PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Uso de Técnicas de Deep Learning Para a Classificação de Vídeos de Exames Médicos

Rafael Lavatori Caetano de Bastos

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Engenharia da Computação

Rio de Janeiro, RJ, Dezembro de 2024

Rafael Lavatori Caetano de Bastos

Uso de Técnicas de Deep Learning Para a Classificação de Vídeos de Exames Médicos

Projeto Final, apresentado ao programa Engenharia da Computação da PUC-Rio como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientador: Paulo Ivson Netto Santos
Departamento de Informática

Rio de Janeiro

Dezembro de 2024.

Resumo

Lavatori Caetano Bastos, Rafael. Ivson Netto Santos, Paulo. Uso de Técnicas de Deep Learning Para a Classificação de Vídeos de Exames Médicos. Rio de Janeiro, 2024. 65p. Projeto Final de Curso - Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro

Este trabalho aborda o emprego de técnicas de Deep Learning para o treinamento e uso de rede neural na classificação de exames de videofluoroscopia, empregados no diagnóstico de disfagia, como sintoma de doenças neurodegenerativas ou tumores. As arquiteturas implementadas baseiam-se em redes convolucionais (CNN), associadas ou não com memórias de longo e curto prazo (long short-term Memory – LSTM). A arquitetura que apresentou melhor resultado, muito próximo ao estado da arte, foi o modelo baseado em VGG16 multi-classe, treinado a partir de 100 vídeos de videofluoroscopia, disponibilizados pelo INCA.

Palavras-Chaves
Disfagia, videofluoroscopia, aprendizado profundo, CNN, LSTM, VGG16

Abstract

Lavatori Caetano Bastos, Rafael. Ivson Netto Santos, Paulo. Uso de Técnicas de Deep Learning Para a Classificação de Vídeos de Exames Médicos. Rio de Janeiro, 2024. 65p. Projeto Final de Curso II - Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro

This work addresses the use of Deep Learning techniques for the training and use of a neural network in the classification of videofluoroscopy exams, used in the diagnosis of dysphagia, as a symptom of neurodegenerative diseases or tumors. The implemented architectures are based on convolutional neural networks (CNN), associated or not with long short-term memory (LSTM). The architecture that presented the best results, very close to the state of the art, was the model based on multi-class VGG16, trained from 100 videofluoroscopy videos, made available by INCA

Keywords
Dysphagia, videofluoroscopy, Deep Learning, CNN, LSTM, VGG16

Sumário

Lista de Figuras

Lista de Tabelas

1. Introdução

A deglutição é o ato de engolir alimentos, ou seja, transportar o bolo alimentar da boca até o estômago. Este processo incluí contrações e relaxamentos coordenados dos músculos da língua, faringe, laringe e esôfago, que são comandados pelo sistema nervoso central (SNC) [1-3].

A disfagia, ou dificuldade de deglutição, é um problema de saúde severo relacionado a essa dificuldade do transporte seguro da boca até o estômago, ocasionado por problemas neurológicos variados, sendo um sintoma clínico comum em pacientes idosos e/ou com doenças cerebrovasculares, neuromusculares, neurodegenerativas e/ou com câncer de cabeça e pescoço [4-7]. As consequências da disfagia quando não tratada são má nutrição, redução da força muscular, desidratação, falha do sistema imunológico, imobilidade, aspiração e pneumonia, todas com potencial de diminuir a qualidade de vida dos pacientes e aumentar o risco de mortalidade [4-7]. Dessa forma, o estudo da deglutição vem sendo realizado ao longo dos anos para melhorar a qualidade de vida desses pacientes e evitar mortes.

O exame de Videofluoroscopia da deglutição (VFSE) é atualmente o padrão ouro para diagnosticar precisamente e analisar quantativamente a disfagia [8]. Esse exame envolve a captura de um vídeo criado por meio de uma sequência de imagens de raio-X enquanto o paciente ingere líquidos, de diferentes volumes e viscosidades, misturados com bário, permitindo uma visualização mais detalhada do processo de deglutição. No entanto, esta análise dos vídeos é complexa e demorada, demandando expertise de fonoaudiólogos e radiologistas treinados, que realizam uma análise quadro a quadro de parâmetros espaço-temporais e quantitativos do vídeo para determinar a causa e o tratamento (dieta) adequado. Além do grande tempo gasto nesta etapa de análise, este procedimento está sujeito à variabilidade interpessoal na interpretação dos resultados.

Em abril de 2019, a Professora Catriona Steele e seus colegas publicaram o artigo “Reference Values for Healthy Swallowing Across the Range from Thin to Extremely Thick Liquids” no Journal of Speech, Language e Hearing Research [9]. Este trabalho estabeleceu valores de referência quantitativos para diversos parâmetros da fisiologia da deglutição em adultos saudáveis, desde líquidos finos até líquidos mais viscosos, conforme os níveis 0 a 4 da Iniciativa Internacional de Padronização de Dietas para Disfagia (IDDSI). A pesquisa de Steele et al. [9], também

definiu a criação e uso de um protocolo padronizado para coletar dados de estudo para o VFSE, e implementou um procedimento operacional padrão (SOP) para analisar o mecanismo de deglutição.

O resultado deste trabalho foi o desenvolvimento do Método ASPEKT (Analysis of Swallowing Physiology: Events, Kinematics, and Timing) que inclui definições padronizadas para toda a terminologia e medidas quantitativas necessárias para a pesquisa da videofluoroscopia da deglutição [9].

Nesse sentido, dado a complexidade do exame, a quantidade de tempo e esforço gastos pelos profissionais da saúde na realização deste procedimento e da análise dos resultados, a crescente oferta de recursos computacionais e os avanços de estudos das técnicas de IA no campo da medicina, como o deep learning, é possível usar inteligência artificial na classificação de exames de videofluoroscopia da deglutição, aumentando a precisão e automatizando os diagnósticos de disfagia, além de diminuir o tempo de análise do exame, permitindo que os profissionais da saúde concentrem seus esforços nos casos mais críticos, acelerando o tratamento e salvando vidas.

2. Contexto Atual

O Instituto Nacional de Câncer (INCA) realiza o estudo da deglutição através do exame de Videofluoroscopia utilizando o método ASPEKT. A disfagia possui correlação com o desenvolvimento do câncer de cabeça e pescoço (CCP), sendo o 5º tipo de câncer mais frequente em homens e o 13º mais frequente em mulheres no Brasil, no ano de 2020 [10] . Por esse motivo, o INCA vem desenvolvendo pesquisas nessa área que estão alinhadas com o desenvolvimento deste projeto.

O método ASPEKT, visto na *Figura 1*, inclui uma série de definições padronizadas para toda a terminologia e medições quantitativas utilizadas na análise do VFSE, com o objetivo de proporcionar consistência e precisão nas pesquisas sobre a fisiologia da deglutição.

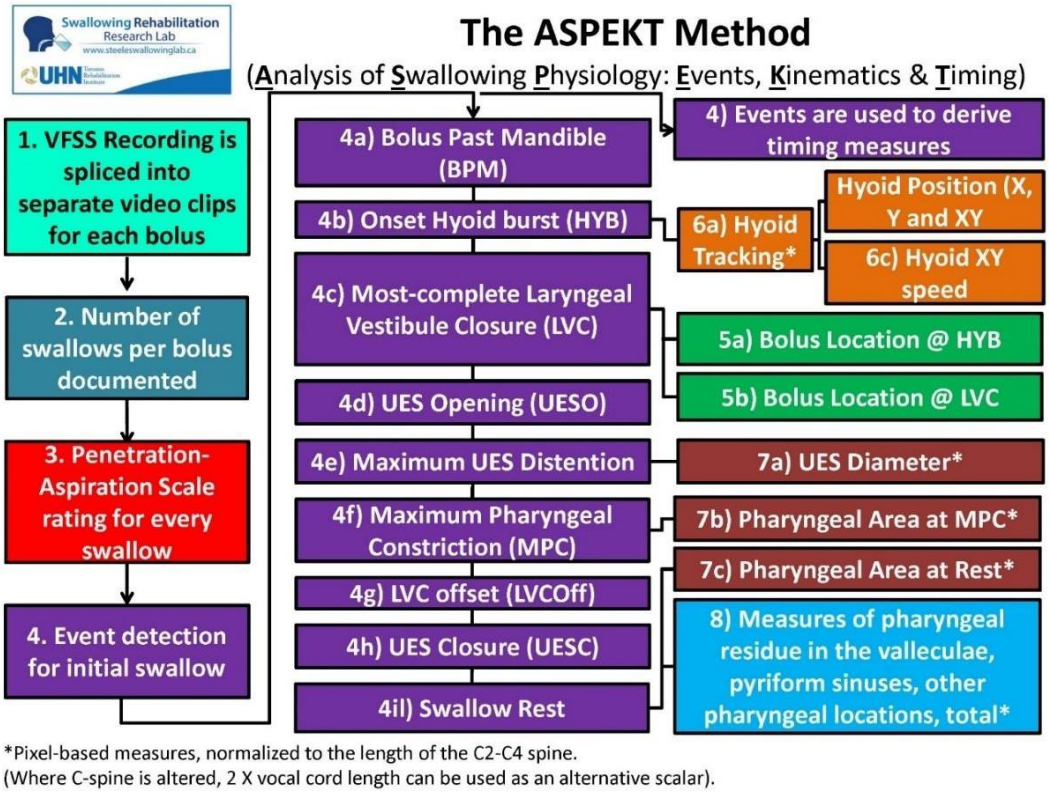


Figura 1 - Método ASPEKT (Steele, Peladeau-Pigeon et al., 2019)

Baseado nessa metodologia, foi desenvolvido para uso clínico o método ASPEKT-C, visto na *Figura 2*, sendo um processo mais curto e mais prático que o método ASPEKT completo [11]. O INCA utiliza este método para auxiliar os clínicos na identificação dos mecanismos que comprometem a segurança e a eficiência da

deglutição nos exames de Videofluoroscopia. O aspecto da segurança refere-se à capacidade de proteger as vias aéreas, enquanto a eficiência refere-se à capacidade de limpar o material residual através da faringe.

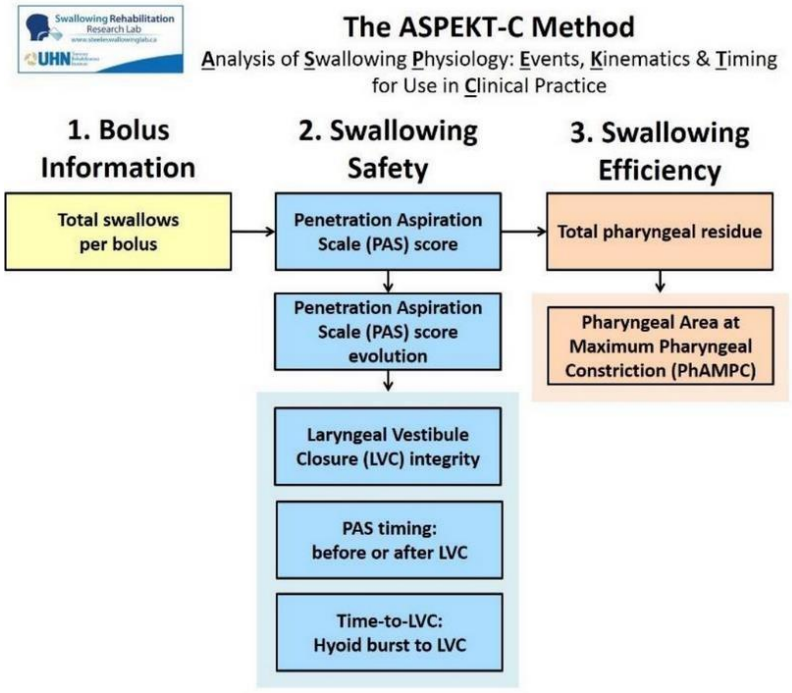


Figura 2 - Método ASPEKT-C (Steele Swallowing Lab © 2019)

O ASPEKT-C foca nos oito parâmetros-chave da deglutição vistos na Figura 2, que foram selecionados com base em dados que mostram sua relevância na explicação da aspiração traqueal do material ingerido e da penetração, que é a retenção de resíduos pós-deglutição, em pessoas com disfagia [11]. Além disso, inicialmente são determinados valores de referência para posterior análise dos resultados, que são obtidos sobre condições específicas e que determinam vários aspectos sobre o exame de Videofluoroscopia, tais como: a concentração de Bário, para dar o contraste na imagem, o volume e a viscosidade dos líquidos ingeridos e a deglutição espontânea do paciente [11]. Posteriormente, os dados obtidos após a realização do método, sobre as mesmas condições explicitadas anteriormente, são comparados com os valores de referência para a definição do diagnóstico.

Para a implementação deste método, o INCA realiza o VFSE capturando imagens a 30 frames por segundo, com uma resolução de 720x480 pixels por quadro. Na sequência, o clínico analisa cada imagem com o software ImageJ, buscando

destacar os eventos descritos no método ASPEKT, caso ocorram, e realizando medições sempre que necessário. Por fim, os resultados da análise são colocados em uma planilha padrão, que auxilia o clínico no diagnóstico e na definição do tratamento do paciente.

Outras instituições em diversos países também utilizam os métodos ASPEKT e/ou ASPEKT-C e realizam pesquisas visando automatizar o processo ou partes dele. Uma dessas partes do método que possui um grande foco é na área da segurança da deglutição, mais especificamente a etapa de classificação de um exame de um paciente saudável ou de um paciente com disfagia, isto é, um paciente com penetração (quando o alimento entra na laringe, mas não ultrapassa as cordas vocais, com retenção de resíduo no local) e/ou com aspiração (quando o alimento ultrapassa as cordas vocais e entra na traqueia, podendo chegar até os pulmões).

No estudo de Patrick Wilhelm et al. [12], os dados de 107 indivíduos, armazenados em 1154 VSFE, foram utilizados para treinar uma Rede neural convolucional recorrente de longo prazo (LRCN) com o objetivo de criar um classificador capaz de distinguir exames de pacientes normais (sem disfagia) e anormais (com disfagia, possível aspiração ou penetração). Segundos os autores, a acurácia do método de classificação foi de 85%.

No estudo de Jeoung Kun Kim et al. [13], os dados de 190 participantes com disfagia foram coletados e um total de 10 imagens de um processo de deglutição foram selecionados (cinco imagens da fase superior e cinco imagens da fase inferior da deglutição) para a aplicação de deep learning em um VFSE de um paciente com disfagia. Foi implementado uma rede neural convolucional (CNN) para a classificação dos achados do VFSE em três classes: deglutição normal, penetração ou aspiração. A classificação foi determinada tanto nas imagens da fase superior quanto nas da fase inferior. Posteriormente, as duas classificações obtidas isoladamente foram integradas em uma classificação final. Segundo os autores, a acurácia do modelo foi de 0,942 para achados normais, 0,878 para penetração e 1,000 para aspiração.

Já os estudos de Andrea Bandini et al. [14] e de Handenur Caliskan et al. [15] são pesquisas voltadas para a detecção e segmentação do bolus alimentar e apesar de não estarem voltadas para a classificação de eventos de disfagia, estes estudos podem servir de material auxiliar para essa etapa.

Por fim, no estudo desenvolvido por Eunhee Park et al. [16], foram investigados diferentes modelos de aprendizado profundo para a detecção automática de

penetração e aspiração em vídeos de estudos de deglutição videofluoroscópica (VFSS). Os autores conduziram uma análise comparativa entre diversas arquiteturas de Deep Learning, com resultados que demonstram o estado da arte para a tarefa. Estes resultados servirão como referência para comparar o desempenho dos modelos desenvolvidos neste projeto.

3. Proposta e Objetivos do Trabalho

A proposta deste projeto é construir modelos de deep learning para a análise automatizada de imagens/vídeos de exames de videofluoroscopia. O objetivo principal é classificar os casos de pacientes em três categorias: normais (sem disfagia), pacientes com penetração e pacientes com aspiração. Será utilizado a metodologia ASPEKT para padronizar as medições e as definições dos eventos fisiológicos de deglutição, garantindo a consistência e a precisão dos resultados.

Para este propósito, este projeto conta com o apoio do INCA, que disponibilizou uma base de dados composta por 100 vídeos de videofluoroscopia para o treinamento dos modelos.

Dessa forma, para alcançar o objetivo principal, alguns objetivos específicos são definidos abaixo.

- Treinamento e validação de rede neural para classificação das três categorias citadas anteriormente, utilizando os dados disponibilizados pelo INCA.
- Medir a acurácia do modelo utilizado.
- Validação da aplicabilidade clínica com o apoio de especialistas do INCA.

Os avanços buscados, em relação ao trabalho realizado nas instituições que realizam o estudo da disfagia, como por exemplo o INCA, permitirão:

- automatização do processo de trabalho, reduzindo a necessidade de intervenção manual na análise do VFSE, aumentando a eficiência e reduzindo o tempo de diagnóstico;
- aumento da precisão, utilizando redes neurais convolucionais para classificação, minimizando erros de diagnóstico; e
- acessibilidade, a partir do desenvolvimento de uma ferramenta que possa ser facilmente integrada em clínicas e hospitais, democratizando o acesso a diagnósticos avançados de disfagia.

Cabe ainda destacar que fonoaudiólogos, médicos especialistas em disfagia, profissionais da saúde que trabalham com o tratamento de deglutição e os pacientes com problemas de deglutição serão as pessoas influenciadas por este projeto.

4. Fundamentos Teóricos

Nas seções a seguir vão ser apresentados todos os conceitos teóricos utilizados no projeto. O entendimento desses fundamentos é necessário para compreender o método proposto.

4.1. Redes Neurais Artificias

Uma rede neural artificial (RNA) é um programa de aprendizado de máquina, ou modelo, que toma decisões de uma forma semelhante ao cérebro humano, utilizando processos que imitam a maneira como os neurônios biológicos trabalham juntos para identificar fenômenos, avaliar opções e chegar a conclusões[17]. Elas são compostas por neurônios artificiais que são as camadas de nós da rede, possuindo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada neurônio se conecta com os outros possuindo seu próprio peso e limiar (ou viés) associado e eles são ativados se a sua saída estiver acima do seu limiar, enviando os dados para a próxima camada da rede, caso contrário, nenhum dado é passado para a próxima camada. Essas redes precisam de dados de treinamento para melhorar sua precisão ao longo do tempo podendo recalcular seus pesos usando a técnica de retropropagação (backpropagation).

O funcionamento de um neurônio pode ser visto na *Figura 3*, quando os dados de entrada são recebidos pelas entradas (sinapses) do neurônio processador, os pesos são atribuídos as entradas, determinando a importância de cada entrada em relação as demais, com os maiores valores contribuindo de forma mais significativa para a saída. O processamento desses dados é uma combinação linear entre os pesos e as entradas, e o resultado é passado pela função de ativação que verifica se esse valor ultrapassou ou não o limiar estabelecido. Se o resultado ultrapassou o limiar, o neurônio dispara o valor 1 na saída binária, fazendo com que a saída de um nó seja a entrada do próximo nó, se não ultrapassar, a saída fica passiva em 0.

Esse processo de passagem de dados de uma camada para a próxima define a rede neural como uma rede feedforward ou perceptrons de múltiplas camadas (MLPs), no qual o fluxo da informação é unidirecional. Esse tipo de neurônio cuja saída é 0 ou 1 é chamado de perceptron e utiliza como função de ativação a função Heaveside (função de escada), onde a saída g(u) = 1 se x>= 0 ou g(u) = 0 caso

contrário. Porém, a maioria das redes neurais utilizam neurônios sigmoides que utilizam valores entre 0 e 1, cuja função de ativação é a função sigmoidal, que gera uma saída não-linear contínua. Desse modo, a RNA é uma entidade de processamento que calcula uma função de saída a partir das entradas e dos pesos, com uma função de ativação predefinida.

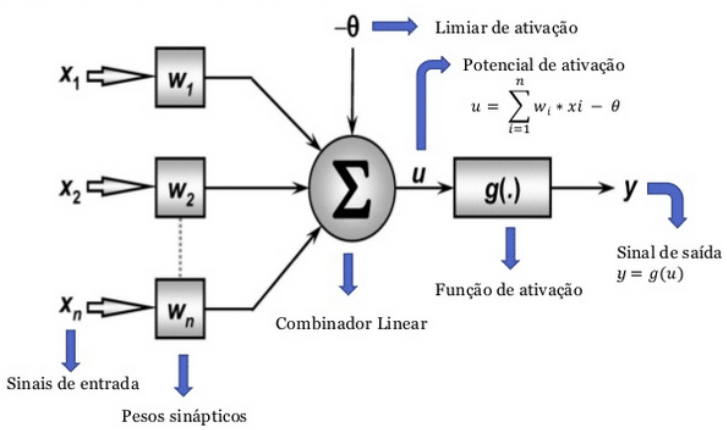


Figura 3 - Funcionamento de um Neurônio Artificial

O potencial e flexibilidade do cálculo baseado em redes neurais vêm da criação de conjuntos de neurônios que estão interligados entre si. Esse paralelismo de elementos com processamento local cria a “inteligência” global da rede. Um elemento da rede recebe um estimulo nas suas entradas, processa esse sinal e emite um novo sinal de saída para fora que por sua vez é recebido pelos outros elementos[18].

Uma RNA possui dois elementos principais, a arquitetura e o algoritmo de aprendizagem. Desse modo, na construção do modelo é essencial definir o tipo de rede que será utilizada para resolver o problema e qual o algoritmo utilizado para ajustar os pesos da rede, isto é, para treinar o modelo. As redes podem ser classificadas em tipos de acordo com a topologia dos seus neurônios e a forma de propagação da informação.

As redes de feedforward, ou perceptrons de múltiplas camadas, como descrito anteriormente, propagam a informação para frente em um fluxo unidirecional, onde neurônios que recebem informações simultaneamente pertencem a mesma camada e as camadas que não estão ligadas nem a entrada e nem a saída são chamadas de camadas ocultas. Esse tipo de rede é a base para a visão computacional e processamento de linguagem natural.

As redes neurais convolucionais (CNN’s) são parecidas com as redes feedforward mas são normalmente utilizadas para visão computacional e reconhecimento de imagens, utilizando princípios da álgebra linear, especialmente a multiplicação de matrizes, para identificar padrões dentro de uma imagem. Essa rede será detalhada em mais detalhes posteriormente.

As redes neurais recorrentes (RNN’s) são caracterizadas pelos loops de feedback, com ligações entre os neurônios sem restrição. São redes tipicamente usadas com dados de séries temporais para previsões futuras, onde o aspecto temporal é fundamental nesse tipo de rede. Essa rede também será detalhada em mais detalhes na sua seção mais à frente.

Definindo o tipo de rede, o próximo passo é estabelecer o tipo de algoritmo de aprendizagem utilizado para recalcular e adaptar os pesos entre os neurônios de maneira ótima durante o treinamento. Os dois tipos principais utilizados pelos sistemas com capacidade de adaptação são a aprendizagem supervisionada e a aprendizagem não supervisionada.

Na aprendizagem supervisionada são utilizados dados de treinamento rotulados para ensinar o modelo, ajustando os pesos para produzir o resultado esperado. Com a relação de entradas e suas respectivas saídas esperadas, o modelo consegue aprender com o tempo, medindo sua precisão pela função de perda, ajustando-se até que o erro seja minimizado. Conforme o modelo ajusta seus pesos e viés, ele utiliza a função de custo e o aprendizado por reforço para alcançar o ponto de convergência, ou o mínimo local. O processo pelo qual o algoritmo ajusta seus pesos é através do gradiente descendente, permitindo que o modelo determine a direção a tomar para reduzir os erros (ou minimizar a função de custo). Com cada exemplo de treinamento, os parâmetros do modelo se ajustam para convergir gradualmente para o mínimo[19]. Esse tipo de aprendizado é muito utilizado em problemas de classificação e previsão.

Na aprendizagem não supervisionada são utilizados dados não rotulados, usando algoritmos de machine learning para descobrir padrões que ajudam a resolver problemas de agrupamento ou associação. A principal característica desse tipo de aprendizagem é descobrir semelhanças e diferenças nas informações, sendo a solução ideal para análise exploratória de dados, estratégias de vendas cruzadas, segmentação de clientes e reconhecimento de imagem[20].

4.2. Redes Neurais x Deep Learning

Sistemas de deep learning, ou aprendizado profundo, são redes neurais mais complexas com uma arquitetura com muitas camadas ocultas. O conceito de rede neural está mais relacionada ao modelo feedforward, isto é, aos sistemas de alimentação direta. Nesse sistema, os dados passam de um nó de entrada para o próximo nó de saída e todos os nós de uma camada estão conectados a cada nó da próxima camada, possuindo apenas uma camada oculta além das camadas de entrada e saída. Este tipo de rede neural também é chamada de rede neural básica. Uma arquitetura de uma rede neural simples pode ser vista na *Figura 4*.

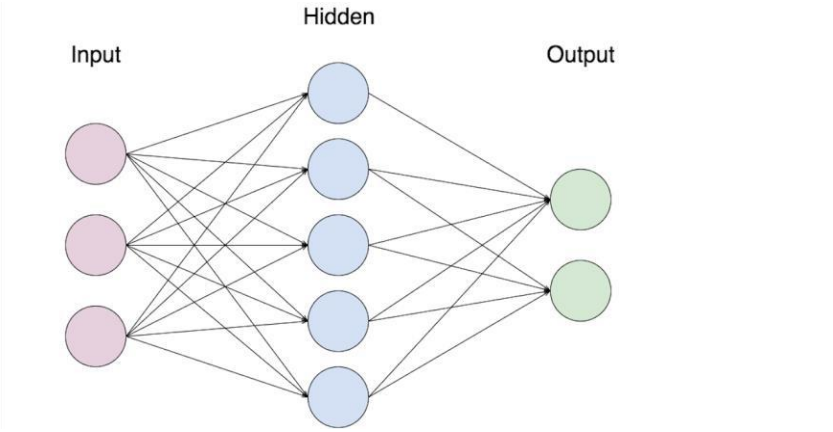


Figura 4 - Arquitetura de uma Rede Neural Simples

Por sua vez, os sistemas de deep learning possuem várias camadas ocultas o que os tornam profundos. Além disso, as arquiteturas mudam bastante de uma para a outra, seja em quantidade de camadas ocultas, seja no nível de complexidade, isto é, a quantidade de parâmetros da rede, incluindo pesos e vieses. Ademais, existem dois tipos principais de arquitetura de sistemas de aprendizado profundo, sendo elas as redes neurais convolucionais e as redes neurais recorrentes. Uma arquitetura de um sistema de deep learning pode ser vista na *Figura 5*.

Comparativamente, as redes de deep learning são mais complexas que as redes neurais simples, pois possuem mais camadas e mais parâmetros, sendo computacionalmente mais exigentes, podendo possuir estruturas altamente complicadas como memória de longo prazo (LSTM) e codificadores automáticos. Na parte de performance, um algoritmo de aprendizado profundo pode resolver problemas complexos em grandes volumes de dados, quando comparado as redes

neurais simples que funcionam melhor na resolução de problemas mais simples. Por fim, avaliando o custo de treinamento, treinar um algoritmo de aprendizado profundo custa mais recursos computacionais, mais dinheiro e mais tempo do que redes neurais simples, tendo mais capacidade para aprender padrões mais complexos em grandes volumes de dados[21]. A comparação das arquiteturas de uma rede neural simples e de um sistema de deep learning está presente na *Figura 6*.

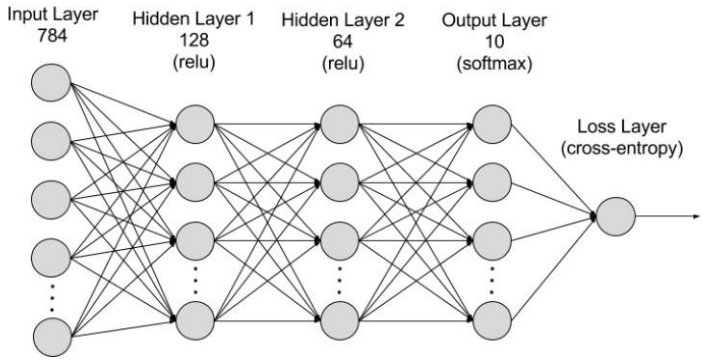


Figura 5 - Arquitetura de uma Rede Deep Learning

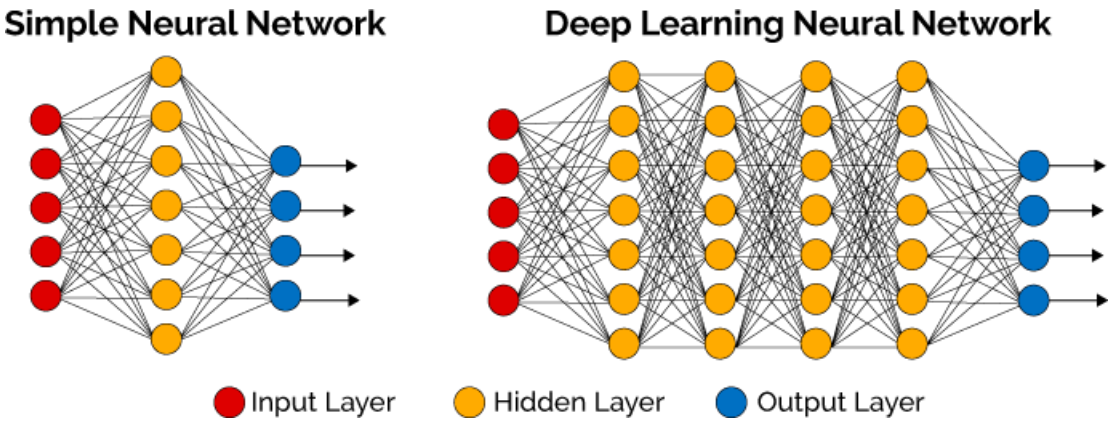


Figura 6 - Comparação Rede Neural Simples x Deep Learning

4.3. Redes Neurais Convolucionais

As redes neurais convolucionais (Convolutional Neural Network ou CNNs) utilizam dados tridimensionais para realizar tarefas de visão computacional, muito utilizadas para realizar a classificação de imagens e reconhecimento de objetos, possuindo um histórico de alta acurácia na resolução desses problemas [22]. Essas redes usam princípios de álgebra linear, como a multiplicação de matrizes, para

reconhecer padrões das imagens e realizar a extração de características principais, o que antes era feito de modo manual e demorado, sendo uma alternativa mais precisa e escalável. Por outro lado, uma CNN possui muitas camadas e muitos parâmetros, pois é uma rede de deep learning, sendo necessário muitos recursos computacionais, exigindo unidades de processamento mais robustos como GPU's (Unidades de Processamento Gráfico) e tempo para treinar um modelo.

O princípio básico de funcionamento de uma rede convolucional é filtrar linhas, curvas e bordas e em cada camada acrescida transformar essa filtragem em uma imagem mais complexa. Nas camadas iniciais, a rede se concentra em filtrar elementos simples das imagens, como cores e extremidades, e ao avançar pelas camadas, a rede identifica partes maiores, reconhecendo os elementos e formas características de um objeto de modo a classificá-lo ou identificá-lo corretamente. Além disso, as CNN's possuem normalmente três tipos de camadas, as camadas convolucionais, as camadas de agrupamento (Pooling) e camadas totalmente conectadas (Fully Connected). Desse modo, a camada convolucional é a camada de entrada dessa rede, podendo ser complementada por outras camadas convolucionais adicionais ou por uma camada de agrupamento, e a camada totalmente conectada é a camada final.

Para melhor compreender o funcionamento dessa a arquitetura e a função de cada camada é necessário entender o formato de entrada da rede. Para a classificação e reconhecimento de imagens, as entradas são usualmente matrizes tridimensionais com altura e largura (de acordo com as dimensões da imagem) e profundidade, determinada pela quantidade de canais de cores. Em geral, as imagens utilizam três canais RGB com os valores de cada pixel. Um exemplo de dado de entrada pode ser visto na *Figura 7*, com o formato de quatro pixels de largura e altura e o canal de três cores RGB para cada pixel.

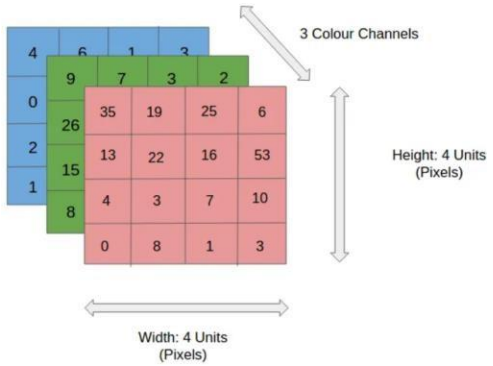


Figura 7 - Exemplo de Entrada de CNN

A camada convolucional é onde ocorre a maioria dos cálculos e funciona como um filtro que enxerga quadrados. Esse filtro vai passando por toda a imagem captando os traços mais marcantes, cujo processo é chamado de convolução, visto na *Figura 8*. Para seu funcionamento, essa camada precisa de alguns elementos como os dados de entrada, um filtro e um feature map. O filtro, também chamado de kernel, é formado por uma matriz bidimensional de pesos, que representa parte da imagem. Os pesos são inicializados aleatoriamente, sendo atualizados a cada nova entrada durante o processo de backpropagation. Dessa forma, o filtro se move pelos campos da imagem detectando se uma feição está presente nessa entrada.

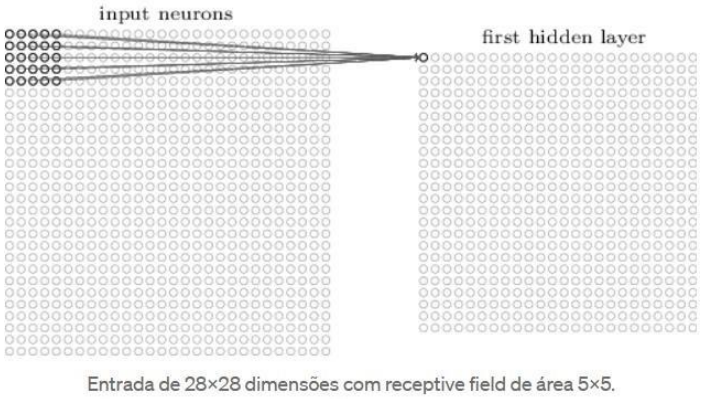


Figura 8 - Exemplo de Convolução de uma CNN

O tamanho dos filtros pode variar mais normalmente são usados matrizes de 3x3, o que também define o tamanho do campo receptivo, que é a pequena região da entrada onde o filtro é aplicado. No processo de convolução, quando o filtro é aplicado em uma região da imagem, um produto escalar é calculado entre o campo receptivo e o filtro, visto *Figura 9*, resultando em um número alto se tiver compatibilidade entre o filtro aplicado e essa parte da imagem, tendendo a números mais próximos de zero se não houver compatibilidade.

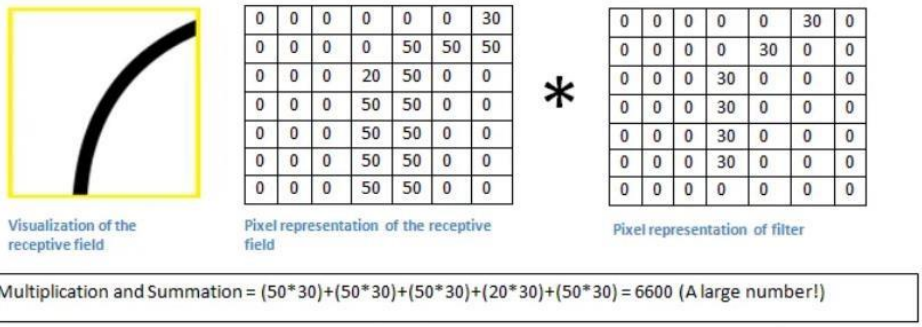


Figura 9- Exemplo de um Filtro Aplicado na convolução

Esse produto escalar calculado alimenta uma matriz de saída, visto *Figura 10*, e o filtro se movimenta por um avanço de acordo com uma quantidade predeterminada de saltos, chamado de stride, repetindo o processo até o filtro ter passado por toda a imagem. A matriz de saída alimentada pelo produto escalar do campo receptivo pelo filtro é chamado de feature map ou activation map.

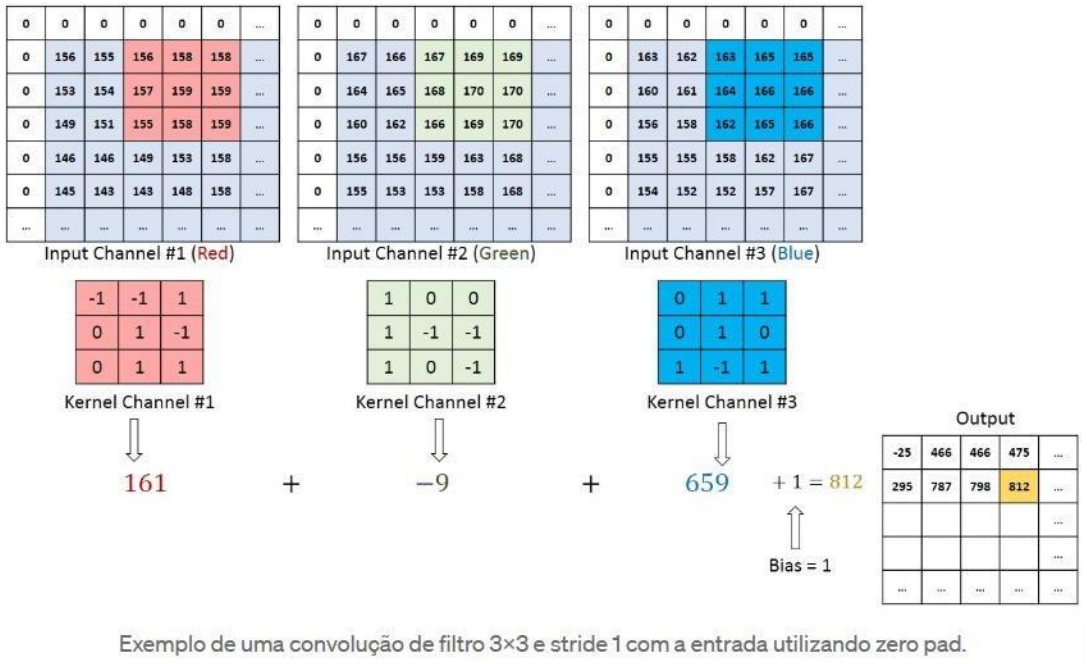


Figura 10 - Processo de Convolução de uma CNN e Construção do Feature Map

Além do tamanho e quantidade de filtros, e do número de strides da convolução, também deve ser definido previamente como hiperparâmetro o número do padding. O mais comumente utilizado é o Zero-padding, quando os filtros não se encaixam na imagem de input, definindo todos os elementos que ficam fora da matriz de input como zero, produzindo um resultado de tamanho maior ou igual. O padding serve para que as camadas não diminuam muito mais rápido do que é necessário para o aprendizado.

Após cada convolução, a CNN aplica uma transformação no feature map por meio de uma função de ativação. As funções de ativação servem para trazer a não-linearidades ao sistema, para que a rede consiga aprender qualquer tipo de funcionalidade. Existem muitas funções de ativação como a sigmoid, tanh e softmax, mas a mais utilizada é a ReLU (Unidade Linear Retificada) por ser mais eficiente computacionalmente.

Uma camada convolucional adicional pode seguir uma camada convolucional inicial, tornando a estrutura da CNN hierarquizada, permitindo que as camadas posteriores visualizem os pixels dentro do campo receptivo das camadas anteriores. Esse processo é semelhante aos princípios de divisão e conquista, dividindo o processamento em etapas menores e menos complexas com o objetivo de resolver um problema mais complexo. Neste caso, uma imagem é constituída por uma soma de suas partes, por exemplo, uma imagem que contém uma bicicleta é composta por frames (quadros) de guidões, rodas, pedais, entre outros. Cada parte individual compõe um nível inferior e menos complexo na rede neural e a soma de cada parte individual compõe um nível superior e mais complexo na rede neural. Portanto, o intuito final é permitir que a rede tenha a capacidade de interpretar a imagem e extrair os padrões relevantes.

As camadas de agrupamento, também conhecidas como Downsampling ou Pooling, servem para simplificar a informação da camada anterior de convolução, reduzindo dimensionalidade, isto é, reduzindo o número de parâmetros na entrada. Semelhante como ocorre na convolução, um filtro é aplicado no campo receptivo da saída da camada anterior, que é a entrada da camada de agrupamento, com o intuito de resumir a informação dessa área em um único valor, de acordo com uma função de agrupamento. Como o filtro dessa camada não possui pesos, o objetivo dessa camada é somente realizar a sumarização dos dados de modo a diminuir a complexidade, reduzindo a quantidade de pesos a seres aprendidos durante o treinamento e para evitar overfitting (sobreajuste). O overfitting ocorre quando o modelo aprende com boa precisão os dados de treinamento mas não consegue generalizar para outros dados como os de validação e teste. Como função de agrupamento, o método mais utilizado é o Max Pooling, no qual apenas o maior número da unidade onde o filtro está sendo aplicado é passado para a saída. Uma alternativa é utilizar o Average Pooling, no qual é calculado o valor médio dos valores dentro do campo receptivo.

A última camada de uma CNN é a camada totalmente conectada (Fully Connected). Nessa camada, cada nó está conectado a um nó da camada anterior e a saída são N neurônios, com N sendo a quantidade de classes do modelo para finalizar a classificação. Essa camada realiza a classificação final utilizando como base as características principais extraídas das imagens pelas camadas anteriores, utilizando uma função de ativação softmax para classificar as classes corretamente,

produzindo uma probabilidade entre 0 e 1 para cada uma das N classes do modelo.

Existem diversos tipos de redes convolucionais pré-treinadas e com suas respectivas arquiteturas. Dentre as principais, as CNN's que vão ser detalhadas e que foram utilizadas neste projeto são a VGG16 e a ResNet50.

4.4. VGG16

O VGG16 é uma rede neural convolucional muito utilizada para classificação de imagens. Ele foi criado pelos pesquisadores, Karen Simonyan e Andrew Zisserman, do Visual Geometry Group da universidade de Oxford que apresentaram este modelo no paper intitulado "*VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*" que ganhou o prêmio de primeiro e segundo lugares na detecção e classificação de objetos no evento ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) em 2014, que apresenta desafios de modelos de visão computacional[23]. Esse modelo se destaca pelo design mais simples e que utiliza filtros convolucionais menores (3x3), sendo muito popular por ser fácil de integrar em projetos de classificação de imagens pela técnica de transfer learning, utilizando os pesos do modelo já pré-treinado, possuindo uma alta acurácia na classificação de imagens da ImageNet.

A ImageNet é um projeto de um grande banco de dados visual desenvolvido para auxiliar pesquisas de software de reconhecimento visual de objetos. Essa base de dados possui mais de 14 milhões de imagens anotadas e mais de 20000 categorias que indicam qual objeto está presente na imagem[24].

A arquitetura do VGG16 pode ser vista na *Figura 11*, sendo composta por 16 camadas com pesos, com 13 camadas convolucionais e 3 camadas densas. Além disso, o VGG16 possui 5 camadas com Max Pooling, totalizando um total de 21 camadas, das quais 16 são camadas com parâmetros para aprendizado e treinamento. Ele recebe como entradas imagens de 244x244x3, respectivamente, altura, largura e canal de cores RGB.

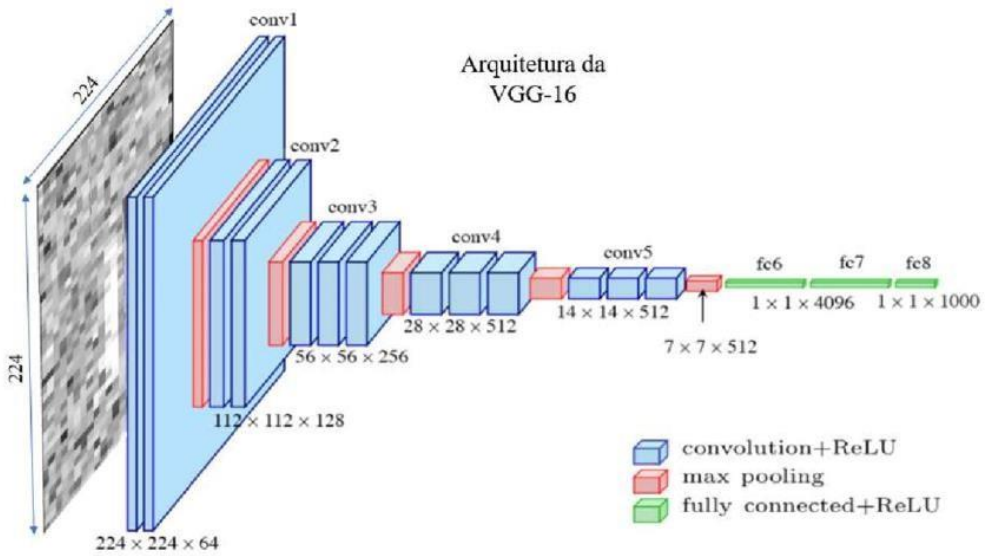


Figura 11 - Arquitetura VGG16

As camadas convolucionais possuem filtros de tamanho 3x3 e com stride de 1, as camadas de Max Pooling com filtros de tamanho 2x2 com stride de 2. Essas camadas estão sempre juntas por toda a arquitetura, visto *Figura 12*, de modo que a camada Conv-1 possui 64 filtros, Conv-2 possui 128 filtros, Conv-3 possui 256 filtros, Conv-4 e Conv-5 possuem 512 filtros. As duas primeiras camadas densas, totalmente conectadas ou fully connected, possuem 4096 canais cada e a última possui 1000 canais para realizar a classificação de cada classe das imagens da ImageNet com função de ativação softmax. As camadas convolucionais e as duas primeiras camadas densas possuem função de ativação ReLU.

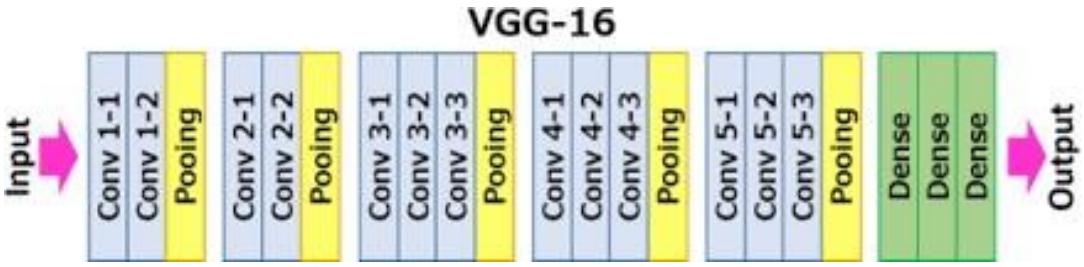


Figura 12 - Arquitetura VGG16 Simplificada

4.5. ResNet50

A Resnet50, abreviação de “Residual Network”, é uma rede neural convolucional desenvolvida pela Microsoft Research no artigo intitulado “*Deep Residual Learning for Image Recognition*” em 2015 [25]. Essa arquitetura possui 50 camadas e é muito poderosa para realizar a classificação de imagens, podendo ser treinada e alcançar resultados de estado da arte. Uma das principais inovações dessa arquitetura são os blocos residuais, que permitem treinar redes muito mais profundas do que era previamente possível, sem sofrer o problema do desaparecimento do gradiente[25].

Esse problema ocorre no treinamento de redes neurais profundas, quando os gradientes de parâmetros nas camadas profundas se tornam muito pequenos, dificultando ou impedindo com que a rede aprenda na hora da atualização dos pesos, de modo que o resultado pode ser cada vez pior conforme a rede vai ficando mais profunda. A Resnet50 resolve esse problema utilizando os blocos residuais com conexões de atalho (skip connections), permitindo que a informação e os gradientes saltem algumas camadas, estabelecendo um fluxo da entrada até a saída da rede, auxiliando no treinamento e aprendizagem da rede.

Na arquitetura, as skip connections são utilizadas nas camadas convolucionais e nos “Identity Block”, que preservam a entrada original sem alterar sua dimensionalidade.

A arquitetura da Resnet50 vista na *Figura 13,* aplica na entrada o Zero Padding, adicionado bordas de zeros ao redor das imagens para preservar dimensões. Além disso, o primeiro estágio (Stage 1) possui uma camada convolucional inicial com filtros 7x7 e stride de 2, uma normalização de lotes (Batch Norm) para estabilizar o treinamento, com função de ativação ReLU, e uma camada de Max Pooling para reduzir dimensionalidade espacial. Os estágios residuais (Stages 2 a 5), possuem vários blocos residuais cada um, com cada bloco sendo por composto por convoluções 1x1, 3x3 e 1x1, além de Normalização e ativação ReLU após cada camada convolucional. Ademais, o estágio 2 contém 3 blocos residuais com saídas 64,64 e 256 canais, o estágio 3 contém 4 blocos com saídas 128,128 e 512 canais, o estágio 4 contém 6 blocos com saídas 256,256 e 1024 canais, e o estágio 5 contém 3 blocos residuais com saídas 512,512 e 2048 canais. Por fim, a parte final da rede possui uma camada de global Average Pooling, uma camada de Flattening, que

achata a matriz resultante em um vetor, e uma camada totalmente conectada, ou Fully Connected, para realizar a classificação final, com número de saídas igual ao número de classes do problema.

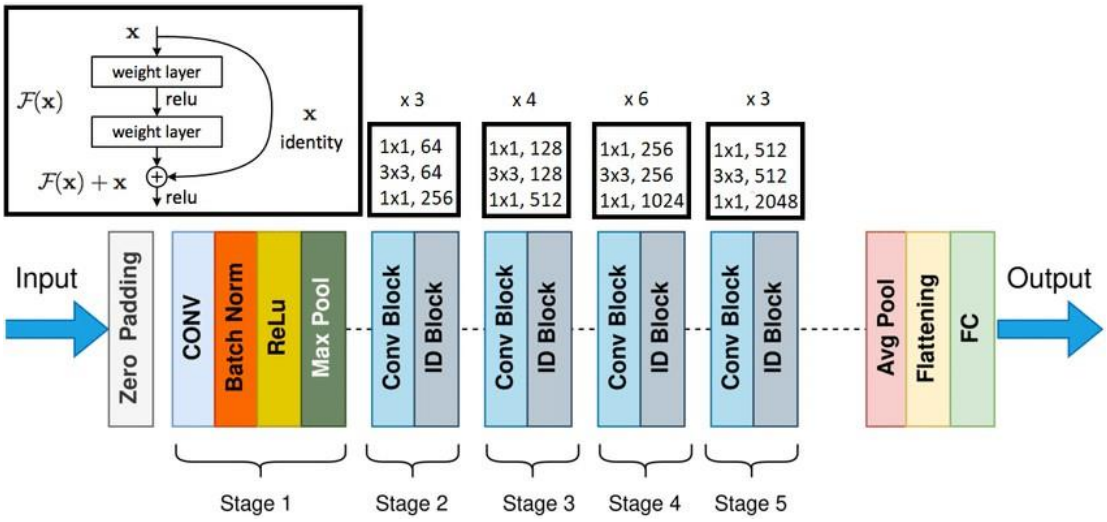


Figura 13 - Arquitetura ResNet50

4.6. Redes Neurais Recorrentes

Uma rede neural recorrente (RNN), *Figura 14*, é uma rede de deep learning que é treinada em dados sequenciais ou de séries temporais, de modo a fazer previsões ou conclusões sequenciais com base em inputs sequenciais[26]. Ela foi criada por pesquisadores do *Massachusetts Institute of Technology* (MIT) entre 1980 e 1990. As RNN's são utilizadas na resolução de problemas como reconhecimento de fala, tradução de textos, processamento de linguagem natural e classificação de vídeos.

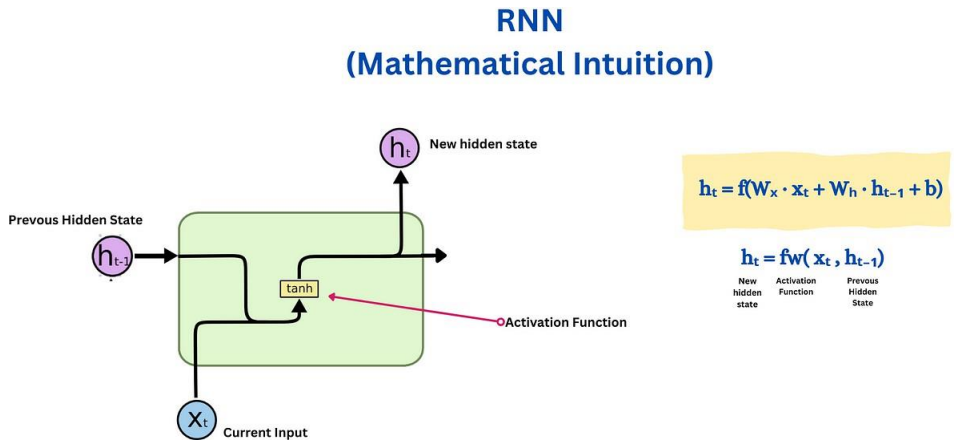


Figura 14 - Arquitetura RNN

A diferença principal da rede recorrente para as demais redes apresentadas é o fluxo de trabalho em loop automático ou recorrente, possuindo uma “memória”. Essa rede não considera apenas a entrada atual para produzir a saída atual, mas sim, considera entradas anteriores nos inputs e outputs atuais, de modo que a saída de uma camada oculta utilize a entrada atual mais a memória armazenada. Outra diferença é que essas redes compartilham os mesmos pesos em cada camada da rede, não atribuindo pesos diferentes para cada nó, de forma que esses pesos são atualizados pela retropropagação e gradiente descendente. A retropropagação utilizada pelas redes recorrentes é diferente do tradicional, utilizando algoritmos de retropropagação no tempo (BPTT) que reverte a saída para a etapa de tempo anterior, recalculando a taxa de erro, podendo identificar qual estado oculto na sequência está causando um erro significativo, e assim, reajustar o peso para reduzir a margem de erro.

A funcionalidade principal da utilização de uma RNN é tirar conclusões com base no aspecto temporal dos dados. A seguir, nas *Figuras 15 e 16*, temos dois exemplos de uso dessa rede na classificação de vídeos de ações humanas, onde o modelo recebe como entrada os frames de um vídeo de modo sequencial e tenta classificar a ação com base nesse input sequencial, utilizando o aspecto temporal para distinguir e detectar as ações.

Exemplo 1 - Entrada: Um vídeo com dois frames de uma pessoa sentando-se na cadeira. Sequência na direção da esquerda para a direita.

Figura 15 - Primeiro Exemplo de Uso de RNN

Exemplo 2 - Entrada: Um vídeo com dois frames de uma pessoa levantando-se da cadeira. Sequência na direção da esquerda para a direita.

Figura 16 - Segundo Exemplo de Uso de RNN

Nesses exemplos, é evidente a importância do modelo recorrente para classificação, de modo que a diferença entre as ações de se sentar e levantar está na ordem sequencial dos frames. Para a ação de levantar-se, temos uma pessoa sentada na cadeira e depois, no frame seguinte, ela está em pé na frente da cadeira. De forma análoga, o ato de se sentar é o oposto de levantar-se, sendo o primeiro frame da sequência a pessoa em pé na frente da cadeira e depois o frame dela sentada na cadeira. Desse modo, analisar a temporalidade é essencial para distinguir as ações, uma tarefa que uma rede recorrente faz muito bem.

Existem alguns tipos de arquiteturas nas redes neurais recorrentes, sendo o mais comum a arquitetura um para um, vista na *Figura 17*, de modo que uma sequência de entrada é associada a uma saída[27].

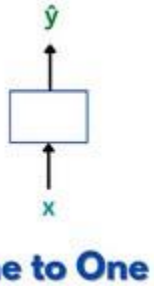


Figura 17 - Arquitetura RNN um para um

Outras arquiteturas são um para muitos, *Figura 18*, muitos para muitos, *Figura 19*, e muitos para um, *Figura 20*. Uma RNN um para muitos produz várias saídas a

partir de uma entrada, sendo muito utilizada para geração de frases a partir de uma única palavra-chave.

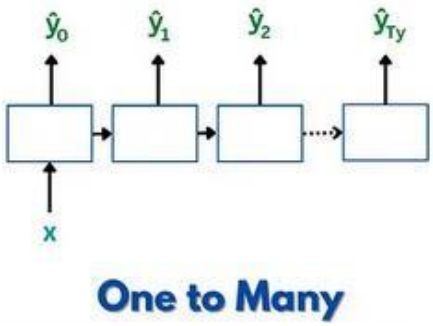


Figura 18 - Arquitetura RNN um para muitos

A arquitetura muitos para muitos gera várias saídas a partir de várias entradas, normalmente o número de inputs e outputs é o mesmo e é muito utilizada para tradução de idiomas, recebendo uma frase e traduzindo as palavras em uma estrutura correta em outro idioma.

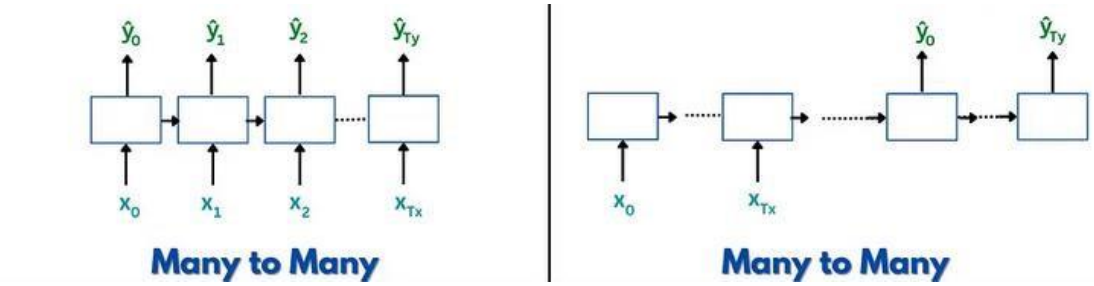


Figura 19 - Arquitetura RNN muitos para muitos

Por último, a arquitetura muitos para um gera uma saída a partir de várias entradas, muito utilizada para classificação de vídeos, recebendo os diversos frames de um vídeo como entrada e produzindo uma classe de saída.

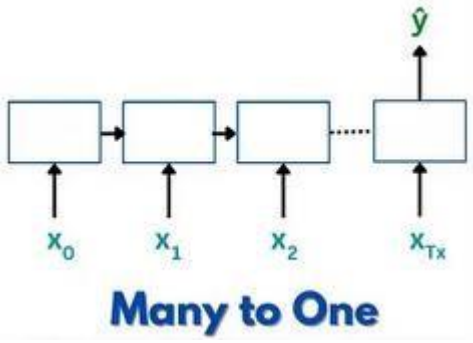


Figura 20 - Arquitetura RNN muitos para um

As principais limitações de uma rede neural recorrente são o problema do gradiente de desaparecimento, já explicitado na seção da arquitetura da ResNet50, e o problema do gradiente em explosão. Esse último problema ocorre quando o gradiente, a inclinação da função de perda ao longo da curva de erro, é muito grande, fazendo com que a rede tenha baixa acurácia no início do treinamento, necessitando de muitas épocas de treinamento para ajustar os pesos do modelo com o intuito de reduzir o erro, criando um modelo instável. Uma das consequências principais desse problema é o overfitting, que reduz a performance da rede, sendo necessário diminuir o número de camadas ocultas para reduzir a complexidade. De modo geral, as RNN's precisam de muitos recursos computacionais e de tempo para serem treinadas.

Entre as arquiteturas variantes da RNN que compartilham seu princípio de retenção de memória e melhoraram sua funcionalidade original é possível destacar as redes neurais recorrentes bidirecionais (BRNN's), a memória de curto e longo prazo (LSTM) e as unidades recorrentes fechadas (GNU's). Neste projeto, só foi utilizada e detalhada a arquitetura LSTM.

4.7. LSTM

A memória de curto e longo prazo, *Figura 21*, é uma arquitetura de RNN criada por Sepp Hochreiter e Juergen Schmidhuber em 1997 para resolver o problema do gradiente de desaparecimento[28]. Eles buscaram solucionar o problema de dependências de longo prazo, porque a RNN tradicional só consegue lembrar da entrada anterior imediata para influenciar na saída atual. Dessa forma, o LSTM adicionou um bloco de memória espacial chamado de células (cells) nas camadas ocultas, permitindo usar entradas de várias sequências anteriores, que não estão apenas no passado recente, para melhorar a saída atual.

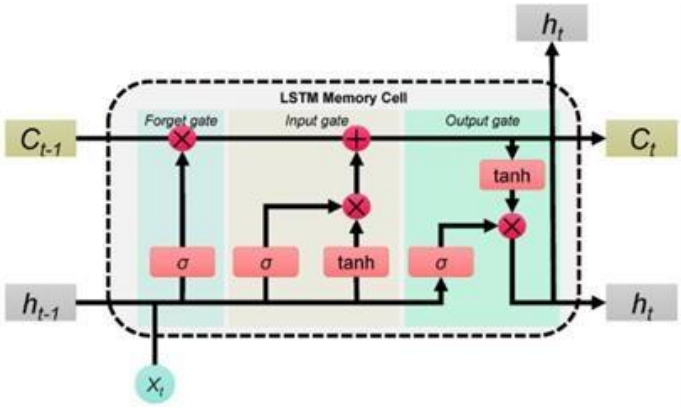


Figura 21 - Arquitetura LSTM

Cada uma dessas células de memória são controladas por uma porta de entrada, uma porta de saída e uma porta de esquecimento, possibilitando com que o modelo se lembre de informações úteis, controlando o fluxo de informações necessárias para produzir uma saída melhor. A porta de esquecimento decide quais informações armazenadas na célula de memória serão descartadas, baseando-se na entrada atual e no estado oculto anterior. A porta de entrada controla quais novas informações da entrada atual serão adicionadas à célula. Por último, a porta de saída determina quais informações da célula serão usadas para atualizar o estado oculto e gerar a saída atual.

4.8. Redes CNN com LSTM

Em problemas de classificação de vídeos de exames médicos são frequentemente utilizados modelos que combinam as redes convolucionais com as redes recorrentes. A ideia é aplicar o modelo CNN em cada uma das imagens de um vídeo, de modo a extrair as características e traços principais dos frames, e passar a saída dessa rede pelo LSTM, que irá analisar a temporalidade de cada frame com o objetivo de entender o aspecto sequencial do vídeo para realizar a classificação final dele.

No artigo intitulado "*Classification of benign and malignant subtypes of breast cancer histopathology imaging using hybrid CNN-LSTM based transfer learning*" desenvolvido por Mahati Munikoti Srikantamurthy et al., é demonstrado um exemplo da combinação dos modelos, *Figura 22*, para classificar os subtipos de câncer de mama benignos e malignos em imagens histopatológicas, possuindo uma acurácia de 99% para classificação binária entre as duas classes de câncer[29].

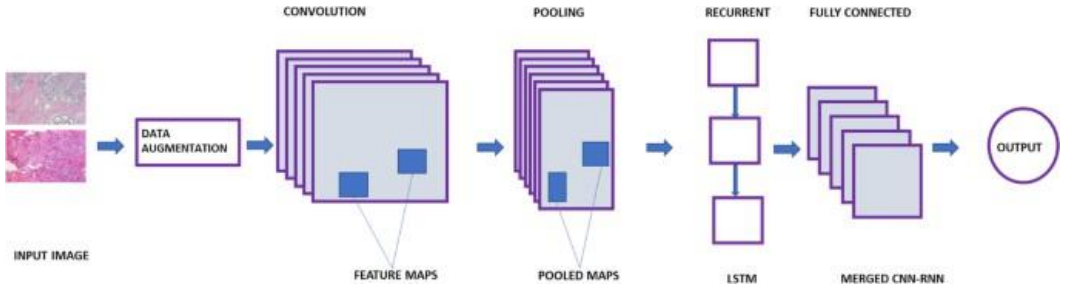


Figura 22 - Primeiro Exemplo de Uso da Arquitetura CNN com LSTM

Outro exemplo pode ser observado no artigo intitulado "*Pulmonary COVID-19:*

Learning Spatiotemporal Features Combining CNN and LSTM Networks for Lung Ultrasound Video Classification" desenvolvido por Bruno Barros et al., que combina ambas as arquiteturas, visto *Figura 23*, e classifica vídeos de ultrassom pulmonar para diagnosticar COVID-19 em pacientes, possuindo uma acurácia de 93% [30].

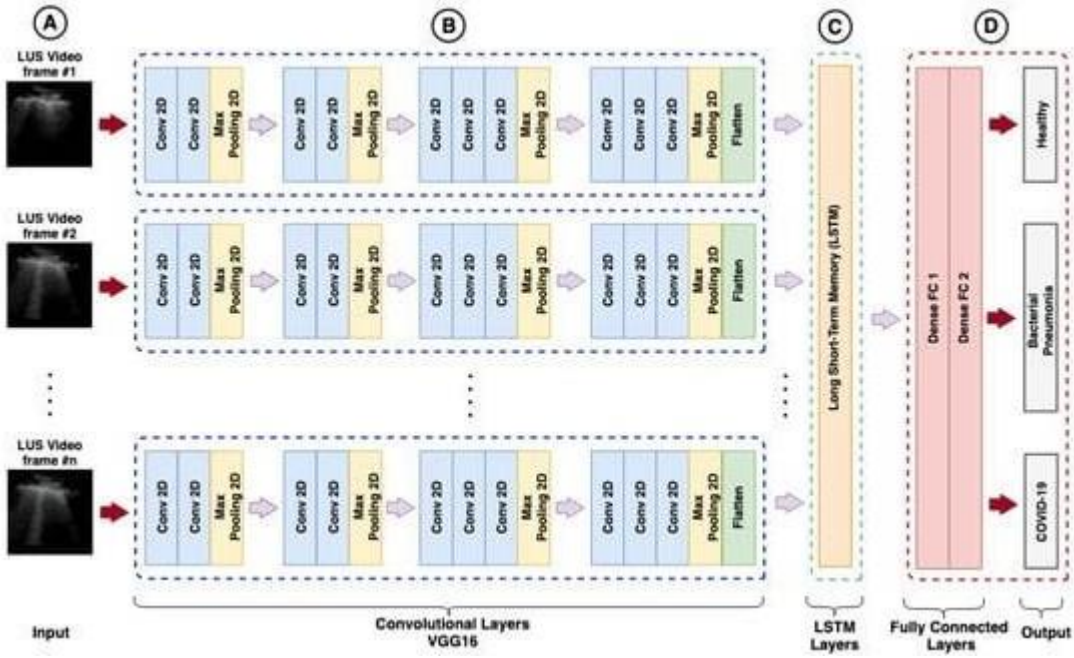


Figura 23 - Segundo Exemplo de Uso da Arquitetura CNN com LSTM

Esses artigos demonstram a força ao combinar o CNN e o LSTM para classificação de vídeos na área médica, uma alternativa testada neste projeto.

5. Metodologia

Este projeto propõe diversos métodos, utilizando modelos de deep learning, para realizar a classificação de vídeos de videofluoroscopia da deglutição. Os vídeos e os rótulos foram disponibilizados pelo INCA e mais de 50 modelos foram treinados visando a otimização da acurácia da classificação. Foram utilizados vários tipos de pré-processamento dos dados e algumas redes pré-treinadas como backbone, como VGG16, Resnet50, EfficientNetB0, entre outras. Desses mais de 50, foram selecionados modelos de destaque, que mesmo alguns não possuindo boa acurácia de classificação, são importantes pois possuem aspectos interessantes para comparação e compreensão do problema, e colaboraram para a construção da melhor solução encontrada. O ambiente computacional utilizado para realizar os experimentos foi o sistema operacional do Windows 11, com Python 3.11 como linguagem de programação e com TensorFlow 2.18 e Keras 3.6 como frameworks.

5.1. Dataset

O dataset utilizado foi o mesmo para todos os modelos, modificando apenas a quantidade de vídeos utilizados para treino, validação e teste. O INCA disponibilizou 100 vídeos de videofluoroscopia da deglutição e uma planilha Excel, *Figura 24*. Essa planilha possui uma coluna com o número de cada vídeo, de 1 a 100, além de colunas contendo informações sobre cada vídeo. Essas colunas são a coluna do PAS, escala de 1 a 8 que avalia o grau de penetração e aspiração da deglutição, a coluna do PAS FRAME, que indica o número do frame do vídeo que evidencia a ocorrência do PAS e o possível comprometimento das vias áreas, a coluna FRAME DE MÁXIMA CONSTRIÇÃO FARÍNGEA, que indica o número do frame onde a faringe está na sua contração máxima e a coluna FRAME DE REPOUSO que indica o número do frame do final da deglutição e relaxamento da faringe. Dentre todas as colunas, apenas a coluna FRAME DE REPOUSO não foi usada no trabalho. Além dos 100 vídeos, também foi feito data augmentation com o intuito de ampliar o dataset. As técnicas de augmentation aplicadas foram variação de brilho, parâmetros de 0.8 e 1.2, variação de contraste, escala entre 0.8 e 1.2, deslocamento horizontal e vertical entre -5 e +5 pixels, zoom de escala entre 0.9 e 1.1, rotação entre -10 e +10 graus e ruído gaussiano de escala entre 5 e 15.

VIDEO	PAS	PAS FRAME	FRAME DE MÁXIMA	FRAME DE REPOUSO
1	1	N/A	104	116
2	1	N/A	66	99
3	1	N/A	63	96
4	1	N/A	54	80
5	8	70	80	176
6	2	42	51	SEQ
7	1	N/A	28	124
8	8	125	144	175
9	1	N/A	75	93
10	1	N/A	46	76
11	1	N/A	44	75
12	1	N/A	72	94
13	8	82	92	SEQ
14	8	56	65	96
15	1	N/A	105	135
16	8	70	82	115
17	1	N/A	33	60
18	1	N/A	98	SEQ
19	1	N/A	37	51
20	4	46	44	SEQ

Figura 24 - Planilha com Informações dos Vídeos do Inca

Para definir o rótulo das classes no treinamento supervisionado entre normal, penetração e aspiração, foi utilizado a coluna do PAS, de modo que os vídeos de PAS 1 e 2 receberam a classe normal, de PAS de 3 até 6 receberam classe de penetração e de PAS 7 e 8 receberam classe de aspiração. Desse modo, ao final da definição das classes para cada vídeo é possível observar que o dataset é desbalanceado, *Figura 25*, com uma proporção de 71 vídeos com a classe normal, 15 vídeos com a classe de aspiração e 14 vídeos com classe de penetração, visualizando de outro modo, uma divisão de quase 70% normal e 30% anormal.

```
normal      71
aspiracao   15
penetracao  14
```

Figura 25 - Proporção das Classes no Dataset

Após a definição dos rótulos para cada vídeo, um pré-processamento foi feito para cada modelo, cada um com suas particularidades na formação dos conjuntos de treino, validação e teste; na seleção dos frames dos vídeos; no tipo de arquitetura definida e no tamanho do input de dados das redes.

5.2. Modelo VGG16 com LSTM com janelas deslizantes de 25 frames

Para o modelo VGG16 com LSTM foi utilizado um pré-processamento específico utilizando o conceito de janelas deslizantes. Primeiramente, os 100 vídeos foram divididos em conjuntos de treino, validação e teste. Como o conjunto de dados é desbalanceado para a classe normal a técnica de undersampling foi aplicada na classe majoritária, reduzindo a quantidade de vídeos normais no conjunto de treinamento. Dessa forma, o conjunto de treino ficou com 15 vídeos com a mesma quantidade de vídeos de cada classe, sendo 5 vídeos normais, 5 de aspiração e 5 de penetração. De forma análoga, o conjunto de validação ficou com 15 vídeos com a mesma proporção. Por último, o conjunto de teste ficou com o restante dos vídeos, possuindo 70 vídeos no total, sendo 61 normais, 4 de penetração e 5 de aspiração.

Após a definição dos conjuntos, a técnica de janelas deslizantes foi aplicada nos vídeos de cada conjunto. Essa técnica consiste em agrupar frames consecutivos em um vetor (janela ou window) e deslocar essa janela de acordo com o número de steps, de modo que cada janela recebe um label para treino, validação e teste e o modelo faz a classificação dessas janelas. Por exemplo, para um vídeo de 100 frames com um tamanho de janela (window size) de 10 frames e step de 1, com os frames começando do índice 0 até o índice 99, a primeira janela pegaria os frames do índice 0 até o índice 9 e receberia um label1, a segunda janela pegaria os frames do índice 1 até o índice 11 e receberia um label2, e assim sucessivamente até a última janela, que pegaria os frames do índice 90 até o índice 99.

O objetivo dessa abordagem é passar o vídeo todo como input do modelo. Com o intuito de reduzir o custo computacional de memória e de tempo de treinamento, o tamanho da entrada foi padronizando. Outro ponto importante é que o modelo vai se especializar em classificar as janelas do vídeo, possibilitando classificar o vídeo completo no teste com a técnica de votação das janelas. A votação funciona da seguinte forma, se uma determinada quantidade de janelas consecutivas possuírem uma das classificações de problema, isto é, de penetração ou aspiração, então essa será a classificação final do vídeo, senão a classificação final é a normal. Desse modo, utilizar as janelas deslizantes permite em teoria que o modelo possua uma boa acurácia no teste mesmo não tendo uma boa acurácia no treino e validação, além de conseguir identificar em qual região, ou janelas, foi identificado o problema.

Este modelo utilizou um tamanho de janela de 25 frames com step de 1. Na *Figura 26*, é exposta uma janela de 25 frames do vídeo número 5 do conjunto de treino com rótulo de aspiração.

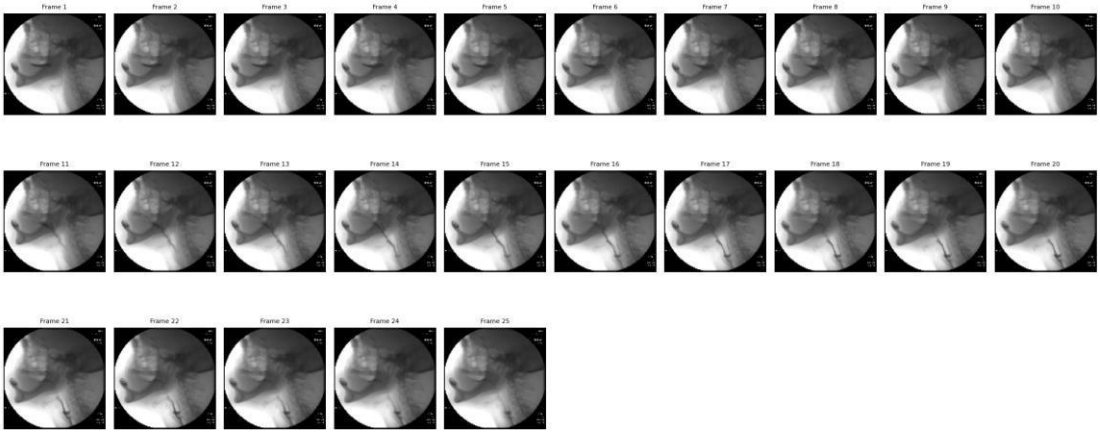


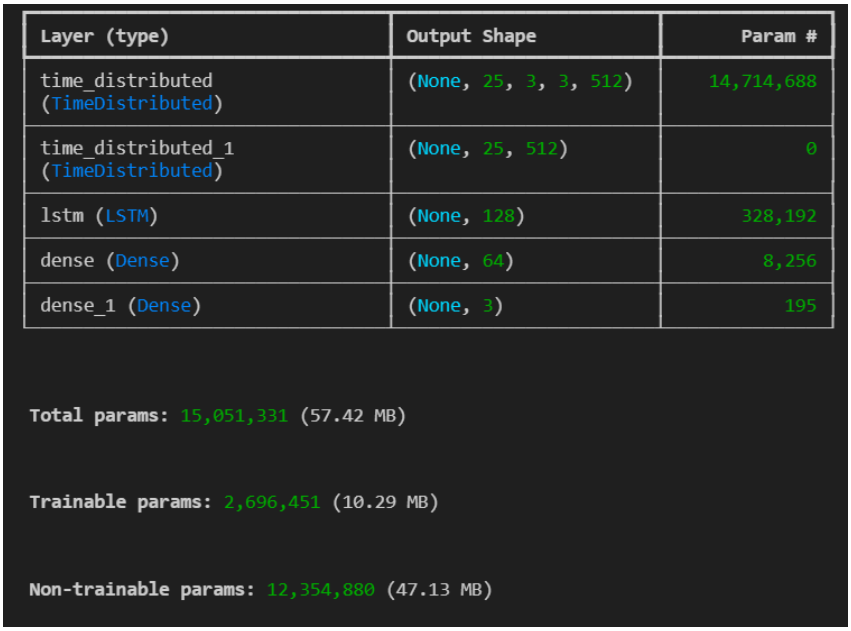
Figura 26 - Janela Deslizante de 25 Frames

Nesse modelo, todas as janelas de um vídeo normal receberam o label normal. Por outro lado, nos vídeos de aspiração e penetração, todas as janelas que possuíssem o PAS frame receberam o label referente ao seu vídeo, aspiração ou penetração, e todas as outras janelas receberam o label normal.

Essas janelas criadas para todos os vídeos vão ser a entrada do modelo descrito a seguir. Primeiramente, o modelo recebe como entrada um tensor de 5 dimensões com o seguinte formato (quantidade de janelas totais, tamanho da janela ou quantidade de frames por janela, altura das imagens, largura das imagens, número de canais de cor de cada frame) e os labels referentes a cada entrada com o formato (quantidade de janelas totais, número de classes). Para o treino, o formato dos dados de entrada e dos labels foi de, respectivamente, (2460, 25, 224, 224, 3) e (2460, 3). Para a validação, o formato da entrada e dos labels foi de, respectivamente, (2207, 25, 224, 224, 3) e (2207, 3).

A arquitetura do modelo utiliza o VGG16 pré-treinado, sem camada de topo e com os pesos da imagenet, como a parte convolucional da rede, sendo a primeira estrutura da arquitetura. Das 16 camadas do VGG16, apenas as duas últimas camadas foram descongeladas para treinamento, permitindo o ajuste de seus pesos. Essa primeira camada é seguida de uma camada de Pooling, a GlobalAveragePooling2D, e tanto essa camada quanto a anterior são envolvidas por uma camada de TimeDistributed, que aplica a rede convolucional VGG16 e o Pooling

em cada janela de entrada. A camada de Pooling tem como sucessor 128 unidades de LSTM com return sequences como falso. Por fim, a rede termina com 2 duas camadas densas, a primeira com 64 unidades e função de ativação relu e a última com 3 unidades, uma para cada classe, com ativação softmax para realizar a classificação e produzir como saída um tensor com a probabilidade de cada classe para cada janela. Essa arquitetura pode ser vista na *Figura 27*, com 15 milhões de parâmetros totais dos quais quase 2,7 milhões são treináveis.



Layer (type)	Output Shape	Param #
time_distributed (TimeDistributed)	(None, 25, 3, 3, 512)	14,714,688
time_distributed_1 (TimeDistributed)	(None, 25, 512)	0
lstm (LSTM)	(None, 128)	328,192
dense (Dense)	(None, 64)	8,256
dense_1 (Dense)	(None, 3)	195

Total params: 15,051,331 (57.42 MB)

Trainable params: 2,696,451 (10.29 MB)

Non-trainable params: 12,354,880 (47.13 MB)

Figura 27 - Arquitetura VGG16 com Janelas Deslizantes de 25 Frames

Para treinar essa rede foram utilizados como hiperparâmetros o otimizador Adam com métrica de acurácia e a função de loss (perda) categorical crossentropy. A rede foi treinada por 30 épocas com batch size de 16 e com EarlyStopping, salvando o modelo com melhor acurácia de treino e validação. Os resultados referentes a esse modelo podem ser vistos na seção 6.1.

5.3. Modelo VGG16 com LSTM com 15 janelas deslizantes de 25 frames

Esse modelo é uma variação do anterior com grandes diferenças no pré-processamento e algumas diferenças na arquitetura. A primeira grande mudança está na separação dos conjuntos de treino, validação e teste e no processo de seleção

das janelas. Durante o treinamento e no teste foi observado uma grande dificuldade de evolução do modelo anterior no quesito de aprendizagem e generalização. Esse problema está relacionado à pequena quantidade de dados no treino e a uma escolha ruim das janelas deslizantes. A maioria das janelas tinham a classificação normal, pois na maior parte dos vídeos não acontecia nada, isto é, tinham muitas janelas onde o paciente não havia iniciado o processo de deglutição ou ele já havia concluído este processo, e uma minoria de janelas onde o processo da deglutição ocorria, 25 janelas especificamente. Desse modo, a quantidade de vídeos foi aumentada no conjunto de treino e não foi mais passado o vídeo todo como entrada, mas sim, apenas janelas da fase faríngea, região do vídeo onde ocorre a deglutição do bolo alimentar.

O novo conjunto de treino agora possui 70 vídeos, sendo 50 normais, 10 de aspiração e 10 de penetração, o conjunto de validação agora possui 15 vídeos, sendo 10 normais, 3 de aspiração e 2 de penetração e o conjunto de teste também possui 15 vídeos, sendo 11 normais, 2 de aspiração e 2 de penetração.

Esse modelo também usa CNN mais LSTM para classificar janelas contendo uma sequência temporal de 25 frames, mas agora são 15 janelas obtidas deslizando janelas de 25 frames entorno de um frame de referência. O frame de referência depende da classe do vídeo. Para vídeos normais, o frame de referência é o frame de repouso, obtido pela planilha Excel, e para os vídeos de aspiração e penetração, o frame de referência é o PAS frame.

Todas as janelas de um mesmo vídeo recebem o mesmo label de classificação (normal: 0, penetração: 1 e aspiração: 2). Sendo assim, seriam 1050 janelas na entrada, mas na prática o número foi de 1044 janelas, tendo em vista que algumas janelas não conseguiram obter 25 frames completos. Desse modo, a entrada da rede recebe os dados e os labels, respectivamente, no formato (1044, 25, 224, 224, 3) e (1044, 3) no treino e no formato (225, 25, 224, 224, 3) e (225, 3) na validação.

A arquitetura agora é formada pelo VGG16 com todas as camadas congeladas, utilizando os pesos da rede pré-treinada, sem camada de topo e com GlobalAveragePooling2D aplicado na saída. Além disso, também possui o LSTM com 128 unidades, do tipo “many to one” e com dropuot de 0.5 na entrada. O dropout é uma técnica de regularização utilizada em redes neurais para reduzir o overfitting durante o treinamento, introduzindo uma forma de "ruído" na rede, desligando (zerando) aleatoriamente um percentual das unidades (neurônios) durante cada passo de treinamento. Por último, duas camadas densas, a primeira com 64

neurônios e ativação relu e a última com 3 neurônios, ativação softmax e um dropout de 0.5 na sua entrada. Essa arquitetura pode ser vista na *Figura 28*, com 15 milhões de parâmetros totais dos quais quase 337 mil são treináveis.

Layer (type)	Output Shape	Param #
time_distributed (TimeDistributed)	(None, 25, 7, 7, 512)	14,714,688
time_distributed_1 (TimeDistributed)	(None, 25, 512)	0
dropout (Dropout)	(None, 25, 512)	0
lstm (LSTM)	(None, 128)	328,192
dense (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 3)	195

Total params: 15,051,331 (57.42 MB)

Trainable params: 336,643 (1.28 MB)

Non-trainable params: 14,714,688 (56.13 MB)

Figura 28 - Arquitetura VGG16 com 15 Janelas Deslizantes de 25 Frames

Para treinar essa rede foram utilizados como hiperparâmetros o otimizador Adam com métrica categorical accuracy, com learning rate de 10^{-4} e com função de loss categorical crossentropy, com pesos de 0.46550179211469533 para a classe normal, 2.3088888888888888 para classe de aspiração e 2.3885057471264366 para a classe de penetração, com o intuito de compensar o conjunto desbalanceado de treinamento. A rede foi treinada por 30 épocas com batch size de 16 e com EarlyStopping, salvando o modelo com melhor acurácia de treino e validação.

Outros modelos CNN com LSTM foram utilizados com modificações nos hiperparâmetros, no pré-processamento e na escolha da rede convolucional, tais como a ResNet50 e a EfficientNetB0. Porém, os resultados foram semelhantes ou piores do que os apresentados nesse projeto. Os resultados referentes a esse modelo podem ser vistos na seção 6.2.

5.4. Modelo VGG16 para classificação de frames multi-classe

Outro modelo utilizado foi para realizar a classificação dos frames em si e não do vídeo todo por meio de janelas. Dessa forma, a nova arquitetura utiliza apenas a rede convolucional sem LSTM, pois o aspecto temporal não é mais relevante.

Os conjuntos de treino, validação e teste vão ter a mesma proporção de, respectivamente, 70, 15 e 15. Por outro lado, a seleção dos frames também vai ser em torno de um frame de referência, mas os dados de input não serão mais as janelas deslizantes, mas sim os frames individuais, de modo que cada frame selecionado de um determinado vídeo recebe o mesmo label do vídeo.

Para um vídeo normal, foram selecionados o frame de referência e os 20 frames anteriores a esse frame, totalizando 21 frames, com o frame de referência sendo o frame de repouso. Para os vídeos de penetração e aspiração foram selecionados o frame de referência e os 10 frames anteriores e os 10 frames posteriores a esse frame, totalizando 21 frames, com o frame de referência sendo o PAS frame. Desse modo, a entrada da rede agora recebe os frames individuais e os seus respectivos labels no formato de um tensor de 4 dimensões, sendo eles, respectivamente, (quantidade de frames, altura das imagens, largura das imagens, número de canais de cor de cada frame) e (quantidade de frames, número classes). Logo, no treinamento, o formato da entrada dos frames e seus labels é, respectivamente, (1470, 224, 224, 3) e (1470, 3), e na validação o formato dos dados é (315, 224, 224, 3) e dos labels é (315, 3).

A arquitetura dessa rede é formada pelo VGG16 com todas as camadas congeladas, utilizando os pesos da rede pré-treinada, sem camada de topo e com GlobalAveragePooling2D aplicado na saída. Por fim, uma camada densa com 3 neurônios, ativação softmax e um dropout de 0.85 na sua entrada. Essa arquitetura pode ser vista na *Figura 29*, com aproximadamente 14,7 milhões de parâmetros totais dos quais 1539 são treináveis.

Para treinar essa rede foram utilizados como hiperparâmetros o otimizador Adam com métrica de categorical accuracy, com learning rate de 10^{-4} e com função de loss categorical crossentropy, com pesos de 0.4666666666666667 para a classe normal e 2.3333333333333335 para as classes de aspiração e penetração, com o intuito de compensar o conjunto desbalanceado de treinamento. A rede foi treinada por 100 épocas no total com batch size de 16 e com EarlyStopping, salvando o

modelo com melhor acurácia de treino e validação. Os resultados específicos desse modelo podem ser vistos na seção 6.3.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 3)	1,539

Total params: 14,716,227 (56.14 MB)

Trainable params: 1,539 (6.01 KB)

Non-trainable params: 14,714,688 (56.13 MB)

Figura 29 - Arquitetura VGG16 para Classificação de Frames muiti-classe

5.5. Modelo Resnet50 para classificação de frames multi-classe

Esse modelo é semelhante ao modelo anterior em todos os aspectos, com a mudança apenas da rede convolucional utilizada, que agora é a Resnet50. A sua arquitetura pode ser vista na *Figura 30*, com aproximadamente 23,6 milhões de parâmetros totais dos quais 6147 são treináveis. Os resultados específicos desse modelo podem ser vistos na seção 6.4.

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 3)	6,147

Total params: 23,593,859 (90.00 MB)

Trainable params: 6,147 (24.01 KB)

Non-trainable params: 23,587,712 (89.98 MB)

Figura 30 - Arquitetura ResNet50 para Classificação de Frames muiti-classe

5.6. Modelo VGG16 para classificação binária de frames multi-label

Esse modelo é uma variação do multi-classe que ao invés de realizar uma classificação entre as 3 classes, essa rede vai realizar duas classificações binárias, 2 bits, com o primeiro bit representando um vídeo normal ou anormal e o segundo bit representando a gravidade do problema. Dessa forma, os labels para as classes normal, penetração e aspiração são respectivamente (0,0), (1,0) e (1,1).

Como explicitado anteriormente, o primeiro rótulo indica a presença ou ausência de qualquer problema de deglutição, enquanto o segundo rótulo indica a gravidade. A classificação multi-label é adotada nesta abordagem devido à convicção de que as classes não são completamente independentes, mas sim correlacionadas.

A divisão dos conjuntos de treino, validação e teste é semelhante ao do modelo multi-classe, assim como o método de seleção dos frames, com 21 frames totais para cada vídeo usando o frame de referência, como consequência, o formato de entrada também é igual. O que muda é o formato dos labels que agora é binário, sendo eles (1470, 2) para treino e (315, 2) para validação.

Além da definição dos labels, outra grande diferença vai estar nos cálculos da métrica de acurácia e na função de perda. As métricas padrões do Tensorflow realizam o cálculo de accuracy e de loss como sendo dois labels (bits) independentes, o que não é o correto para esse tipo de rede. Por exemplo, supondo uma classificação de três frames com essas métricas, onde os labels verdadeiros são [0,0], [1,0] e [1,1] e que os rótulos previstos pelo modelo treinado seja [1,0], [1,1] e [0,1], a acurácia total calculada pelas funções do Tensorflow será de 50%, porque ele acertou um bit de cada um dos 3 frames, sendo que a acurácia correta é de 0%, pois o modelo errou todas as classes. Dessa forma, para essa rede foi necessário construir funções customizadas para o cálculo da accuracy e do loss.

As duas funções customizadas recebem dois tensores, um com os labels verdadeiros e o outro com os labels preditos. Cada um desses tensores possui duas colunas, uma para cada label. Sendo assim, a primeira etapa em cada uma das novas funções é extrair um tensor que represente cada label isoladamente, tanto para os labels verdadeiros, quanto para os labels preditos, gerando as variáveis listadas a seguir:

- y1true: tensor com os primeiros bits do tensor de labels verdadeiros (y_true);
- y2true: tensor com os segundos bits do tensor de labels verdadeiros (y_true)
- y1pred: tensor com os primeiros bits do tensor de labels preditos (y_pred)
- y2pred: tensor com os segundos bits do tensor de labels preditos (y_pred)

Para a função que calcula a acurácia (acc), é criado um tensor de tamanho N, igual ao *batch_size*, que representa o resultado da equação:

$$condition = tf.logical\_or(condition1, condition2)$$

A primeira condição (*condition1*) presente na equação acima verifica para cada elemento dentro do batch se a classe verdadeira é normal e se houve acerto na predição desses elementos. A equação para o cálculo da primeira condição é:

$$condition1 = tf.logical\_and(tf.equal(y1true, 0), tf.equal(y1pred, y1true))$$

Repare que o segundo label foi desconsiderado completamente, pois ele é irrelevante para elementos pertencentes à classe normal.

Já a segunda condição (*condition2*) é utilizada para detectar os casos de acerto na predição das classes não normais (penetração e aspiração), comparando tanto o primeiro quanto o segundo label verdadeiros com suas respectivas predições. A próxima equação apresenta o referido cálculo.

$$condition2 = tf.logical\_and(tf.equal(y1true, y1pred), tf.equal(y2true, y2pred)$$

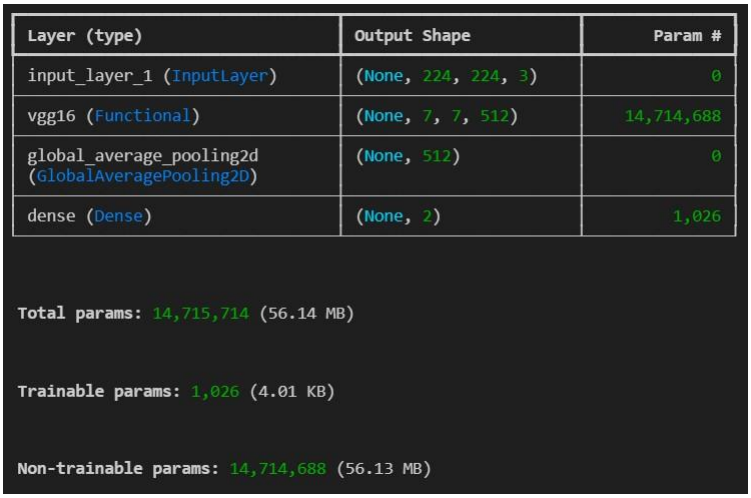
Dessa forma, a saída *condition* será um vetor com zeros e uns, representando respectivamente os erros e acertos na classificação de cada elemento do batch, bastando tirar a média dos acertos pela equação:

$$acc = tf.reduce\_mean(tf.cast(condition, dtype = tf.float32))$$

Já para a função de perda (loss), ela é calculado segundo a próxima equação, onde a função Binary Crossentropy padrão (bce) é utilizada para calcular a perda referente aos primeiros e aos segundos label, porém o segundo termo é multiplicado pelo primeiro label verdadeiro. A motivação para isso é que para vídeos normais o segundo bit é irrelevante, sendo necessário levá-lo em consideração apenas se o frame (ou vídeo) pertencer às classes não normais (penetração ou aspiração), cujo primeiro label verdadeiro é sempre 1. Desse modo, o loss é calculado para todos os elementos do batch e depois dividido pela quantidade de elementos (*N=batch_size*), obtendo-se a média das perdas individuais.

$$Loss = \frac{1}{N}\sum_{k=1}^{N} bce(y1_{true}[k], y1_{pred}[k]) + y1_{true}[k] * bce(y2_{true}[k], y1_{pred}[k])$$

A arquitetura dessa rede é formada pelo VGG16 com todas as camadas congeladas, utilizando os pesos da rede pré-treinada, sem camada de topo e com GlobalAveragePooling2D aplicado na saída. Por fim, uma camada densa com 2 neurônios e ativação sigmoid para realizar a classificação binária. Essa arquitetura pode ser vista na *Figura 31*, com aproximadamente 14,7 milhões de parâmetros totais dos quais 1026 são treináveis.



Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 2)	1,026

Total params: 14,715,714 (56.14 MB)

Trainable params: 1,026 (4.01 KB)

Non-trainable params: 14,714,688 (56.13 MB)

Figura 31 - Arquitetura VGG16 para Classificação de Frames muiti-label

Para treinar essa rede foram utilizados como hiperparâmetros o otimizador Adam com learning rate de 10^{-4} e com as métricas customizadas. A rede foi treinada por 100 épocas com batch size de 16 e com EarlyStopping, salvando o modelo com melhor acurácia de treino e validação. Os resultados específicos desse modelo podem ser vistos na seção 6.5.

5.7. Modelo ResNet50 para classificação binária de frames multi-label

Esse modelo é semelhante ao modelo anterior em todos os aspectos, com a mudança apenas da rede convolucional utilizada, que agora é a Resnet50. A sua arquitetura pode ser vista na *Figura 32*, com aproximadamente 23,6 milhões de parâmetros totais dos quais 4098 são treináveis. Os resultados específicos desse

modelo podem ser vistos na seção 6.6.

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0
dense_1 (Dense)	(None, 2)	4,098

Total params: 23,591,810 (90.00 MB)

Trainable params: 4,098 (16.01 KB)

Non-trainable params: 23,587,712 (89.98 MB)

Figura 32 - Arquitetura ResNet50 para Classificação de Frames muiti-label

6. Resultados

Essa seção apresenta os resultados dos modelos apresentados neste projeto. No final também é feito uma comparação entre os resultados obtidos e o estado da arte.

6.1. Resultado Modelo VGG16 com LSTM com janelas deslizantes de 25 frames

O melhor resultado obtido para esse modelo após o treinamento está ilustrado na tabela a seguir. Relembrando que esse foi o único modelo presente no relatório que usou a distribuição de 15 vídeos para o conjunto de treino, 15 vídeos para validação e 70 vídeos para o teste.

Tabela 1 - Desempenho sobre os conjuntos de treinamento/validação

	TREINAMENTO		**VALIDAÇÃO**	
Epoch	**Acurácia (%)**	**Perda**	**Acurácia (%)**	**Perda**
12/25	90,69	0,3776	89,13	0,4196

Na época 12, o modelo encontrou o valor máximo da acurácia sobre o conjunto de validação e de treino. Além disso, é importante destacar que embora o treinamento estivesse programado para acabar na época 30, devido a configuração de parada antecipada (EarlyStopping) com paciência de 10 épocas sem melhoras na acurácia sobre o conjunto de validação, o treinamento terminou antes do planejado.

Um ponto importante de destacar sobre esse modelo é que os valores de acurácia são altos porque a maioria das janelas são da classe normal, de modo que a acurácia no conjunto de treino e de validação se manteve praticamente constante em torno de 90%. Isso demonstra que o modelo não conseguiu aprender bem e não conseguiu generalizar, classificando quase todas as janelas como normal, caracterizando o fenômeno de overfitting.

Quanto ao tempo de treinamento, utilizando a máquina já especificada na metodologia, foram gastos aproximadamente 20h30m sobre um conjunto com shape de (2460, 25, 224, 224, 3) de dados e (2460, 3) para os labels.

Depois de treinado, o modelo encontrado com a melhor acurácia de validação

foi utilizado para predizer a classificação das janelas dos 70 vídeos do conjunto de teste.

Para fazer a classificação do conjunto de teste foi feito uma votação das janelas de modo que se 5 janelas consecutivas apresentassem uma das condições de problema, penetração ou aspiração, na região de interesse, durante a fase faríngea, então a classificação final do vídeo receberia essa condição. Caso contrário, o vídeo foi classificado como normal. Utilizando esse método, a acurácia no conjunto de teste foi de 83%. Como o modelo estava predizendo a maioria das janelas com a classe normal, a acurácia ideal seria de 87,50% mas com o limite brando de apenas 5 janelas consecutivas, o modelo classificou algumas janelas normais como penetração, resultando em falso positivo e abaixando a acurácia.

Esse modelo foi um dos primeiros modelos a serem construídos e ele foi útil para a evolução da solução na concepção dos modelos a seguir. Dessa forma, foi possível concluir que era necessário ampliar o conjunto de treinamento e que não precisava passar o vídeo todo para o modelo, mas sim, apenas a região de interesse da fase faríngea, tendo em vista que a maior parte dos vídeos não tinha nada acontecendo e que os problemas só poderiam ser observados durante a deglutição. Logo, essas mudanças ajudaram o modelo a treinar e predizer melhor.

6.2. Resultado Modelo VGG16 com LSTM com 15 janelas deslizantes de 25 frames

Os melhores resultados obtidos para esse modelo após o treinamento estão ilustrados na tabela a seguir.

Tabela 2 - Desempenho sobre os conjuntos de treinamento/validação

	TREINAMENTO		**VALIDAÇÃO**	
Epoch	**Acurácia (%)**	**Perda**	**Acurácia (%)**	**Perda**
18/28	89,59	0,2354	**62,22**	1,2960
28/28	**96,71**	0,0910	59,56	1,7202

Na época 18, o modelo encontrou o valor máximo da acurácia sobre o conjunto de validação, enquanto na época 28, encontrou o melhor desempenho para a acurácia sobre o conjunto de treinamento. É importante destacar que embora o treinamento estivesse programado para acabar na época 30, devido a configuração

de parada antecipada (EarlyStopping) com paciência de 10 épocas sem melhoras na acurácia sobre o conjunto de validação, o treinamento terminou antes do planejado.

Outro dado relevante é que a partir da época 18, a acurácia de validação oscilou em torno de 55%, mostrando estagnação, enquanto a acurácia no treinamento evolui constantemente de aproximadamente 90% para 97%. Isso demonstra que o modelo conseguiu aprender bem sobre o conjunto de treinamento, mas não demonstrou capacidade de generalização, ou seja, não foi capaz de predizer corretamente elementos não vistos durante o treinamento, caracterizando o fenômeno de overfitting.

Quanto ao tempo de treinamento, utilizando a máquina já especificada na metodologia, foram gastos aproximadamente 28h38m sobre um conjunto com shape de (1044, 25, 224, 224, 3) de dados e (1044, 3) dos labels.

Depois de treinado, o modelo encontrado com a melhor acurácia de validação foi utilizado para predizer a classificação de 225 janelas, referente a 15 vídeos, pertencentes ao conjunto de teste. Cada janela de um vídeo contém 15 janelas, totalizando 15 x 15 = 225 janelas completas, ou seja, com exatamente 25 frames obtidos no “entorno” de um frame de referência (durante a fase faríngea).

O resultado do desempenho desse modelo sobre o conjunto de teste é mostrado na tabela a seguir.

Tabela 3 - Desempenho sobre o conjunto de teste

TESTE	
Acurácia (%)	**Perda**
73,21	1,1184

Embora o resultado para o conjunto de teste tenha sido melhor do que o obtido sobre o conjunto de validação, é possível perceber, a partir da matriz de confusão associada e apresentada na *Figura 33*, que o modelo treinado não foi capaz de generalizar, pois quase todas as 225 janelas foram classificadas como normal, e apenas uma janela normal foi erroneamente classificada como penetração. No entanto, como cada vídeo possui 15 janelas, a classificação final do vídeo se daria pelo esquema de votação, sendo o vídeo associado à classe dominante. Em outras palavras, os 15 vídeos seriam todos classificados como normal.

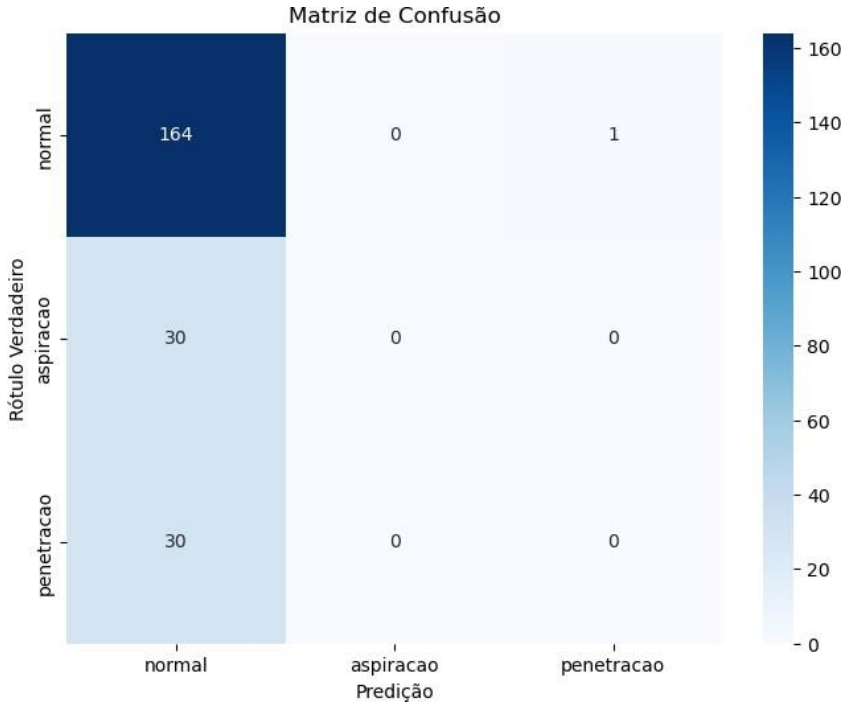


Figura 33 - Matriz de confusão – Mod VGG16/LSTM/ 15 janelas x 25 frames

Sendo assim, é possível constatar novamente a ocorrência de overfitting. Tal resultado deve-se ao conjunto pequeno de dados de entrada frente a complexidade da arquitetura utilizada, apesar das simplificações introduzidas pelas técnicas de GlobalAveragePooling e Dropout.

6.3. Resultado Modelo VGG16 para classificação de frames multi-classe

Esse modelo foi treinado por 100 épocas no total com o treinamento dividido em duas partes de 50 épocas. Quanto ao tempo de treinamento, utilizando a máquina já especificada na metodologia, foram gastos aproximadamente 5h2m.

O conjunto de teste de 15 vídeos possui o formato (315, 224, 224, 3). Nas primeiras 50 épocas a acurácia no conjunto de teste foi de 73,65% na classificação dos frames, acertando 232 frames de 315, com a matriz de confusão destacada na *Figura 34*.

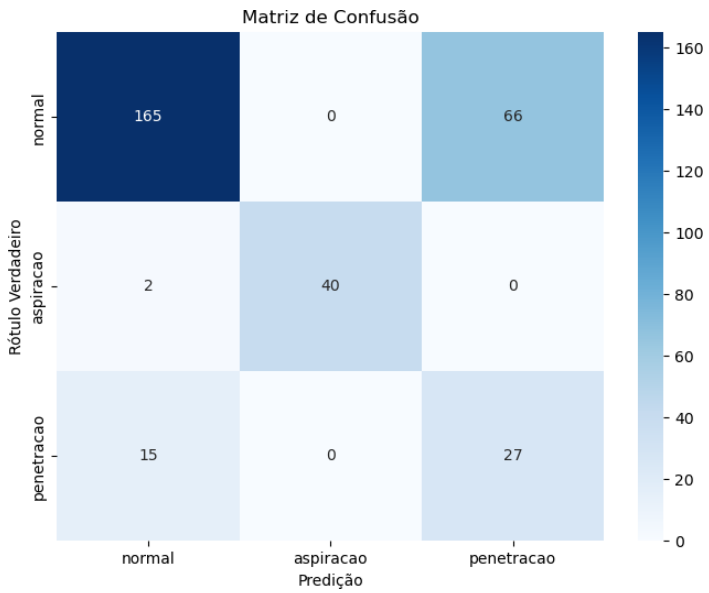


Figura 34 - Matriz de confusão por frame – Mod VGG16 multi-classe 315 frames e 50 épocas

Dos 15 vídeos o modelo acertou 11 vídeos o que resulta em uma acurácia de 73,33%, acertando os 2 vídeos de aspiração, 1 vídeo de penetração e 8 vídeos normais. No entanto, o modelo teve 3 falsos positivos, classificando 3 vídeos que eram normais como penetração, além de ter 1 falso negativo, classificando um vídeo de penetração como normal. A matriz de confusão dessa análise para a classificação dos vídeos de teste pode ser vista na *Figura 35*.

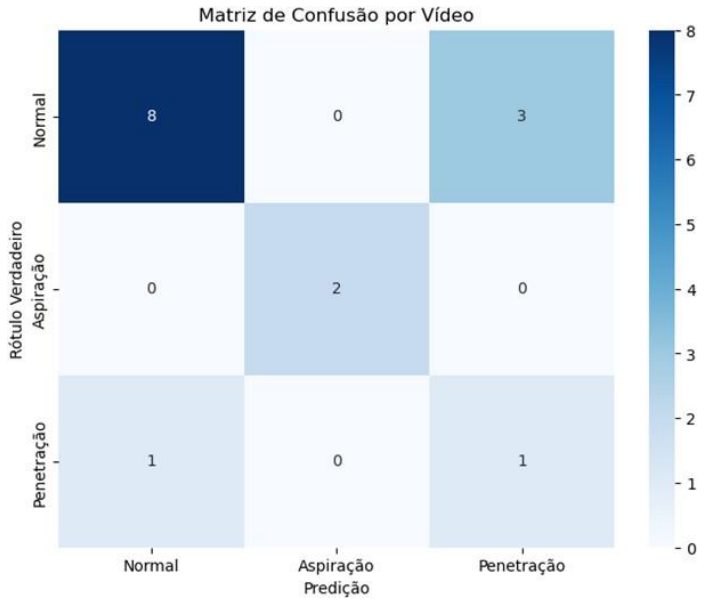


Figura 35 - Matriz de confusão por vídeo – Mod VGG16 multi-classe 315 frames e 50 épocas

Após a segunda parte do treinamento, com mais 50 épocas, a acurácia no conjunto de teste foi de 67,62% na classificação dos frames, acertando 213 frames de 315, com a matriz de confusão destacada a seguir, *Figura 36*.

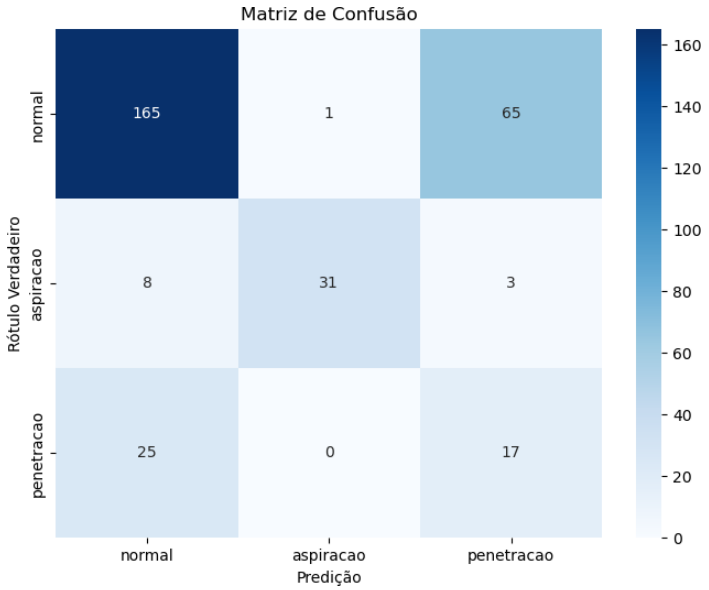


Figura 36 - Matriz de confusão por frame – Mod VGG16 multi-classe 315 frames e 100 épocas

Nesse caso, dos 15 vídeos o modelo também acertou 11 vídeos o que resulta em uma acurácia de 73,33%. A grande diferença desse modelo, após 100 épocas de treinamento, do modelo anterior, com as primeiras 50 épocas, é que ele errou mais frames de aspiração, ficando mais distribuído entre as outras classes. No entanto, a classificação final do vídeo continuou sendo de aspiração, porque a maioria dos 21 frames receberam a predição dessa classe. A matriz de confusão dessa análise da classificação dos vídeos do teste pode ser vista na figura a seguir, *Figura 37*.

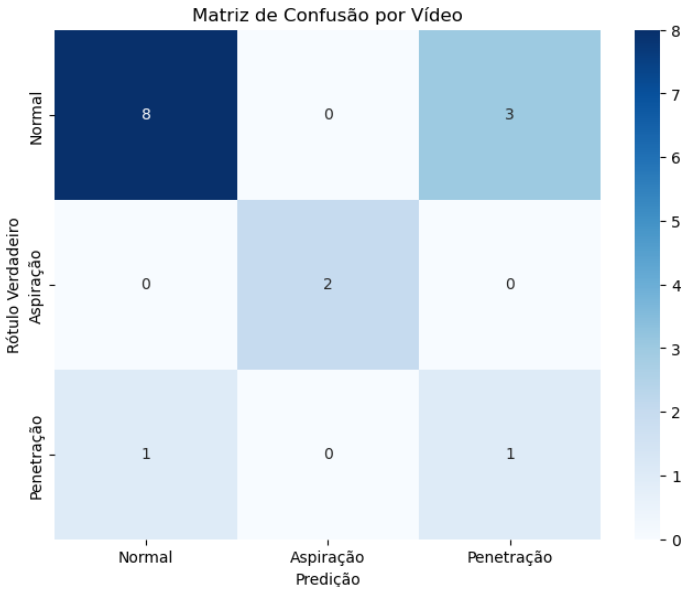


Figura 37 - Matriz de confusão por vídeo – Mod VGG16 multi-classe 315 frames e 100 épocas

Esse modelo foi mais consistente no treinamento com o seu desempenho total na classificação dos frames, após 100 épocas, demonstrado na tabela a seguir.

Tabela 4 - Desempenho geral do modelo na classificação dos frames

	TREINAMENTO		**VALIDAÇÃO**		**TESTE**	
Epoch	**Acurácia (%)**	**Perda**	**Acurácia (%)**	**Perda**	**Acurácia Frames (%)**	**Acurácia Vídeo (%)**
50/100	88,10	1,2304	61,90	0,8827	73,65	**73,33**
99/100	96,73	0,7049	67,30	0,8146	67,62	**73,33**

Um ponto interessante é que a classificação dos frames é de modo geral muito agrupada por vídeo, isto é, ou o modelo classifica corretamente quase todos os frames do vídeo ou ele erra a maioria deles. Além disso, a curva de acurácia da validação e do teste tende aos 67% na classificação dos frames, com a validação aumentando sua precisão com o tempo e com o teste reduzindo.

Ademais, esse modelo teve um melhor desempenho do que os modelos CNN com LSTM anteriores, tendo em vista que ele generalizou melhor, não tendendo a classificar apenas uma classe específica. Esse resultado de 73,33% no conjunto de

teste se aproxima do resultado de estado da arte existente para esse tipo de modelo de classificação de vídeos de videofluoroscopia da deglutição apresentado no artigo intitulado "*Comparative Analysis of Deep Learning Architectures for Penetration and Aspiration Detection in Videofluoroscopic Swallowing Studies*" desenvolvido por Eunhee Park et all. [16], publicado em setembro de 2023 na IEEE. Os resultados do estado da arte para o modelo de classificação de frames muiti-classe pode ser visto na *Figura 38*, tendo alcançado 76,88% de acurácia quando utilizando arquitetura baseada no VGG16.

TABLE 2. Comparison of models based on TOP-1 accuracy (TOP-1), F1 Score (F1), cohen-kappa score (CK), number of frames (FR), number of parameters (PAR), and giga floating point operations per second (GF).

MODEL	TOP-1(%)	F1(%)	CK(%)	FR	PAR(M)	GF
Single 2D-CNN Multi-Class Classifier						
VGG16	76.88	67.76	51.89	10	14.72	307.1
ResNet50	**79.75**	**68.91**	**56.47**	10	23.59	77.5
EfficientNet V1	76.52	62.72	46.76	10	4.05	**8.0**
EfficieentNet V2	75.27	61.45	46.38	10	5.92	14.6
MobileNet	67.92	41.96	20.34	10	**3.23**	11.5

Figura 38 - Resultados do Estado da Arte Para Classificação de Frames multi-classe

Por fim, o resultado desse modelo foi interessante porque conseguiu chegar próximo da faixa de acurácia da solução de estado da arte com quase 20 vezes menos vídeos disponíveis para treino, validação e teste.

6.4. Resultado Modelo ResNet50 para classificação de frames multi-classe

Esse modelo foi treinado por 100 épocas no total. Quanto ao tempo de treinamento, utilizando a máquina já especificada na metodologia, foram gastos aproximadamente 2h.

O conjunto de teste de 15 vídeos possui o formato (315, 224, 224, 3). A acurácia no conjunto de teste foi de 55,87% na classificação dos frames, acertando 176 frames de 315, com a matriz de confusão destacada na *Figura 39*.

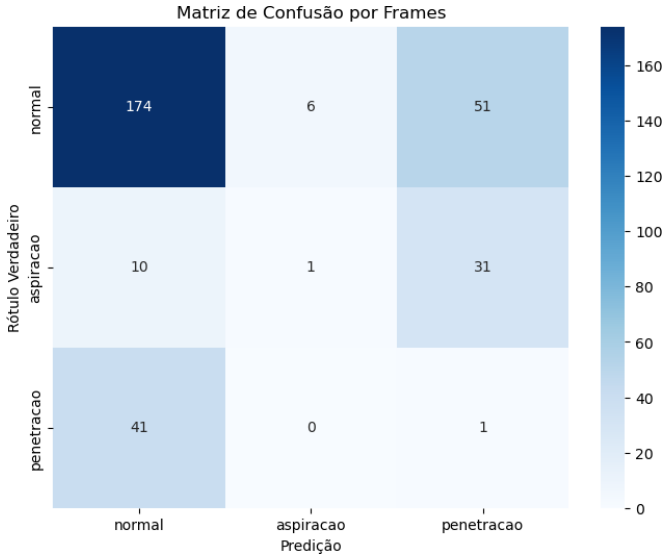


Figura 39 - Matriz de confusão por frame – Mod Resnet50 multi-classe 315 frames e 100 épocas

Dos 15 vídeos o modelo acertou 9, o que resulta em uma acurácia de 60,00%. Diferente do VGG16, esse modelo só acertou alguns frames normais, errando muito mais frames do que o VGG16, acertando apenas 1 frame de cada classe anormal, além de 2 vídeos classificados como falso positivo (era normal e classificou como penetração), 2 vídeos de aspiração ele classificou como penetração e 2 vídeos classificados como falso negativo (era penetração e classificou como normal). A matriz de confusão dessa análise da classificação dos vídeos de teste pode ser vista na *Figura 40*.

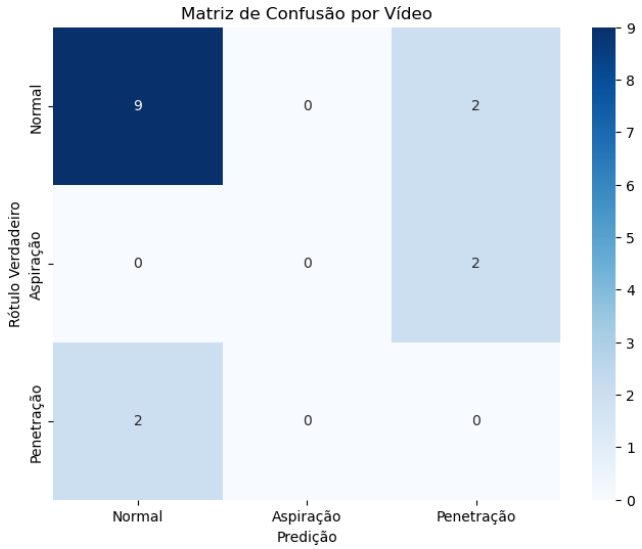


Figura 40 - Matriz de confusão por vídeo – Mod Resnet50 multi-classe 315 frames e 100 épocas

A acurácia máxima no conjunto de treino e validação após 100 épocas foi de, respectivamente, 83,80% e 60,63%, indicando overfitting. Esse modelo pode ter tido o mesmo problema dos modelos CNN com LSTM, ficando distante do estado da arte, devido à grande complexidade de sua rede, com aproximadamente 23 milhões de parâmetros, para um conjunto de dados muito pequeno, dificultando seu aprendizado e generalização.

6.5. Resultado Modelo VGG16 para classificação binária de frames multi-label

Esse modelo foi treinado por 100 épocas e o conjunto de teste possui 15 vídeos com o formato (315, 224, 224, 3) para os dados e (315, 2) para os labels. Os melhores resultados obtidos para esse modelo após o treinamento estão ilustrados na tabela a seguir.

Tabela 5 - Desempenho sobre os conjuntos de treinamento/validação

	TREINAMENTO		**VALIDAÇÃO**	
Epoch	**Acurácia (%)**	**Perda**	**Acurácia (%)**	**Perda**
75/100	81,71	0,4970	**70,00**	0,9616
89/100	**95,27**	0,2041	65,31	0,9396

Na época 75, o modelo encontrou o valor máximo da acurácia sobre o conjunto de validação, enquanto na época 89, encontrou o melhor desempenho para a acurácia sobre o conjunto de treinamento.

Quanto ao tempo de treinamento, utilizando a máquina já especificada na metodologia, foram gastos aproximadamente 5h15m.

Depois de treinado, o modelo encontrado com a melhor acurácia de validação foi utilizado para predizer a classificação dos frames pertencentes ao conjunto de teste, obtendo acurácia de 69,84% na classificação dos frames, acertando 220 frames de 315, com o seu desempenho demonstrado na matriz de confusão a seguir, *Figura 41*.

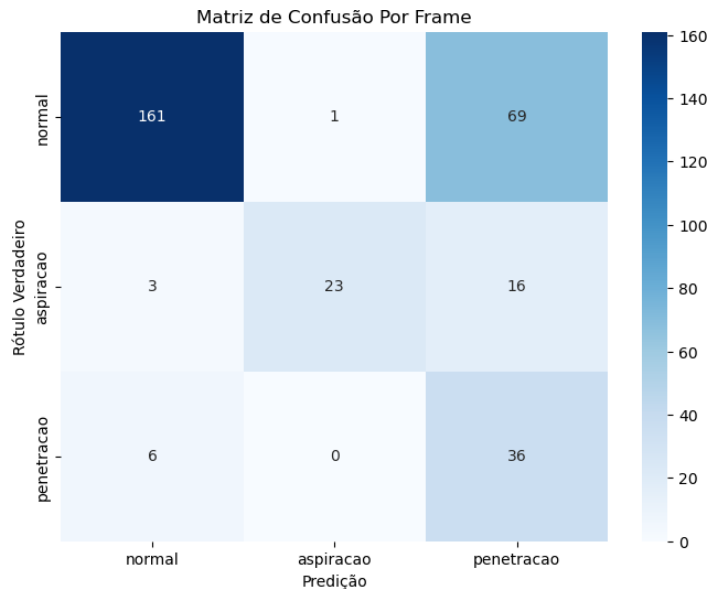


Figura 41 - Matriz de confusão por frame – Mod VGG16 multi-label 315 frames e 100 épocas

Dos 15 vídeos o modelo acertou 11 vídeos o que resulta em uma acurácia de 73,33%, acertando o 1 vídeo de aspiração, 2 vídeos de penetração e 8 vídeos normais. No entanto, o modelo teve 4 falsos positivos, classificando 3 vídeos que eram normais como penetração, e classificando um vídeo de aspiração como penetração. Um fato importante é que diferente do modelo multi-classe, esse modelo multi-label não teve falsos negativos, o que é interessante pensando na usabilidade, onde é melhor um médico rever um vídeo normal com uma classificação de anormal do que não verificar uma vídeo anormal. A matriz de confusão dessa análise da classificação dos vídeos do teste pode ser vista na *Figura 42*.

O modelo multi-label obteve a mesma acurácia para os vídeos do teste do que o modelo multi-classe, mas nesse caso, o resultado ficou um pouco mais distante da solução do estado da arte. Isso ocorreu devido ao overfitting e a pequena quantidade de dados disponibilizados pelo INCA, que foram 100 vídeos, levando em consideração os 2815 vídeos da solução ideal. A solução ótima pode ser vista na *Figura 43*.

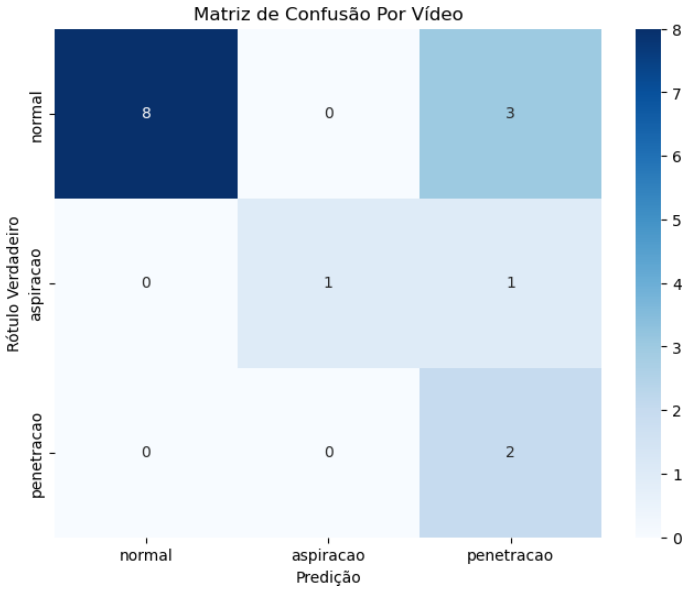


Figura 42 - – Matriz de confusão por vídeo – Mod VGG16 multi-label 315 frames e 100 épocas

TABLE 2. Comparison of models based on TOP-1 accuracy (TOP-1), F1 Score (F1), cohen-kappa score (CK), number of frames (FR), number of parameters (PAR), and giga floating point operations per second (GF).

MODEL	TOP-1(%)	F1(%)	CK(%)	FR	PAR(M)	GF
2D-CNN Multi-label Binary Classifier						
VGG16	84.41	80.84	68.93	10	14.72	307.1
ResNet50	**85.13**	**81.90**	**70.07**	10	23.59	77.5
EfficientNet V1	84.59	80.65	68.38	10	4.05	**8.0**
EfficientNet V2	83.69	79.19	66.26	10	5.92	14.6
MobileNet	83.69	76.36	66.16	10	**3.23**	11.5

Figura 43 - Resultados do Estado da Arte Para Classificação de Frames multi-label

6.6. Resultado Modelo ResNet50 para classificação binária de frames multi-label

Esse modelo foi treinado por 100 épocas no total. Quanto ao tempo de treinamento, utilizando a máquina já especificada na metodologia, foram gastos aproximadamente 2h24m. O conjunto de teste de 15 vídeos possui o formato (315, 224, 224, 3). A acurácia no conjunto de teste foi de 60,0% na classificação dos frames, acertando 189 frames de 315, com a matriz de confusão destacada na *Figura 44*.

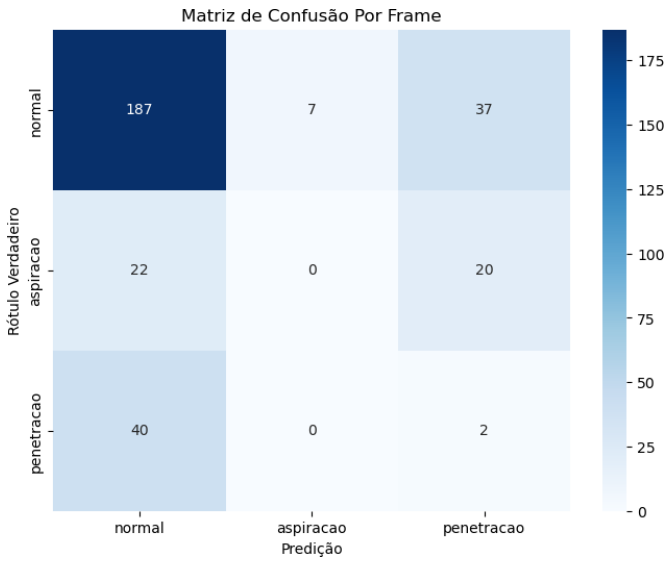


Figura 44 - Matriz de confusão por frame – Mod Resnet50 multi-label 315 frames e 100 épocas

Dos 15 vídeos o modelo acertou 9 vídeos o que resulta em uma acurácia de 60,00%. Diferente do VGG16 multi-label, esse modelo acertou alguns frames normais, 2 frames de penetração e nenhum frame de aspiração. Além disso, 2 vídeos foram classificados como falso positivo (era normal e classificou como penetração), 1 vídeo de aspiração foi classificado como penetração e 3 vídeos foram classificados como falso negativo (2 de penetração foram classificados como normal e 1 de aspiração foi classificado como normal). A matriz de confusão dessa análise da classificação dos vídeos de teste pode ser vista na *Figura 45*.

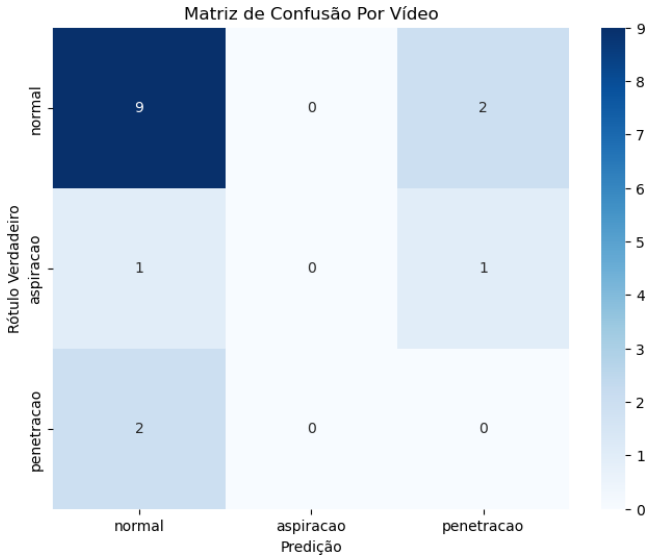


Figura 45 - Matriz de confusão por vídeo – Mod Resnet50 multi-label 315 frames e 100 épocas

A acurácia máxima no conjunto de treino e validação após 100 épocas foi de, respectivamente, 98,84% e 66,56%, indicando overfitting. Esse modelo pode ter tido o mesmo problema dos modelos CNN com LSTM e do modelo ResNet50 multi-classe, ficando distante do estado da arte, devido à grande complexidade de sua rede, com aproximadamente 23 milhões de parâmetros, para um conjunto de dados muito pequeno, dificultando seu aprendizado e generalização.

7. Conclusão

A classificação dos vídeos de videofluoroscopia da deglutição representa um desafio de grande relevância clínica e científica. Este processo é essencial para identificar alterações como penetração e aspiração, contribuindo para diagnósticos mais precisos e intervenções adequadas no tratamento de disfagias. No entanto, a tarefa é inerentemente complexa devido à natureza dinâmica e detalhada dos vídeos, exigindo modelos que captem nuances temporais e espaciais.

Neste projeto, foram construídos modelos baseados em arquiteturas avançadas, como CNN combinadas com LSTM, modelos CNN multi-classe e CNN multi-label. Embora essas arquiteturas sejam conhecidas por sua capacidade de capturar padrões complexos, os resultados obtidos ficaram quase sempre abaixo do estado da arte. Este desempenho pode ser atribuído a dois fatores principais como complexidade dos modelos e pequeno conjunto de dados, ocsionando problemas de overfitting.

Primeiramente, as redes profundas são muitas custosas computacionalmente e demandam muito tempo para serem treinadas, sendo difícil analisar seus resultados rapidamente. Além disso, a complexidade dessas redes também aumentam o risco de overfitting, principalmente quando o tamanho do conjunto de dados é limitado. Ademais, O uso de apenas 100 vídeos disponibilizados pelo INCA limitou a capacidade generalização dos modelos. Em comparação, o artigo “*Comparative Analysis of Deep Learning Architectures for Penetration and Aspiration Detection in Videofluoroscopic Swallowing Studies*”, que alcançou resultados de estado da arte, utilizou um conjunto significativamente maior, com 2815 vídeos, o que forneceu uma base sólida para treinar modelos robustos. Logo, A combinação de modelos complexos com poucos dados resultou em um ajuste excessivo aos dados de treinamento, prejudicando o desempenho em cenários de validação e teste.

Apesar dessas limitações, o modelo baseado na arquitetura CNN com VGG16, configurado para uma abordagem multi-classe, apresentou resultados satisfatórios, destacando-se ao atingir um desempenho muito próximo ao estado da arte para este tipo de arquitetura, mesmo com a quantidade reduzida de vídeos. Isso demonstra que a transferência de aprendizado, aliada a uma arquitetura bem projetada, pode mitigar parcialmente os efeitos da limitação de dados e capturar padrões essenciais para a resolução do problema.

Além disso, o modelo CNN com VGG16 configurado para uma abordagem multi-label também apresentou desempenho satisfatório, com o adicional de alcançar 100% de acerto na classificação de vídeos de classe anormal (penetração ou aspiração), ou seja, sem a presença de falsos negativos.

Portanto, os resultados deste projeto reforçam a importância de se equilibrar a complexidade do modelo com o tamanho e a qualidade do conjunto de dados. Para trabalhos futuros, é necessário aumentar a quantidade de dados disponíveis, aumentando também a quantidade de vídeos das classes minoritárias, ou realizar processos de cross-validation (validação cruzada) para analisar o desempenho dos modelos de forma iterativa e buscar estratégias para lidar com o overfitting. O desempenho promissor do modelo VGG16, principalmente quando comparado com o modelo Resnet50, também sugere que essa arquitetura pré-treinada pode ser um caminho viável e eficaz para problemas de classificação de vídeos em contextos médicos.

8. Referências

[1] STEEL, C. M.; MILLER, A. J. **Sensory Input Pathways and Mechanisms in Swallowing: A Review. Dysphagia**, v. 25, n. 4, p. 323-333, 2010. Disponível em: https://doi.org/10.1007/s00455-010-9301-5. Acesso em: 28 maio. 2024.

[2] ERTEKIN, C. **Electrophysiological evaluation of oropharyngeal dysphagia in Parkinson's disease**. J Mov Disord, v. 7, n. 2, p. 31-56, 2014. Disponível em: https://www.e-jmd.org/journal/view.php?doi=10.14802/jmd.14008. Acesso em: 28 maio. 2024.

[3] PARK, S.; CHO, J. Y.; LEE, B. J.; HWANG, J. M.; LEE, M.; HWANG, S. Y.; et al. **Effect of the submandibular push exercise using visual feedback from pressure sensor: an electromyography study**. Sci Rep, v. 10, n. 1, p. 11772, 2020. Disponível em: https://www.nature.com/articles/s41598-020-68738-0. Acesso em: 28 maio. 2024.

[4] YEOM, J.; SONG, Y. S.; LEE, W. K.; OH, B. M.; HAN, T. R.; SEO, H. G. **Diagnosis and clinical course of unexplained dysphagia**. Ann Rehabil Med, v. 40, n. 1, p. 95-101, 2016. Disponível em: https://doi.org/10.5535/arm.2016.40.1.95. Acesso em: 28 maio. 2024.

[5] CHANG, M. C.; PARK, J. S.; LEE, B. J.; PARK, D**. Effectiveness of pharmacologic treatment for dysphagia in Parkinson's disease: a narrative review**. Neurol Sci, v. 42, n. 2, p. 513-519, 2021. Disponível em: https://doi.org/10.1007/s00455-010-9301-5. Acesso em: 28 maio. 2024.

[6] YU, K. J.; PARK, D. **Clinical characteristics of dysphagic stroke patients with salivary aspiration: a STROBE-compliant retrospective study**. Medicine (Baltimore), v. 98, n. 12, p. e14977, 2019.

[7] SEJDIĆ, E.; MALANDRAKI, G. A.; COYLE, J. L. **Swallowing and swallowing disorders: Recent advances and challenges**. IEEE Signal Processing Magazine, v. 36, n. 1, p. 138-150, 2019. Disponível em: https://doi.org/10.1109/MSP.2018.2875863. Acesso em: 28 maio. 2024.

[8] STEELE, C.M. et al. **Challenges in preparing contrast media for videofluoroscopy. Dysphagia**, v. 28, n. 3, p. 464-467, 2013. Disponível em: https://doi.org/10.1007/s00455-013-9476-7. Acesso em: 28 maio. 2024.

[9] STEELE, C.M. et al. **Reference values for healthy swallowing across the range from thin to extremely thick liquids. Journal of Speech, Language, and Hearing Research**, v. 65, n. 5, p. 1338-1363, 2019. Disponível em: https://doi.org/10.1044/2019_JSLHR-S-18-0448. Acesso em: 28 maio. 2024.

[10] Instituto Nacional de Câncer José Alencar Gomes da Silva**. Estimativa 2020: incidência de câncer no Brasil**. Rio de Janeiro: INCA; 2019 [acesso 28 maio. 2024]. Disponível em: https://www.inca.gov.br/publicacoes/livros/estimativa-2020-incidencia-de-cancer-no-brasil.

[11] Swallowing Rehabilitation Research Lab. **Instruction Manual for ASPEKT-C Method**. Disponível em: https://steeleswallowinglab.ca/srrl/ .Acesso em: 28 maio. 2024

[12] WILHELM, P.; REINHARDT, J. M.; VAN DAELE, D. **A deep learning approach to video fluoroscopic swallowing exam classification.** 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). Anais...IEEE, 2020.

[13] KIM, J. K. et al. **Deep learning analysis to automatically detect the presence of penetration or aspiration in videofluoroscopic swallowing study**. Journal of Korean medical science, v. 37, n. 6, p. e42, 2022.

[14] BANDINI, A.; SMAOUI, S.; STEELE, C. M. **Automated pharyngeal phase detection and bolus localization in videofluoroscopic swallowing study: Killing two birds with one stone? Computer methods and programs in biomedicine**, v. 225, n. 107058, p. 107058, 2022.

[15] CALISKAN, H. et al. **Automated bolus detection in videofluoroscopic images of swallowing using Mask-RCNN**. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference, v. 2020, p. 2173–2177, 2020.

[16] PARK, E. et al. **Comparative analysis of deep learning architectures for penetration and aspiration detection in videofluoroscopic swallowing studies.** IEEE Access, v. 11, p. 102843-102851, 2023.

[17] IBM. **O que é uma rede neural?**. Disponível em: https://www.ibm.com/br-pt/topics/neural-networks. Acesso em: 13 novembro. 2024

[18] W. RAUBER, T. **Redes neurais artificiais**. Disponível em: https://www.researchgate.net/profile/ThomasRauber2/publication/228686464_Redes_neurais_artificiais/links/02e7e521381602f2bd000000/Redes-neurais-artificiais.pdf. Acesso em: 13 novembro. 2024

[19] IBM. **O que é aprendizado supervisionado?**. Disponível em: https://www.ibm.com/br-pt/topics/supervised-learning. Acesso em: 13 novembro. 2024

[20] IBM. **O que é Aprendizado não Supervisionado?** Disponível em: https://www.ibm.com/br-pt/topics/unsupervised-learning. Acesso em: 13 novembro. 2024

[21] AMAZON. **Qual é a diferença entre aprendizado profundo e redes neurais?** Disponível em: https://aws.amazon.com/pt/compare/the-difference-between-deep-learning-and-neural-networks/. Acesso em: 13 novembro. 2024

[22] IBM. **O que são redes neurais convolucionais?** Disponível em: https://www.ibm.com/br-pt/topics/convolutional-neural-networks. Acesso em: 13 novembro. 2024

[23] SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. Disponível em: https://arxiv.org/abs/1409.1556. Acesso em: 13 novembro. 2024

[24] ULTRALYTICS. **Conjunto de dados ImageNet**. Disponível em: https://docs.ultralytics.com/pt/datasets/classify/imagenet/. Acesso em: 13 novembro. 2024

[25] HE, K. et al. **Deep Residual Learning for Image Recognition**. Disponível em: https://arxiv.org/abs/1512.03385. Acesso em: 13 novembro. 2024

[26] IBM. **O que é uma rede neural recorrente (RNN)?**. Disponível em: https://www.ibm.com/br-pt/topics/recurrent-neural-networks. Acesso em: 13 novembro. 2024

[27] AMAZON. **O que é RNN (Rede neural recorrente)?**. Disponível em: https://aws.amazon.com/pt/what-is/recurrent-neural-network/. Acesso em: 13 novembro. 2024

[28] HOCHREITER, S; SCHMIDHUBER, J. **Long short-term memory. Neural Computation**, v. 9, n. 8, p. 1735-1780, 1997. Disponível em: https://www.bioinf.jku.at/publications/older/2604.pdf. Acesso em: 13 novembro. 2024

[29] SPRINGER. **Classification of benign and malignant subtypes of breast cancer histopathology imaging using hybrid CNN-LSTM based transfer learning**. Disponível em: https://link.springer.com/article/10.1186/s12880-023-00964-0. Acesso em: 13 novembro. 2024

[30] BARROS, B. et al. **Pulmonary COVID-19: Learning Spatiotemporal Features Combining CNN and LSTM Networks for Lung Ultrasound Video Classification**. Disponível em: https://www.mdpi.com/1424-8220/21/16/5486. Acesso em: 13 novembro. 2024