

# REDE NEURAL CONVOLUCIONAL PARA CARACTERIZAÇÃO DE 10 CLASSES DIFERENTES

Túlio Castro Silva

## Caracterização do problema:

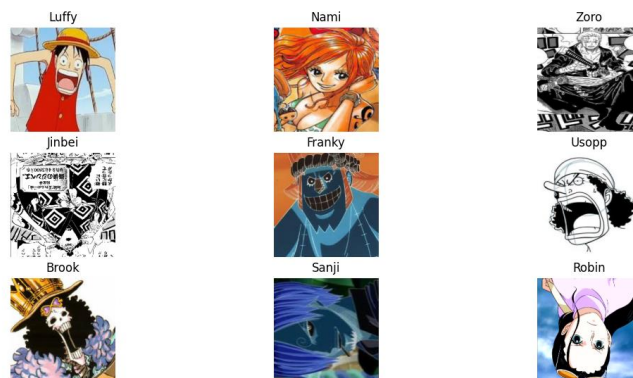
10 classes diferentes devem ser classificadas usando uma rede neural convolucional, com o intuito de conseguir classificar um desenho feito à mão.

## Dataset:

As imagens estão dispostas em um diretório da seguinte maneira :

```
.
├── Data/
│   ├── Brook/
│   │   ├── 1.jpg
│   │   ├── 2.jpg
│   │   ├── 1_inverted.jpg
│   │   └── ...
│   ├── Chopper/
│   │   ├── 1.jpg
│   │   ├── 2.jpg
│   │   ├── 1_inverted.jpg
│   │   └── ...
│   ├── Franky/
│   │   ├── 1.jpg
│   │   ├── 2.jpg
│   │   ├── 1_inverted.jpg
│   │   └── ...
│   ├── Jinbei/
│   │   ├── 1.jpg
│   │   ├── 2.jpg
│   │   ├── 1_inverted.jpg
│   │   └── ...
│   ├── Luffy/
│   │   ├── 1.jpg
│   │   ├── 2.jpg
│   │   ├── 1_inverted.jpg
│   │   └── ...
│   └── ...
```

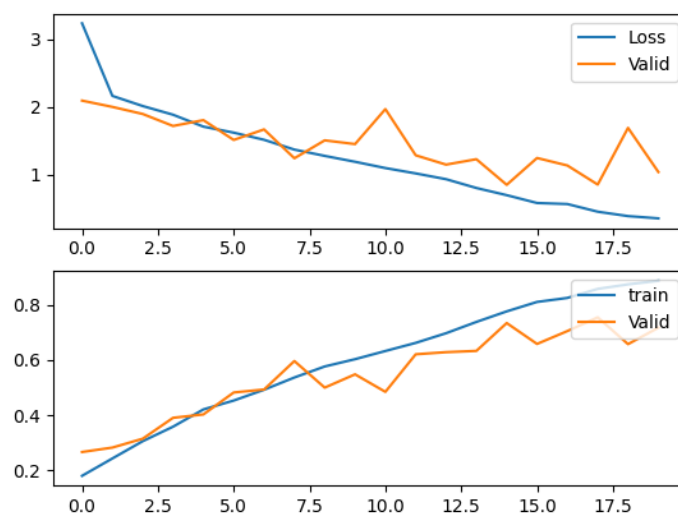
Aqui estão alguns exemplos das imagens:



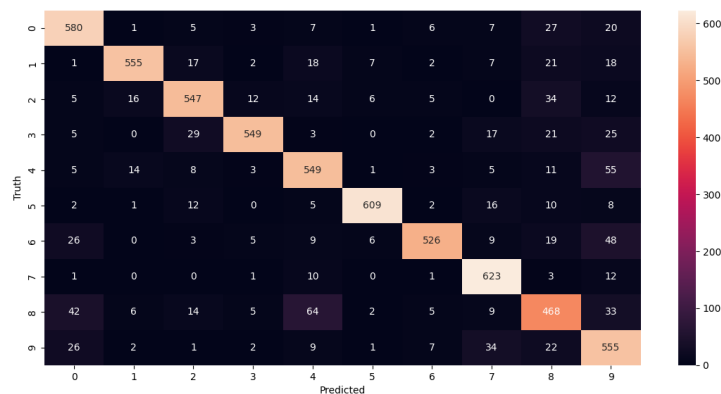
Dentro do dataset existem imagens coloridas, imagens em preto e branco, imagens com as cores invertidas, imagens invertidas horizontalmente e verticalmente.

### Primeira abordagem para o problema:

O primeiro modelo criado foi um modelo sequencial simples com 5 camadas de convolução, 5 camadas de MaxPooling e 5 camadas de batchnormalization, o output consiste em uma camada densa com 512 neurônios de ativação ReLU, ligada a outra camada densa com 256 neurônios de ativação ReLU, finalizando com uma camada densa com 10 neurônios com ativação SoftMax. O otimizador escolhido foi o Adam, a função de perda Categorical Crossentropy.



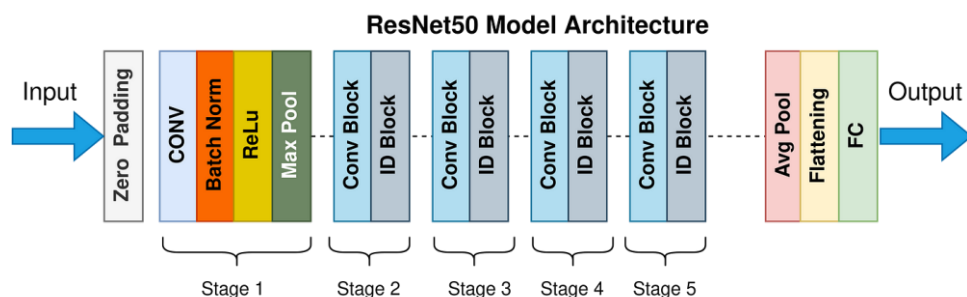
Ao fim do treinamento, o modelo alcançou uma acurácia de 0.8510, isto pode ser visualizado na matriz de confusão a seguir:



### Segunda abordagem para o problema:

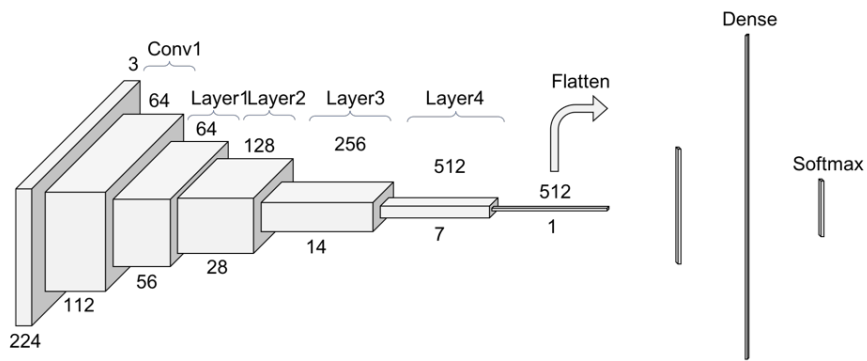
Agora a estratégia escolhida seria utilizar um modelo já existente e apenas aplicá-lo neste problema com o intuito de maximizar os resultados.

O modelo escolhido foi a Resnet(Residual Network), mais especificamente sua variante com 50 camadas, este modelo e os pesos desenvolvidos no projeto ImageNet pode ser carregado diretamente do modulo Keras.



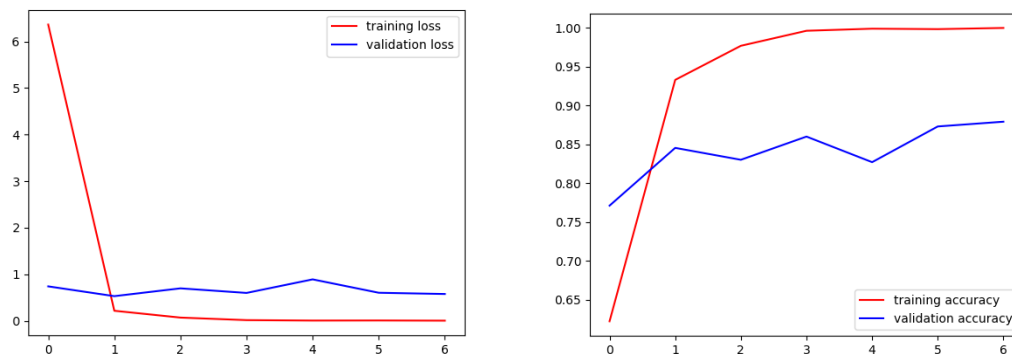
Arquitetura da ResNet50

A ResNet sera capaz de extrair as características que queremos das imagens, ela não pode ser usada para classificar uma imagem, para isso, transformamos a matriz fornecida pela Resnet em um vetor num e o introduzimos uma rede neural sequencial para classificar essa imagem. Podemos visualizar isso no seguinte diagrama:

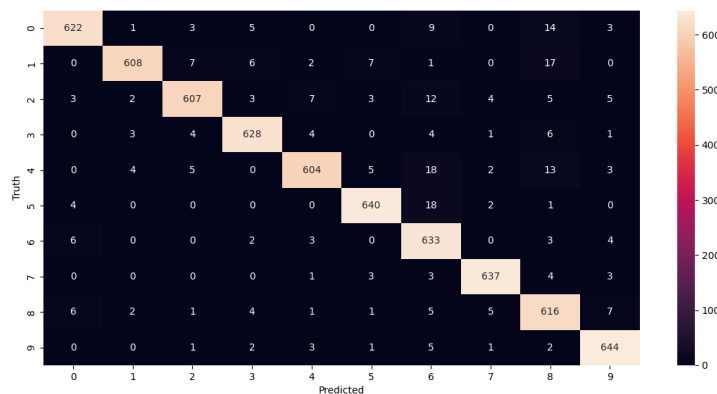


Para implementar esta rede no Tensorflow, importamos a Resnet como um modelo base diretamente do Keras e adicionamos uma rede sequencial comum no output do nosso modelo base. É importante congelar os neurônios da resnet para impedir que seus pesos alterem durante o treinamento, isto não só diminui o tempo de treinamento como também ajuda na acurácia da rede.

Ao fim do treinamento obtivemos os seguintes resultados:

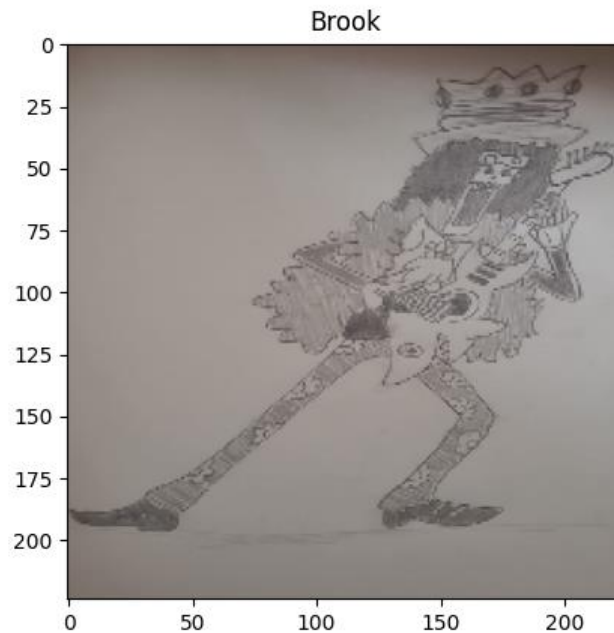


Ao fim do treinamento, o modelo atingiu uma acurácia de 0.9547, isto pode ser visualizado na matriz de confusão a seguir>



### Conclusão:

Ambos os modelos atingiram valores de acurácia satisfatórios, apenas a Resnet foi capaz de classificar corretamente o desenho feito a mão



Desenho por: Gabriel Rodrigues

<https://www.instagram.com/reel/Ctx0GBYNdtI/?igshid=MzRIODBiNWFIZA==>