



AI Project PulmoScan

Integrantes:

**1- Vinicius Rodrigues Cardoso Silva
(202000632)**

**2- Paloma Corrêa Alves
(202000716)**

**3- Davi Souza da Silva Veras
(202000212)**

**4- Juliete C. de Albuquerque Sérvio
(202003843)**

**5- João Vitor Lemos
(202002342)**

Introdução



Esse projeto se baseia em classificar as doenças de Câncer de pulmão.

Importância do diagnóstico precoce e avanços no tratamento.

Papel das imagens de raio-X no diagnóstico do câncer de pulmão.

Definição de Problema

Objetivo do projeto: desenvolver um modelo de CNN para auxiliar no diagnóstico do câncer de pulmão.

Dataset utilizado: imagens de raio-X de pulmão rotuladas em quatro categorias.

Utilização de técnicas de aprendizado de máquina, como CNNs, para auxiliar no diagnóstico.

Preparação do DataSet

Divisão do conjunto de dados em treinamento, teste e validação.

Importância de um conjunto de dados balanceado.

Pré-processamento dos dados: carregamento, redimensionamento, normalização e codificação one-hot dos rótulos das classes.

Pré - Processamento

Pré-processamento dos dados:

- Carregamento
- Redimensionamento
- Normalização
- Codificação one-hot dos rótulos das classes

Configuração a Arquitetura da CNN

Utilização de Transfer Learning para aproveitar conhecimento prévio.

Escolha da CNN Inception como base do modelo.

Explicação das camadas utilizadas na construção do modelo (Conv2D, GlobalAverage Pooling2D, Flatten, Dense, Dropout).

Construção do Modelo Sequential com Camadas

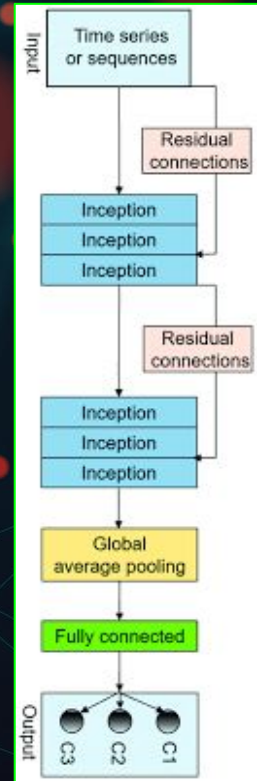


Para construir o modelo CNN, utilizamos a abordagem Sequential da API do Keras.

O Sequential permitiu adicionar camadas sequencialmente, como um conjunto de blocos de construção, para formar a arquitetura da rede neural convolucional.

Inception V3 architecture

O Inception v3 é um modelo de reconhecimento de imagem que alcança uma boa precisão no conjunto de dados do ImageNet.

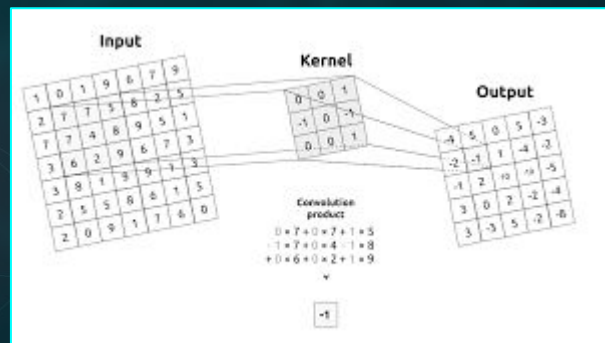


Camadas Utilizadas na Construção do Modelo:

Conv2D

Essa camada aplicar filtros (kernels) que percorrem a imagem para extrair características importantes.

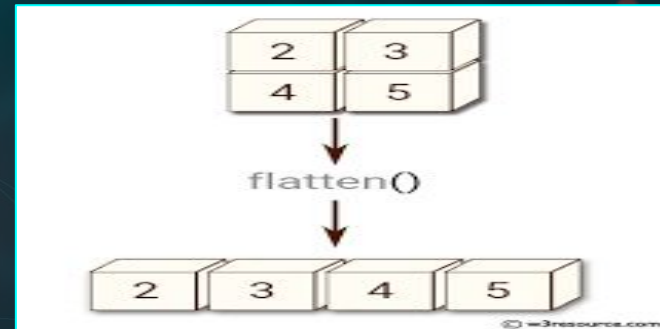
Cada filtro gera um mapa de características convolucionais, capturando diversos padrões presentes nas imagens de raio-X.



Flatten

Esta camada é responsável por transformar o mapa de características 2D em um vetor unidimensional.

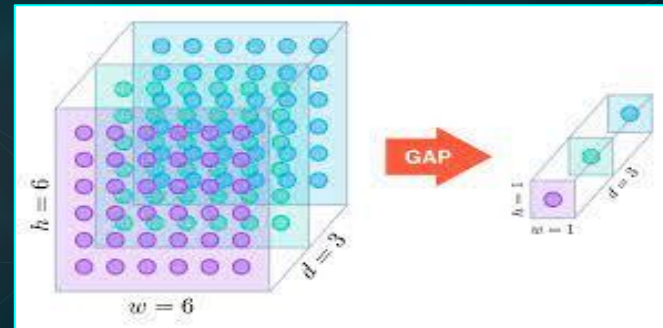
Ela "achata" os dados para que possam ser alimentados em uma camada densamente conectada.



GlobalAverage Pooling2D

A camada GlobalAveragePooling2D reduz a dimensionalidade dos mapas de características 2D das camadas convolucionais, selecionando o valor médio em cada região do mapa.

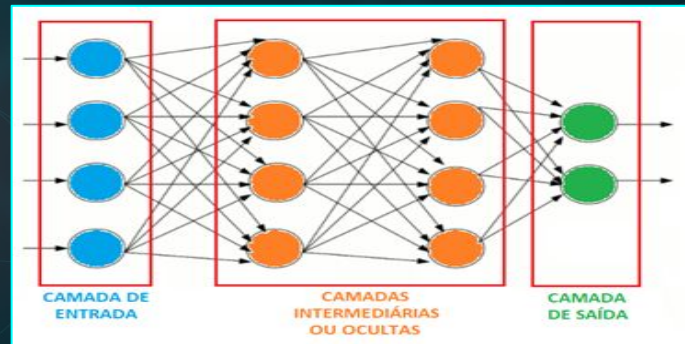
Isso reduz a resolução, mantendo as informações relevantes.



Dense

A camada totalmente conectada é responsável pela classificação final, pois todas as entradas estão conectadas a cada neurônio.

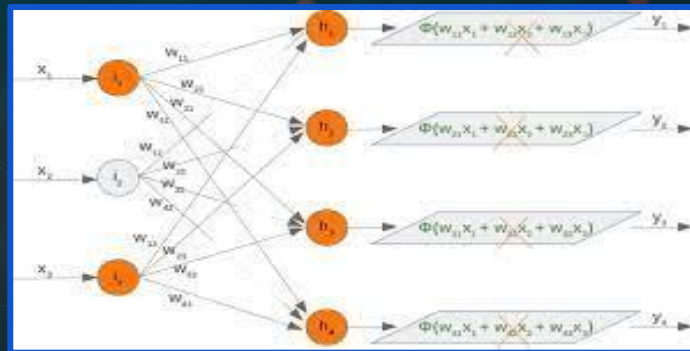
Ela utiliza as características extraídas pelo modelo para realizar essa classificação.



Dropout

Durante o treinamento, essa camada reduz o overfitting descartando conexões entre as camadas de forma aleatória.

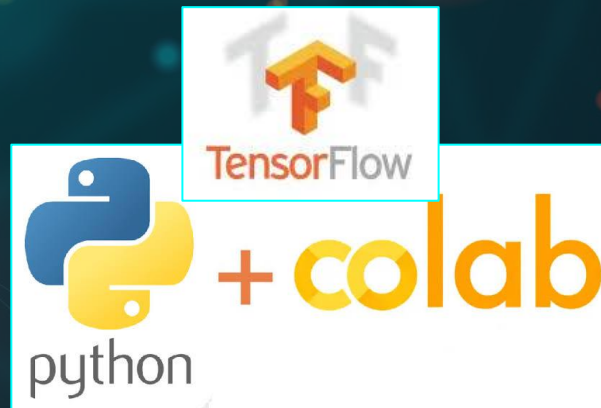
Isso promove a regularização do modelo e melhora sua capacidade de generalização.



Ambiente de Desenvolvimento

—
Google Collab

Recursos fornecidos pelo Google
Colab para treinamento eficiente
do modelo.



Treinamento do Modelo

Utilização do conjunto de treinamento para otimizar o modelo.

Minimização da função de perda com otimizador Adam
Monitoramento de métricas de desempenho durante o treinamento.



Validação do Modelo

Utilização do conjunto de validação para garantir a generalização correta do modelo.

Avaliação do Modelo

Utilização do conjunto de teste para avaliar a capacidade de classificação do modelo.

Matriz de confusão e métricas de desempenho (acurácia, precisão, recall, F1-score).

Ajuste e Otimização do Modelo

Técnicas de regularização, aumento de dados e ajuste de hiperparâmetros.

Importância de explorar diferentes configurações e técnicas para otimizar o desempenho do modelo.

Regularização

Foram usadas técnicas de dropout para evitar o overfitting e melhorar a generalização do modelo.

Essas técnicas reduzem a complexidade limitando os pesos ou desativando conexões aleatoriamente durante o treinamento.

Aumento de dados (data augmentation)

Foram usadas técnicas de dropout e aumento de dados para evitar overfitting e melhorar a generalização do modelo.

O dropout limita pesos e desativa conexões, enquanto o aumento de dados introduz variações nas imagens existentes, aumentando a robustez do modelo.

Ajuste dos hiperparâmetros:

Foram realizados ajustes nos hiperparâmetros do modelo, como taxa de aprendizado, tamanho do lote, número de épocas de treinamento e arquitetura da CNN.

Através de experimentação e validação cruzada, buscou-se encontrar a combinação ideal de hiperparâmetros que otimizam o desempenho do modelo.

Outras técnicas

Otimização do algoritmo (utilizando Adam) e utilização da técnica de transfer learning que possuem arquiteturas mais avançadas, como redes residuais (ResNet) ou redes densas (DenseNet), e a Inception.

Foi essencial explorar diferentes configurações e técnicas para otimizar o desempenho do modelo e melhorar a precisão do diagnóstico do câncer de pulmão.

Implantação e Uso

Integração do modelo H5 de classificação de câncer de pulmão no frontend, permitindo que os usuários carreguem imagens e obtenham resultados de classificação precisos.

Interface gráfica amigável que facilita o upload de imagens e a visualização dos resultados de classificação em tempo real.

Implantação e Uso

Parte 2

Pré-processamento de imagens e uso do modelo H5 para classificar as imagens, fornecendo informações claras sobre o tipo de câncer de pulmão identificado.

Referências:

- *A guide to transfer learning with Keras using ResNet50 | by Kenneth Cortés Aguas | Medium*
- *InceptionV3 (keras.io)*
- *<https://www.kaggle.com/code/kanncaal/convolutional-neural-network-cnn-tutorial>*
- *<https://www.tensorflow.org/js/guide/conversion?hl=pt-br>*

Visualização do Projeto:

1. Collab

2. Server Local