

Reconhecimento de imagens utilizando métodos de extração de características e classificadores de classes

Rafael Rampim Soratto¹

Michel Gomes De Souza²

¹ Universidade Tecnológica Federal do Paraná - UTFPR// (Campus Campo Mourão) - Brasil - Campo Mourão - PR

soratto@alunos.utfpr.edu.br e michels@alunos.utfpr.edu.br

Abstract. The objective of this work is to extract characteristics of images referring to 90 lungs infected by COVID 19 and 90 healthy lungs to train this data in classifiers. In this way it is possible to compare the result and accuracy of the classifiers in relation to the extracted features. The best result was obtained with an accuracy of 94.4387 % using the extraction of deep RESNET characteristics, the normalization of z-score data, the MLP classification algorithm with the configuration of 100 neurons.

Resumo. O objetivo deste trabalho é extrair características de imagens referentes a 90 pulmões infectados pelo COVID 19 e 90 pulmões saudáveis para treinar esses dados em classificadores. Desta maneira é possível comparar o resultado e acurácia dos classificadores em relação as features extraídas. O melhor resultado obtido foi com acurácia de 94,4387% utilizando a extração de características profundas RESNET, a normalização de dados z-score, o algoritmo de classificação MLP com a configuração de 100 neurônios.

1 Objetivos

O objetivo é classificar uma imagem de pulmão em apenas duas classes: Com COVID-19 e Sem COVID-19. Para isso serão utilizadas imagens de pacientes com o vírus e pacientes sem o vírus.

Exemplo de imagem de um paciente com COVID-19:

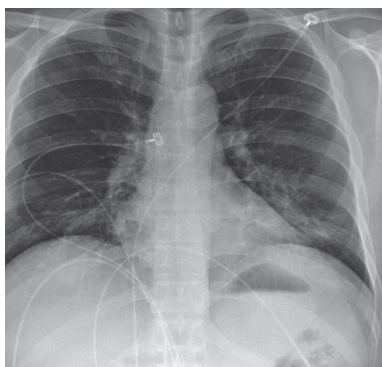


Figura 1: Imagem de um pulmão infectado pelo vírus.

Exemplo de imagem de um paciente não infectado:

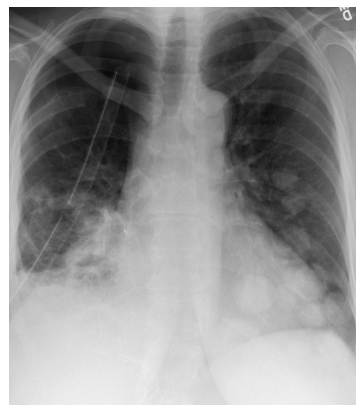


Figura 2: Imagem de um pulmão não infectado pelo vírus.

2 Metodologia

São utilizados métodos de extração de características profundas em imagens de pulmões infectados e saudáveis tais como: vgg16, vgg19, mobilenet, inception, resnet, xception. É possível comparar a acurácia durante o reconhecimento de pulmões infectados pelo COVID-19 utilizando os classificadores k-NN, SVM, MLP e Random Forest e as features extraídas.

Serão utilizados algoritmos de classificação de dados com a finalidade de obter a melhor acurácia (mais próxima de 100%) durante o reconhecimento do vírus utilizando as características extraídas das imagens. A implementação deste trabalho foi realizada utilizando a linguagem Python. A entrada do algoritmo são dois arquivos de dados de teste e de treino gerados a partir das imagens dos pulmões.

As imagens são divididas em imagens de teste e imagens de treinamento, das 180 imagens (90 infectadas e 90 normais) foram selecionadas 70% para treinamento e 30% para teste.

2.1 Arquitetura de Extração de características de uma imagem

Existem arquiteturas específicas para extrair “features” de uma imagem. Este é o primeiro processo necessário para captar dados interessantes sobre as classes para o classificador.

VGG16 - Rede Convolucional para Classificação e Detecção VGG16 é um modelo de rede neural convolucional proposto por K. Simonyan e A.

Zisserman da Universidade de Oxford no artigo “Redes Convolucionais Muito Profundas para Reconhecimento de Imagens em Grande Escala”. Ele melhora o desempenho ao substituir grandes filtros de tamanho de kernel (11 e 5 na primeira e segunda camadas convolucionais, respectivamente) por vários filtros de tamanho de kernel 3×3 , um após o outro.

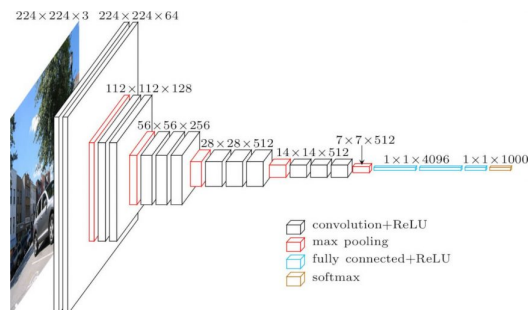


Figura 3: VGG16 - Rede Convolucional para Classificação e Detecção [1]

De acordo com a figura 3 três camadas Fully-Connected (FC) seguem uma pilha de camadas convolucionais (que tem uma profundidade diferente em diferentes arquiteturas): as duas primeiras têm 4096 canais cada, a terceira realiza a classificação ILSVRC de 1000 vias e, portanto, contém 1000 canais (um para cada aula). A camada final é a camada soft-max. A configuração das camadas totalmente conectadas é a mesma em todas as redes.

Desvantagens:

- Lento para treinar.
- Pesos da arquitetura de rede são bastante grandes (em relação ao disco / largura de banda).

2.2 MobileNet

A grande ideia por trás do MobileNet V1 é que as camadas convolucionais, que são essenciais para tarefas de visão por computador, mas são muito caras para computar, podem ser substituídas pelas chamadas convoluções separáveis em profundidade.

Arquitetura que apresenta aprimoramentos na performance de tempo real de sistemas de aprendizagem profunda com hardware limitado, pois reduz o número de parâmetros sem sacrificar a acurácia[2];

2.3 Inception

De acordo com [3] as vantagens dessa arquitetura são :

1. Ganho de alto desempenho em redes neurais convolucionais;
2. Utilização eficiente de recursos de computação com aumento mínimo na carga de computação para a saída de alto desempenho de uma rede Inception.
3. Capacidade de extrair recursos de dados de entrada em escalas variadas por meio da utilização de tamanhos variados de filtro convolucional. rede.

2.4 Resnet

ResNet, abreviatura de Rede Residual, é um tipo específico de rede neural que foi introduzido em 2015 por Kaiming He, Xiangyu Zhang, Shaoqing Ren e Jian Sun em seu artigo “Deep Residual Learning for Image Recognition”. Principalmente para resolver um problema complexo, empilhamos algumas camadas adicionais nas Redes Neurais Profundas que resultam em precisão e desempenho aprimorados. A lógica por trás da adição de mais camadas é que essas camadas aprendem progressivamente recursos mais complexos [4].

Por exemplo, no caso de reconhecimento de imagens, a primeira camada pode aprender a detectar bordas, a segunda camada pode aprender a identificar texturas e da mesma forma a terceira camada pode aprender a detectar objetos e assim por diante. Mas foi descoberto que há um limite máximo para profundidade com o modelo de rede neural convolucional tradicional. Aqui está um gráfico que descreve a porcentagem de erro nos dados de treinamento e teste para uma rede de 20 camadas e uma rede de 56 camadas. A rede ResNet usa uma arquitetura de rede simples de 34 camadas inspirada no VGG-19, na qual a conexão de atalho é adicionada. [4].

2.5 Xception

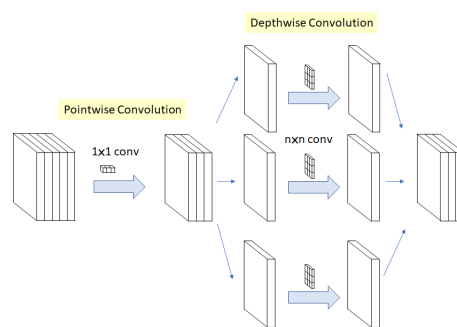


Figura 4: Xception

3 Fetures extraídas

Arquitetura	Total de características
Inception	2049
vvg16	513
vvg19	513
resnet	2049
mobilenet	1025
xception	2049

Tabela 1: Total de características extraídas das 180 imagens dos pulmões por arquitetura

4 Normalizacao dos dados

Utilizando as features extraídas anteriormente, iremos normalizar essas características para simplificar o funcionamento dos classificadores.

A normalização dos dados tenta proporcionar a todos os atributos um peso igual. Normalização é particularmente útil para algoritmos de classificação envolvendo redes neurais ou medições de distância, como classificação de vizinho mais próximo (k-NN) e "clustering".

Existem diversos métodos de normalização, neste trabalho serão utilizados os métodos Min-Max e Z-score. Para isto, utilizaremos um vetor com n elementos de $V_1 \dots V_n$.

4.1 Normalização Min-Max

Normalização que executa uma transformação linear nos dados originais. Cada elemento do vetor é normalizado utilizando o valor máximo e mínimo do vetor. De acordo com a fórmula para definir cada elemento de um vetor A normalizado dentro de um intervalo $[0.0, 1.0]$:

$$v_i = \frac{v_i - \min_a}{\max_a - \min_a} * 1 \quad (1)$$

4.2 Normalização z-score

Considera a média e o desvio padrão durante a normalização de acordo com a formula.

$$v_i = \frac{v_i - \bar{A}}{\sigma_A} \quad (2)$$

sendo \bar{A} a média e σ_A o desvio padrão.

5 Classificadores

Um classificador utiliza dados de treino e teste para classificar um elemento de acordo com uma classe. De acordo com uma bateria de testes ele cria uma matriz de confusão de onde os resultados são medidos pela acurácia. Alguns classificadores são k-NN(k Nearest Neighbors), SVM (Support Vector Machine), MLP (A multilayer perceptron) e Random Forest.

5.1 k-NN

1. Inicialmente, calcula-se a distância entre o exemplo desconhecido x e todos os exemplos do conjunto de treinamento ;
2. Identifica-se os k vizinhos mais próximos utilizando a distância escolhida (euclidiana, manhattan, minkowski e chebyshev);
3. A classificação é feita associando o exemplo desconhecido x à classe que for mais frequente, entre os k exemplos mais próximos de x ;

Utiliza o voto majoritário para definir a classe mais frequente.

5.2 Distância entre os vizinhos no k-NN

Existem algumas distâncias que podem ser testadas no algoritmo e de vizinhos mais próximos tais como:

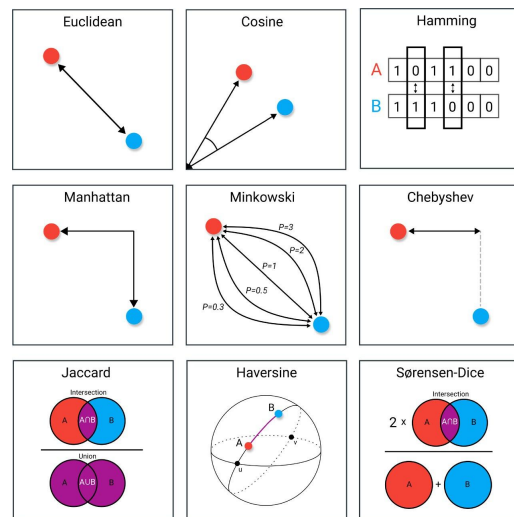


Figura 5: Exemplos de distâncias

5.3 SVM

Support vector machine é utilizado no treinamento e classificação de um dataset. Ele funciona com um hiperplano que separa graficamente dois ou mais conjuntos de dados. Support Vector Machine é uma fronteira que melhor segrega as duas classes (hiperplano / linha).

5.4 MLP

Perceptron Multicamadas (PMC ou MLP — Multi Layer Perceptron) é uma rede neural com uma ou mais camadas ocultas com um número indeterminado de neurônios. A camada oculta possui esse nome porque não é possível prever a saída desejada nas camadas intermediárias. Para treinar a rede MLP, o algoritmo comumente utilizado é o de retropropagação (Backpropagation). Aumentar o número de camadas e neurônios nem sempre é a melhor solução para uma melhoria de performance/acurácia.

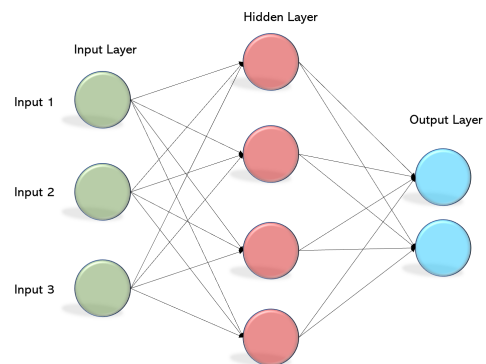


Figura 6: Funcionamento de uma rede perceptron multicamadas MLP

5.5 Random Forest.

A floresta aleatória é um algoritmo de aprendizado supervisionado que funciona de maneira gerar uma árvore de decisão precisa. A floresta aleatória constrói várias árvores de decisão e as mescla para obter uma previsão mais precisa e estável.

6 Resultados

Foram utilizadas 90 imagens de treino e 90 imagens de teste onde se realizou a extração de características profundas em imagens infectadas e saudáveis utilizando os métodos: vgg16, vgg19, mobilenet, inception, resnet, xception. É possível comparar a acurácia durante o reconhecimento de pulmões infectados pelo COVID-19 utilizando os classificadores k-NN, SVM, MLP e Random Forest e as features extraídas. O modelo que possui a melhor acurácia possui também uma matriz de confusão com maior número de acertos, ou seja, maior número de verdadeiros positivos e falsos verdadeiros.

6.1 Matriz de confusão

Tabela que mostra as frequências de classificação para cada classe do modelo, neste caso serão as classes com vírus e sem o vírus. A matriz de confusão pode ser gerada recebendo como parâmetros um array de valores reais $R[]$ e outro array de predições $P[]$. O resultado é a frequência de:

- **verdadeiros positivos (TP)**
($P[i] == 1 \& R[i] == 1$),
- **falsos positivos (FP)**
($P[i] == 1 \& R[i] == 0$),
- **falso verdadeiro (TN)**
($P[i] == 0 \& R[i] == 0$);
- **falso negativo (FN)**.
($P[i] == 0 \& R[i] == 1$);

6.2 Acurácia

É o resultado da matriz de confusão que diz a percentagem de acerto das predições.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (3)$$

A melhor matriz de confusão obtida gerou uma acurácia de 94% utilizando o algoritmo MLP. Sabendo disso testamos rodar novamente este classificador para ver se o resultado se repetia e obtemos o resultado de 85%:

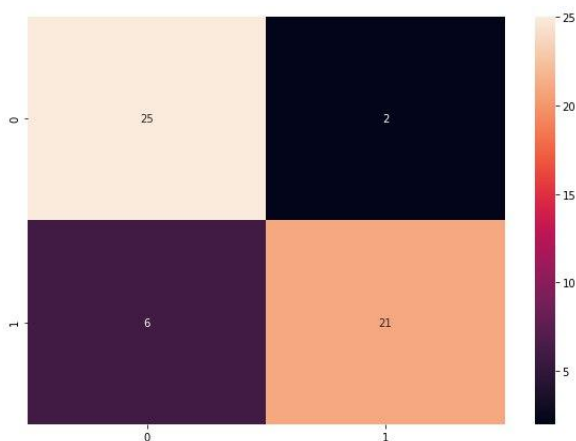


Figura 7: matriz de confusão utilizando MLP

De acordo com a Figura 7 é possível visualizar que o melhor classificador acertou 21 verdadeiros positivos e 25 falsos verdadeiros:

$$accuracy = \frac{21 + 25}{21 + 25 + 6 + 2} = 0.85 \quad (4)$$

7 Conclusão

O melhor resultado obtido foi com acurácia de 94,4387% utilizando a extração de características RESNET, a normalização de dados z-score, o algoritmo de classificação MLP com a configuração de 100 neurônios:

```
1 +-----+
2 |Extraction Feature: RESNET
3 |ACC: 0.9443871115697741
4 |Classificador: MLP
5 |Normalizacao: z-score
6 |Parametros: MLPClassifier(activation='tanh',
   |             hidden\_layer\_sizes=(50, 100, 50), random\_
   |             _state=0)
7 +-----+
8
9 =====BEST VALUES--DO--KNN=====
10 ACC: 0.8493584748844216
11 Distancia: manhattan
12 K: 3
13 Normalizacao: z-score
14 =====
15
16 =====BEST VALUES--DO--SVM=====
17 ACC: 0.9259259259259259
18 Normalizacao: min-max
19 Kernel: linear
20 Best Parametros: Pipeline(steps=[('scaler',
   |                               MinMaxScaler()),
21 |                               ('svm', SVC(C=0.03125, gamma=8.0, kernel='linear
   |                               ', probability=True))])
22 =====
23
24 =====BEST VALUES--DO--RF=====
25 ACC: 0.9259259259259259
26 Normalizacao: z-score
27 Best Parametros: RandomForestClassifier(
   |               min_samples_leaf=2, min_samples_split=5,
   |               n_estimators=200)
```

Para trabalhos futuros poderiam se testar junção de características e métodos de remoção de características insignificantes para otimizar ainda mais acurácia dos classificadores.

Referências

- [1] Roland Hewage. Extract features, visualize filters and feature maps in vgg16 and vgg19 cnn models. <https://towardsdatascience.com/extract-features-visualize-filters-and-feature-maps-in-vgg16-and-vgg19-cnn-models-d2da6333edd0>, 2020.
- [2] Yiting Li, Haisong Huang, Qingsheng Xie, Liguoyao, and Qipeng Chen. Research on a surface defect detection algorithm based on mobilenet-ssd. *Applied Sciences*, 8(9):1678, Sep 2018.
- [3] Richmond Alake. Deep learning: Understanding the inception module. <https://towardsdatascience.com/deep-learning-understand-the-inception-module-56146866e652>, 2020.
- [4] Hussain Mujtaba. Introduction to resnet or residual network. <https://www.mygreatlearning.com/blog/resnet/>, 2020.