

UNIVERSIDADE PRESBITERIANA MACKENZIE
CURSO DE TECNOLOGIA EM CIÊNCIA DE DADOS

CARLOS ANTONIO BATISTA

PEDRO COSTA DIAS

ADMILDO JOSE MARTINHO

PROJETO APLICADO II

SÃO PAULO / RIO DE JANEIRO

2023

SUMÁRIO

1. Introdução.....	3
2. Definição do objetivo do trabalho.....	3
3. Cronograma.....	4
4. Definição do escopo do trabalho	4
5. Definição das responsabilidades dos membros do grupo.....	4
6. Referência de aquisição do dataset.....	5
7. Descrição do dataset	5
8. Projeto	5
9. Definição da linguagem de programação usada no projeto.....	5
10. Análise exploratória da base de dados escolhida.....	6
11. Tratamento da base de dados (Preparação e treinamento).	6
12. Definição e descrição das bases teóricas dos métodos.	6
13. MODELO VGG16	11
14. Definição e descrição de como será calculada a acurácia.	13
15. apresentação do projeto	15
16. Resultado	15
17. Referências	15

1. INTRODUÇÃO

Está na moda falar de inteligência artificial ultimamente. Ligamos a TV, logamos nas redes sociais, abrimos algumas revistas e, inevitavelmente, logo nos encontramos com o termo. Essa nova relevância também não é sem motivos, os últimos anos trouxeram diversas implementações desta área em nosso dia a dia, detecção de fraudes no cartão, recomendação de filmes e livros, reconhecimento de câncer com análises de tomografias, os exemplos são tantos que poderíamos gastar páginas e mais páginas numerando. É por esse motivo que a faculdade presbiteriana Mackenzie nos proporcionou a oportunidade de mais um treino no assunto, que tem a forma desse projeto. Mas afinal, o que é inteligência artificial? (Em tal lugar, ou tal pessoa, colocar onde achou a referência) conceitua da seguinte forma: “Inteligência Artificial (IA) é um ramo da ciência da computação que tem como objetivo criar sistemas que possam realizar tarefas que, normalmente, exigem inteligência humana.”

Como se pode notar, IA tem um conceito abrangente, ela é composta de vários campos e subáreas. Aqui vamos nos ater ao aprendizado de máquina, utilizando técnicas de aprendizado supervisionado.

2. DEFINIÇÃO DO OBJETIVO DO TRABALHO

O objetivo do trabalho é desenvolver um sistema capaz de reconhecer imagens de um banco de dados com alta precisão e eficiência. Escolhemos para esse primeiro projeto imagens de cães e gatos por dois motivos: a simplicidade maior de usar um modelo com apenas duas classes e a quantidade de material disponível. A pesquisa “cats dogs” no Kaggle gera 6.641 resultados sendo 18 de 20 na primeira página projetos semelhantes, achamos sensato utilizar os caminhos de pessoas com mais experiência na área e, sendo esse exercício tão praticado, acreditamos ser benéfico praticá-lo também. Nossa ideia era alcançar o máximo de acurácia possível, mas inicialmente chegamos ao valor de 70% como aceitável.

3. CRONOGRAMA

O cronograma de trabalho inclui as datas de início e término de cada etapa do projeto, bem como as datas de reuniões regulares para acompanhamento do progresso do trabalho e tomada de decisões.

Cronograma do Projeto Aplicado II

Etapas	Atividades	Data	Status
Etapa 1	Momento com o Professor	03/mar	Realizado
	Reunião do Grupo	12/mar	Realizado
	Entrega da Atividade	13/mar	Realizado
Etapa 2	Momento com o Professor	31/mar	Realizado
	Reunião do Grupo	15/abr	Realizado
	Entrega da Atividade	17/abr	Realizado
Etapa 3	Momento com o Professor	28/abr	Realizado
	Reunião do Grupo	13/mai	Realizado
	Entrega da Atividade	15/mai	Realizado
Etapa 4	Momento com o Professor	26/mai	Realizado
	Reunião do Grupo	27/mai	Realizado
	Entrega da Atividade	29/mai	Realizado

4. DEFINIÇÃO DO ESCOPO DO TRABALHO

O escopo do trabalho inclui a seleção e preparação do banco de dados de imagens, a escolha de algoritmos de reconhecimento de imagem adequados, a implementação do sistema de reconhecimento de imagem e a avaliação da performance do sistema.

5. DEFINIÇÃO DAS RESPONSABILIDADES DOS MEMBROS DO GRUPO

Cada membro do grupo terá responsabilidades compartilhadas, tais como: seleção e preparação do banco de dados de imagens, implementação de algoritmos de

reconhecimento de imagem, testes do sistema, documentação, e outras tarefas relacionadas ao projeto.

6. REFERÊNCIA DE AQUISIÇÃO DO DATASET

<https://www.kaggle.com/datasets/samuelcortinhas/cats-and-dogs-image-classification>

Samuel Cortinhas (Dono)

Database Aberta

7. DESCRIÇÃO DO DATASET

- 69.26 MB de arquivos JPG de imagens de cães e gatos.
- 70 imagens de gatos e 70 imagens de cachorros para teste.
- 279 imagens de gatos e 278 imagens de cachorro para treino.

8. PROJETO

O projeto está disponível para consulta no:

<https://github.com/PedroCosDi/ProjApl2Mack.git>

9. DEFINIÇÃO DA LINGUAGEM DE PROGRAMAÇÃO USADA NO PROJETO.

‘os’: Biblioteca para interagir com o sistema operacional.
‘PIL’: Biblioteca para a manipulação e processamento de imagens.
‘matplotlib.pyplot’: Biblioteca para visualização de gráficos e plotagem de dados.
‘tensorflow’: Biblioteca para criação e treinamento de redes neurais.
‘tensorflow.keras’: Submódulo do tensorflow que fornece uma API para construir e treinar modelos de deep learning.
‘ImageDataGenerator’: Classe para gerar lotes de dados de imagens com aumento de dados.

“Input”, “Convo2d”, “MaxPooling2d”, “Flatten”, “Dense”, “Dropout”, “BatchNormalization”: Camadas de rede neural para construir o modelo.
 ‘Model’: Classe para criar e treinar modelos de aprendizado profundo.
 ‘VGG16’: Modelo pré treinado de reconhecimento de imagens.
 ‘ReduceLROnPlateau’: Callback para reduzir a taxa de aprendizado durante o treinamento.

10. ANÁLISE EXPLORATÓRIA DA BASE DE DADOS ESCOLHIDA.

Análise exploratória das imagens de teste:

Tamanho mínimo: 191 x 158 pixels

Tamanho máximo: 4735 x 3157 pixels

Tamanho médio: 886.74 x 612.11 pixels

Tamanho mediano: 759 x 500 pixels

Análise exploratória das imagens de treino:

Tamanho mínimo: 133 x 133 pixels

Tamanho máximo: 4272 x 4272 pixels

Tamanho médio: 952.46 x 657.20 pixels

Tamanho mediano: 800 x 549 pixels

11. TRATAMENTO DA BASE DE DADOS (PREPARAÇÃO E TREINAMENTO).

<https://github.com/PedroCosDi/ProjApl2Mack.git>

12. DEFINIÇÃO E DESCRIÇÃO DAS BASES TEÓRICAS DOS MÉTODOS.

Em relação a definição dos métodos, ainda não conseguimos escolher um. Estamos fazendo testes com modelos de redes neurais convolucionais pré-treinados chamados VGG16, ResNet e InceptionV3, até agora sem muito sucesso. A ideia é de que o uso desses modelos nos permitiria começar a fazer algo com menos conhecimento e ir preenchendo esse déficit com o tempo. Ainda não dominamos a base teórica por trás

dos códigos usados, mas vamos continuar o esforço em relação a isso. O que entendi até agora é o seguinte.

O algoritmo que estamos implementando é uma rede neural convolucional, que difere da rede neural da seguinte forma:

Rede Neural x Rede Convolucional

Uma rede neural é uma estrutura que **conecta pequenas unidades, os neurônios, de forma organizada**. Através desta organização, a combinação das operações unitárias simples realizadas por cada neurônio levará a soluções de problemas complexos.

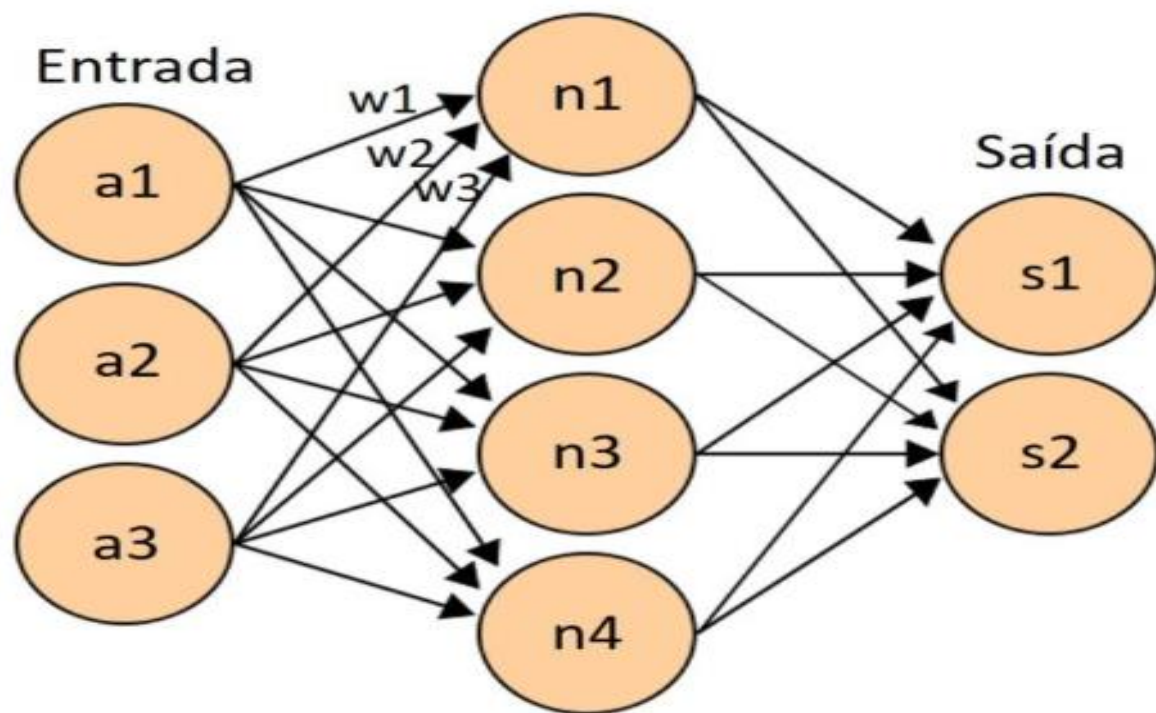


Figura 1 - Rede Neural

Uma rede neural profunda possui muitas camadas ocultas

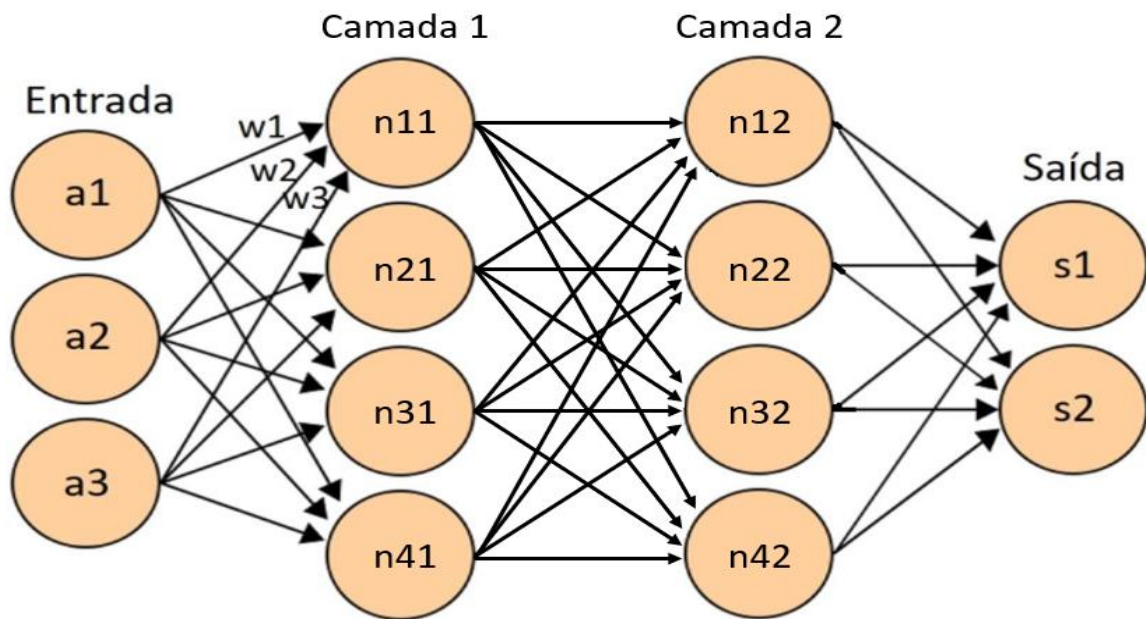


Figura 2 - Rede Neural Profunda

Numa rede neural profunda, são totalmente conectadas, a quantidade de cálculos se torna muito grande.

A rede neural convolucional não são totalmente conectadas, as ligações dos neurônios não são completas.

Arquitetura de uma Rede Neural Convolucional - CNN ("Convolutional Neural Networks")

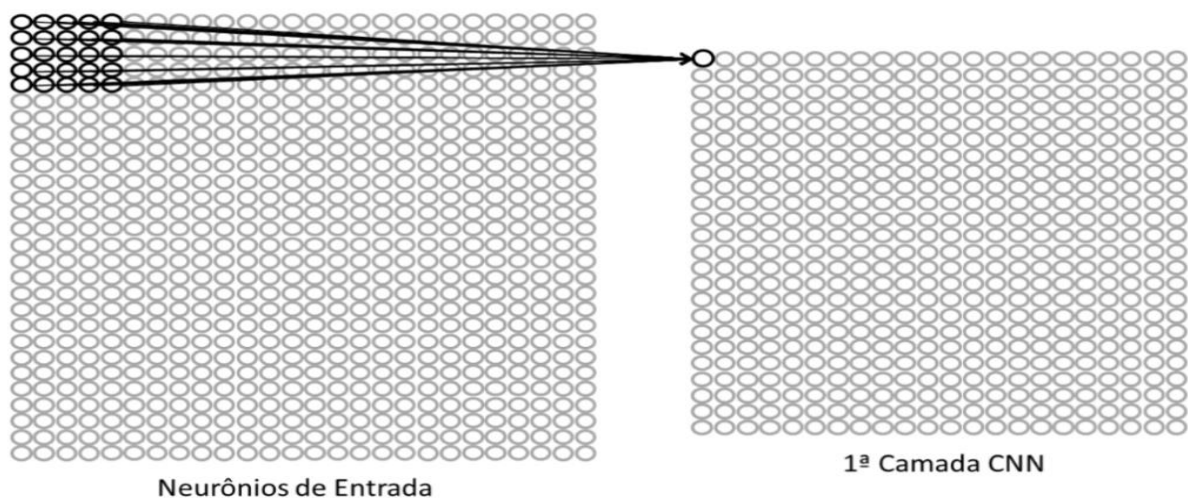


Figura 3 - Ligações em uma Rede Neural Convolucional

Na figura 3 temos apenas 25 neurônios conectados com um neurônio da 1ª camada CNN.

Numa rede neural densa, todos os 784 neurônios de entrada estariam conectados com cada um dos neurônios da primeira camada.

O próximo neurônio da 1ª camada CNN se conectará com 25 neurônios de entrada de maneira deslocada. Assim como a figura 4.

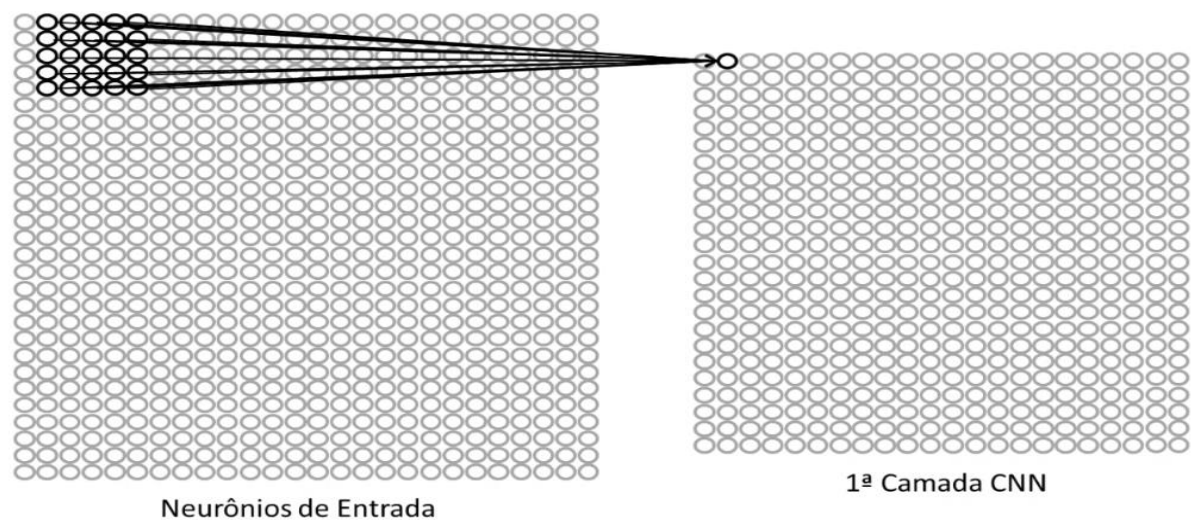


Figura 4 - Ligação do próximo neurônio

As CNN's funcionam muito bem para reconhecimento de imagens.

Abaixo segue um exemplo de um erro que uma rede neural totalmente conectada poderia gerar ao tentar reconhecer uma imagem (figura 5).

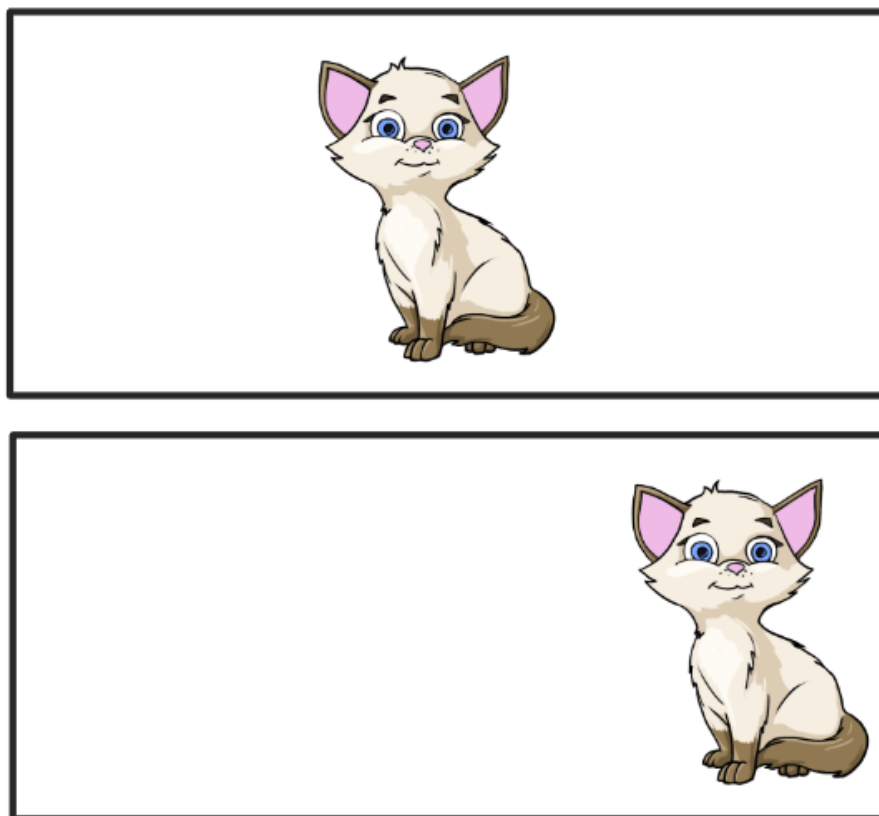


Figura 5 - Gato centralizado e não centralizado

As redes neurais densas (totalmente conectadas), não “enxerga” as duas figuras acima como um gato, pois cada um dos neurônios está aprendendo especificamente aquele pixel que está entrando, assim como a figura está deslocada o neurônio que estava aprendendo uma parte da figura deixou de aprender e passou a aprender outra coisa, por causa do deslocamento.

No caso da Rede Convolucional, cada retângulo que varre a figura é uma espécie de filtro, e ele tenta extrair da figura alguma característica, então podemos ter um filtro que se especializa em identificar os olhos do gato.

E outro filtro identifica o nariz, outro filtro identifica as patas, e assim uma CNN tem vários filtros, e todos varrem a figura.

Após varrer a figura, cada filtro cria uma camada de neurônios, camada chamada de feature map (mapa de características – Figura 6)

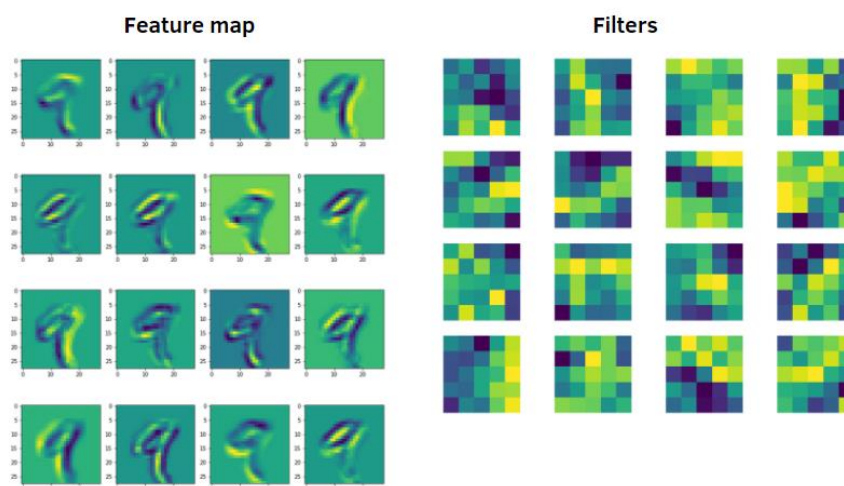


Figura 6 - Feature Map

Os filtros como são independentes, varrem a figura inteira, e cada um segue buscando uma característica que “aprendeu”.

Após isso teremos mais uma camada que une as peças das características das imagens criadas para concluir do que se trata.

13.MODELO VGG16

O modelo VGG16 é uma rede neural convolucional (CNN) amplamente conhecida e utilizada na área de visão computacional. Foi desenvolvido por um grupo de pesquisadores do Visual Geometry Group (VGG) na Universidade de Oxford.

O nome "VGG16" se refere à sua arquitetura, que consiste em 16 camadas, incluindo camadas convolucionais, camadas de pooling e camadas totalmente conectadas. Essa arquitetura se destacou por sua simplicidade e eficácia na extração de características de imagens.

A ideia por trás do VGG16 é utilizar várias camadas convolucionais com filtros de tamanho reduzido (3x3) em vez de filtros maiores, o que permite uma representação mais profunda e complexa das características da imagem. A presença de várias

camadas convolucionais e de pooling permite a extração de características em diferentes níveis de abstração.

No VGG16, as camadas convolucionais são seguidas por camadas de pooling, que reduzem a dimensionalidade das características e ajudam a capturar características invariantes à escala. Após as camadas de pooling, a rede é composta por camadas totalmente conectadas, que aprendem a classificar as características extraídas.

O treinamento do VGG16 envolve o uso de grandes conjuntos de dados rotulados, como o conjunto de dados ImageNet, que contém milhões de imagens em várias categorias. O VGG16 foi pré-treinado no conjunto de dados ImageNet, o que significa que ele já havia aprendido a extrair características úteis de imagens antes de ser aplicado a outras tarefas.

Devido à sua eficácia e facilidade de uso, o VGG16 se tornou um modelo popular e referência em visão computacional. Muitas vezes, é usado como uma base para transferência de aprendizado, onde o modelo pré-treinado é ajustado para tarefas específicas, como classificação de imagens, detecção de objetos ou segmentação semântica.

É importante mencionar que o VGG16 é relativamente grande em termos de quantidade de parâmetros, o que pode exigir mais recursos computacionais para treinar e implantar em comparação com modelos mais recentes e leves, como o MobileNet ou o EfficientNet. No entanto, seu desempenho e qualidade de extração de características ainda são altamente valorizados em muitas aplicações.

Como uma maior variedade de imagens é benéfica para nosso treinamento, são usadas algumas técnicas para se criar artificialmente essa variedade. São normalizados os pixels das imagens para ficar entre 0 e 1, aplicadas rotações, deslocamento horizontal e vertical, zooms aleatórios entre outras técnicas de transformação que vão ser apresentadas com mais detalhes no código.

Temos 2 classes, cachorro e gato, já separadas em pastas de treinamento e validação. Essa separação nos ajuda a medir melhor o treino, já que o código pode adaptar bem as informações de treino, mas lidar mal com informações novas, no nosso caso imagens.

As imagens sofrem uma mudança de tamanho para serem padronizadas e colocadas nesses modelos pré treinados. Escolhemos o número de batchs (que são uma quantidade de arquivos que vão ser treinados juntos e as épocas, o número de vezes que as imagens vão passar nesses processos de treinamento.)

14. DEFINIÇÃO E DESCRIÇÃO DE COMO SERÁ CALCULADA A ACURÁCIA.

Como vai ser medida a acurácia:

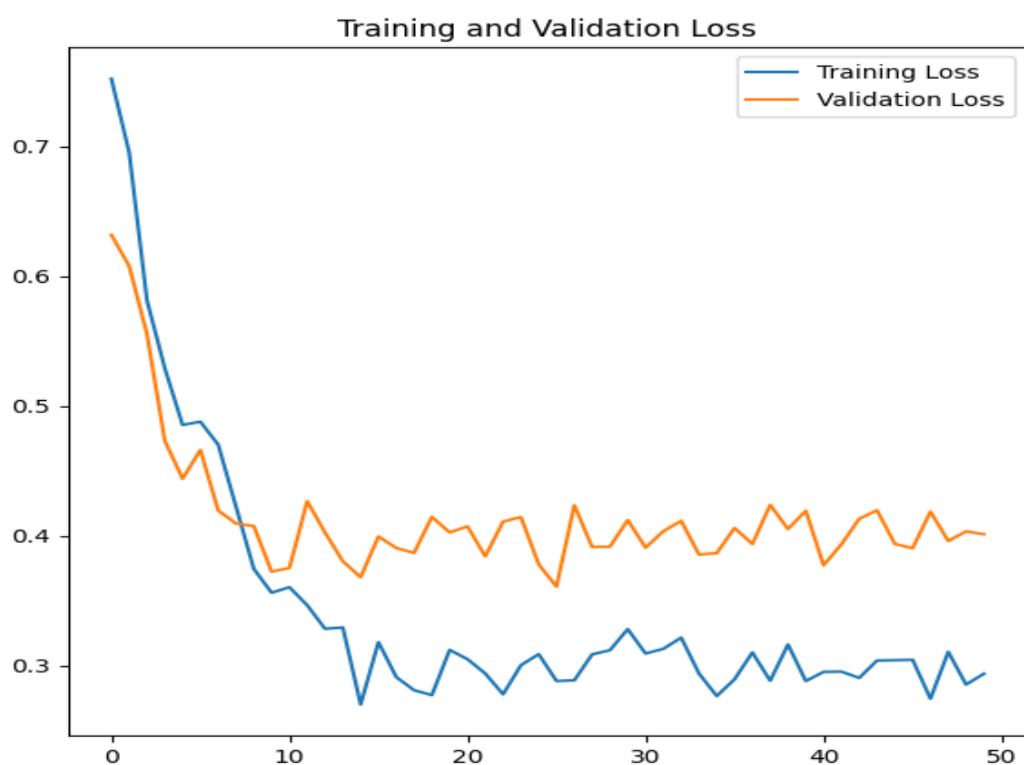
A biblioteca Keras tem embutida as seguintes métricas que estão sendo usadas para medir nossos treinamentos.

Loss: A medida da diferença entre as previsões do modelo e os valores reais dos dados de treinamento. Quanto menor o valor de perda, melhor o modelo está em prever os rótulos certos. Calculada no treinamento.

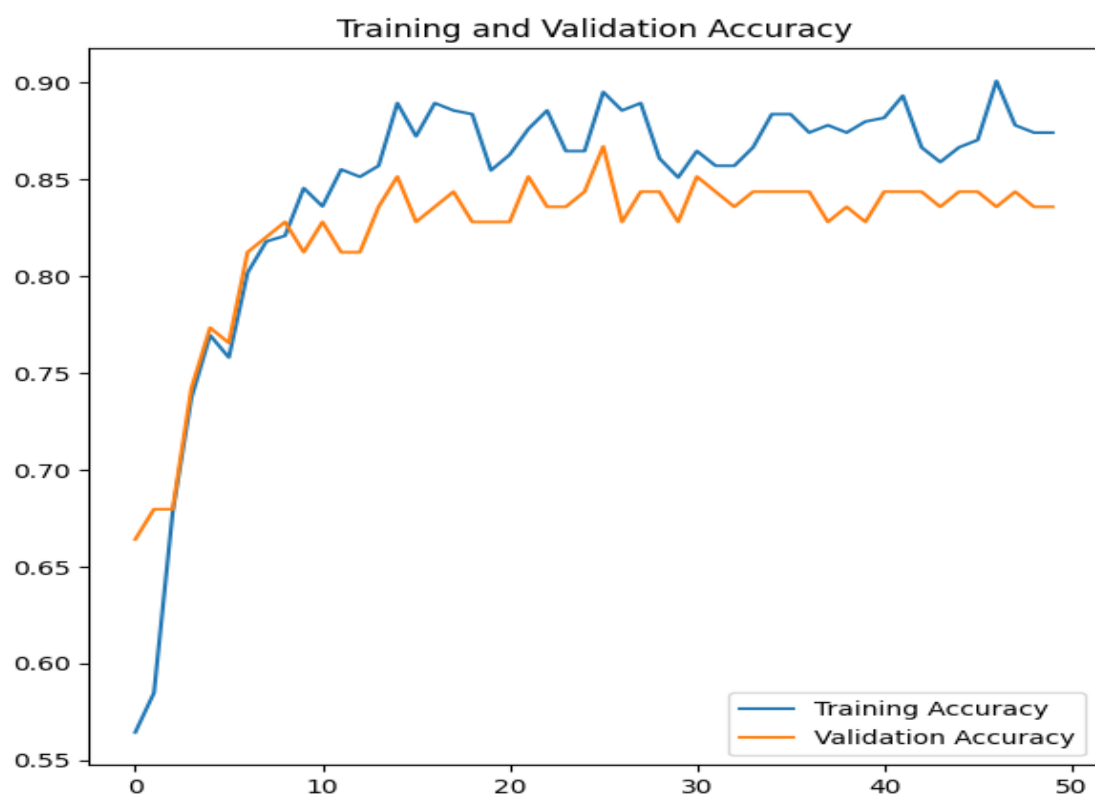
Accuracy: Medida da proporção de previsões corretas do modelo em relação ao número total de previsões totais. Calculada no treinamento.

Test accuracy: 0.8428571224212646

Val_loss: É a loss nos dados de validação em vez dos dados de treinamento.



Val_Accuracy: Acurácia calculada nos dados de validação em vez dos de treinamento.



Aqui nos interessam como resultado final os valores de Val_loss e Val_Accuracy.

15. APRESENTAÇÃO DO PROJETO

<https://youtu.be/wYp6gtT7xZY>

16. RESULTADO

Neste trabalho, apresentamos um modelo de classificação de imagens de gatos e cachorros usando redes neurais convolucionais. O modelo foi treinado em um conjunto de dados com mais de 500 imagens de gatos e cachorros, divididas em duas classes: gato e cachorro. O objetivo foi desenvolver um sistema capaz de reconhecer automaticamente se uma imagem contém um gato ou um cachorro, o que pode ter aplicações práticas em diversas áreas, como segurança, saúde animal e entretenimento.

O modelo foi avaliado usando uma métrica chamada acurácia, que mede a porcentagem de imagens classificadas corretamente pelo sistema. O resultado obtido foi de **0.8428571224212646**, o que significa que o modelo acertou cerca de 84% das imagens do conjunto de teste. Esse valor é considerado satisfatório, mas ainda há espaço para melhorias, como aumentar o tamanho do conjunto de dados, variar os parâmetros da rede neural ou aplicar técnicas de aumento de dados.

Em conclusão, o modelo proposto demonstrou ser capaz de classificar imagens de gatos e cachorros com uma acurácia razoável, mas ainda sujeito a erros.

17. REFERÊNCIAS

Understanding VGG16: Concepts, Architecture, and Performance: Disponível em:

[https://datagen.tech/guides/computer-vision/vgg16/#:~:text=What%20is%20VGG16%3F,CNN\)%20model%20supporting%2016%20layers.](https://datagen.tech/guides/computer-vision/vgg16/#:~:text=What%20is%20VGG16%3F,CNN)%20model%20supporting%2016%20layers.)