

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA
E TECNOLOGIA DO TRIÂNGULO MINEIRO

Vinícius Silveira Bisinoto

**Reconhecimento Automático de Valores em Dados de
RPG Utilizando Redes Neurais Convolucionais**

UBERABA - MG
2025

Vinícius Silveira Bisinoto

Reconhecimento Automático de Valores em Dados de RPG Utilizando Redes Neurais Convolucionais

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, Campus Uberaba Parque Tecnológico, como requisito parcial para a conclusão do Curso de Engenharia da Computação.

Orientador: Prof. Dr. Marcelo Barreiro

Sumário

1	Metodologia	2
1.1	Ambiente de Desenvolvimento	2
1.2	Base de Dados Inicial	2
1.3	Arquitetura da Rede Neural Convolucional	2
1.4	Planejamento da Base de Dados de RPG	3

1 Metodologia

Esta seção descreve os procedimentos adotados para o desenvolvimento do modelo de reconhecimento automático de valores em dados de RPG utilizando redes neurais convolucionais (CNNs). Inicialmente, foram conduzidos testes com uma base de dados pública, composta por imagens de cães e gatos, a fim de validar a arquitetura da rede e o fluxo de processamento. Em seguida, planejou-se a criação de uma base própria com dados de RPG, respeitando aspectos como visibilidade das faces e condições variadas de iluminação.

1.1 Ambiente de Desenvolvimento

O modelo foi desenvolvido utilizando a biblioteca TensorFlow, com a API `tf.keras`, em um ambiente baseado em Jupyter Notebook. O código foi escrito em Python 3. A execução dos experimentos foi realizada em uma máquina com suporte a GPU, utilizando bibliotecas auxiliares como NumPy, Matplotlib e scikit-learn.

1.2 Base de Dados Inicial

Para os testes preliminares, utilizou-se uma base pública contendo imagens rotuladas de cães e gatos, com distribuição balanceada entre as classes. Essa base serviu para a validação da arquitetura inicial da CNN e do pipeline de treinamento. As imagens foram redimensionadas para 180×180 pixels, com normalização dos valores de pixel no intervalo $[0, 1]$.

1.3 Arquitetura da Rede Neural Convolucional

A rede neural convolucional desenvolvida neste trabalho foi projetada para realizar tarefas de classificação de imagens com eficiência e baixa complexidade. A seguir, apresenta-se a descrição da arquitetura utilizada para o problema inicial de classificação binária (gato ou cachorro).

- **Pré-processamento:** A primeira camada realiza a normalização dos valores dos pixels, convertendo-os para o intervalo $[0, 1]$ através da operação `Rescaling(1./255)`.
- **Camada Convolucional 1:** Conv2D com 16 filtros, tamanho de kernel 3×3 , ativação ReLU e preenchimento `same`, mantendo a dimensão espacial.
- **Max Pooling 1:** MaxPooling2D, responsável por reduzir a dimensionalidade da imagem.
- **Camada Convolucional 2:** Conv2D com 32 filtros e mesmas configurações da primeira camada convolucional.
- **Max Pooling 2:** Segunda operação de MaxPooling2D.
- **Camada Convolucional 3:** Conv2D com 64 filtros.
- **Max Pooling 3:** Redução final da dimensionalidade espacial.
- **Camada de Achplanamento:** Flatten, transforma a saída tridimensional em um vetor unidimensional.

- **Camada Densa Oculta:** Dense com 128 unidades e ativação ReLU.
- **Camada de Saída:** Dense(1) com ativação `sigmoid`, utilizada para classificar a imagem entre duas classes.

A rede foi compilada com os seguintes parâmetros:

- **Otimizador:** Adam, com taxa de aprendizado de 0,001.
- **Função de perda:** `binary_crossentropy`.
- **Métrica:** Acurácia.

A Figura 1 apresenta um diagrama ilustrativo da arquitetura da CNN descrita.



Figura 1: Arquitetura da CNN para classificação binária de imagens.

1.4 Planejamento da Base de Dados de RPG

O próximo passo do projeto envolve a construção de uma base de imagens contendo fotos reais de dados de RPG, priorizando faces de fácil visualização. Decidiu-se evitar o uso de dados tetraédricos (como o d4), pois a face superior nem sempre é bem definida visualmente. Pretende-se focar em dados como d6, d8, d10, d12 e d20, capturando múltiplas fotos de cada valor possível, em diferentes condições de iluminação e ângulos. A arquitetura será ajustada para lidar com múltiplas classes conforme o dado utilizado (por exemplo, 12 classes para um d12).

Referências