UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Classificação de imagens de estilos arquitetônicos usando aprendizado profundo

Lucas Serafim da Silva

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Classificação de imagens de estilos arquitetônicos usando aprendizado profundo

Lucas Serafim da Silva

USP - São Carlos

2024

Lucas Serafim da Silva

Classificação de imagens de estilos arquitetônicos usando aprendizado profundo

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Fernando Pereira dos Santos

USP - São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

D111c

Da Silva, Lucas Serafim
Classificação de imagens de estilos arquitetônicos usando aprendizado profundo / Lucas Serafim Da Silva; orientador Fernando Pereira Dos Santos. -- São Carlos, 2024.
55 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Inteligência Artificial. 2. Aprendizado de Máquina. 3. Aprendizado Profundo. 4. Arquitetura. I. Dos Santos, Fernando Pereira , orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

DEDICATÓRIA

A todos aqueles que acreditaram no potencial de transformação social por meio da educação e, por meio dela, mudaram as suas vidas.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus pela vida e por me sustentar no até então maior desafio acadêmico ao qual eu enfrentei. Enquanto arquiteto e urbanista recém formado, via a enorme necessidade de estar cada vez mais próximo do universo que é Inteligência Artificial e *Big Data*, por conta disso procurei uma pós-graduação que pudesse me levar a esse encontro. Desse modo, fique registrado que tive a felicidade de receber uma bolsa de 50% no valor das mensalidades, que me possibilitaram a realização desse curso, de modo que agradeço a Universidade de São Paulo (USP) e o Instituto de Ciências Matemáticas e de Computação (ICMC) pela iniciativa. Expresso também o meu agradecimento as Coordenadoras do curso, Roseli Aparecida Franceli Romero e especialmente a Solange Oliveira Rezende, que por diversos momentos atuou com palavras de motivação e encorajamento em meio as dificuldades que se levantaram no decorrer dessa trajetória. Ao corpo docente, em nome do meu orientador Fernando Pereira dos Santos, por aceitar colaborar com um trabalho que interseccionou arquitetura e inteligência artificial. Ademais, agradeço à minha namorada e futura esposa, Isabella Dias Chagas, pelo companheirismo e constante incentivo; à minha irmã, Sara Serafim da Silva, por acreditar no potencial transformador que o estudo pode trazer para nossas vidas; e por fim, à minha mãe Eucivan Lima Serafim (Vona), pelas orações e palavras de confiança que me levam a ter certeza que dias melhores sempre virão.

EPÍGRAFE

“Todo ser humano nasce com a eternidade no coração e anseia realizar grandes coisas. Nasce com uma sede por significado e propósito, mas nem sempre o significado é claro, nem sempre as respostas são certas, nem sempre entendemos os porquês. Queremos lutar na frente de batalhas, mas somos convocados a cantar vitórias que nunca vimos e das quais não participamos. Queremos ser reconhecidos por grandes feitos, mas somos convocados a cuidar de jardins desconhecidos. Queremos ser livres para viver a vida com o que sempre sonhamos, mas somos convocados a proteger um portão que parece ter sido esquecido.

Tiago Arrais

RESUMO

DA SILVA L. S. **Classificação de imagens de estilos arquitetônicos usando aprendizado profundo**. 2024. 55 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

A presente pesquisa inclinou-se em investigar a classificação de imagens de estilos arquitetônicos usando aprendizado profundo. Tendo como objetivo geral a aplicação de uma rede neural pré-treinada. Além disso, também foram alvo da investigação: a seleção dos estilos de arquitetura a serem trabalhados, bem como a coleta de imagens na internet por meio da técnica de raspagem de dados (*web scraping*); curadoria das imagens coletadas e construção de um conjunto de imagens para aplicação nos ensaios propostos e utilização da técnica de aumento de dados (*data aumengtation*). O desenvolvimento dos testes começou com a construção de um *baseline* que visou testar o desempenho das redes neurais ResnNet 50, Inception V3 e Efficient Net por meio da extração de características do conjunto de imagens e classificação com o algoritmo SVM. Ao se constatar que o melhor desempenho foi o apresentado pela rede Efficient Net, foi realizada a segunda etapa com novos 07 experimentos por meio da técnica de *fine-tuning*, que permite modificações nos parâmetros da penúltima camada da rede, bem como observar os desempenhos alcançados. O experimento 02 foi o que obteve melhores resultados pelas métricas de avaliação que estiveram entre 82 e 84% de aproveitamento. Também foram utilizados para análise de resultados a matrizes de desempenho, algoritmo Grad-CAM, grafo e gráfico de correlação. Como resultados, além da confecção de um conjunto de 3899 imagens distribuídos entre 16 estilos arquitetônicos, foi obtido um rendimento satisfatório na classificação e nos casos em que o houve falha no classificador, foi possível traçar paralelos históricos de proximidade ou inspirações entre as diferentes épocas da arquitetura que podem ter causado tal equívoco.

Palavras-chave: inteligência artificial; aprendizado de máquina; aprendizado profundo; arquitetura.

ABSTRACT

DA SILVA L. S. **Architectural style image classification using deep learning**. 2024. 55 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

This research aimed to investigate the classification of images of architectural styles using deep learning. The general objective was to apply a pre-trained neural network. In addition, the following were also targeted: the selection of architectural styles to be worked on, as well as the collection of images on the internet through the web scraping technique; curation of the collected images and construction of a set of images for application in the proposed tests and use of the data augmentation technique. The development of the tests began with the construction of a baseline that aimed to test the performance of the ResnNet 50, Inception V3 and Efficient Net neural networks by extracting features from the set of images and classifying them with the SVM algorithm. When it was found that the best performance was presented by the Efficient Net network, the second stage was carried out with new 07 experiments through the fine-tuning technique, which allows modifications in the parameters of the penultimate layer of the network, as well as observing the performances achieved. Experiment 02 was the one that obtained the best results according to the evaluation metrics, which were between 82 and 84% of use. Performance matrices, Grad-CAM algorithm, graph and correlation graph were also used for the analysis of results. As a result, in addition to the creation of a set of 3899 images distributed among 16 architectural styles, a satisfactory performance was obtained in the classification and in cases where the classifier failed, it was possible to draw historical parallels of proximity or inspirations between the different periods of architecture that may have caused such a mistake.

Keywords: artificial intelligence; machine learning; deep learning; architecture.

LISTA DE ILUSTRAÇÕES

Figura 1- Algoritmo com AM 15
Figura 2- Algoritmo Tradicional 15
Figura 3- Aprendizado Supervisionado 15
Figura 4- Modelo de Neurônio Biológico 16
Figura 5- Modelo de Neurônio Artificial de McCulloch e Pitts (1943) 17
Figura 6- Modelo Perceptron de Frank Rosemblatt (1957) 18
Figura 7- Camadas Totalmente Conectadas 19
Figura 8- Limitação no Perceptron no problema do Ou Exclusivo 20
Figura 9- Modelo MLP 21
Figura 10- Gradiente Descendente 21
Figura 11 – Etapas de Processamento de uma rede CNN 24
Figura 12 - Camadas Convolucionais 25
Figura 13 - O Papel dos Filtros na Convolução 26
Figura 14 - Obtenção de Características nas Imagens 26
Figura 15 - Exemplificação do Processo de Convolucão em Uma Imagem RGB 27
Figura 16 - Estilos Arquitetônicos Selecionados 29
Figura 17 - Imagens Duplicadas Por Estilo 30
Figura 18 - Imagens Baixadas Por Estilo 30
Figura 19 - Imagens Selecionadas Por Estilo 31
Figura 20 - Exemplo de Matriz de Confusão 33
Figura 21- Utilização do Grad-CAM 34
Figura 22 - Matriz de Confusão Extração de Características: Efficient Net. 37
Figura 23 - Repartição do Conjunto de Dados para Fine-Tuning. 38
Figura 24 - Curvas de Perda e Precisão do Experimento 02 41
Figura 25 - Matriz de Confusão Fine-Tuning: Experimento 02 42
Figura 26 - Correlação Entre Quantidade de Imagens e Percentual de Acertos 45
Figura 27 - Grafo de Erros de Classificação Entre os Estilos Arquitetônicos. 45
Figura 28 - Algoritmos Grad-CAM Aplicado Aos Estilos Arquitetônicos Selecionados. 46

LISTA DE TABELAS

Tabela 1 – Resultados Obtidos Para a Extração de Características. 36
Tabela 2 - Parâmetros Utilizados nos Experimentos 39
Tabela 3 - Resultados Obtidos Para os Experimentos Realizados com *Fine-Tuning*. 40
Tabela 4 - Quadro Síntese dos Resultados Obtidos 43
Tabela 5 - Comparação Entre os Melhores Resultados Obtidos Por Classe 44

LISTA DE EQUAÇÕES

Equação 1- Equação do modelo de Neurônio Artificial de McCulloch e Pitts (1943) 17
Equação 2 - Regra de Delta 19
Equação 3 - Equação da Rede Neural Convolucional 23
Equação 4 - Cálculo da Precisão (Precision) 32
Equação 5 - Cálculo da Revocação (Recall) 32
Equação 6 - Cálculo da F1-Score 33

SUMÁRIO

1 INTRODUÇÃO 11
1.1. Motivação 11
1.2. Objetivos 12
2 FUNDAMENTAÇÃO TEÓRICA 13
2.1 Contexto da Visão Computacional 13
2.2 Inteligência Artificial e o Aprendizado de Máquina 14
2.3 Redes Neurais Artificiais 15
2.4 Aprendizado Profundo 22
2.5 Redes Neurais Convolucionais 22
2.5.1. Componentes de uma Rede Neural Convolucional 24
2.5.2. Camadas Convolucionais 25
3 METODOLOGIA 28
3.1 Construção do Conjunto de Dados 28
3.1.1. Coleta e curadoria das imagens 30
3.2 Métricas de Desempenho: Precision, Recall e F1-Score 31
3.3 Matriz de Confusão 33
3.4 Grad-CAM 34
3.5 Metodologia 35
4 EXPERIMENTOS REALIZADOS 36
4.1 Extração de Características 36
4.2 Transferência de Aprendizado com Fine-Tuning 37
4.2.1. Experimentos realizados 39
4.2.2. Resultado dos Experimentos 40
4.3 Conclusões dos Resultados 42
5 CONCLUSÕES 47
REFERÊNCIAS 50

1 INTRODUÇÃO

1.1.Motivação

A arquitetura tem origem ainda na pré-história, as primeiras civilizações em desenvolvimento já criavam abrigos que pudessem as proteger do mundo exterior. Em meados de 12000 a.C., os seres humanos ocupavam todo o globo terrestre, partindo do oeste da África até a extremidade meridional da América do Sul. Através do desenvolvimento da agropecuária e da caça, que permitiram a fixação desses grupos junto a córregos, o conhecimento das estações do ano e o fortalecimento cultural sendo repassado de geração em geração, que a construção e seus usos propósitos religiosos ou comunitários passaram a se fortalecer e ter cada vez mais protagonismo social (CHING; ECKLER, 2013).

A etimologia da palavra arquitetura nasce com os gregos, a partir da necessidade de diferenciar as construções mais simples tecnicamente (onde os cidadãos moravam), dos edifícios de maior significado social (templos). Desse modo, uniu-se o termo *tektonicos* que diz respeito ao profissional construtor e a edificação, ao radical *arché* que corresponde a: principal, autoridade ou origem. As obras arquitetônicas principais passam a ser o centro da esfera social, por meio delas foram apresentados aos cidadãos os deuses, as histórias e as bases éticas do seu povo (BRANDÃO, 1999; HEIDEGGER, 1986).

Conceitualmente, deve-se considerar a arquitetura a junção da arte e da ciência, pois trata-se de uma disciplina artística inventiva que se utiliza de técnicas de construção específicas para se materializar, tendo como prioridade as pessoas que habitarão naquele lugar (CHING; ECKLER, 2013). Bruno Zevi (1996, p.17), diferencia a arquitetura das outras artes a partir do tratamento espacial indissociável a ela, “como uma grande escultura escavada, cujo interior o homem penetra e caminha”. Desse modo, para julgar se uma arquitetura é boa ou não, já não se deve analisar apenas uma avaliação estética das suas fachadas e ornamentos, mas considerar principalmente se os seus espaços atendem as necessidades dos usuários.

Além disso, existem outros elementos que perpassam todo o processo projetivo das edificações, tais como o contexto histórico, social, econômico, político, religioso, estéticos e das tecnologias de construção. Ademais, cabe a reflexão de que a arquitetura não é mera expectadora do meio ao qual foi concebida, mas coautora do mesmo (CHING; JARZOMBEK; PRAKASH, 2019; ZEVI, 1996). Por conta disso, ao longo dos séculos, diferentes civilizações se manifestaram através das suas construções de diferentes formas, o que contribuiu para o surgimento de diversos estilos arquitetônicos.

A tarefa de utilizar edifícios para testar algoritmos de aprendizado de máquina (AM), tem atraído a atenção de diversos pesquisadores. Em 2007, os pesquisadores Berg, Grabler e Malik propuseram a segmentação semântica de cenas arquitetônicas, atribuindo rótulos para cada pixel com base em características visuais e contextuais. Também foram propostas soluções baseadas na recuperação de instâncias parecidas em imagens e depois buscar classificá-las em estilos arquitetônicos (GOEL; JUNEJA; JAWAHAR, 2012). Além desses, Xu *et al*. (2014), utilizaram a Regressão Logística Latente Multinomial (MLLR) para classificar um conjunto de imagens elaborado pelos autores, com aproximadamente 5000 imagens de 25 estilos arquitetônicos.

Partindo de tarefas mais simples, como classificar elementos de patrimônios históricos edificados para ajudar em sua documentação (LLAMAS, 2017). Aos desafios mais complexos, como proposto por Wang *et al.* (2019), investigaram a precisão de uma rede de ramificação dupla, para aprender as principais características de edificações contidas no estilo gótico e como se manifestam em diferentes países (Inglaterra, França e Itália). Por fim, foi possível verificar pesquisadores apresentando a classificação de obras dos seus países de origem, Sharma (2017) modelou uma rede neural própria para classificar monumentos indianos com rede neural própria e Darbandy, Zoajaji e Sani (2023), que classificaram estilos arquitetônicos iranianos, utilizando aprendizagem por transferência e comparando 5 arquiteturas populares.

1.2. Objetivos

De posse do contexto apresentado anteriormente, a pesquisa possui como objetivo geral aplicar uma rede neural pré-treinada de aprendizado profundo capaz de classificar imagens de diferentes estilos arquitetônico. Em termos específicos, esse objetivo pode ser detalhado nos seguintes tópicos:

- Selecionar Estilos Arquitetônicos historicamente relevantes para esta pesquisa e realizar coleta de imagens da internet utilizando técnica de raspagem de dados (*web scraping*);
- Realizar curadoria das imagens coletas e preparar conjunto de dados que será utilizado;
- Aplicar aprendizado profundo através de modelos de redes neurais pré-treinadas no conjunto de imagens.
- Aplicar técnicas de aumento de dados (*data augmentation*) no conjunto de imagens;

2 FUNDAMENTAÇÃO TEÓRICA

O capítulo 2 apresenta a sustentação teórica observada para o desenvolvimento do projeto, inicia com a contextualizado a área de Visão Computacional. Em sequência, foi apresentado o aprendizado de máquina e a sua diferenciação em relação aos métodos tradicionais de programação. Também, será apresentado o temporal das redes neurais artificias, partindo da ideia do neurônio artificial até o a criação das redes neurais profundas.

2.1 Contexto da Visão Computacional

Uma das características mais marcante dos seres humanos é a capacidade de visualizar o mundo com a percepção tridimensional e identificar diferentes objetos, ambientes, texturas, transparência, luz e sombra com certa naturalidade (SUCAR; GOMES, 2011; SZELINSKI, 2022). Inclinando-se a investigar sobre a visão dos homens e animais, pesquisadores buscaram formas de reproduzir matematicamente o mundo em ambiente virtual. A visão computacional pode ser entendida como uma análise de imagens capaz de extrair caraterísticas e interpreta-las por meio do computador (BROWN, 1984; SUCAR; GOMES, 2011, SZELINSKI, 2022).

Tal ideia parecia um sonho distante, no entanto, os últimos anos trouxeram consigo grandes avanços para área, mas ainda existem lacunas no conhecimento a serem superadas. O desafio para os pesquisadores é fazer com que o computador possa compreender um contexto limitando-se ao que pode ser visto, que em alguns casos não é o suficiente. Para solucionar essa característica, podem ser utilizados modelos probabilísticos amparados na física, ou aprendizado de máquina a partir de grandes conjuntos de exemplos (SZELINSKI, 2022).

Os estudos embrionários em visão computacional são datados da década de 1950, com o reconhecimento estatístico de padrões, para classificar uma imagem em um pequeno conjunto de classes, sendo exemplo disso a identificação de caracteres. (BROWN, 1984). O que há atualmente, são aplicações que podem auxiliar na tomada de decisão em diferentes setores da sociedade. Como exemplos relevantes de utilização dessa tecnologia temos algoritmos de detecção e reconhecimento de pessoas, com potencial de utilização na segurança pública (KONNO JÚNIOR; MOURA, 2023); aprendizado profundo para auxiliar na interpretação de imagens médicas (RAJPURKAR *et al.,* 2022) ou ainda, reconstrução de cenários urbanos em 3D a partir de coleções de fotos disponíveis na internet (AGARWAL *et al.*, 2011). De posse disso, fica evidenciado o papel primordial que a visão computacional tem e terá cada vez mais com o desenvolvimento e popularização de seu uso.

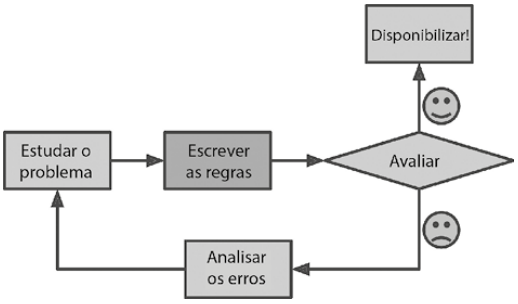
2.2 Inteligência Artificial e o Aprendizado de Máquina

A inteligência artificial (*artificial intelligence* ou IA) se tornou um tema em voga nos últimos anos, seja pela sua capacidade de realizar tarefas que outrora eram exclusivas dos humanos com alta precisão e velocidade, seja pelos riscos de seu uso desregulado. Para adentrar nesta discussão, é interessante compreender o conceito amplo de inteligência, como característica de um sistema biológico ou artificial capaz de estimar o nível de efetividade na resolução de um problema (GABRIEL, 2022). A autora Gabriel (2022, p.54), define que "inteligência de um sistema é sua capacidade de processar fluxos de informação, aprender e se modificar para otimizar resultados na solução de problemas ou para alcançar objetivos específicos."

Em sistemas orgânicos (onde estão inseridos os seres humanos), o cérebro biológico possui a capacidade de processar e aprender com os dados que são obtidos através dos sentidos e enviados ao cérebro por meio do sistema nervoso, além da habilidade de adaptação ao meio em que se foi exposto como forma de obter melhor desempenho. Em sistemas artificiais, o "pensamento" normalmente é feito pelo computador, que processa dados a partir de diferentes formas como sensores, bancos de dados, digitalização, etc.; com possibilidade de aprende e/ou se modificar conforme forem programados (GABRIEL, 2022). Embasado nisso, a inteligência artificial pode ser entendida como "a área da Ciência da Computação que lida com o desenvolvimento de máquinas/computadores com capacidade de imitar a inteligência humana" (*ibid.*, p.56).

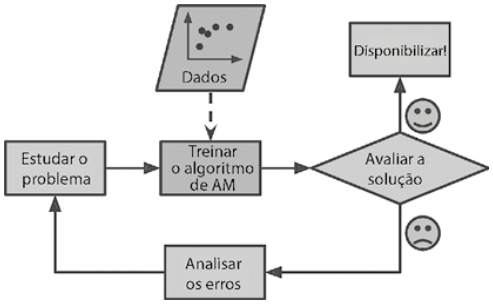
O aprendizado de máquina (*machine learning* ou AM) pode ser visto como uma subárea da inteligência artificial, que foi iniciada pela publicação do artigo *Some Studies In Machine Learning Using the Game of Checkers* por Arthur Samuel (1959), cuja máquina foi programada para vencer humanos em jogos de damas. O autor descreveu o AM como uma programação de computador treinado para se comportar como se fosse realizado por seres humanos ou animais, que envolve processo de aprendizado pela experiência, que reduz a necessidade de codificar todas as etapas (Figura 1 e Figura 2). Esse entendimento pode ser complementado por Géron (2021, p.16) "aprendizado de máquina é a ciência (e a arte) da programação de computadores de modo que eles possam aprender com os dados".

Figura 1- Algoritmo Tradicional



Fonte: Géron (2021).

Figura 2- Algoritmo com AM



Fonte: Géron (2021).

Os algoritmos de AM podem ser classificados em relação quantidade e tipo de supervisão que recebem no decorrer do treinamento, as quatro principais categorias são: supervisionada, não supervisionada, semisupervisionada e aprendizado por reforço. Como enfoque desta pesquisa, será utilizada a aprendizagem supervisionada (AS), nessa classe, as instâncias (*inputs*) do conjunto de dados (*dataset*) são rotuladas (*labels*), o que torna o processo mais demorado, por demandar maior acurácia na revisão dos dados utilizados. O objetivo é que o modelo seja capaz de generalizar o aprendizado (Figura 3), ou seja, classificar corretamente novas instâncias (*outputs*) não utilizadas no treinamento (GÉRON, 2021; SANTAELLA, 2023; TAULLI, 2020).

São exemplos de algoritmos dessa categoria K-ésimo vizinho mais próximo (KNN), Máquinas de vetores de suporte (SVMs), Regressão linear, Árvore de decisão, Floresta de decisão e Redes Neurais Artificiais, que vem sendo amplamente utilizadas a partir de 2012.

Figura 3- Aprendizado Supervisionado

Fonte: Géron (2021).

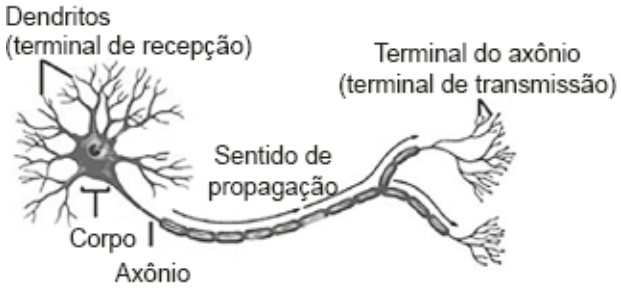
2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (*Artificial Neural Networks* ou RNAs) são exemplos de como observações da natureza podem ser fundamentais para solucionar problemas relevantes para a sociedade. O estudo da estrutura cerebral, possibilitou a construção desse modelo de aprendizado de máquina que se destaca por sua capacidade de realizar tarefas robustas e complexas, como classificar bilhões de imagens, reconhecimento de fala, sistemas de

recomendação baseados no perfil do usuário, etc. (GABRIEL, 2022; GÉRON, 2021). O inventor dos primeiros neurocomputadores, Dr. Robert Hecth-Nielsen definiu rede neural como "um sistema computacional feito de uma quantidade de elementos de processamento simples, altamente conectados, que processam informação por meio dos seus estados dinâmicos de resposta a sinais externos a ele" na publicação *Neural Network Primer: Part I* por Maureen Caudill, AI Expert, em fevereiro de 1989 (GABRIEL, 2022).

Para melhor compreensão desse conceito, é fundamental estudar o neurônio biológicos e seus principais elementos (Figura 4). Trata-se de uma célula encontrada em cérebros animais (não necessariamente humano), composto essencialmente pelo I) corpo do neurônio: responsável por receber e combinar informações de outros neurônios; II) axônio: fibra tubular que transmite estímulos para as demais células e III) dendritos: que recebem os estímulos vindos de outros neurônios (GABRIEL, 2022). De forma sintética, essas informações são transmitidas através de impulsos elétricos, conhecidos como Potenciais de Ação (PAs) pelos axônios e fazem com que os terminais de transmissão (ou terminais sinápticos) emitam sinais químicos, chamados de neurotransmissores que ao receberem grande quantidade de estímulos em milissegundos disparam seus próprios impulsos elétricos, ou seja, são ativados (GÉRON, 2021).

Figura 4- Modelo de Neurônio Biológico

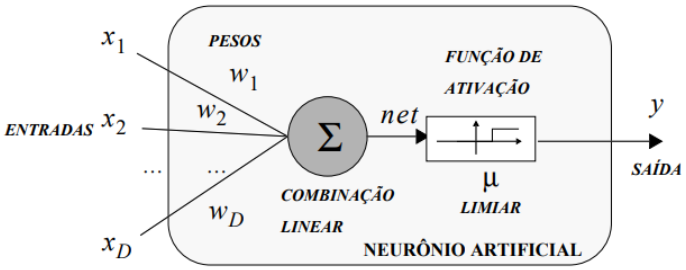


Fonte: Gabriel (2022).

O primeiro neurônio artificial (Figura 5) foi proposto pelo neurofisiologista Warren McCulloch e pelo matemático Walter Pitts, no artigo *A Logical Calculus of Ideas Immanent in Nervous Activity* em 1943 (GABRIEL, 2022; GÉRON, 2021). Em sua equação (Equação 1) seu modelo possui *D* entradas (x_j) binárias (ativar/desativar) no neurônio de processamento que são associados a pesos (w_j) que indicam a importância de cada entrada resultando em uma função linear de valor *net*, resultando em uma combinação linear, que é ativa o neurônio a partir quando do alcance de um limiar (μ), apresentando o valor 1 na saída binária *y*, em caso negativo o valor exibido é $y = 0$. A função de *Heaveside* (ou função de degrau) é utilizada para comparar o valor

de *net* com o limiar μ resultando em $\theta(x) = 1$ se $x \geq 0$ se ativado e $\theta(x) = 0$ caso não seja alcançado (RAUBER, 2005). Essa publicação serviu como lastro para comprovar que seria possível construir uma rede neural artificial que calculasse qualquer problema de lógica proposicional (GABRIEL, 2022).

Figura 5- Modelo de Neurônio Artificial de McCulloch e Pitts (1943)



Fonte: Rauber (2005).

Equação 1- Equação do modelo de Neurônio Artificial de McCulloch e Pitts (1943)

$$y = \theta\left(\sum_{J=1}^{D} w_j \, x_{j-}\mu\right)$$

Fonte: Rauber (2005).

Anos mais tarde (1949), Donald Hebb publicou em seu livro *The Organization of Behavior* uma hipótese baseada em estudos biológicos do cérebro, que quando um neurônio aciona outro com determinada frequência, a conexão entre eles fica mais forte, fazendo com que os pesos dos neurônios sejam ativados simultaneamente (GÉRON, 2021; RAUBER, 2005). Para Rauber (2005, p.11) "a regra de Hebb define um algoritmo de adaptação dos pesos, porém sem a definição de um objetivo a atingir, por exemplo, minimizar um erro entre um valor desejado e calculado". As redes Perceptrons utilizaram uma variação dessa teoria para corrigir os erros de predição, ao reforçar conexões corretas.

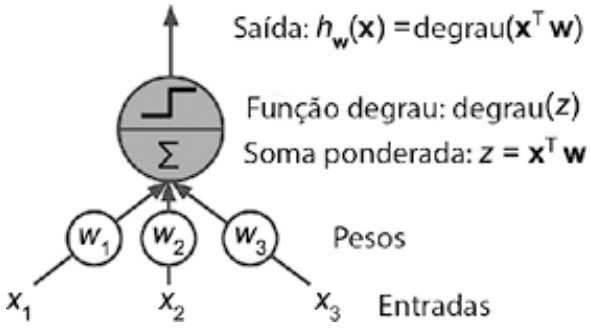
Criadas em 1957 pelo psicólogo e pioneiro no campo da inteligência artificial, Frank Rosenblatt, as redes Perceptrons (Figura 6) são uma das mais simples RNAs por possuírem apenas uma única camada, propagação para frente[1] (*feedfoward*) e saída de valores binários (GÉRON, 2021; MUELLER; MASSARON, 2019; RAUBER, 2005). A ideia do cientista era "criar um computador capaz de aprender a partir de tentativa e erro, assim como os humanos"

[1] "O fluxo de informação é sempre unidirecional, ao contrário de redes com realimentação (*backpropagation*)" (RAUBER, 2005, p. 12).

(MUELLER; MASSARON, 2019, p.146). O Perceptron foi o primeiro modelo capaz separar linearmente classes por conta da sua capacidade de aprendizado dos pesos de cada categoria a partir dos exemplos de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016).

A arquitetura utilizada é ligeiramente diferente do neurônio artificial, chamado de unidade lógica de limiar (TLU). Possui um peso[2] associado a cada conexão de entrada que por sua vez são números, bem como as saídas. A TLU calcula a soma ponderada das entradas ($z = w_1x_1 + w_2x_2 + \cdots + w_nx_n = \boldsymbol{x^T w}$), aplica uma função de degrau (*Heaviside*) ao valor somado resultando em: hw(x) = degrau(z), sendo que $z = \boldsymbol{x^T w}$, (GÉRON, 2021).

Figura 6- Modelo Perceptron de Frank Rosemblatt (1957)



Fonte: Géron (2021).

A atualização dos pesos do Perceptron básico ocorre por meio de um algoritmo iterativo[3], por se tratar de uma aprendizagem supervisionada, os valores alvo são conhecidos e comparados com a classificação obtida. Todos os objetos que com erro de classificação são utilizados para modificar os pesos, até que desapareçam ou diminuam ao máximo. A arquitetura ADALINE (*Adaptative Linear Neuron*), foi criada por Windrow e Hoff em 1960 e se difere por permitir classificações lineares contínuas, com valores reais. Os autores também apresentaram a regra de aprendizagem Delta (Equação 2), para cálculo de erro (e_i) de classificação no TLU ($_i$) pela diferença entre o predito (y'_i) e o valor desejado (y_i), de modo que serão modificados os pesos das entradas que mais contribuem para o erro serão modificados proporcionalmente à função de ativação, erro e escalado por uma taxa de aprendizado (η), minimizando o erro iterativamente pelos exemplos fornecidos (RAUBER, 2005).

---

2 Inicializado aleatoriamente.

3 Ou seja, o conjunto de objetos/instâncias passa pela rede neural por um número finito de vezes buscando uma separação linear (RAUBER, 2005).

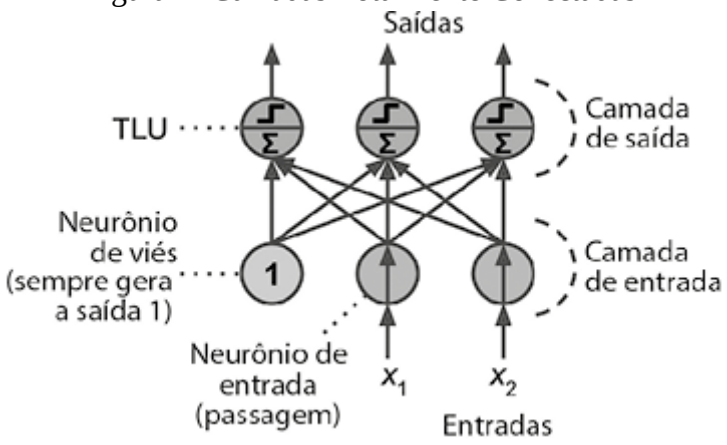
Equação 2 - Regra de Delta

$$\Delta w_{ij} = \eta e_i \, y_j = \eta(y_i - y'_i)y_j$$

Fonte: Rauber (2005).

Um perceptron possui apenas uma camada de TLUs, e cada um deles está conectada a todas as entradas. São chamadas de camadas densas ou totalmente conectadas (*fully connected layer*) quando todos os neurônios de uma camada se conectam a todos os neurônios da camada anterior[4] (Figura 7). Os valores de entrada passam por neurônios especiais de entrada que geram a saída de qualquer entrada fornecida. Também, é adicionada uma característica de viés extra[5] ($x_0 = 1$) em um neurônio de viés, que em todo tempo gera saída 1. Quando a camada de saída possui vários neurônios capazes de classificar instâncias simultaneamente em classes binárias distintas, recebe o nome de classificador de saída múltipla (GÉRON, 2021).

Figura 7- Camadas Totalmente Conectadas

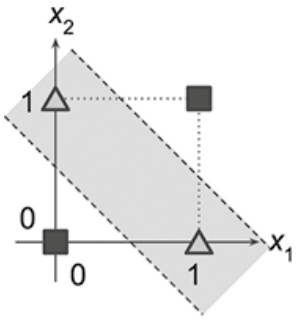


Fonte: Géron (2021).

As evoluções no estudo das redes neurais foram paralisadas em 1969, com a publicação Perceptrons de Marvin Minsky e Seymour Papert. Os autores provaram a limitação do algoritmo em resolver alguns problemas corriqueiros, como o problema de classificação do Ou Exclusivo (*Exclusive OR* ou XOR) (Figura 8). Em verdade, nenhum classificador linear consegue solucionar esse problema, no entanto, se esperava que o Perceptron pudesse representar um avanço nesse sentido, o que levou a comunidade científica a abandonar o estudo das redes neurais artificiais por aproximadamente 20 anos (GÉRON, 2021; RAUBER, 2005).

[4] Isso é, da camada de entrada.
[5] Também chamado de *bias*

Figura 8- Limitação no Perceptron no problema do Ou Exclusivo



Fonte: Géron (2021).

As limitações apresentadas só seriam superadas a partir da utilização de redes neurais perceptrons de múltiplas camadas (*multilayer perceptron* ou MLP) (Figura 9), que são compostas por uma camada de entrada (passagem), uma ou mais camadas de TLUs ocultas (*Hidden Layers*), ambas possuem um neurônio de viés e estão totalmente conectadas com a camada seguinte, ainda, possui uma última camada de TLUs, chamada de saída (GÉRON, 2021; HAYKIN, 2007). Para Géron (2021), seu sucesso está associado a utilização do algoritmo de retropropagação (*backpropagation*), criado em 1986 por David Rumelhart, Geoffrey Hinton e Ronald Williams em artigo intitulado *Learning Internal Representations by Error Propagation*. Esse algoritmo atingiu tamanha relevância que até os dias de hoje é utilizado para resolver problemas de complexos de classificação.

Tem como suporte teórico o gradiente descendente (Figura 10) “um algoritmo de otimização genérico que consegue identificar ótimas soluções para um leque amplo de problemas. A ideia geral do gradiente descendente é ajustar iterativamente os parâmetros com intuito de minimizar a função de custo” (GÉRON, 2021, p.60). Simplificadamente, o aprendizado por retropropagação do erro possui duas etapas:

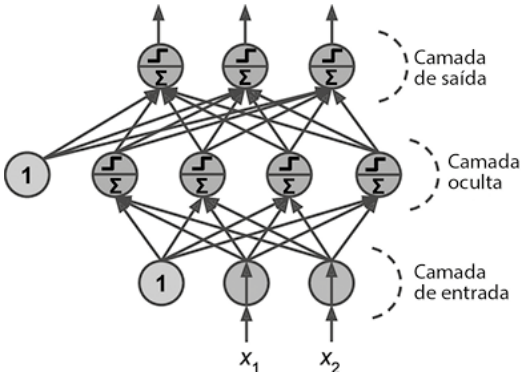
I) O passo para a frente (*forward pass*), os pesos sinápticos são aleatoriamente fixados e as entradas passam por todas as camadas até a saída, produzindo o erro inicial da rede;

II) Sua correção é feita por meio do passo para trás (*reverse pass*), os pesos são atualizados levando em consideração a sua contribuição para erro produzido e finalmente são ajustados por retropropagação para reduzir o erro até o valor mínimo possível (GÉRON, 2021; HAYKIN, 2007).

De acordo com Géron (2021), outro ponto importante a ser destacado foi a utilização da função logística (sigmóide) em detrimento da função degrau, que possibilitou segmentos não planos e progressão adequada do gradiente descendente, ademais, o algoritmo *backpropagation*

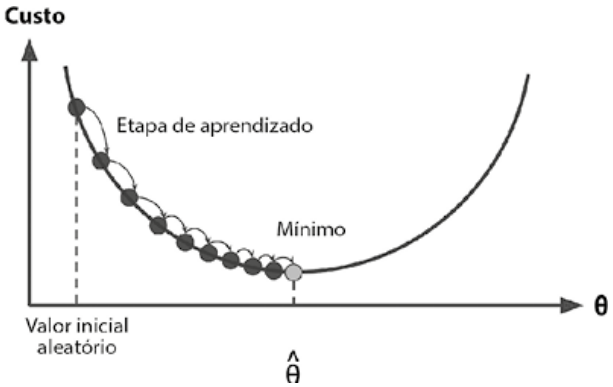
mostrou-se eficiente também ao utilizar as funções tangente hiperbólica (tanh) e unidade linear retificada (ReLU).

Figura 9- Modelo MLP



Fonte: Géron (2021).

Figura 10- Gradiente Descendente



Fonte: Géron (2021).

Por conta disso, é fundamental que os dados escolhidos para esse processo sejam de boa qualidade e bem tratados para que possam generalizar adequadamente, sem que haja um viés forte que inviabilize sua utilização em novos dados. Os autores Ponti *et al*. (2021), elencaram alguns cuidados essenciais com o conjunto de dados trabalhado:

A. Lista de verificação básica: Trata-se de uma lista com sete itens que abordam I) Se os dados de entrada de fato representam os padrões alvo e seu potencial de reconhecimento pelo ser humano; II) A adequada normalização dos dados, isso é, uma padronização dos intervalos numéricos; III) A qualidade dos dados utilizados, considerando que dados rotulados erroneamente prejudicam a capacidade dos modelos; IV) Uso de função de perda e métricas de avaliação; V) Estrutura razoável dos recursos projetados que permita identificar forte separabilidade das classes; VI) Ajuste e validação do modelo no conjunto de treinamento, nunca utilizando o conjunto de teste e; VII) Uso de validação interna e externa.

B. Conjuntos de dados pequenos com fraca convergência: Nesses casos, é indicado que seja realizado aprendizagem por transferência (*transfer learning*) por meio de redes neurais profundas (DNNs) ou aumento de dados (*data augmentation*), se os dados originais estiverem bem representados.

C. Desequilíbrio de dados alvo em tarefas supervisionadas: Em um cenário perfeito, é indicado que haja um equilíbrio no número de amostras das diferentes classes no conjunto de treinamento, podendo trazer engano para a função de perda e métricas de avaliação. Uma solução possível é fornecer pesos as classes, tornando aquelas com menos instâncias mais presentes no conjunto de treinamento.

D. Complexidade dos modelos, *overfitting* e *underfitting*: É importante saber reconhecer cenários indesejados de aprendizagem como a ocorrência de *overfitting* e *underfitting*, que possuem relação com a complexidade dos modelos.

E. Ataques: Quando a rede profunda utiliza padrões incorretos dos dados de treinamento para minimizar a perda e características incorretas são aprendidas, resultando em classificações equivocadas como sendo de alta confiabiabilidade.

2.4 Aprendizado Profundo

O desenvolvimento do Aprendizado Profundo (*Deep Learning*), representou uma quebra de paradigma no Aprendizado de Máquina devido a sua capacidade de aprender características a partir de uma robusta base dados (MUELLER; MASSARON, 2019). Para Ponti *et al*. (2017), o aprendizado profundo via de regra envolve o uso de uma série de camadas que processam cada entrada em busca de representações hierárquicas que servem com instância para a próxima camada. De modo que, a cada camada, as características principais de cada classe do conjunto de treinamento são identificadas e refinadas pela rede neural para serem testados no conjunto de testes.

Com o advindo do *Deep Learning* alguns problemas enfrentados pela IA clássica como reconhecimento de imagens, fala e tradução automática passaram a obter resultados melhores resultados em comparação com os algoritmos tradicionais de ML. Dentre os métodos de aprendizado profundo, as Redes Neurais Convolucionais (CNNs) se destacam pela sua capacidade de uso em tarefas de Visão Computacional e Processamento de Imagens (portanto, ênfase desta pesquisa). Também, as Redes Adversariais Generativas (GANs), Redes Siamesas e Trigêmeas e Auto-Encoders (AEs) também compõem o rol de algoritmos dessa categoria (MUELLER; MASSARON, 2019; PONTI *et al*., 2017).

2.5 Redes Neurais Convolucionais

As redes CNNs tiveram como precursor a publicação *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position* por Fukushima em 1980, onde foi apresentada uma rede neural artificial hierárquica e multicamada capaz de reconhecer padrões visuais baseados na similaridade geométrica. Enquanto, a utilização do aprendizado de máquina alcançava resultados sobre-humanos, sendo exemplo disso o supercomputador Deep Blue da IBM ter vencido Gary Kasparov, xadrezista campeão mundial em 1996, em atividades que envolviam tarefas reconhecimento de imagens, os

resultados obtidos não superavam nem mesmo crianças das mais tenras idades. Para avançar nessa área, novamente os pesquisadores se voltaram ao estudo do cérebro humano, mais especificamente ao córtex visual.

O real despertar das Redes Neurais Convolucionais para classificação de imagens se deu a partir o artigo *Gradient-based learning applied to document recognition* proposto por LeCun *et al*. (1998). A rede LeNet-5 era capaz de reconhecer dígitos manuscritos do banco de dados MINIST que foi organizado pelos autores. Os resultados obtidos foram animadores, tendo sido alcançado uma taxa de erro de apenas 0.95% no conjunto de testes, após 20 iterações.

A maior popularização dessa técnica aconteceu após o a criação da competição anual de classificar o banco de imagens ImageNet (DENG *et al*., 2009), iniciada em 2010. Foi notável a evolução das arquiteturas vencedoras, que são consideradas referência no que diz respeito a reconhecimento de objetos e classificação de imagens tais como AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), VGGNet (SIMONYAN; ZISSERMAN, 2015), GoogleNet (SZEGEDY, 2015) e ResNet (HE *et al.,* 2016). Além da disponibilidade de novos conjuntos de dados rotulados, essa evolução também foi proporcionada pelo desenvolvimento dos *hardwares* de computador que permitiram acelerar o tempo de processamento dos cálculos (PONTI *et al*., 2017).

A principal diferença das CNNs é a organização dos neurônios em três dimensões, que são a largura e altura da entrada e sua profundidade, isso é, o volume de camadas (O'SHEA; NASH, 2015). O termo convolucional trata-se de uma operação matemática explicada por Goodfellow, Bengio e Courville (2016, p.330) "é um tipo especializado de operação linear". Redes Convolucionais são simplesmente redes neurais que usam convolução no lugar da matriz geral de multiplicação em pelo menos uma de suas camadas. Conforme Ponti *et al.* (2017) pode ser escrita matematicamente como uma sequência de funções $f_l(.)$ que recebem um vetor (x_l) como entrada e um conjunto de parâmetros W_l, resultando em um vetor x_{l+1} (Equação 3):

Equação 3 - Equação da Rede Neural Convolucional

$$f(x) = f_L(...f_2(f_1(x_1, W_1); W_2)....), W_L)$$

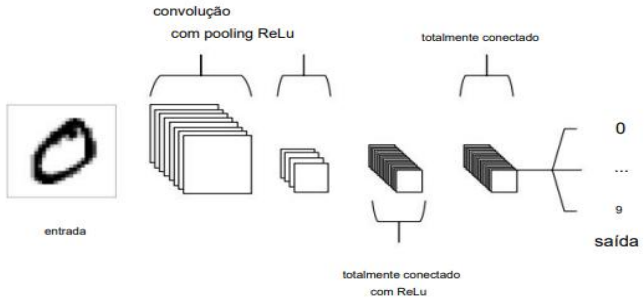
Ponti *et al*. (2017).

2.5.1. Componentes de uma Rede Neural Convolucional

Existem blocos que são considerados básicos na construção de uma rede CNN, que são camadas de convolução (*convolutional layers*), operadores de *pooling*, funções de ativação e camadas totalmente conectadas[6] (PONTI *et al*., 2017). Os autores O'shea e Nash (2015, p.04) detalharam o processamento de uma rede CNN (Figura 11) em cada etapa:

1. A camada de entrada não altera o valor do pixel das imagens, assim como as demais redes neurais artificiais;
2. Na camada convolucional, a saída dos neurônios está conectada a pequenas regiões da entrada, como produto do cálculo escalar entre os pesos e a região conectada ao volume de entrada, resultando em um mapa de características (*feature map*) mais relevantes. Em geral, utiliza-se a função ReLU como ativação e os parâmetros das camadas ficam concentrados nos *kernels* que podem aprendidos com a ativação da camada anterior;
3. Camadas de *pooling* são responsáveis por reduzir a resolução (dimensionalidade espacial) da entrada, filtrando ainda mais o número de parâmetros dentro da ativação;
4. As camadas totalmente conectadas são responsáveis pela classificação a partir das ativações, com base nas características aprendidas no conjunto de treinamento. Também é recomendado o uso de função ReLU nesta etapa para melhoria do desempenho de previsão.

Figura 11 – Etapas de Processamento de uma rede CNN



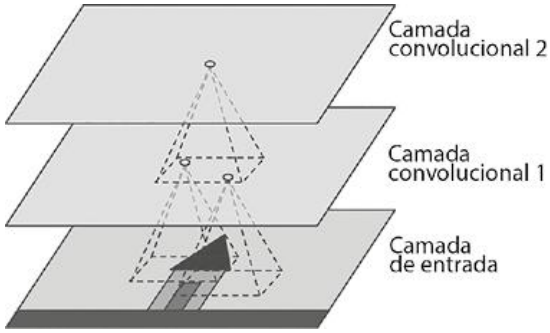
O'shea e Nash (2015).

[6] Ou camadas densas, conforme apresentadas anteriormente.

2.5.2. Camadas Convolucionais

Pode-se afirmar que as camadas de convolução são os elementos mais importantes em uma CNN, na primeira camada, os neurônios não possuem interligação com todos os pixels, mas apenas em seus campos receptivos. No entanto, na segunda camada convolucional há interligações apenas com um retângulo pequeno da primeira camada. Esse processo permite que a rede possa identificar características de baixo nível na primeira camada oculta e nas camadas seguintes haja um refino para reconhecimento de características de maiores níveis suscetivelmente (GÉRON, 2021). É um processo que se aproxima da visão humana, no sentido de identificar o todo, isso é, classificar quase que automaticamente o que está sendo visto (uma pessoa, um carro, uma casa) e em seguida perceber detalhes mais característicos que distinguem aquele alvo dos demais (como a cor dos cabelos, altura, cor de pele) (Figura 12).

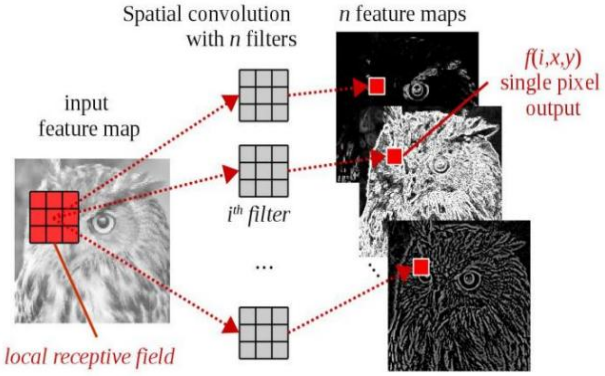
Figura 12 - Camadas Convolucionais



Fonte: Géron (2021).

Essas camadas são compostas por um conjunto de filtros (*kernels*), cada um aplicado a todo o vetor de entrada. Esses filtros são em síntese uma matriz K x K de pesos $W_{i,}$ possibilitando uma combinação linear dos valores dos pixels na região de influência do filtro (Figura 13). Um campo receptivo local pode ser entendido como uma vizinhança de influência do *kernel* possibilitando que a camada se saída seja a combinação de pixels de entrada pelo campo receptivo local (PONTI *et al*., 2017).

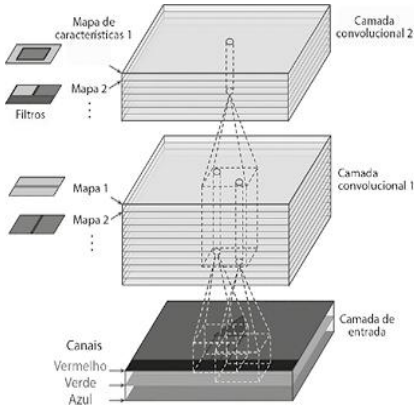
Figura 13 - O Papel dos Filtros na Convolução



Ponti *et al*. (2017).

As convoluções podem capturar diferentes características em cada entrada, como linhas horizontais, curvas ou padrões especiais que possam ajudar na classificação. O local em que aparecem na imagem pouco influencia, tendo o vista a propriedade de invariância, uma das características que distinguem as CNNs das demais redes neurais (Figura 14). Idealmente, cada camada convolucional contribui para a formação de padrões mais complexos, que facilitará a classificação de novas imagens (MUELLER; MASSARON, 2019).

Figura 14 - Obtenção de Características nas Imagens



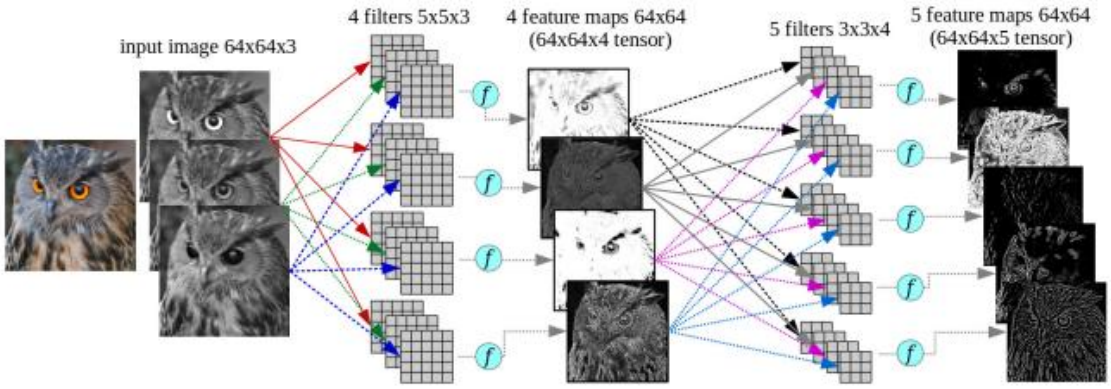
Fonte: Géron (2021).

Para melhor exemplificação, as imagens RGB possuem 3 canais de cores[7], para configurar uma camada convolucional com 4 filtros de tamanho 5 x 5, deve se notar que ele terá na verdade uma dimensão 5 x 5 x 3, para atuar em todas as camadas. Ainda, é necessário colocar uma dimensão para a entrada e a quantidade de canais, que 64 x 64 x 3. Como resultado, será obtida uma saída com 4 matrizes empilhadas que formarão um tensor dimensionado com

[7] Enquanto as imagens em Escala de Cinza, frequentemente utilizadas em CNNs possuem apenas um canal (GÉRON, 2021).

64 x 64 x 4. Além dessa, mais uma camada convolucional com 5 filtros 3 x 3 x 4 pode ser colocada para resultar em um tensor 64 x 64 x 5. Após cada um dos filtros, recomenda-se a utilização de uma função de ativação, como por exemplo a ReLU (PONTI, *et al*., 2017) (Figura 15).

Figura 15 - Exemplificação do Processo de Convolucão em Uma Imagem RGB



Ponti *et al*. (2017).

Uma técnica bastante utilizada lidar com imagens é o preenchimento com zeros (ou *zero padding*) nas bordas nas imagens, garantindo que todas as entradas possuam as mesmas alturas e largura e que nenhuma informação importante será ignorada. Por fim, pode-se destacar o passo de deslocamento (*stride*) do Kernel, ao utilizar o padrão 1, todos os pixels são utilizados na convolução, enquanto que ao utilizar passo 2 apenas os pixels ímpares são processados. Deslocamentos maiores podem otimizar o tempo de processamento da rede neural e reduzir as dimensões das imagens (GÉRON, 2021; PONTI et al., 2017).

3 METODOLOGIA

3.1 Construção do Conjunto de Dados

Após buscas nos principais sites de conjuntos de imagens (*datasets*)[8] para visão computacional, ficou constatado que não existiam exemplares disponíveis com estilos arquitetônicos que fossem suficientemente consistentes com a formação cultural ocidental da Arquitetura e Urbanismo que influenciaram diretamente o desenvolvimento das edificações e cidades no Brasil.

Desse modo, optou-se por uma seleção baseada em obras literárias que culminou na escolha 16 movimentos históricos da arquitetura (Figura 16), buscando estabelecer uma linha temporal que teve como ponto de partida o Egito Antigo, passando pela forte influência da Grécia e Roma Antigas, ao passo que também se desenvolvia a arquitetura Bizantina e posterior declínio do Império Romano, já com mostras do estilo Românico que dividiria parte da Idade Média com seu símbolo máximo, o estilo Gótico.

Após aproximadamente X séculos, o Renascimento gestado na Itália trouxe o homem para o centro do universo, retomando elementos do mundo clássico como a proporção áurea e o desenvolvimento da perspectiva. Chegado o séc. XVI, a igreja Católica retoma o protagonismo o Barroco, até meados do iluminado séc. XVIII, quando a Revolução Francesa inspira um olhar inspirado para o Mundo Antigo, culminando no Neoclássico. Também, a Revolução Industrial e a disponibilidade de novos materiais construtivos abriram os caminhos para novas manifestações artísticas como *Art Nouveau* e *Art Déco*.

Depois do mundo enfrentar o período entreguerras, existia uma grande demanda por romper com o contexto de vida anterior e olhar para a cidade do futuro rapidamente, o movimento Modernista ganhou força com uma proposta funcional e racional que tinha em grandes condomínios de apartamentos a “teoria perfeita” de como se deve morar.

Ocorreu que ainda no séc. XX, diversos estudiosos perceberam que uma única moldura não serve a todas as culturas, e apontaram as inúmeras consequências da adoção desenfreada da arquitetura moderna, tais como a supressão da convivência humana nos espaços urbanos, e passaram a olhar para as cidades que mantiveram suas características, sendo fonte de inspiração para o Pós-Modernismo, que triunfou frente a rigidez anterior, desaguando antagonista dos modernistas, o Desconstrutivismo.

[8] Para mais informações, acesse: https://www.kaggle.com/ e https://datasetsearch.research.google.com/

Para além de uma linhagem temporal, foi percebida a necessidade de valorização de pelo menos um estilo fortemente presente no Brasil, sendo selecionada a Arquitetura Colonial Brasileira, que pode ser vista Centros Históricos do país. Por fim, para representar a influência Oriental na miscigenada cultura brasileira, foi escolhida a Arquitetura Tradicional Chinesa, pelas características marcantes e grande disponibilidade de referências históricas.

Figura 16 - Estilos Arquitetônicos Selecionados

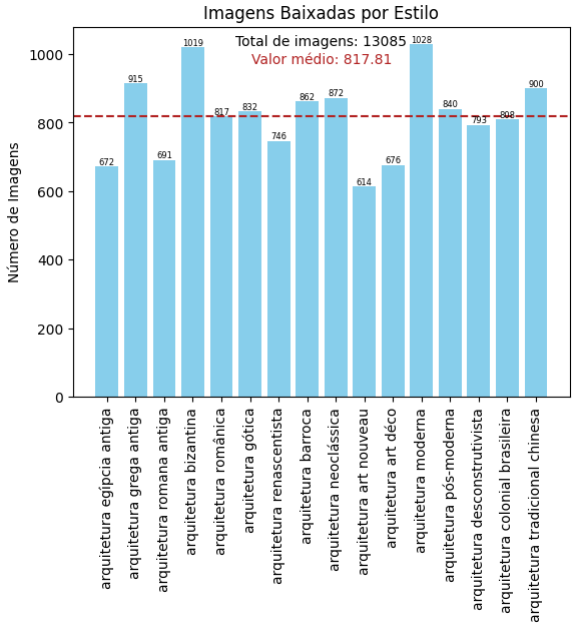
Fonte: Autor (2024).

3.1.1. Coleta e curadoria das imagens

Todas as etapas para a obtenção das imagens foram realizadas no ambiente Google Colab[9], utilizando linguagem de programação Python. Para que a coleta de imagens ocorresse de maneira automatizada, foi adotada uma técnica de raspagem de dados da internet (*web scraping)*, utilizando a biblioteca Bing Image Downloader. Cada estilo foi buscado em português, inglês e mais um idioma de sua origem, com a intenção de ampliar a quantidade de amostras. O desempenho do algoritmo foi excelente, tendo em vista que foram obtidas 13085 imagens de todos os estilos com uma média de 817,81 para cada um deles (Figura 17).

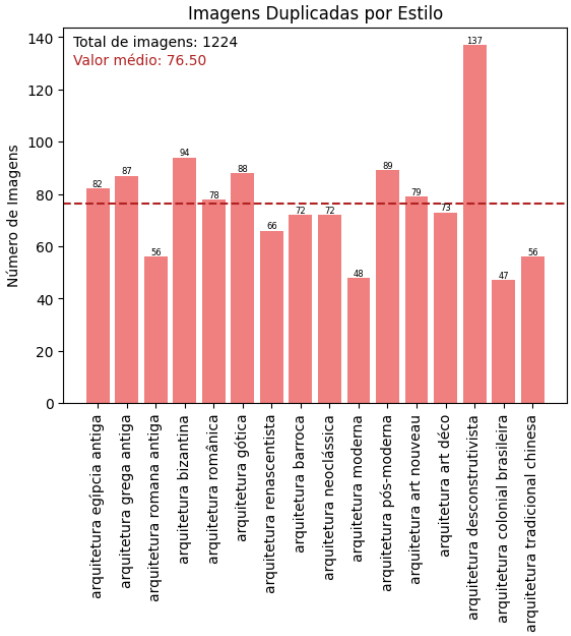
Como era de se esperar, foram baixadas imagens que não estavam de acordo com o interesse da pesquisa e duplicadas, de modo que, o passo seguinte consistiu em agrupar as imagens em diferentes idiomas em uma única pasta e selecionar as imagens repetidas. Para tal, foram utilizadas as bibliotecas OS, Imagehash, PIL e Shutil, cujo código cria uma cadeira de números em cada pixel (*hash*), que a certo grau de compatibilidade mantém apenas uma das imagens e move as outras para uma pasta diferente. Tal técnica colaborou com a redução do tempo de curadoria humana, tendo em vista que identificou 1224 imagens idênticas com uma média de 76,50 para cada estilo (Figura 18).

Figura 17 - Imagens baixadas Por Estilo



Fonte: Autor (2024).

Figura 18 - Imagens Duplicadas Por Estilo

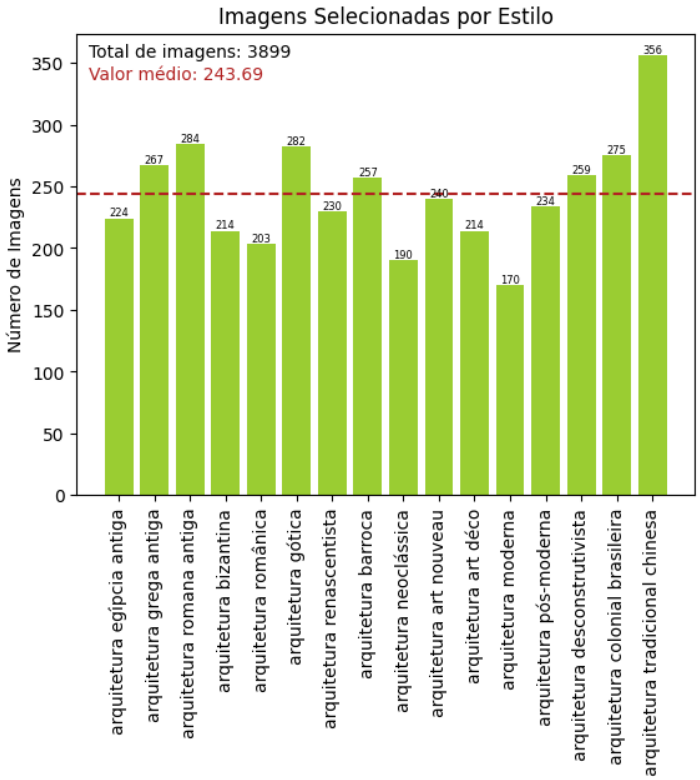


Fonte: Autor (2024).

[9] Para mais informações, acesse: https://colab.research.google.com/

Em seguida, foi necessária uma análise técnica e detalhadas em cada estilo arquitetônico, para selecionar obras que fossem congruentes com o estilo ao qual foram designadas e em segunda instância avaliar as imagens possuíam boa resolução e não constavam marca d'água ou quaisquer outros símbolos ou descrições que pudessem prejudicar o uso da visão computacional em sua classificação. Por fim, o *dataset* resultou em 3899 imagens, com um valor médio de 243,69 por estilo que variaram de 356 (arquitetura tradicional chinesa) a 170 (arquitetura moderna), configurando uma amostra desbalanceada (Figura 19).

Figura 19 - Imagens Selecionadas Por Estilo



Fonte: Autor (2024).

3.2 Métricas de Desempenho: Precision, Recall e F1-Score

O campo do aprendizado de máquina, possui como objetivo realizar previsões, também conhecidas como classificação que podem ser divididas em binária (quando possui apenas duas respostas possíveis) ou multiclasses (quando possui 1, 2, . . . N respostas possíveis). O algoritmo estima a probabilidade de que cada rótulo seja o verdadeiro, atribuindo para todos os dados aquela classe que recebeu maior valor dentre as outras. As métricas de desempenho são úteis para avaliar o desempenho de modelos classificadores e analisar o impacto gerado por diferentes parâmetros no comportamento de uma rede neural (GRANDINI; BAGLI; VISANI,

2020). No contexto dessa pesquisa foram utilizadas como métricas de desempenho: *Precision* (precisão), *recall* (revocação) e *F1-Score*.

A precisão calcula a quantidade de dados classificados como Verdadeiros Positivos (*True Positive* ou TP), que são elementos preditos como positivos e realmente são (pode ser entendido como os acertos de classificação) e os divide pela soma entre TPs e Falsos Positivos (*False Positive* ou FP), que foram as instâncias classificadas como verdadeiras de forma equivocada (pode ser entendido como erros de classificação). Uma maneira mais palpável de interpretar a precisão é entendendo que ela reflete a capacidade de estimar o grau de confiança que a resposta de determinado modelo terá como corretos (GRANDINI; BAGLI; VISANI, 2020) (Equação 4). No entanto, apenas essa métrica não é suficiente compreender de fato se os resultados alcançados são realmente satisfatórios pois para se obter uma precisão em tese perfeita (de 100%), bastaria apenas uma previsão correta, podendo o classificador se abster de classificar demais instâncias. Para reduzir a possibilidade de análises incorretas é importante que seja avaliada em conjunto com o recall (GÉRON, 2021).

Equação 4 - Cálculo da Precisão (*Precision*)

$$Precisão = \frac{TP}{TP + FP}$$

Fonte: Grandini, Bagli, Visani (2020).

Para calcular a proporção de instâncias positivas classificadas de forma correta, é necessário utilizar a métrica revocação (GÉRON, 2021). Possui a capacidade de estimar todos os elementos positivos no conjunto de dados. Seu cálculo é feito pelo número de Verdadeiros Positivos, dividido pela soma entre os Verdadeiros Positivos e os Falsos Negativos (*False Negative* ou FN), que são elementos classificados como negativos, mas que são positivos e juntos computam o total de instâncias rotuladas como positivas (GRANDINI; BAGLI; VISANI, 2020) (Equação 5).

Equação 5 - Cálculo da Revocação (*Recall*)

$$Revocação = \frac{TP}{TP + FN}$$

Fonte: Grandini, Bagli, Visani (2020).

Uma forma de avaliar paralelamente o modelo em relação a Precisão e ao *Recall* é por meio da métrica F1-Score, que nada mais é do que a média harmônica de ambos, indicando

melhor valor igual 1 e pior em 0. Esse cálculo busca ponderar da melhor forma de compensar os índices e oferece pontuação mais alta a classes menores e com métricas desbalanceadas (Equação 6). Pode-se ilustrar a partir do seguinte raciocínio: na média convencional um Modelo A que alcançou Precisão de 100% e *Recall* de 20% (mais próximo a 0) possuiria o a mesma pontuação do modelo B, que marcou 60% em ambos. No entanto, a média ponderada faz com que o modelo B seja considerado melhor em relação ao A, devido ao desequilíbrio constatado (GÉRON, 2021; GRANDINI; BAGLI; VISANI, 2020).

Equação 6 - Cálculo da F1-Score

$$F1 - Score = \left(\frac{2}{precision^{-1} + recall^{-1}}\right) = 2 \times \left(\frac{precision \times recall}{precision + recall}\right)$$

Fonte: Grandini, Bagli, Visani (2020).

3.3 Matriz de Confusão

Desenvolver uma matriz de confusão (*confusion matrix*) é uma maneira de interessante de analisar o desempenho de um classificador com organização visual. Sua elaboração consiste em comparar o conjunto de previsões do conjunto de testes com as respostas reais, onde cada linha representa uma classe verdadeira e cada coluna uma classe prevista, seguindo a mesma ordem (GÉRON, 2021). Os itens classificados como verdadeiros positivos (TP) ficam localizados na diagonal do central da matriz do canto superior esquerdo ao canto inferior direito e o cruzamento entre linha e coluna indica que o modelo acertou (GRANDINI; BAGLI; VISANI, 2020) (Figura 20).

Figura 20 - Exemplo de Matriz de Confusão

		Previsão			
	Classes	a	b	c	d
Verdadeiro	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

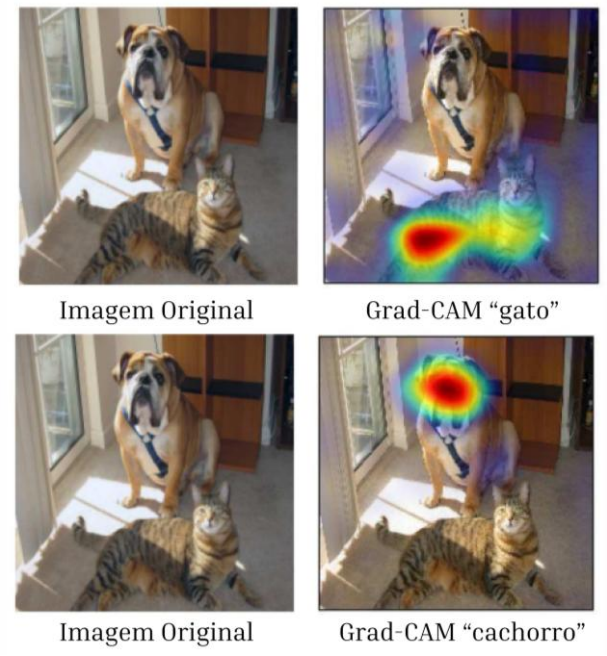
Fonte: Adaptado pelo autor de Grandini, Bagli, Visani (2020).

3.4 Grad-CAM

A massiva utilização de Redes Neurais Convolucionais (CNNs), trouxe consigo o desafio em compreender como os modelos atuam ao capturar as características mais relevantes na classificação correta de uma imagem. Isso ocorre devido ao alto grau de abstração trazido pelo maior número de camadas conectadas que se por um lado melhoraram os resultados obtidos, por outro sacrificaram a sua interpretabilidade. Por conta disso, Sealvaraju *et al*. (2016) propuseram uma maneira mais transparente de visualizar regiões mais importantes (que reúnam características mais relevantes para a classificação) em conjunto de imagens, permitindo uma interpretação visual assertiva (Figura 12).

Em sua metodologia, utilizaram o Mapeamento de Ativação de Classe Ponderada por Gradiente (Grad-CAM), onde são utilizadas informações de gradiente específicas da classe indicada para localizar regiões mais relevantes sem alterações na arquitetura da rede. Com isso, é possível visualizar um mapa de calor (onde vermelho intenso representa maior presença de características), que torna capaz até mesmo a comparação por pessoas leigas de desempenho entre duas redes CNNs que tenham acertado uma classificação, qual delas foi mais certeira (SEALVARAJU *et al*., 2016, 2017).

Figura 21- Utilização do Grad-CAM



Fonte: Adaptado pelo autor de Sealvaraju (2016, 2017).

3.5 Metodologia

Para que pudesse ser escolhida a Rede Neural Profunda que serviria como modelo para o desenvolvimento da pesquisa, foi realizado um *baseline* para extração de características por meio das redes Resnet 50, Inception V3 e Efficient Net, com modelos pré-treinados utilizando a Interface de Programação de Aplicação (API) Keras do Tensor Flow, devido sua interface simples e consistente otimizada, poucas restrições de configuração, que possibilitam desenvolvimento de elementos personalizados[10]. Os resultados foram coletados na penúltima camada de predição das respectivas redes, sendo submetidos na sequência ao classificador SVM.

Uma Máquina de Vetores de Suporte (*Support Vector Machine*, ou simplesmente SVM), é um dos modelos de aprendizado de máquina mais populares para atividades de classificação, adequado para classificar conjunto de dados complexos, de pequeno ou médio porte (GÉRON, 2021). O Algoritmo objetiva realizar uma separação linear entre as classes, com a maximização da distância da linha até as instâncias das categorias, de modo que, os vetores de suporte são os pontos extemos do hiperplano divisor (HARRISON, 2019). Essas características balizaram a escolha pelo classificador para essa pesquisa, tendo em vista que o conjunto de imagens coletado possui uma dimensão compatível com o bom desempenho do algoritmo.

Em uma segunda etapa, foi adotada uma técnica de aprendizado por transferência (*transfer learning*) por meio da rede neural profunda que obteve o melhor desempenho na anteriormente, juntamente com um *fine-tunning* que se trata da modificação na penúltima camada (densa ou *fully connected*) da arquitetura original da rede, cuja função é extrair características das imagens e que passam a possibilitar ajustes de hiperparâmetros e por consequência novos experimentos que visam aprimorar o desempenho de classificação e avaliar o custo computacional x benefícios de processamentos mais robustos, diferentes otimizadores ou a modificação da quantidade de épocas, taxa de aprendizado e aplicação do aumento de dados (*data aumengtation*) visando que a classificação possa obter resultados de testes tão satisfatórios ou até mesmo melhores em relação ao apresentado no *baseline*, com a precaução que não haja ajustes excessivos aos dados (*overfitting*) ou pouca convergência (*underfitting*).

[10] Para mais informações, acesse: https://www.tensorflow.org/guide/keras?hl=pt-br

4 EXPERIMENTOS REALIZADOS

4.1 Extração de Características

A extração de características é uma técnica pela qual é possível comparar o desempenho de um determinado conjunto de dados em redes neurais que já possuem reconhecido destaque para realizar a tarefa pretendida, que neste caso é a classificação das imagens de diferentes estilos arquitetônicos.

Para este experimento foram escolhidas as redes neurais pré-treinadas Resnet 50, Inception V3 e Efficient Net. No ambiente Google Colab, o primeiro passo foi o carregamento do conjunto de imagens diretamente do diretório em nuvem (Google Drive), garantindo que todas possuam três canais de cores (RGB) e redimensionando-as para o formato de entrada adequado para as redes trabalhadas, seguido de sua conversão para uma *array* NumPy.

O passo seguinte consistiu na separação desses dados em treinamento (80% - 3119 imagens) e testes (20% - 779 imagens). Finalmente os modelos pré-treinados escolhidos foram usados para extrair características resultantes de cada imagem do conjunto de treinamento e posterior uso do classificador SVM aplicado ao conjunto de testes para verificação e comparação do desempenho obtido através das métricas escolhidas.

Dado o desafio de escolher apenas um modelo que serviria como *baseline* para o desenvolvimento da etapa seguinte, foi selecionada a rede Efficient Net, devido ao seu desempenho superior que alcançou aproximadamente 80% de aproveitamento em todas as métricas. Os resultados também apontaram para números satisfatórios obtidos pela rede ResNet 50 que margearam os 77% do modelo. Por fim, a performance demonstrada pela rede Inception V3 esteve destacada inferior as demais trabalhadas, alcançando menos de 30% nos índices para as métricas adotadas (Tabela 1).

Tabela 1 – Resultados Obtidos Para a Extração de Características

Modelo Pré-Treinado	Precision	Recall	F1 Score
ResNet 50	0.7732	0.7757	0.7734
Inception V3	0.3035	0.3093	0.3026
Efficient Net	0.8035	0.8052	0.7996

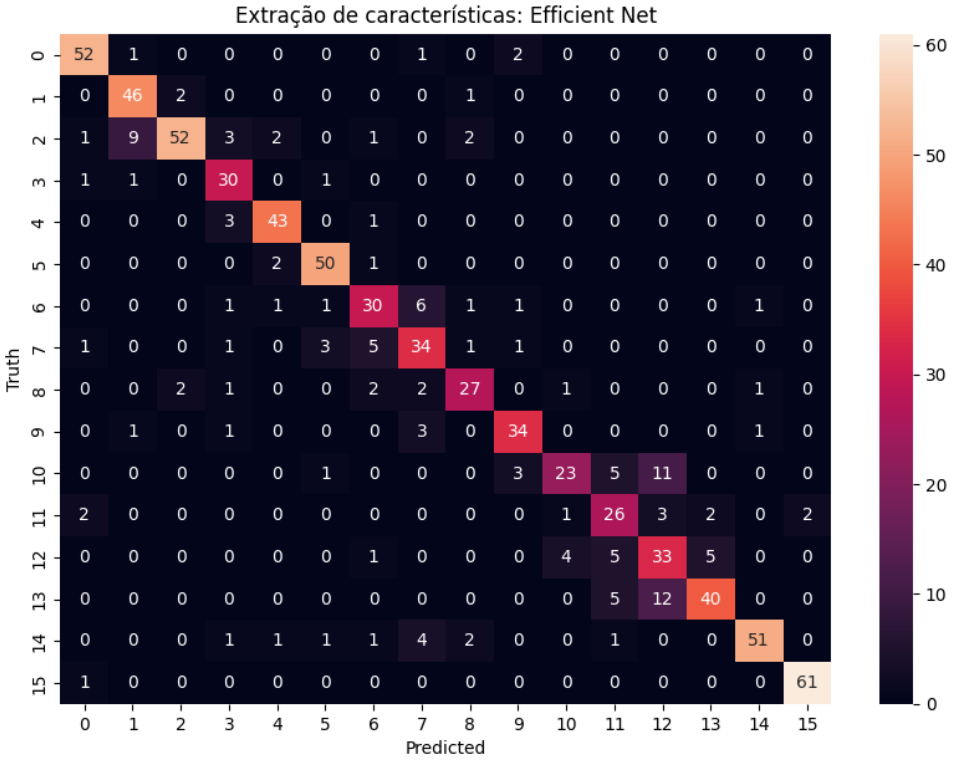
Fonte: Autor (2024).

Para compreender com maior detalhamento como o modelo performou para cada classe, foi gerada a matriz de confusão dessa extração de características (Figura 22). As 780 amostras de testes foram distribuídas proporcionalmente e os melhores desempenhos de acertos ficaram

por conta da arquitetura tradicional chinesa [15][11]: 98.39%, arquitetura gótica [5]: 94.34%, arquitetura grega antiga [1]: 93.88%. Isso quer dizer que esses estilos possuem características próprias que em pouquíssimas vezes foi confundido com os demais apresentados.

Por outro lado, dentre os piores desempenhos apresentados pode-se destacar a arquitetura desconstrutivista [13]: 70.18%, cujos erros de classificação foram com arquitetura moderna (5)[12], e pós-moderna (12). Em sequência, a arquitetura pós-moderna [12] com 68.75%, sendo classificados com equívoco arquitetura renascentista (1), *art déco* (4), moderna (5) e desconstrutivista (5). O estilo que obteve rendimento inferior foi o *art déco* [10], obtendo 53.49% de acertos e sendo confundida com arquitetura gótica (1), *art nouveau* (3), moderna (5) e pós-moderna (11).

Figura 22 - Matriz de Confusão Extração de Características: Efficient Net



Fonte: Autor (2024).

4.2 Transferência de Aprendizado com Fine-Tuning

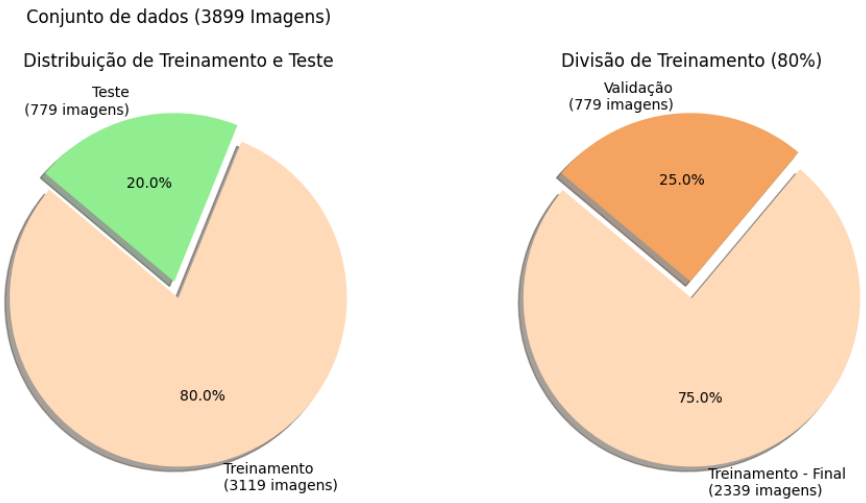
Conforme planejado, após identificar que a rede Efficient Net apresentou melhor desempenho no baseline, chegou o momento de utiliza-la juntamente com a técnica de *fine-tuning*, que possibilita experimentos a partir da exploração de hiperparâmetros associado ao

[11] Interprete [ ] como: Número correspondente a classe na matriz de confusão.
[12] Interprete ( ) como: Número correspondente a quantidade de vezes em que ocorre na matriz de confusão.

processo de aumento de dados, em busca de índices superiores aos alcançados anteriormente e computacionalmente viável.

A inicialização dessa etapa é bastante similar com a anterior até o momento da partição do conjunto de imagens que desta vez foi de 80% (3119) para treinamento e 20% para testes (779), considerando que o conjunto de treinamento foi subdividido em treinamento – final (2339) e validação (779) (Figura 23).

Figura 23 - Repartição do Conjunto de Dados para *Fine-Tuning*



Fonte: Autor (2024).

Em sequência, a rede foi carregada e sua penúltima camada foi alterada para permitir diferentes experimentos com seus hiperparâmetros. A etapa de treinamento do modelo foi iniciada com definição da taxa de aprendizado (*leaning rate*) e otimizador, seguida da codificação dos rótulos inteiros Ytrain e Yval em vetores *one-hot* que possibilitam a classificação em 16 classes e definição da métrica F1 Score, que precisou ser personalizada já que não é nativa do keras.

O compilador utilizou a função de perda *categorical_crossentropy*, indicada para problemas que envolvam mais de duas classes e atua medindo a diferença entre a classificação feita pelo modelo e o rótulo real da imagem. Também são adicionados o otimizador instanciado acima e as métricas de desempenho Precisão, Recall e F1-Score.

Em sequência, foi implementado o *callback EarlyStopping* para monitorar a perda de validação (*val_loss*) e interromper o treinamento não haja melhoria no desempenho após 10 épocas consecutivas, uma técnica que ajuda a combater a possibilidade de *overfitting* e reduzir

o tempo de treinamento. A taxa de aprendizado foi configurada para se manter constante nas 10 primeiras épocas e ser ajustada diminuindo exponencialmente durante pelo *callback learning_rate_scheduler*, colaborando para melhorar a eficiência e convergência do modelo. Também, foi necessário determinar o número de épocas (*epochs*) que ocorre uma iteração completa da rede com cada exemplo do conjunto de treinamento. Vale ressaltar a escolha pela não utilização de *mini-batchs*, tendo em vista que o conjunto original ser reduzido e o potencial de processamento de todo o conjunto na plataforma Google Colab ter apresentado bom desempenho.

4.2.1. Experimentos realizados

Ao todo, foram realizados 07 experimentos com diferentes configurações de número de épocas, troca de otimizadores, mudança na taxa de aprendizado e implementação do aumento de dados. Incialmente, para as experiências E01 e E02, foram adotados padrões comumente usados nesse tipo de pesquisa, apenas variando o número de épocas de 30 para 50, com otimizador Adam, taxa de aprendizado 0.001 e ainda sem o aumento de dados.

Adiante, na tentativa E03, houve uma alteração na taxa de aprendizado para 0.005 como uma forma de identificar quais impactos positivos e/ou negativos seriam perceptíveis no modelo. Logo após, nos ensaios E04 e E05, a solução pretendia foi alterar o otimizador para SGD (*Stochastic Gradient Descent*) e número de épocas que variou entre 50 e 100, buscando visualizar se os resultados obtidos seriam superiores aos alcançados anteriormente.

Para finalizar, nos experimentos E06 e E07 foi adotada a técnica de aumento de dados, no primeiros as configurações utilizadas foram as mesmas no teste E05 e para o último a opção foi de retornar aos parâmetros utilizados em E02 (Tabela 2).

Tabela 2 - Parâmetros Utilizados nos Experimentos

Experimento	Épocas	Otimizador	Taxa de aprendizado	Aumento de dados
E01	30	Adam	0.001	Não
E02	50	Adam	0.001	Não
E03	50	Adam	0.005	Não
E04	50	SGD	0.001	Não
E05	100	SGD	0.001	Não
E06	100	SGD	0.001	Sim
E07	50	Adam	0.001	Sim

Fonte: Autor (2024).

4.2.2. Resultado dos Experimentos

Os resultados obtidos apontaram para melhor desempenho do experimento E02, que se mantiveram entre 81 e 84%, com pequena vantagem em relação ao E01, cuja mudança em termos de parâmetros foi apenas o número de épocas. Em seguida, o teste E07 também apresentou métricas desbalanceadas, que resultaram em um F1-Score na casa dos 80%, dando indícios que a implementação do aumento de dados não surtiu o efeito desejado.

O ensaio E03, foi o último a apresentar um desempenho aceitável, mesmo sendo inferior aos demais, tendo em vista a mudança em sua taxa de aprendizado, com métricas que variaram de 76 a 74%, mas ainda apresentando um certo equilíbrio entre Precisão e *Recall*. Os demais resultados (E04, E05 e E06) foram destacadamente inferiores, muito por conta da mudança no otimizador de Adam para SGD, a alta precisão combinada com o baixo *Recall*, indica que o modelo está com alto índice de acertos, mas deixando de classificar boa parte dos exemplos apresentados, grau de indecisão que prejudica a generalização de modo geral, não sendo interessante para o desafio proposto na pesquisa (Tabela 3).

Tabela 3 - Resultados Obtidos Para os Experimentos Realizados com *Fine-Tuning*

Experimento	Precision	Recall	F1 Score
E01	0.8389	0.8109	0.8246
E02	0.8401	0.8160	0.8279
E03	0.7655	0.7453	0.7553
E04	0.9704	0.0403	0.0773
E05	1.0000	0.0417	0.0800
E06	1.0000	0.0277	0.0540
E07	0.8099	0.7919	0.8008

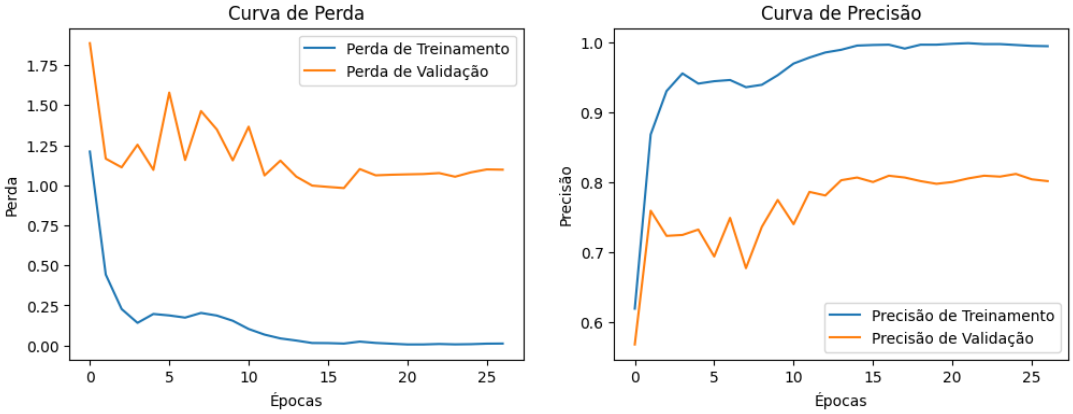
Fonte: Autor (2024).

O gráfico da curva de perda (*loss curve*) apresenta a perda de treinamento em um movimento de declínio que aponta para bom aprendizado do modelo. Com relação a perda de validação, foi possível verificar uma inconstância na curva até aproximadamente a 12ª época, quando finalmente os pesos foram ajustados e a partir daí há redução constante dessa curva até que o modelo não apresente mais grandes evoluções na época 27.

Além disso, também foi gerado um gráfico com a curva de precisão, cuja etapa de treinamento apresentou uma elevação constante, como é recomendado. Ao analisar a precisão, novamente há instabilidade até a 12ª época, para daí em diante finalmente ascender e

demonstrar o potencial de aprendizado do modelo. Por meio desses gráficos foi possível visualizar o trajeto de aprendizado do modelo, bem como a atualização de seus pesos em busca de um melhor desempenho (Figura 24).

Figura 24 - Curvas de Perda e Precisão do Experimento 02



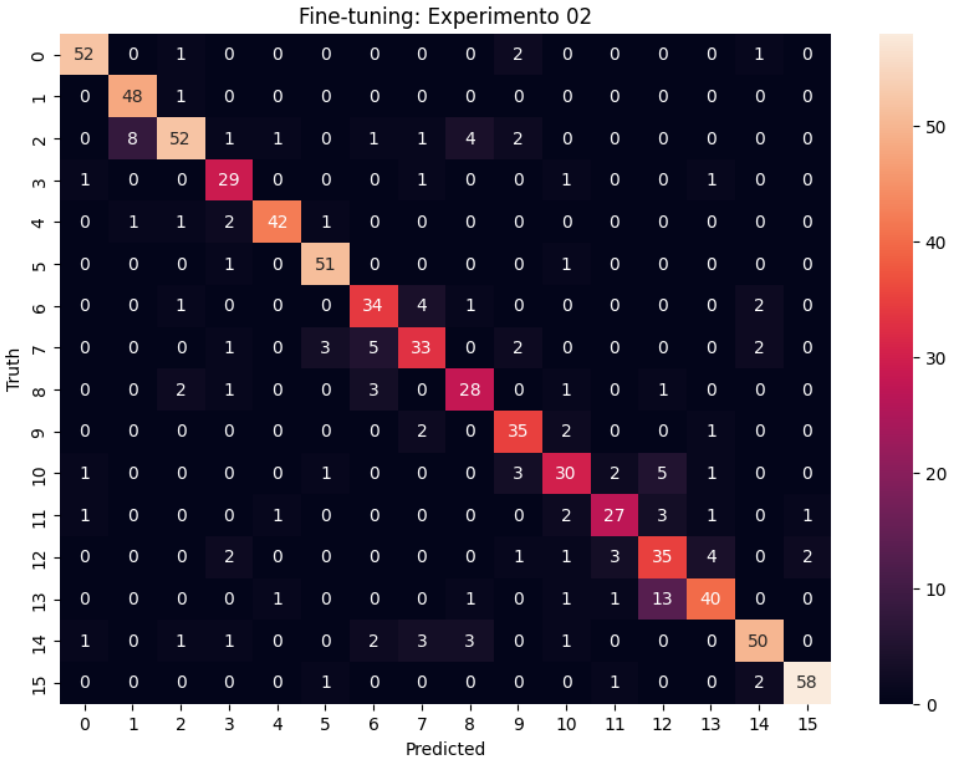
Fonte: Autor (2024).

A matriz de confusão gerada por esse experimento elucidou possíveis dúvidas quanto ao rendimento do modelo relacionado a cada uma das classes (Figura 25). Os três estilos com maior percentual de acerto foram: arquitetura grega antiga [1][13] que alcançou 97.96%, seguida da arquitetura gótica [5] com 96.23% e arquitetura tradicional chinesa [1] que chegou a 93.55%. A leitura que se pode ter para as classes com índices tão altos é que o modelo está conseguindo identificar com primor as suas principais características e distingui-las dos demais estilos.

Por outro lado, nem todas as classes obtiveram considerável êxito, que indica dificuldade na classificação correta. Esse foi o caso da arquitetura barroca [7], com percentual na casa dos 71.74%, que por sua vez confundiu itens com arquitetura bizantina (1), gótica (3), renascimento (5), *art déco* (2) e colonial brasileira (2). Também, com a arquitetura desconstrutivista [13], com índice 70.18% e previsões incorretas para os estilos arquitetônicos românica (1), neoclássica (1), *art déco* (1), moderna (1) e destacadamente pós moderna (13). Ademais, o menor rendimento ficou por conta do estilo *art déco* [10], que com marcou 69.77%, sendo confundida com arquitetura egípcia antiga (1), gótica (1), *art nouveau* (3), moderna (2), pós-moderna (5) e desconstrutivista (1).

[13] Interprete [ ] como: Número correspondente a classe na matriz de confusão.

Figura 25 - Matriz de Confusão Fine-Tuning: Experimento 02



Fonte: Autor (2024).

4.3 Conclusões dos Resultados

Transcorrido o percurso experimental da pesquisa, foi possível observar que com relação ao processo de extração de características utilizando uma rede neural pré-treinada foi bastante interessante na perspectiva do *baseline* e definição de qual seria o modelo a ter melhor aproveitamento e consequentemente seguir para a próxima etapa de investigação.

Além disso, proporcionou uma primeira ideia da capacidade de treinamento e testes utilizando o conjunto de dados desenvolvido para este estudo. Como mencionado, o melhor resultado obtido foi por meio da rede Efficient Net e classificador SVM, com métricas que marcaram em média 80% de aproveitamento, que possibilitou uma avaliação positiva, tanto em relação ao método quanto ao conjunto de imagens.

O prosseguimento da investigação foi por meio da técnica de transferência de aprendizado com *fine-tuning*, que propiciou uma série de experimentos com diferentes parâmetros até que fosse possível alcançar índices melhores do que os apresentados anteriormente e que ainda assim fossem computacionalmente viáveis. Dentre os 7 testes realizados o melhor dentre eles foi o E02, que teve como configuração 50 épocas (com ativação do *early stopping* na 27ª), otimizador Adam, *learning rate* 0.001 e sem aumento de dados. Os

resultados apontaram para métricas consistentes que variaram de 81 a 84%, valores superiores em relação ao obtido no experimento feito simplesmente utilizando redes pré-treinadas.

Uma comparação global no valor obtido pelo F1-Score entre todos os testes realizados aponta que os experimentos E02, E01 e E07 (que contou com aumento de dados) figuraram nas primeiras colocações, seguidos finalmente pela rede Efficient Net com classificador SVM, que conseguiram manter um bom equilíbrio entre as métricas apresentas. Na 5ª colocação ficou a rede ResNet 50 com SVM, que apresentou índices pouco melhores que o E03. Os demais experimentos não foram considerados exitosos, seja pelo baixo rendimento no conjunto de dados (caso da rede Inception V3) ou pela alta precisão desbalanceada em relação ao recall que demonstra pouca capacidade de generalização do modelo (Tabela 4).

Tabela 4 - Quadro Síntese dos Resultados Obtidos

Experimento	Precision	Recall	F1 Score
E02	0.8401	0.8160	0.8279
E01	0.8389	0.8109	0.8246
E07	0.8099	0.7919	0.8008
Efficient Net	0.8035	0.8052	0.7996
ResNet 50	0.7732	0.7757	0.7734
E03	0.7655	0.7453	0.7553
Inception V3	0.3035	0.3093	0.3026
E06	1.0000	0.0277	0.0540
E04	0.9704	0.0403	0.0773
E05	1.0000	0.0417	0.0800

Fonte: Autor (2024).

Além disso, as análises geradas tendo como base as matrizes de confusão serviram inicialmente para ratificar o melhor desempenho do experimento 02 e vantagem da técnica que utilizou transferência de aprendizado por *fine-tuning* por meio da comparação com o melhor rendimento obtido através da extração de características e classificador SVM, com a rede Efficient Net. Foi possível constatar que em 9 dos 16 estilos arquitetônicos estudados o percentual de acertos em E02 foi maior, enquanto em 4 oportunidades foi inferior e por 3 vezes ficaram empatados.

Em suma, ambos modelos apresentaram um equilíbrio no percentual de classificações corretas, em 14 das classes trabalhadas a diferença ficou abaixo de 5%, no entanto, ao se tratar da arquitetura renascentista (06), esse número subiu para 9.52% e para *art déco* alcançou

16.25%, que demonstram uma boa evolução na classificação desses estilos que foi proporcionada por meio do experimento 02 (Tabela 5).

Tabela 5 - Comparação Entre os Melhores Resultados Obtidos Por Classe

Classe	Efficient Net	Experimento 02	Diferença
00	92.86%	92.86%	0.00%
01	93.88%	97.96%	4.08%
02	74.29%	74.29%	0.00%
03	90.91%	87.88%	3.03%
04	91.49%	89.36%	2.13%
05	94.34%	96.23%	1.89%
06	71.43%	80.95%	9.52%
07	73.91%	71.74%	2.17%
08	75.00%	77.78%	2.78%
09	85.00%	87.50%	2.50%
10	53.49%	69.77%	16.28%
11	72.22%	75.00%	2.78%
12	68.75%	72.92%	4.17%
13	70.18%	70.18%	0.00%
14	82.26%	80.65%	1.61%
15	98.39%	93.55%	4.84%

Fonte: Autor (2024).

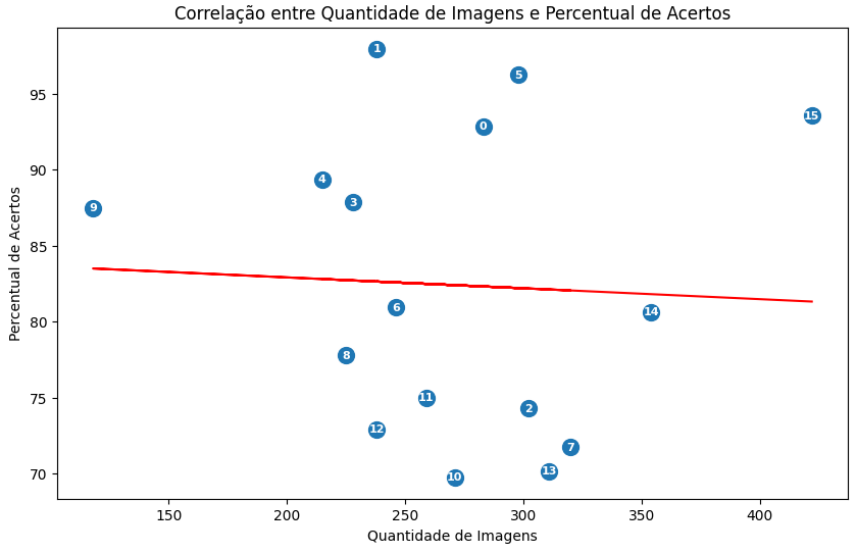
Seguindo as investigações, uma dúvida que surgiu em relação ao aproveitamento do experimento 02, foi se a quantidade de imagens do conjunto de dados estaria diretamente relacionada com a porcentagem de acertos. O questionamento ecoou muito por conta do desbalanceamento entre as imagens de cada estilo arquitetônico. O gráfico de correlação linear descartou essa possibilidade apresentando pontos dispersos sem relevante correlação, fato positivo para a pesquisa, pois demonstra o potencial de generalização do modelo (Figura 26).

Além disso, uma outra correspondência percebida foi de movimentos que buscaram em inspiração em estilos da era antiga e se apropriaram de parte dos seus elementos, como foi o caso da arquitetura neoclássica [08] que buscou inspiração nas obras do antigo Império Romano [02] e o interessante caso da arquitetura colonial brasileira [14], que devido ao seu histórico de ocupação com enorme influência da cultura portuguesa, esse estilo apresentou correlações com

a arquitetura barroca [07] (movimento que vigorava na época da construção dos primeiros centros urbanos no Brasil) e renascentista [06], que foram gerados na Europa.

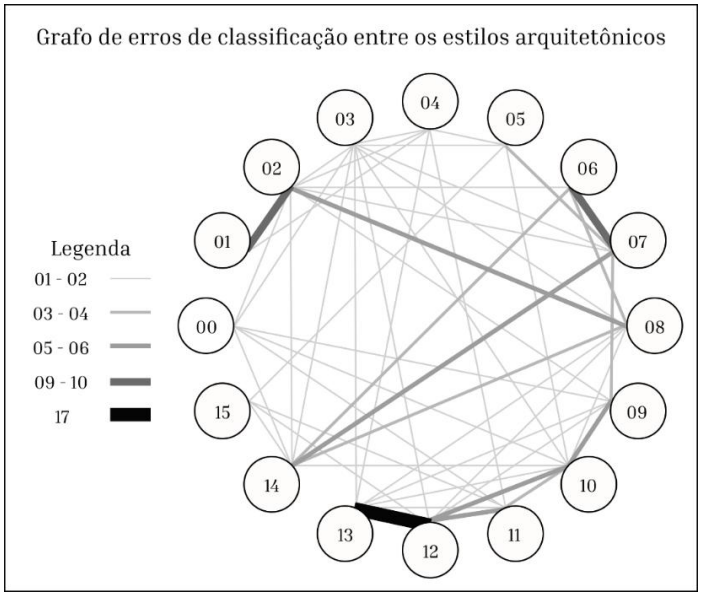
Por fim, outro ponto de vista a ser observado é quais dentre essas classes possuem maiores níveis de legibilidade, ou seja, sendo pouquíssimas vezes confundido com os outros estilos, que em primeiro plano foi a arquitetura tradicional chinesa, com seus telhados imponentes, seguida pela arquitetura egípcia, românica e gótica (Figura 27).

Figura 26 - Correlação Entre Quantidade de Imagens e Percentual de Acertos



Fonte: Autor (2024).

Figura 27 - Grafo de Erros de Classificação Entre os Estilos Arquitetônicos.



Fonte: Autor (2024).

O algoritmo Grad-CAM foi escolhido para compor a análise de desempenho do modelo consagrado como de melhor desempenho (E02), pela sua capacidade de criar mapas de calor nas imagens das edificações onde a rede neural pode aprender características relevantes de cada um dos 16 estilos arquitetônicos, cujas ocorrência mais demonstrativas foram selecionadas. Em algumas situações, quase que todos os elementos construtivos foram destacados, mostrando um enorme potencial em separar os planos de fundo da construção que era o alvo a ser analisado.

Em outros casos, foi possível perceber que o modelo focou em características que são muito marcantes e representativas daquela classe, como por exemplo alguns telhados, cúpulas ou torres. Ainda, foi possível notar que por vezes surgiram mapas de calor centralizados no meio da imagem que se estendeu por ela inteiramente. Como era de se esperar, também ocorreram casos em que o modelo teve dificuldade em interpretar os elementos ou simplesmente o mapa de calor não destacou elementos arquitetônicos, seja pela distorção trazida no pré-processamento das imagens, presença de elementos naturais que podem indicar a dificuldade de generalização para esses casos, resultando em erros de classificação (Figura 28).

Figura 28 - Algoritmos Grad-CAM Aplicado Aos Estilos Arquitetônicos Selecionados

Fonte: Autor (2024).

5 CONCLUSÕES

A humanidade sempre buscou formas de se expressar através dos séculos de sua existência, a arquitetura é parte disso, pois está diretamente ligada a fatores como contexto social, político, religioso estético e das tecnologias disponíveis de quando foi concebida. Naturalmente, no decorrer do tempo as construções ganharam diferentes formas, materiais e significado, sendo um dos mais interessantes elementos a ser estudados quando se busca entender a cultura de um povo.

Além disso, vale ressaltar que a linha temporal que delimita o início e fim desses estilos arquitetônicos muitas vezes não é linear, fazendo com hajam sobreposições entre eles, bem como elementos que se misturaram em meio aos anos e agora compõem um grande mosaico social. Por conta disso, é bastante comum que pesquisadores da teoria e história da arquitetura e do urbanismo tenham se inclinado em escrever sobre as principais características descritivos das construções de cada época, até que um robusto acervo fosse alcançado.

O presente contexto de evolução das tecnologias de aprendizado de máquina, redes neurais e visão computacional têm inspirado uma nova geração de pesquisadores a utilizar esses meios para alcançar resultados que sejam ao menos próximas ao alcançado por humanos especialistas na área, por meio da inteligência artificial.

De posse de todo esse contexto, a presente pesquisa se propôs a investigar a classificação de imagens de estilos arquitetônicos usando aprendizado profundo. De fato, houve êxito no objetivo geral, tendo em vista que foi possível obter resultados de classificação cujos melhores desempenhos alcançaram aproveitamentos maiores que 80% na classificação de imagens.

Para que isso fosse possível, inicialmente foi realizada uma busca de conjunto de dados (*datasets*), que fossem adequados para esse tipo de investigação, no entanto, foi constatado que não existia a disposição conjuntos de imagens que fossem suficientemente relevantes levando em consideração o contexto de formação arquitetônica, social e cultural do Brasil. Por conta disso, foram escolhidos 16 estilos históricos de construção que serviram como referência para a construção do banco de dados por meio da coleta de imagens da internet utilizando técnica de raspagem de dados (*web scraping*).

Além de obter essas imagens, também foi necessário realizar uma curadoria para selecionar aquelas que por ventura possa ter sido baixada duplicadas ou com alto grau de semelhança, para tal, foi realizado um processamento que transformou o pixel das imagens em números (*hash*), que colaborou para melhorar a qualidade dos dados. Não obstante, foi

necessária a análise humana para identificar possíveis erros de *download* (que não fossem edificações ou que estivessem em estilos arquitetônicos equivocados). Esse processo rendeu um total de 3899 registros distribuídos de maneira desbalanceada entre as classes, sendo um relevante produto elaborado.

A pretensão de aplicar aprendizado através de redes pré-treinadas no conjunto de imagens se mostrou bem sucedida, à medida que o *baseline* comparou 3 modelos robustos e consolidados para processar as imagens e classifica-las por meio do algoritmo SVM. O melhor desempenho obtido por meio da rede Efficient Net serviu de base para o desenvolvimento de outros 7 experimentos por meio de transferência de aprendizado com *fine-tuning*, que obtiveram resultados ainda melhores.

Por se tratar de um conjunto desbalanceado, surgiu a hipótese de que a aplicação da técnica de aumento de dados (*data aumengtation*), pudesse trazer relevantes melhorias de desempenho, mas o que foi observado é que na realidade não houve significativo impacto positivo.

Conforme mencionado, a metodologia utilizada partiu de uma rede de neural profunda pré-treinada base e por meio da modificação de parâmetros da penúltima camada foi observada qual configuração seria a mais adequada em relação ao potencial computacional utilizado e resultado alcançado. Nesse sentido, pode-se dizer que a avaliação foi positiva, pois as técnicas que necessitaram de mais épocas de processamento não foram as que melhor generalizaram os dados, tornando o modelo adequado aos limites fornecidos gratuitamente pela plataforma Google Colab.

Dentre as considerações que podem ser feitas aos demais pesquisadores, ressalto que para se obter resultados ainda melhores e mais consistentes, seria necessária uma força tarefa de pesquisadores e estudantes que pudessem revisar e implementar o conjunto de imagens utilizado, bem como a implementação de novos estilos arquitetônicos e se possível que haja balanceamento entre as classes. Com relação a análise de dados, seria muito interessante uma comparação no desempenho de classificação de estudantes e do modelo de rede neural, já que até mesmo para os humanos há dificuldade em reconhecer com precisão transição entre alguns períodos e suas principais características e correlações existentes entre eles. No mais, futuramente eu entendo que esforço não seja apenas para classificar as imagens como um todos, mas que será de segmentar os elementos das imagens de edificações histórias e quem sabe sirvam de base para modelagem da informação de construção (BIM).

Por fim, ressalto que toda essa pesquisa não teve como intuito enaltecer modelos inteligência artificial em detrimento dos pesquisadores que por meio da teoria e crítica

arquitetônica construíram as bases para o conhecimento pregresso das edificações, muito pelo contrário, se até mesmo para a máquina essa tarefa é consideravelmente difícil pois existem contextos que somente anos de análises sociais podem responder, há de se enaltecer as excelentes contribuições deixadas por cada um deles. Talvez em um futuro (distante ou não), pode ser que existam modelos que sejam capazes de descrever perfeitamente estilos arquitetônicos e classifica-los, mas isso não seria possível sem que se estivéssemos apoiados em ombros de gigantes, como diria Isaac Newton.

REFERÊNCIAS

AGARWAL, S. et al. Building Rome in a day. **Communications of the ACM**, v. 54, n. 10, p. 105-112, 2011.

BERG, A.C.; GRABLER, F.; MALIK, J. Parsing images of architectural scenes. In: **IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 11., 2007.** Proceedings […]. New York: IEEE, 2007. p. 1-8.

BRANDÃO, C.A.L. **A formação do homem moderno vista através da arquitetura**. Editora UFMG, 1999.

BROWN, C.M. Computer vision and natural constraints. **Science**, v. 224, n. 4655, p. 1299-1305, 1984.

CHING, F.D.K.; ECKLER, J.F. Introdução à arquitetura. Porto Alegre - RS: Grupo A, 2013. E-book. ISBN 9788582601020. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788582601020/. Acesso em: 11 jan. 2024.

CHING, F.D.K.; JARZOMBEK, M.; PRAKASH, V. **História global da arquitetura**. Porto Alegre - RS: Grupo A, 2019. E-book. ISBN 9788582605127. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788582605127/. Acesso em: 11 jan. 2024.

DARBANDY, M. T.; ZOJAJI, B.; SANI, F.A. Iranian architectural styles recognition using image processing and deep learning. In: **INTERNATIONAL CONFERENCE ON THE DYNAMICS OF INFORMATION SYSTEMS, 2023, Cham.** Proceedings […]. Cham: Springer Nature Switzerland, 2023. p. 69-82.

DENG, J. et al. ImageNet: a large-scale hierarchical image database. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2009.** Proceedings […]. New York: IEEE, 2009. p. 248-255.

GOEL, A.; JUNEJA, M.; JAWAHAR, C. V. Are buildings only instances? Exploration in architectural style categories. In: **INDIAN CONFERENCE ON COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, 8., 2012.** Proceedings […]. New York: IEEE, 2012. p. 1-8.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Aprendizado profundo**. Cambridge: MIT Press, 2016. Disponível em: http://www.deeplearningbook.org.

HE, K. et al. Deep residual learning for image recognition. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016.** Proceedings […]. New York: IEEE, 2016. p. 770-778.

HEIDEGGER, M. A origem da obra de arte. Tradução e apresentação de CAMPOS, M.J.R. **Revista Kriterion**. Belo Horizonte, v.XXVII, n.76, p.185-210, 1986.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998.

LLAMAS, J. et al. Classification of architectural heritage images using deep learning techniques. **Applied Sciences**, v. 7, n. 10, p. 992, 2017.

SHARMA, S. et al. Classification of Indian monuments into architectural styles. In: **NATIONAL CONFERENCE ONM COMPUTER VISION, PATTERN RECOGNITION, IMAGE PROCESSING, AND GRAPHICS, 6., 2017, Mandi.** Proceedings […]. Singapore: Springer, 2018. p. 540-549.

SIMONYAN, K.; ZISSERMAN, A. Redes convolucionais muito profundas para reconhecimento de imagens em larga escala. In: **International Conference on Learning Representations (ICLR)**, 2015.

SZEGEDY, C. et al. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition,** p. 1-9, 2015.

WANG, R. et al. Intra-class classification of architectural styles using visualization of CNN. In: **INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND SECURITY, 5., 2019, New York.** Proceedings […]. New York: Springer International Publishing, 2019. p. 205-216.

ZEVI, B. **Saber ver a arquitetura**. 5ª edição. São Paulo: Martins Fontes, 1996.

PONTI, M.A. et al. Training deep networks from zero to hero: avoiding pitfalls and going beyond. In: **SIBGRAPI CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 34., 2021.** Proceedings […]. New York: IEEE, 2021. p. 9-16.

PONTI, M.A. et al. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: **SIBGRAPI CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES TUTORIALS, 30., 2017.** Proceedings […]. New York: IEEE, 2017. p. 17-41.

RAJPURKAR, P. et al**.** AI in health and medicine. **Nature Medicine**, v. 28, n. 1, p. 31-38, 2022.

FUKUSHIMA, K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, n. 4, p. 193-202, 1980.

GÉRON, A. **Mãos à Obra**: Aprendizado de Máquina com Scikit-Learn, Keras & TensorFlow: Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes. Rio de Janeiro: Editora Alta Livros, 2021. E-book. ISBN 9786555208146. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786555208146/. Acesso em: 29 conjuntos. 2024.

GABRIEL, M. **Inteligência Artificial**: Do Zero ao Metaverso. Rio de Janeiro: Atlas, 2022. E-book. ISBN 9786559773336. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786559773336/.

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. **arXiv preprint**, arXiv:2008.05756, 2020.
HARRISON, M. **Machine Learning–Guia de referência rápida**: trabalhando com dados estruturados em Python. Novatec Editora, 2019.

HAYKIN, S. **Redes Neurais**: Princípios e Prática. Porto Alegre: Bookman, 2007. E-book. ISBN 9788577800865. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788577800865/.

HEBB, D. O. **The organization of behavior**: a neuropsychological theory. New York: Psychology Press, 2005.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In**: RUMELHART, D. E.; MCCLELLAND, J. L.** (Eds.). Parallel distributed processing: explorations in the microstructure of cognition. v. 1. Cambridge: MIT Press, 1986. p. 318-362.

KONNO JÚNIOR, J.; MOURA J. D. Inteligência Artificial no reconhecimento facial em Segurança Pública: dados sensíveis e seletividade penal. **Revista Eletrônica Direito & TI**, [S. l.], v. 1, n. 15, p. 61–80, 2023.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. **Advances in Neural Information Processing Systems**, v. 25, 2012.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, p. 115-133, 1943.

MINSKY, M.; PAPERT, S. **Perceptrons**: an introduction to computational geometry. Cambridge: MIT Press, 1969.

MULLER; P., J.; MASSARON, L. **Inteligência Artificial para leigos**. Rio de Janeiro: Editora Alta Livros, 2019. E-book. ISBN 9788550808505. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788550808505/.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint**, arXiv:1511.08458, 2015.

RAUBER, T. W. Redes neurais artificiais. **Universidade Federal do Espírito Santo**, v. 29, 2005.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210-229, 1959.

SANTAELLA, L. **A inteligência artificial é inteligente?**. São Paulo: Edições 70, 2023. E-book. ISBN 9786554270588. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786554270588/.

SELVARAJU, R.R. et al. Grad-CAM: Why did you say that?. arXiv preprint arXiv:1611.07450, 2016.

SUCAR, L. E.; GÓMEZ, G. Visión computacional. **Instituto Nacional de Astrofísica, Óptica y Electrónica**, México, 2011.

TAULLI, T. **Introdução à inteligência artificial. Uma abordagem não técnica**, TEIXEIRA, L.A. (trad.). São Paulo: Novatec, 2020.

WIDROW, B. et al. **Adaptive" adaline" Neuron Using Chemical" memistors."**. 1960.

XU, Z. et al. Architectural style classification using multinomial latent logistic regression. In: **Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13**. Springer International Publishing, 2014. p. 600-615.

SZELISKI, Richard. **Computer vision: algorithms and applications**. Springer Nature, 2022.