modelo MobileNetV2.



Fonte: Autoria própria.

5. CONSIDERAÇÕES FINAIS

Os resultados desse trabalho foram satisfatórios, pois foi possível atingir performances para os modelos baseados em MLP competitivas com a de modelos estado da arte, como CNNs e *Transformers*.

Como contribuição pode-se citar a avaliação dos modelos MLP-Mixer e gMLP na tarefa de classificação de imagens, especificamente de objetos presentes no cotidiano das pessoas. Além disso, foi possível chegar a resultados semelhantes aos encontrados por Tolstikhin et al. e Liu et al., demonstrando a competitividade dos modelos baseados em MLP.

5.1 Trabalhos Futuros

Os modelos aqui testados poderiam ser treinados com bancos de imagens maiores, a fim de verificar a capacidade de melhoria quando submetidos a mais informações durante o treinamento. A resolução das imagens de entrada também poderia ser ampliada para verificar o impacto no desempenho. Não foi possível nesse trabalho utilizar resoluções maiores pois aumentaria o custo de treinamento e impossibilitaria a utilização do *Google Colab*.

Poderiam ser feitas também outras formas de pré-processamento das imagens que potencializassem os resultados.

Um incremento desse trabalho poderia ser a aplicação desses modelos na classificação de imagens em uma cena visual dinâmica, em que expostos a diferentes objetos na entrada respondesse com a classe do objeto.

Outra forma de verificar e aprimorar os resultados dos experimentos seria a realização de testes estatísticos e a avaliação dos modelos por validação cruzada, fortalecendo a validade dos dados.

5.2 Limitações

As restrições de uso do *Colab* limitaram o desenvolvimento do trabalho, pois os tempos de treinamento são longos e a plataforma exige que o usuário esteja ativo usando a ferramenta. Além disso, quando as unidades computacionais acabavam era necessário esperar alguns dias para utilizar novamente as GPUs do serviço.

Um banco de imagens mais robusto poderia potencializar os resultados gerados pelos modelos estudados nesse trabalho. Seria necessário encontrar um *dataset* que contenha imagens de objetos como os utilizados nesses experimentos, com mais imagens e mais variações dos objetos em questão.

6. AGRADECIMENTOS

Aos meus pais, pelo apoio, incentivo e amor incondicional. Às minhas irmãs, pela divertida companhia que são. Aos meus avós, que me deram carinho e sabedoria. A todos os meus familiares que me formaram como pessoa. Aos meus professores, do curso de Ciência da Computação e da minha formação escolar que me conduziram até aqui e serviram de referência. Ao professor Herman, pela orientação no trabalho e pelas boas discussões. Aos meus amigos que são uma fortaleza na minha vida. Aos colegas de curso pelos momentos divertidos e leves compartilhados.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., & Dosovitskiy, A.. 2021. MLP-mixer: An all-MLP architecture for vision. Advances in neural information processing systems, 34, 24261-2427.

[2] Liu, H., Dai, H., So, D. R., & Le, Q. V.. 2021. Pay attention to MLPs. Advances in neural information processing systems, 34, 9204-9215.

[3] Ismail, A., Ahmad, S. A., Soh, A. C., Hassan, M. K., & Harith, H. H.. 2020. MYNursingHome: A fully-labelled image dataset for indoor object classification. Data in Brief, 32, 106268.

[4] Mol, M. E.. Importância da robótica assistiva para o auxílio da humanidade. 2022. Monografia (Graduação em Engenharia de Controle e Automação). Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto.

[5] Peña-Barragán, J. M., Gutiérrez, P. A., Hervás-Martínez, C., Six, J., Plant, R. E., & López-Granados, F.. 2014. Object-Based Image Classification of Summer Crops with Machine Learning Methods. Remote Sensing, 115(6), 5019-5041.

[6] Meng, Z., Zhao, F., Liang, M.. 2021. SS-MLP: A Novel Spectral-Spatial MLP Architecture for Hyperspectral Image Classification. Remote Sensing, 13(20), 4060.

[7] Zhang, H., Dong, Z., Li, B., He, S.. 2022. Multi-Scale MLP-Mixer for image classification. Knowledge-Based Systems, 258, 109792.

[8] Wang, P.. 2019. On Defining Artificial Intelligence. Journal of Artificial General Intelligence, 10(2). 1-37.

[9] Dobrev, D.. 2012. A definition of artificial intelligence. arXiv preprint arXiv:1210.1568. Retrieved from https://arxiv.org/abs/1210.1568

[10] Zhou, Z.H., Liu, S.. 2021. Machine Learning. Springer Nature, Singapore.

[11] Haykin, S.S.. 2009. Neural Networks and Learning Machines, Pearson International Edition. Pearson.

[12] Almeida, L.. Multilayer perceptrons. Handbook of Neural Computation. C12-1.

[13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & L. C., Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition.

[14] Keras. Keras Applications. Retrieved from https://keras.io/api/applications/

Sobre os autores:

Pedro Raimundo dos Santos Neto. Graduando em Ciência da Computação.

Herman Martins Gomes. Professor orientador.