Universidade de Brasília – UnB
Campus Gama – FGA
Engenharia Eletrônica

Classificação de imagens de ressonância funcional para auxílio ao diagnóstico de transtornos mentais com uso de redes convolucionais

Alan Müller e Matheus Moreira da Silva Vieira

Orientador: Prof. Dr. Cristiano Jacques Miosso

UnB – Universidade de Brasília

FGA – Faculdade Gama

Engenharia Eletrônica

Classificação de imagens de ressonância funcional para auxílio ao diagnóstico de transtornos mentais com uso de redes convolucionais

Alan Müller e Matheus Moreira da Silva Vieira

Orientador: Prof. Dr. Cristiano Jacques Miosso

Trabalho de Conclusão de Curso
Engenharia Eletrônica

Brasília/DF, Fevereiro de 2023

UnB – Universidade de Brasília

FGA – Faculdade Gama

Engenharia Eletrônica

Classificação de imagens de ressonância funcional para auxílio ao diagnóstico de transtornos mentais com uso de redes convolucionais

Alan Müller e Matheus Moreira da Silva Vieira

Trabalho de Conclusão de Curso submetido à Faculdade UnB Gama da Universidade de Brasília, como parte dos requisitos necessários para a obtenção do grau de bacharel em engenharia eletrônica

Aprovada por:

Prof. Dr. Cristiano Jacques Miosso

(Orientador)

Prof. Dr. Fabiano Araújo Soares

Prof. Dr. Marcus Vinícius Chaffim Costa

Brasília/DF, Fevereiro de 2023

Ficha Catalográfica

Müller, Alan; Vieira, Matheus M. S.
Classificação de imagens de ressonância funcional para auxílio ao diagnóstico
de transtornos mentais com uso de redes convolucionais
[Distrito Federal], 2022.
83p., 210 × 297 mm (FGA/UnB Gama, Bacharelado em Engenharia Eletrônica, 2022).
Trabalho de Conclusão de Curso, Faculdade UnB Gama, Engenharia Eletrônica

1. Imagens Médicas	2. Psiquiatria
3. Redes Convolucionais	4. Processamento de Sinais
I. FGA UnB/UnB.	II. Título (série)

Referência

Müller, Alan; Vieira, Matheus M. S. (2023). Classificação de imagens de ressonância funcional para auxílio ao diagnóstico de transtornos mentais com uso de redes convolucionais. Trabalho de Conclusão de Curso, Engenharia Eletrônica, Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, 83p.

Cessão de Direitos

---

alanmuller@unb.br; matheus.silvadf@gmail.com

Brasília, DF – Brasil

Resumo

Introdução: A psicopatologia busca formas de definir critérios para os desvios da normalidade. A neuropsicologia contribuiu na investigação, eventualmente se utilizando do imageamento na busca por marcadores biológicos que possam colaborar no processo diagnóstico. Já há registros de associações de diferenciação anatômica em alguns cenários de transtornos mentais e problemas neurológicos. O imageamento funcional busca avançar nas descobertas, avaliando as respostas dos indivíduos com e sem diagnóstico clínico, a fim de caracterizar possíveis distinções. Este trabalho busca avaliar a classificação automática de imagens de ressonância magnética funcional, através do uso de redes convolucionais, quanto ao diagnóstico em esquizofrenia.

Fundamentação teórica: Os transtornos esquizofrênicos fazem parte de um grupo de distúrbios mentais que não possui sintomas específicos. Desde a década de 1970, o imageamento anatômico do encéfalo tem colaborado com o diagnóstico ou sua exclusão para diversas doenças, dentre estas a esquizofrenia. O uso de aprendizagem de máquina tem se mostrado útil na classificação de imagens de ressonância magnética. Por sua versatilidade, escalabilidade e poder de classificação, as redes neurais são a atual base do aprendizado profundo, com destaque para as convolucionais (CNN). A análise de padrão em *multivoxel* utiliza aprendizagem através de dados inseridos para treinamento e validação, sendo uma das abordagens para a avaliação de imagens de ressonância magnética funcional.

Materiais e Métodos: Bases de dados disponíveis em Schizconnect.com e OpenNeuro.com foram utilizadas como material de avaliação. As imagens de ressonância magnética funcional foram categorizadas pelas rotinas realizadas no exame (*tasks*). Rotinas de pré processamento segmentam estas imagens, estabelecendo uma sequência temporal associada a uma região do encéfalo (*slice*) e aos perfis de maior ativação neural. Redes convolucionais são treinadas a fim de encontrar os melhores parâmetros e definirem um modelo de predição. As predições são executas, com os melhores modelos treinados, e avaliadas suas métricas de desempenho. São comparados os resultados, relacionando o desempenho às *tasks* e as segmentações espaciais.

Resultados e discussões: Foram encontradas métricas de desempenho de até 85% de acurácia para determinadas *tasks* e *slices*, com predição baseada em imagens não utilizadas nos treinamentos. Muitos dos modelos de bom desempenho não apresentam métricas de treinamento e validação como curvas de tendências claras de crescimento de acurácia e redução de "perdas", usualmente associadas ao aprendizado crescente. No modelo base, das 88 rotinas, 8 demostraram taxa de *F1 score* acima de 75% e, pelo menos, o mesmo valor de acurácia. O melhor desempenho após o processo de *data augmentation* foi semelhante ao modelo base não apresentando melhoria significativa nos demais modelos. Rotinas de *tuning* eventualmente encontraram modelos menores – em complexidade – com predição

semelhante aos iniciais ou modelos maiores com métricas superiores. A validação cruzada (*cross validation*, CV) teve desempenho aquém dos treinamentos com conjuntos estáticos, mas ainda superior a 70% de acurácia em alguns dos cenários avaliados.

Considerações: Algumas regiões do encéfalo e rotinas se destacam nos resultados considerados positivos. As segmentações realizadas aparentam ser adequadas para reduzir a necessidade de utilizar enormes volumes de dados gerados nos exames para a rotina de classificação. Convém avançar as pesquisas, pela inclusão de explicações sobre os dados utilizados pelos modelos (*explainable artificial intelligence, X-AI*), para apresentar maiores evidências sobre a qualidade das métricas na abordagem sugerida. São passíveis de serem avaliadas abordagens correlatas, com segmentação de sinais distintos de ativação. O baixo quantitativo de exemplos disponíveis nos bancos de imagens para exames de ressonância magnética funcional associados a transtornos mentais é um ponto limitante na pesquisa.

Abstract

Introduction: Psychopathology aims to develop criteria for deviations from normality. Neuropsychology contributes to the research, eventually through imaging techniques to describe biomarkers to support the diagnosis. Some mental issues and neurological illnesses have anatomical differentiation already described. Functional imaging is expected to reach new findings, by evaluating neural responses for individuals with diagnostic established or excluded, in order to define differences between those. This study uses Convolutional Neural Networks (CNNs) to evaluate Functional Magnetic Resonance Imaging (fMRI) related to schizophrenia and healthy controls.

Theoretical background: Schizophrenic disorders are mental illnesses with no specific symptoms to support the diagnosis. Since the 1970s, anatomical imaging of the human brain has contributed to establishing or excluding diagnosis in several diseases, schizophrenia included. Machine Learning has been useful to classify MRI, because of how versatile, scalable, and powerful it is. Neural Networks are the actual basis of deep learning, convolutional in the spotlight. Multi-voxel Pattern Analysis is an approach that uses machine learning, through training and validation of input data to evaluate fMRI.

Materials and Methods: Data in use is collected from databases available on Schizconnect.com and OpenNeuro.com. Images (fMRI) are classified according to the tasks performed during exams. Preprocessing performs segmentation to define a time series and thresholding emphasizes higher blood oxygen level-dependent (BOLD) signals. Training CNN tries to fit the models with the best parameters to perform predictions. These predictions are done with the best models saved, in order to evaluate their metrics. Results are associated with the images' tasks and slices (spatial segmentation).

Results and discussion : Up to 85% of accuracy was reported, for some tasks and slices – predictions performed with images different from the ones used in the training and validation process. Curves of training and validation in the best metrics models have no clear tendencies – growing in accuracy and decreasing in losses – usually related to growing learning. The basis model reached more than 75% in F1 score and, at least, equivalent accuracy for 8 of 88 runs. Data augmentation is not crucial to model performance, compared to the basic model. Tuning reported improvements in some runs, with less complex models with similar metrics or higher complexity with better metrics. Cross validation did not establish the same metrics as basic models, but higher than 70% accuracy was still found.

Conclusions: Some regions of the brain and some tasks stand out in positive results.

Slicing and signal highlighting (thresholding) reduce the amount of data, from the huge volume generated by image reconstruction on fMRI, used to perform classification. Explainable Artificial Intelligence must be implemented, for better evidencing the quality of the results found on the models, to validate the suggested approach. Correlated approaches, like different levels of thresholding, are encouraged. Low quantities of images associated with mental illnesses available on public datasets restrict some aspects of the research.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Nomenclaturas e Abreviações

2D	–	Duas dimensões / bidimensional / plano
3D	–	Três dimensões / tridimensional / espaço
4D	–	Quatro dimensões / espaçotemporal / hiperespaço
Acc	–	Acurácia
ALFF	–	Flutuação de Amplitude em Baixa Frequência
API	–	Interface de Programação da Aplicação
BART	–	Tarefa de [avaliação de] Risco por Analogia do Balão
BHT	–	Tarefa de Prender a Respiração
BOLD	–	[Sinal com] Nível Dependente de Oxigenação Sanguínea
CID/*ICD*	–	Classificação Internacional de Doenças
CNN	–	Redes Convolucionais/Redes Neurais Convolucionais
CNP	–	Consórcio para [o Estudo] de Fenômenos Neuropsiquiátricos
COINS	–	Suíte Colaborativa em Informática e Neuroimagem
Conv	–	Convolução
CPU	–	Unidade de Processamento Central / Processador
DCM	–	Modelo Causal Dinâmico
DDR	–	*Double Data Rate*
DF	–	Distrito Federal
Dr	–	Doutor / Doutorado
DSM	–	Manual Diagnóstico e Estatístico em Saúde Mental
Enc	–	Codificação / Encoding
ELU	–	Unidade Exponencial Linear
fMRI	–	Imagem de Ressonância Magnética Funcional
GB	–	Gigabyte(s)
GCM	–	Modelo de Causalidade de Granger
GHz	–	Bilhões de ciclos por segundo
GLM	–	Modelo Linear Geral
GPU	–	Unidade de Processamento Gráfico
ICA	–	Análise de Componentes Independentes
IDE	–	Ambiente de Desenvolvimento Integrado
kNN	–	K'ésimo Vizinho mais Próximo

LeNet	–	Nome da rede neural criada por LeCun (1990)
LIT	–	[Sistema] Linear e Invariante no Tempo
MB	–	Megabyte
MLP	–	Percéptron Multicamada
MRI	–	Imagem de Ressonância Magnética
MVPA	–	Análise de Padrão Multivoxel
Neg.	–	Negativo
NIfTI	–	*Neuroimaging Informatics Technology Initiative* Refere-se ao formato de imagem padronizado pelo grupo
PAM	–	Memória de Associação de Pares
PCA	–	Análise de Componente Principal
PCDT	–	Protocolo Clínico e Diretrizes Terapêuticas
PET	–	Tomografia por Emissão de Pósitrons
PPI	–	Interação Psicofisiológica
Pos	–	Positivo
Prec	–	Precisão
Prof.	–	Professor
RAM	–	Memória de Acesso Randômico
ReHo	–	Homogeneidade Local/Regional
Ret	–	Recupração / *Retrieval*
RNN	–	Redes Neurais Recorrentes
RELU	–	Unidade Linear Retificada
SbFC	–	Conectividade Funcional baseada em semente
SCAP	–	Capacidade Espacial da Memória de Trabalho
SELU	–	Unidade Exponencial Linear em Escala
SEM	–	Modelo de Equação Estrutural
Sens	–	Sensibilidade
Spec	–	Especificidade
SVM	–	Máquina de Vetor de Suporte
SSD	–	Disco/Armazenamento em Estado Sólido
UCLA	–	Universidade da Califórnia em Los Angeles

1 Introdução

Esta seção visa familiarizar o leitor com conceitos elementares sobre o problema investigado e o cenário em que se apresenta. Mais adiante, são apresentados aspectos da abordagem proposta, com a descrição dos objetivos pretendidos pelos autores.

1.1 Contextualização

1.1.1 O estudo dos transtornos mentais

No estudo das funções inerentes ao ser humano social contemporâneo, a psicopatologia é a área que objetiva a observação de características que descrevam aspectos do comportamento, pensamentos e sentimentos dos sujeitos. Recentemente, foram publicadas atualizações dos manuais de diagnóstico utilizados em saúde mental – o Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM) em sua quinta edição [4] e a Classificação Internacional de Doenças (CID) em sua décima primeira [28]. Estes dois manuais ampliam e esclarecem os fatores que permitem a identificação de perfis de comportamento e sintomatologia associados àqueles transtornos.

A semiologia psicopatológica é o estudo dos sinais e sintomas produzidos pelos transtornos mentais. Dentro desta, a semiotécnica refere-se a técnicas e procedimentos específicos de observação e coleta de sinais e sintomas, assim como da descrição de tais sintomas. Enquanto a semiogênese investiga origem e mecanismos de produção daqueles. Entende-se desta forma que o aprimoramento da semiotécnica pode favorecer a agilidade do diagnóstico e direcionar de formá ágil para um acompanhamento adequado do quadro apresentado [9].

O termo “transtorno mental” passou a ser usado à partir de definições da CID e do DSM, substituindo o termo “doença mental”, do século XX, que por sua vez substituiu o termo “alienação”, do século XIX. A psicopatologia depende das definições do que é saúde versus doença/transtorno para criar critérios de desvio de normalidade nos processos de

diagnóstico [9]. Desta forma, caracterizar adequadamente o transtorno mental e vincular à definição de suas características é parte do processo diagnóstico.

A neuropsicologia, por sua vez, busca compreender quais atividades cerebrais estão relacionadas às funções psicológicas. Houve grande avanço neste campo com o trabalho de Alexander R. Luria (1902-1977), neurologista e neuropsicólogo russo. Ele apresenta a definição de "sintoma" com uma abordagem dinâmica e complexa, não sendo, portanto, possível de ser descrita de forma puramente mecanicista. Apesar do uso em psicopatologia dos testes neuropsicológicos, sua relação também é complexa, visto que os resultados dos testes podem ser implicados por condições físicas, de acuidade visual ou auditiva, por exemplo, e não necessariamente um deficit neuropsicológico [9].

No mesmo sentido, a partir da década de 1960, um novo conceito acerca da estrutura cerebral passou a ser moldado: a neuroplasticidade. O sistema nervoso, antes considerado uma estrutura fixa foi, a partir de então, visto como capaz de transformação e adaptação. Tais mudanças podem ser provocadas por fatores ambientais ou internos, acontecendo ao longo de todo o ciclo vital e não apenas nas fases iniciais de formação [17].

Desta forma, é possível conceber que uma maior agilidade no diagnóstico de um processo de adoecimento em saúde mental pode encontrar um estágio menos avançado de alterações nas estruturas neuronais. Consequentemente, poderia demandar um processo mais curto para uma potencial remissão dos transtornos gerados. Para compreensão das relações entre as funções cerebrais e as estruturas psíquicas, os exames de neuroimagem têm sido incentivados na prática psiquiátrica desde o início do século XXI, por terem grande potencial no auxílio ao diagnóstico diferencial [9].

1.1.2 Uso de imageamento no diagnóstico de transtornos mentais

No ano de 1976 foi publicado o primeiro artigo com uso neuroimagens de transtornos mentais, relacionando a dilatação ventricular como padrão de diferenciação entre pacientes com múltiplos episódios de esquizofrenia em comparação com indivíduos de controle [14]. Desde então, muitos estudos foram feitos no intuito de avaliar as estruturas neurológicas dos pacientes com diagnósticos em saúde mental para, entre outras tarefas, excluir alterações físicas (tumores, inflamações) como causadoras dos transtornos [10].

Quanto às neuropatologias, os marcadores identificados em imageamento estrutural foram primeiro utilizados no diagnóstico, prognóstico e tratamento de Alzheimer, por ter sido encontrado um padrão neurodegenerativo comum. A busca por marcadores semelhantes em transtornos relacionados à psicose, por outro lado, tem demandado maior

tempo em seu desenvolvimento, visto que não há uma clara associação com as estruturas neuronais e ainda há espaço para muito estudo neste campo [10].

A inteligência artificial e aprendizagem de máquina são utilizadas para fins de análise e classificação de imagens neurológicas há quase duas décadas, sendo que as *Support Vector Machines* (SVM) já apontaram 95% de acurácia em distinguir comprometimentos cognitivos leves em relação à população de controle, através de ressonância magnética funcional. Para esquizofrenia, por exemplo, a acurácia também já foi reportada como 92% [18] – neste caso, reclassificando as imagens utilizadas .

O uso de ressonância magnética funcional (fMRI) tem sido apontado como o mais adequado na investigação dos padrões de atividade cerebral, por alguns fatores de destaque. Dentre estes, por não depender de radiação ionizante – como é o caso do PET e dos raios x – e por conseguir identificar maior atividade cerebral relacionada a seu nível de oxigenação em determinadas áreas – sinal BOLD (dependente do nível de oxigenação sanguínea, do inglês *blood oxygen level dependent*). Uma sequência temporal de atividade é descrita durante a aquisição de dados e sua análise posterior pode vir a identificar padrões relacionados com o quadro de saúde mental do indivíduo [24].

Visto que as relações ainda não seguem bem estabelecidas entre as atividades cerebrais relacionadas aos transtornos, não estão claros os padrões que são buscados na imagem de ressonância funcional – como já é o caso da ressonância anatômica [10]. Assim, o uso de CNN, neste caso em particular a biblioteca *Keras* do *Tensorflow*, justifica-se pela facilidade de implementação, alta compatibilidade com unidades de processamento gráficos (GPUs), existência de interface de programação (API) de alto nível e fácil ajuste de parâmetros [19]. Também porque as técnicas de classificação baseadas em aprendizado profundo (*deep learning*) têm apresentado potencial crescente em relação a técnicas tradicionais de aprendizado de máquina [6].

1.2 Definição do problema científico e objeto de pesquisa

Apesar de fazer parte da lista de possíveis exames a serem requisitados no complemento da prática clínica em psiquiatria há quase duas décadas [9], os exames de neuroimagem ainda são pouco utilizados neste contexto. Um dos fatores que se destacam nessa problemática é o pouco treinamento recebido pelos profissionais de medicina quanto a interpretação destes dados [16].

Em adição, ainda se conhece pouco das relações entre as estruturas psíquicas e as respostas dos tecidos neuronais como base destes processos. O uso de imagens anatômicas

foi predominante ao longo de quatro décadas, tendo sido provado muito capaz de apontar diversos diagnósticos [10]. Já o estudo com as imagens funcionais apesar de promissor, ainda carece de um padrão de abordagem que possa ser prescrito como ferramenta padrão no auxílio diagnóstico [29].

Diversas técnicas de abordagem têm sido descritas na avaliação das imagens obtidas por ressonância magnética funcional, algumas delas sendo performadas com o uso de inteligência artificial. Houve enorme avanço em desempenho na classificação destes dados, pela interpretação de padrões que não estão relacionadas diretamente a visualização tridimensional (3D) ou espaçotemporal (4D) daquele conteúdo por um profissional de medicina [29].

Neste contexto, este trabalho utiliza redes convolucionais na busca de padrões, em imagens geradas por exames de ressonância magnética funcional, que possam demonstrar potencial para classificação, inclusão ou exclusão de diagnóstico clínico de pacientes com transtorno mental. Neste escopo, faz uso de imagens de bases de dados tornadas públicas, associadas a indivíduos com diagnóstico clínico de esquizofrenia e outros sem diagnóstico (controle).

1.3 Objetivos

1.3.1 Objetivo geral

Avaliar com que desempenho as redes neurais convolucionais são capazes de classificar imagens de ressonância magnética funcional do encéfalo humano, quando associadas a tarefas pré definidas – estímulos utilizados como geradores de atividade cerebral em análise – para auxílio ao diagnóstico em saúde mental, nos quadros de esquizofrenia.

1.3.2 Objetivos específicos

Para atender o objetivo geral, são abordados os seguintes objetivos específicos:

1. Levantar uma base de dados com imagens de ressonância magnética funcional do cérebro de pacientes com diagnóstico de esquizofrenia e indivíduos de controle, proveniente de bases de dados publicizadas, de forma a alimentar uma rede convolucional para identificação de padrões que possam ser associados ou desassociados do transtorno.

2. Destacar as estruturas que apresentam uma maior ativação neuronal no decorrer do exame, baseado nos níveis de oxigenação dos tecidos, para que os padrões extraído pelas redes convolucionais possam estar associado a estas.

3. Implementar um algoritmo que realize, apenas para o treinamento, técnicas de *data augmentation* baseada em rotação, a fim de gerar aprendizagem capaz de se adaptar as possíveis variações nos dados adquiridos no exame por eventuais movimentos ou diferenciações na posição do crânio do indivíduo.

4. Comparar diferentes configurações possíveis da rede convolucional 3D, pela variação de seus hiperparâmetros, no intuito de verificar as possibilidades de melhora nos resultados pela configuração da rede - número de camadas, tamanho dos filtros e *dropout*.

5. Analisar qual das *tasks* disponíveis no banco de imagens apresenta maior potencial para classificação, pelo resultado de desempenho descrito por métricas típicas das redes convolucionais – acurácia, precisão, sensibilidade, especificidade e *f1 score*.

6. Validar em quais cortes no plano axial possuem maior informação útil, por altos níveis de ativação neuronal, para auxílio em diagnóstico de esquizofrenia.

7. Realizar uma rotina de avaliação de desempenho da mesma rede convolucional 3D para entrada de dados extraídos de imagem anatômica, em região espacial proporcional à utilizada nas imagens funcionais.

2 Fundamentos em saúde mental, imageamento e redes convolucionais

2.1 Aspectos gerais da esquizofrenia

As síndromes são conjuntos de sinais e sintomas que se agrupam de forma recorrente e que são observados na prática clínica diária. Tentar identificar os perfis de síndromes em um paciente é o primeiro passo para ordenar a análise dos sinais e sintomas nele. O diagnóstico sindrômico é um ato clínico estrategicamente importante. Após a caracterização dos sinais e sintomas e sua acomodação em síndromes clínicas, é desejável que se formule hipóteses de diagnósticos relacionados aos transtornos mentais específicos, que, na teoria, teriam fatores etiológicos determinados e fisiopatologia específica [9].

Entre as várias síndromes existentes, as síndromes psicóticas são caracterizadas por alucinações, delírios, desorganização do pensamento e comportamento catatônico (perturbação do comportamento motor, no qual o paciente pode ficar em uma posição rígida, imóvel por horas e até dias). Podem envolver sensação intensa de estar sendo perseguido ou ameaçado. De modo geral, alterações na vida pessoal, familiar e social são indicativos de quadros graves. A principal forma de psicose é a esquizofrenia por sua importância clínica e frequência [9].

A esquizofrenia e os denominados transtornos esquizofrênicos compõem um grupo de distúrbios mentais graves sem sintomas específicos, ou seja, nenhum neste grupo possui um sinal ou sintoma que confirme seu diagnóstico de forma definitiva e o desenvolvimento do transtorno é variável. De acordo com o Protocolo Clínico e Diretrizes Terapêuticas (PCDT) da esquizofrenia, cerca de 30% dos casos apresentam recuperação completa ou quase completa, cerca de 30% com remissão incompleta e prejuízo parcial de funcionamento e aproximadamente 30% com deterioração importante e persistente da capacidade funcional [8].

A incidência de casos novos da doença é cerca de 15 a 42 por cem mil habitantes e a prevalência pontual é de 4,5 indivíduos por mil habitantes (0,45%). A prevalência de

indivíduos com risco de apresentar esquizofrenia alguma vez na vida no Brasil é de 0,8%, ou seja, 8 pessoas a cada mil habitantes. Um pouco maior que a prevalência mundial que é de 0,7% [9].

O diagnóstico clínico de esquizofrenia é apontado pela presença, por período superior a um mês, de pelo menos dois dos sintomas, um deles pertencente ao conjunto de: mania persistente, alucinação persistente, pensamento desorganizado, experiências de influência (ser influenciado por ou influenciar elementos ou pessoas); e outro no conjunto de: sintomas de negatividade, distúrbios psicomotores, comportamento desorganizado capaz de impedir a realização de metas. Ademais, é preciso que haja exclusão de efeitos de outras doenças (como tumores cerebrais), reações adversas de medicação, danos no sistema nervoso e uso de álcool [28].

Entre os sintomas associados, a falta de *insight* é um elemento presente na maioria dos pacientes. Em geral, não reconhecem que têm qualquer transtorno. Atribuem a presença dos sintomas a fatores como nervosismo, influências espirituais ou que são dificuldades que todo mundo tem. Além disso, é frequente a negação de que precisam de tratamento e acompanhamento médico. Mesmo em surtos psicóticos, apresentando comportamentos agressivos e destruição de objetos em casa, os pacientes recusam intervenção [9].

As causas da esquizofrenia ainda são desconhecidas. A forma mais aceita é a associação entre vulnerabilidade e estresse. Ou seja, o risco do desenvolvimento dos sintomas é aumentado quando existe algum tipo de vulnerabilidade prévia. Estes fatores podem ser físicos, ambientais, psicológicos e biológicos, que inclui predisposição genética. A manifestação dos sinais costuma acontecer quando o indivíduo entra em contato com estressores ambientais e falha ao tentar lidar com eles [8].

O tratamento da esquizofrenia não tem duração determinada. A indeterminação é apoiada por estudos analisando o efeito da interrupção do uso de medicação em pacientes estáveis. Foi evidenciado que os pacientes que mantiveram a utilização de medicamento, em curto, médio e longo prazos, tiveram menos chances de reincidência comparados com o grupo que suspendeu o uso. Outros estudos relataram que 25% dos indivíduos com apenas um sintoma psicótico não apresentou episódios depois do tratamento da crise [3].

2.2 Descobertas em imageamento neurológico

Desde 1976, quando foi apresentado em publicação pela primeira vez, o imageamento anatômico (ou estrutural) do cérebro apresentou aprimoramento de diagnóstico ou exclusão de diagnóstico para diversas doenças [10], dentre as quais:

- Alzheimer: caracterizado por perda de volume no lobo temporal, classificado por diferenciação no padrão de atrofia em relação aos indivíduos de controle; a análise de imagem também se apresenta capaz de distinguir degeneração do lobo fronto-temporal e demência por corpos de Lewy; no momento, o volume do hipocampo também é parâmetro para inclusão ou exclusão do diagnóstico;

- Transtorno bipolar: há uma hipótese persistente de que deve haver uma disfunção nos neurotransmissores ou um processo inflamatório associado ao quadro clínico, mas ainda seguem em investigação; destaca-se um aumento do lobo temporal esquerdo, do putâmen e do ventrículo lateral direito em relação ao público saudável;

- Esquizofrenia: destaca-se na literatura o volume reduzido do hipocampo, da amígdala, do tálamo, do núcleo Accumbens e da massa intracraniana, com ventrículo lateral, pálido ventral e putâmen aumentados; os dois últimos associados a duração do tratamento.

O uso de aprendizado de máquina utilizando as imagens de ressonância magnética estrutural parece ser útil na classificação de pacientes com transtorno bipolar (tipos I ou II) em relação a pacientes com esquizofrenia. Num estudo realizado com 66 pacientes em cada grupo – esquizofrenia, bipolar, controle – foi reportado acurácia superior a 86% na diferenciação entre os grupos. A técnica de aprendizagem utilizada foi a máquina de vetor de suporte e os mesmos exames utilizados no treinamento foram reutilizados nas métricas de predição [27].

2.3 Abordagens de aprendizagem de máquina

Três classificações principais são importantes para diferenciar as abordagens em aprendizagem de máquina: supervisão (supervisionado, não supervisionado, semissupervisionado ou por reforço), aprendizado incremental (*online* ou por lotes) e forma de detecção (comparação com elementos conhecidos ou modelo de predição). O aprendizado supervisionado apresenta ao algoritmo os resultados esperados para os dados de treinamento, fazendo da classificação uma aplicação típica para esta abordagem. Os algoritmos de destaque no aprendizado supervisionado incluem regressão linear ou logística, kNN (k'ésimo vizinho mais próximo), SVM (máquinas de vetor de suporte), árvores de decisão e redes neurais [13].

As redes neurais são a base do que se denomina "aprendizado profundo", com alta versatilidade, escalabilidade e enorme poder de classificação – chegando a bilhões de entradas. Podem se apresentar de várias maneiras, em destaque os *Perceptrons* Multica-

madas (MLP), as Redes Neurais Recorrentes (RNN) e as Redes Neurais Convolucionais (CNN) [13]. Destas, a última tem sido descrito como o estado da arte em classificação de imagens, tendo evoluído desde sua primeira apresentação em 1990 (com a LeNet-5) e tornando-se amplamente difundida, principalmente com a possibilidade de se processar os dados com uso de GPUs [25].

2.4 Aspectos das redes convolucionais

As CNN, embora sejam muito utilizadas em visão computacional com imagens bidimensionais, o processo de convolução pode ser unidimensional ou possuir três ou mais dimensões, utilizando vídeos ou sequências de imagens como entrada [22]. É possível construir um rede neural convolucional basicamente com camadas três tipos de camadas: Convolução, *Pooling* e conexão completa [7]. A Figura 2.1 ilustra uma arquitetura genérica típica de uma CNN com os três tipos de camadas.

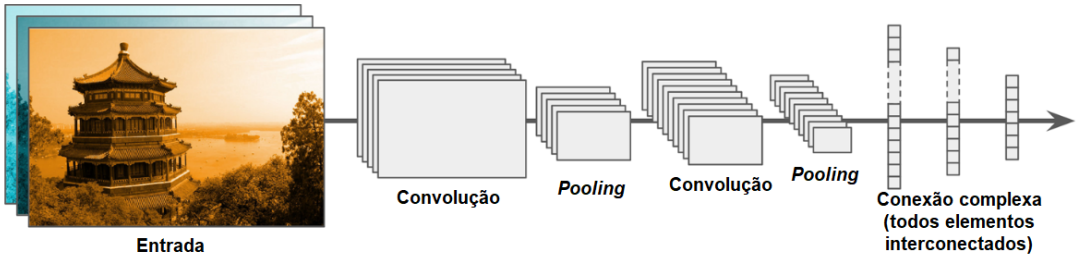

Figura 2.1. Arquitetura de uma CNN com camadas de convolução, *pooling* e conexão entre todos os elementos. Esta construção é comum em funções de classificação e similar aos modelos implementados neste projeto. Adaptado de [13].

A camada de convolução se utiliza de filtros com *kernel* baseado na resposta impulsional de sistemas lineares invariantes no tempo (LIT). Define-se a dimensão do espaço da saída e o *kernel size*, sendo um conjunto de três números inteiros, a profundidade, altura e largura da janela da convolução 3D. O resultado dos treinamentos é dados pelos valores atribuídos aos elementos do filtro e um vetor de *bias* também é criado e adicionado às saídas [7].

Outro atributo da camada de convolução é a escolha da função de ativação. Sua finalidade é fazer com que pequenas alterações nos pesos e *bias* causem uma pequena alteração na saída, principalmente se forem negativos. Entre as funções de ativação disponíveis do *Keras* pode-se citar: ELU, SELU, ReLU, Softmax, *Sigmoid* (sigmoide). A Unidade Linear Exponencial (ELU) estabelece que a entrada seja ela mesma quando

maior que zero e a exponencial da entrada menos 1 multiplicado por uma constante α

$$x = \alpha(exp(x) - 1)$$

quando for negativa [7].

A Unidade Linear Exponencial Escalada (SELU) é análoga à primeira, com a diferença que, tanto para entradas positivas, quanto para negativas a saída é reescalada por uma constante fixa: $scale = 1.0507$. A Unidade Linear Retificada (RELU) é a mais utilizada em redes neurais e retorna o máximo valor entre 0 e a entrada, ou seja, se a entrada for positiva, a saída não é modificada, porém será igual a zero se a entrada for negativa. A função *Softmax* é mais utilizada em classificação multiclasses por converter um vetor de valores em uma distribuição de probabilidades no qual os elementos da saída estão compreendidos entre 0 e 1 e a soma de todos é igual a 1. Já a função *Sigmoid* é mais aplicada em classificação binária, pois retorna 0 para valores de entrada menores que -5 e 1 para entradas maiores que 5. É equivalente a uma *Softmax* de dois elementos [7].

Já a camada de *pooling* é responsável por agrupar blocos para reduzir a saída e acelerar a computação. Pode-se definir o tamanho dos *pools* e a forma de agrupamento. É possível agrupar o valor dos pixeis de uma imagem, por exemplo, substituindo um bloco pelo seu maior valor(*MaxPooling*) ou pela média dos valores do bloco (*AveragePooling*). As duas formas de agrupamento possuem suas versões globais (temporais) em que a escolha dos valores substituídos são feitos ao longo do tempo. A camada *BatchNormalization* realiza uma transformação que mantém a saída média próxima a 0 e o desvio padrão da saída próximo a 1. Isso reduz o deslocamento interno de covariáveis entre as camadas [7].

A camada que faz a conexão completa de cada neurônio com todos os neurônios da camada anterior é a camada *dense*. Alguns dos parâmetros dessa camada são o tamanho do vetor de saída e a função de ativação ao final da camada. A saída é um produto ponto a ponto dos valores da entrada e dos pesos adicionado aos valores de *bias* passando por uma função de ativação. Para diminuir o excesso de ajustes e a quantidade de elementos treináveis em uma rede neural é a inclusão de uma camada de *dropout*. Esta camada iguala a zero, de forma aleatória, entradas até uma taxa determinada e amplia os outros valores em $1/(1 - taxa)$. Esta técnica de abandono no treinamento resulta em uma menor dependência de pesos específicos de neurônios e em uma rede capaz de melhor generalização [7].

Além da utilização dessas camadas, para o treinamento da rede neural, é necessário definir a dimensão do tensor de entrada e de saída. Dois parâmetros importantes são

o *batch_size* e as *epochs*. O primeiro define o número de amostras por atualização de gradiente – quantidade de elementos a serem trabalhados antes que os parâmetros internos sejam atualizados. Por padrão o lote de amostras é 32 e quanto maior o valor do *batch_size* mais espaço em memória será utilizado. O número de épocas define o número de iterações por todo o conjunto de dados do treinamento. Em uma época, cada amostra no conjunto de treinamento atualiza seus parâmetros internos [7].

2.5 Aprendizagem de máquina com imagens de ressonância em transtornos mentais

Há mais de quatro décadas, as imagens neurológicas têm sido usadas em investigações dos mais diversos quadros de saúde relacionados à atividade cerebral. As imagens estruturais tiveram sua análise realizada intensamente durante todo este período, tendo contribuído no diagnóstico e tratamento de várias doenças. Alzheimer, transtorno bipolar, depressão maior e esquizofrenia são exemplos onde o uso das imagens anatômicas tiveram sucesso em apresentar uma classificação adequada ao diagnóstico [10].

Muitos dos achados das pequisas com imagens cerebrais apontam para a negação do diagnóstico pré estabelecido, tendo sido descobertos problemas de formação estrutural, tumores ou problemas vasculares, os quais direcionaram para tratamentos diversos dos de saúde mental, mas que tiveram potencial para salvar a vida daqueles pacientes. Não se dá por esgotado o uso das imagens anatômicas, visto que novas técnicas de análise são desenvolvidas com o avanço da tecnologia, novas formas de abordagem são incentivadas, para ampliar e complementar as descobertas já feitas [10].

Visto já estarem disponíveis vários estudos correlatos, alguns pesquisadores têm verificado a possibilidade de *transfer learning*, para aproveitar treinamentos prévios de aprendizagem de máquina já utilizados na classificação de imagens. Consideram a possibilidade de que dados necessários já podem ter sido assimilados por aquela arquitetura. Esta abordagem, contudo, não tem se mostrado tão efetiva como a reconfiguração de parâmetros (*fine tuning*) das redes convolucionais por reaplicação do treinamento [26].

Por outro lado, revisões sistemáticas apontam que as alterações anatômicas encontradas na literatura ainda não seriam definitivas, visto a baixa prevalência das alterações de forma isoladas [5]. A observação destes desvios anatômicos pode ser talvez melhor utilizado no acompanhamento do curso do transtorno do que na identificação deste [1].

2.6 Investigação em ressonância funcional

Especificamente na análise de dados de imagens de ressonância magnética funcional, têm sido propostas algumas abordagens possíveis a fim de determinar padrões nesta observação, de acordo com suas duas classes principais nestas imagens: *task-based* e *resting-state*. Para o primeiro caso, entende-se que o fluxo sanguíneo gerado após um estímulo - sonoro, visual - identifica áreas em atividade, gerando os marcadores BOLD. Já para o segundo caso, o único comando dado aos voluntários é que fiquem com os olhos abertos, também observando sua atividade neste período [29].

Nas análises realizadas em imagens do tipo *resting-state*, seis métodos de análise se destacam [29]:

- SbFC (seed-based functional connectivity): a partir de um ponto definido (*seed*), avaliando os valores médios do sinal BOLD ali e suas correlações com os valores noutras regiões;
- ReHo (region homogeneity): avalia a concordância de um voxel e sua vizinhança, baseado na sequência temporal destes;
- ALFF (amplitude low-frequency fluctuation): direciona a análise para as atividades espontâneas do cérebro, em espectros de baixa frequência;
- PCA (principal component analysis): em domínios transformados de bases ortogonais, busca-se a maior variância para extração de informações mais significantes;
- ICA (independent component analýsis): uma evolução do PCA, que depende de regras axiomáticas menos estritas para definir a não correlação entre as componentes;
- Graph teory: numa representação em grafo, cada voxel é um nó e uma conexão entre suas vizinhanças, buscando descrever as conexões neurais no encéfalo.

Na análise de imagens do tipo *task-based*, outros seis métodos são apresentados [29]:

- GLM (general linear model): representa o resultado obtido na imagem como um produto de parâmetros pela soma de sinais temporais associados a cada ponto no espaço, associado (soma) a um erro;
- PPI (psychophysiological interaction): busca compreender as conectividades funcionais através de relações causais e mapear as direções das influências;

- SEM (structural equation model): com aspectos semelhantes a PPI, mapeando as regiões do encéfalo que se relacionam durante a interação;
- DCM (dynamic causal model): para além do SEM, leva em consideração a influência da hemodinâmica para a representação e trata as conexões neurais como determinísticas;
- GCM (Granger causality model): diferente dos anteriores, não busca fixar uma interação direta entre as diferentes regiões, mas estabelecer relações entre séries temporais estocásticas;
- MVPA (multi-voxel pattern analysis): utiliza classificadores de padrões, como máquinas de vetor de suporte ou redes neurais, para definir aprendizagem sobre os dados apresentados como treinamento e validação, partindo da hipótese de que o treinamento pode ser capaz de definir os padrões de atividade a serem decodificados.

3 Dados e métodos utilizados na análise de ressonância funcional

3.1 Bases de dados e rotinas de fMRI

São utilizados dados publicizados de estudos prévios, disponíveis em SchizConnect.org e OpenNeuro.org. A primeira base de dados utilizada é oriunda de estudo sobre imagens multimodais sobre esquizofrenia realizado no Mind Research Network, do *Center of Biomedical Research Excellence (COBRE)* [2] e são providos pelo COllaborative Informatics and Neuroimaging Suite Data Exchange tool (COINS; http://coins.mrn.org/dx).

Os estudos levaram em consideração apenas participantes com esquizofrenia (ou transtorno esquizoafetico) e saudáveis. O recrutamento foi feito através divulgação em clínicas psiquiátricas da cidade de Albuquerque, Novo México nos Estados Unidos. Os participantes tinham entre 23 e 51 anos e uma proporção de 81% de homens para o grupo de diagnóstico e 72% de homens para o grupo de controle. Os participantes com esquizofrenia apresentavam os primeiros sintomas aos 21 anos, em média. Foram excluídos indivíduos com histórico de dependência ou abuso de substâncias ativas (feniclidina, anfetamina ou cocaína) nos últimos 12 meses e todos os participantes se abstiveram de fumar por pelo menos uma hora antes do início do exame [2].

A segunda base de dados foi produzida no âmbito do *The Consortium for Neuropsychiatric Phenomics (CNP)* da Universidade da Califórnia em Los Angeles (UCLA), no estudo intitulado LA5c, que objetivava investigar as relações neurais e cognitivas [20]. Os dados das neuroimagens de ressonância magnética funcional (fMRI) são segmentados entre controle (sem diagnóstico associado) e esquizofrenia e pelas rotinas associadas a execução do exame.

Os participantes saudáveis foram recrutados através de anúncios da comunidade de Los Angeles e os com esquizofrenia através de divulgação em clínicas locais e portais *online*. Para os dois grupos, os participantes eram homens e mulheres com idade entre 21 e 50 anos. Eram brancos não hispânicos ou latinos e hispânicos ou latinos, de qualquer

grupo racial. O idioma primário dos indivíduos era inglês ou espanhol e tinham pelo menos 8 anos de educação formal. Era requisito da pesquisa que o resultado do exame de urina fosse negativo para substâncias como cocaína, metanfetamina e morfina. Para estudos de ressonância magnética a pesquisa exclui participantes canhotos, que acreditam estar grávidas ou que tenham claustrofobia.

A seguir são descritas as rotinas de testes a que foram submetidos os voluntários. Algumas rotinas e regiões cerebrais já fazem parte da investigação neuroradiológica descrita na literatura, outras são processadas com objetivo de ampliar a pesquisa. Cada uma pode apresentar ativação ou inibição de regiões específicas do cérebro, por exemplo, com a ativação de regiões subcorticais em atividades relacionadas à memória. Entende se que a avaliação com inteligência artificial pode cooperar na descoberta de atividades e padrões que possam ser incluídos como típicas em transtornos. As rotinas avaliadas neste estudo foram também, previamente, avaliada em seus estudos originais, com objetivos semelhantes. Pode haver ou não potencial de descoberta em diferentes rotinas para a abordagem proposta e, portanto, faz parte da pesquisa apontar tais relações.

3.1.1 Balloon Analog Risk Task - BART

A *task* BART é uma tarefa usada para avaliar o comportamento de um indivíduo a um risco envolvendo uma recompensa. O teste é feito utilizando uma tela que apresenta um balão vazio e dois botões: “encher” e “coletar”. Quanto mais o voluntário clicar em “encher” o balão cresce mais e ele recebe algum valor em um banco temporário. Ele pode continuar enchendo e aumentar o dinheiro no banco ou coletar. Se encher demais o balão estoura e o valor no banco temporário é zerado e quando clicar em “coletar” o valor que ganhou vai para um banco permanente que não é perdido se estourar outros balões no decorrer do teste. BART mede o comportamento de risco de um indivíduo registrando o número médio de cliques em “encher”, total de ganhos e número de balões explodidos [23].

A validade de construto da BART já foi comprovada e mostrou correlações com fatores de comportamento antissocial, psicopatia e impulsividade autorreferida [23]. O teste realizado no estudo da UCLA o participante usaram trinta balões virtuais, com um tempo médio de nove minutos e cada estouro valia cinco pontos [21].

3.1.2 Breath Hold Task - BHT

A rotina BHT consiste basicamente em uma alternância entre respiração normal, preparação para apneia – pode ser uma inspiração prolongada, por exemplo – e apneia propriamente. Podem existir algumas diferenças de acordo com os autores do teste. Na aquisição das imagens em uso, foi apresentada uma tela aos participantes em que aparecia sinais verdes, amarelos e vermelhos. O sinal verde indicava aos participantes para respirarem normalmente, o amarelo, para se prepararem para a apneia e o vermelho para prenderem a respiração por 13,5 segundos. O ciclo foi repetido 8 vezes e foi utilizado um cinto respiratório para medir a respiração e um oxímetro de pulso para dados fisiológicos. O objetivo do teste de retenção de respiração em exames de fMRI é avaliar a contribuição dos ritmos respiratórios para as mudanças no sinal BOLD [21].

3.1.3 Gate Task - GATE

As imagens da *task gate* foram adquiridas da base de dados do *COllaborative Informatics and Neuroimaging Suite Data Exchange tool* [2], que não informou na documentação dos estudos qual a rotina executada dos testes.

3.1.4 Paired Associates Memory Task - PAM

Nesta tarefa foram realizadas duas varreduras para avaliar a codificação e recuperação da memória explícita declarativa utilizando uma tela. As duas varreduras são partes integrantes de um mesmo teste avaliando os sinais BOLD na tentativa de memorização e, em seguida, na tentativa de lembrança dos objetos [21].

3.1.4.1 Encoding - PAM-Enc

A primeira parte do teste foi divida em dois blocos: um de memorização e outro de controle. No teste de memorização foram apresentadas quarenta tentativas em que duas palavras apareciam por um segundo, uma de cada lado da tela. Em seguida apareciam dois objetos que correspondiam às palavras durante três segundos. Um dos objetos foi desenhado em preto e branco e o outro em apenas uma cor. No segundo bloco, na tentativa de controle, vinte e quatro pares de estímulos embaralhados apareciam por dois segundos cada, um preto e branco e outro colorido. Os participantes tinham que pressionar um botão indicando em que lado o objeto colorido estava. Todos foram instruídos a lembrar

os objetos e a relação entre eles. O tempo total do teste de codificação foi de 8,07 minutos [21].

3.1.4.2 Retrieval - PAM-Ret

A tarefa de recuperação avaliou a confiança dos participantes em suas memórias a partir da observação dos pares de objetos. Foram utilizadas 104 tentativas ao todo: 24 tentativas de controle, 40 tentativas corretas e 40 incorretas de acordo com a primeira parte do teste. Os participantes puderam responder de quatro formas: "certamente correto", "talvez correto", "talvez incorreto" e "certamente incorreto". A varredura de recuperação durou 8,93 minutos [21].

3.1.5 Resting State - REST

No exame em estado de repouso, os voluntários são orientados a estar relaxados e manter os olhos abertos. O teste durou aproximadamente 5 minutos e não receberam qualquer outro estímulo[21]. Neste estudo, são utilizados três conjuntos distintos de imagens associadas ao estado de repouso – diferenciadas pela resolução espacial e temporal – sendo identificados os grupos como *rest*, *rest(a)* e *rest(b)*.

3.1.6 Spatial Working Memory Capacity Task - SCAP

A tarefa de capacidade espacial consiste em uma apresentação de alvos de 1, 3, 5 ou 7 círculos amarelos posicionados de forma pseudo-aleatória em torno de uma cruz central. Após um tempo de dois segundos de apresentação ocorre um *delay* variável de 1,5, 3 ou 4,5 segundos. Depois do atraso os participantes viram um círculo verde e foram solicitados a indicar se esse círculo estava na mesma posição em que estava um dos círculos amarelos anteriormente. O tempo de resposta era fixo de três segundos. O teste foi feito com 48 tentativas: 12 por tamanho, com 4 em cada comprimento de atraso para cada conjunto de memória. Metade das tentativas eram verdadeiros positivos e metade verdadeiros negativos [21].

3.1.7 Stopsignal Task

Na tarefa de sinal de parada, os participantes receberam uma série de estímulos “vá” na tela com setas apontando para direita e para esquerda. Eles deveriam pressionar os botões direito e esquerdo de acordo com os estímulos na tela. Porém em alguns estímulos foi apresentado um sinal de parada sonoro em fones de ouvido. O sinal de parada foi tocado com um atraso de 250 milissegundos após um comando ”ir”. Os voluntários foram instruídos a responder o mais rápido possível. O tempo de persistência do sinal de parada foi variado conforme o voluntário acertava ou errava a tentativa. Se ele acertava o sinal, o próximo atrasava mais 50 milissegundos, ficando mais difícil; se errava, o próximo *delay* era menor, no mesmo passo de 50 milissegundos, facilitando a tentativa. Cada experimento foi composto de 128 tentativas, no qual 96 foram tentativas *GO* e 32 tentativas *STOP*, como sinal de parada [21].

3.1.8 Task switch

No teste de comutação de tarefas, os participantes receberam estímulos em uma tela com quatro figuras: Um triângulo vermelho e outro verde, um círculo vermelho e outro verde. Os participantes foram solicitados a responder ou a forma (*shape*) ou a cor (*color*) dos objetos, porém em 1/3 das tentativas tiveram que responder o inverso. Por exemplo: Se aparecesse um círculo verde e a instrução *shape*, o participante deveria responder “verde”. Esta tarefa foi executada para medir alterações no tempo de reação entre os ensaios que exigiram a troca de resposta e os que não exigiram. Foram feitas 96 tentativas em um tempo total de 6 minutos e 52 segundos [21].

3.2 Ferramentas de processamento

3.2.1 Ambientes de Hardware

Os desenvolvimentos de técnicas realizados a partir de testes de aplicação tiveram seus códigos rodados em computador local, de propriedade dos pesquisadores, com configuração geral dada por: processador (CPU) Core i7-11390 (4x5,0GHz, 12MB Cache), 16GB RAM DDR4, armazenamento em estado sólido (SSD), GPU NVidea MX430 (2GB RAM DDR5). Para este bloco, algumas rotinas foram executadas no Google Colab, com ambiente limitado à uso de CPU; como o desempenho era inferior ao computador proprietário, não houve uso significativo.

A partir da necessidade de utilizar maiores recursos de memória e processamento, as rotinas avançadas foram migradas para o Google Colab de forma definitiva, em sua versão gratuita, com uso de GPU tipicamente com 15GB RAM disponível - modelo não declarado. Tais execuções são limitadas em tempo pelo fornecedor, sendo necessário segmentar os códigos para que não prolonguem a execução. Também é demandado interação regular com o ambiente, sob pena de desconexão.

3.2.2 Ambientes de Software

As plataformas computacionais dos pesquisadores utilizam sistemas operacionais Windows 11 – para desenvolvimento e acesso ao Google Colab – e Ubuntu 22.10, exclusivamente para acesso ao Google Colab. O navegador preferencial foi o Google Chrome – até a versão 109.0.5414.76, disponível no momento das execuções.

Os códigos são executados no ambiente de desenvolvimento (IDE) Spyder, versão 5.4.0 para Python 3.9.13. As bibliotecas utilizadas são:

- *random, os, pathlib* e *time*, para randomização de dados, acesso ao sistema operacional, caminhos de arquivos e relógio do sistema, respectivamente;
- *Numpy* 1.23.2, para operações matemáticas;
- *Scipy* 1.9.0 para algumas manipulações de imagens;
- *Matplotlib* 3.5.3 para gerar gráficos de treinamento e validação, bem como salvar estes dados;
- *Nibabel* 4.0.1 para leitura e importação das imagens no formato NIfTI;
- *Tensorflow* 2.9.0, com compilação específica de compatibilidade para instruções da CPU e da GPU, para todas as rotinas envolvendo as redes convolucionais.

3.2.3 Técnicas de preprocessamento

O elemento principal do preprocessamento é o destaque dos níveis de ativação, pela limiarização e estratificação das matrizes. Os níveis indicados nas imagens correspondem diretamente ao nível de oxigenação sanguínea naquele espaço. Tal nível de oxigenação é relacionado a alta demanda por oxigênio dos tecidos neurais para a execução de suas funções. A partir da inspeção visual das imagens antes e após a limiarização, foram definidos os critérios dessa operação. Na fase inicial, os dados com nível menor do que

500 – de um limite superior de aproximadamente 1200 – foram zerados, estratificados os dados restantes em oito níveis. Não tendo havido resultado satisfatório e ainda exigindo demasiado esforço computacional – grandes modelos com dezenas de horas de treinamento – os níveis foram reajustados, com zero para níveis originais abaixo de 900 e dez níveis de estratificação. Após este processo, os dados foram normalizados e convertidos em ponto flutuante. Na Figura 3.1 a imagem original, à esquerda e a imagem com destaque da segmentação utilizada no treinamento das redes neurais – nota-se que apenas os elementos destacados foram utilizados. Outros exemplos são apresentados no Anexo 6.1, assim como exemplo da planificação do tensor montado com estes dados.

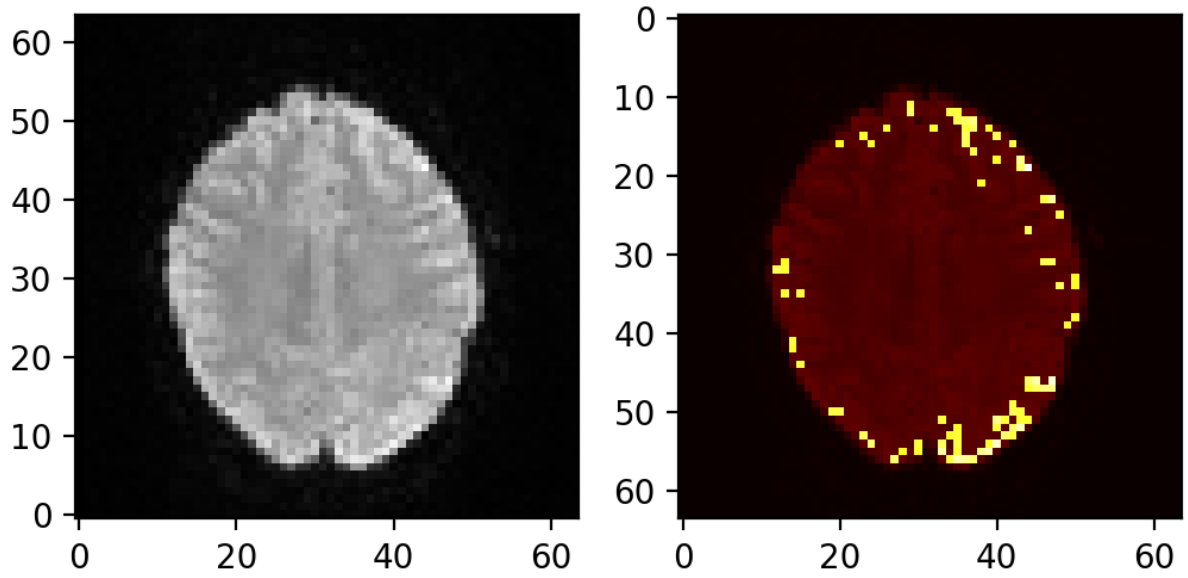

Figura 3.1. Exemplo de segmentação do cérebro no plano axial – Visualização do *slice* 20 extraído da rotina de *Task switch*. À esquerda a imagem original em escala de cinza; À direita a imagem em mapa de calor com destaque ao sinal segmentado. Os pontos amarelos representam o sinal relativo a altos níveis de oxigenação sanguínea. Os eixos representam os *pixels* da imagem recortada.

Da imagem 4D – três dimensões espaciais e tempo – foram escolhidos níveis de altura de interesse. Objetiva-se com esta técnica uma redução da demanda computacional e um processamento mais ágil na ferramenta diagnóstica. Considerados os estímulos realizados durante a realização dos exames funcionais, o tempo é tido como componente principal. Uma avaliação inicial de todo o banco de imagens foi realizado com segmentações espaciais idênticas, a fim de validar que pertenciam a regiões equivalentes nos cérebros dos participantes. A partir de nova inspeção visual, foi observado que os *slices* com alturas entre 40 e 75% do total possuem maior volume de dados de interesse – registro do encéfalo no espaço da imagem. A partir desta observação, os treinamentos foram gerados com a segmentação de um *slice* neste intervalo. O empilhamento de todas as componentes temporais deste *slice* formou o volume de entrada, na forma de um tensor. Para os

modelos sem *data augmentation* propriamente dito, os tensores de cada imagem foram rotacionadas no eixo z, em ângulos randomizados entre -25° e 25°. Para os modelos com *data augmentation*, todo um conjunto de ângulos igualmente espaçados foi utilizado, até o limite da memória do ambiente.

3.2.4 Modelo da Rede Convolucional

O número de camadas da rede foi planejado para evitar um aprendizado excessivo exclusivamente daquele conjunto de imagens, característica chamada de *overfit*. Desta forma, o número de parâmetros treináveis não pode ser demasiado. As configurações preferenciais incluem, após a camada de entrada (*input layer*), dois blocos com camadas sequenciais de convolução, *MaxPooling* e *BatchNormalization* – o segundo bloco convolucional tem o dobro de filtros do primeiro – e um bloco de finalização com *GlobalAveragePooling*, *Dense*, *Dropout* e novo *Dense*.

Algumas configurações foram mantidas constantes e outras variadas ao longo dos treinamentos. Para *MaxPooling*, o tamanho foi fixado em 2; para *Dense*, o primeiro foi fixado em 128 e o segundo em 1; as ativações são sempre do tipo *Sigmoid* para o último *Dense* e *RELU* para todas as demais. Já nos itens variáveis, o tamanho do *input* foi sempre proporcional ao tensor 3D gerado para aquele conjunto de dados; o quantitativo de filtros na primeira camada – e consequentemente o da segunda, por ter o dobro – o tamanho do *kernel* destes filtros e o tamanho do *Dropout* foram ajustados. O resultado em parâmetros treináveis para cada configuração destas variáveis é apresentado na Tabela 3.1, sendo que a variação no *Dropout* não altera este quantitativo, nem o tamanho do *input*.

Tabela 3.1. Quantitativos de parâmetros treináveis com as variações geradas nos modelos, variável dependente do quantitativo de filtros nas camadas de convolução e do tamanho do *kernel* de cada filtro.

Filtros	*Kernel*	Parâmetros treináveis
16	3	18.753
16	5	70.497
32	3	64.897
32	5	268.737
64	3	240.129
64	5	1.049.217
128	3	922.369
128	5	4.146.177

3.2.5 Estrutura do algoritmo

A sequência de construção do algoritmo para o desenvolvimento das rotinas utilizadas neste trabalho tem a seguinte estrutura:

1. Importar as bibliotecas;
2. Definir funções (carregar imagens, segmentação, limiarização e estratificação, normalização, rotação, construção do modelo);
3. Configurar hardware (sempre que possível utilizar a GPU);
4. Definir a *task* e suas parametrizações (tamanhos e quantitativos de imagem);
5. Definir os *slices* de interesse (em laço de repetição ou treinamento único);
6. Definir as localizações de arquivos;
7. Carregar imagens para os tensores;
8. Segmentar treinamento, validação e predição – imagens distintas para cada bloco;
9. Ampliar o *set* de treinamento por *data augmentation*, se constar no escopo daquele treinamento;
10. Configurar carregamento de dados para o treinamento;
11. Compilar o modelo com base nos parâmetros da *task* atual;
12. Configurar o salvamento do modelo de melhor desempenho encontrado no treinamento (*callbacks*);
13. Treinar o modelo e exportar gráficos de evolução;
14. Realizar predições com o modelo de melhor desempenho salvo previamente;
15. Montar uma matriz de confusão e calcular as métricas;
16. Gerar *logs*

Para os casos de *tuning*, onde os parâmetros do modelo são variados na busca por melhores métricas de desempenho, os passos 10 a 16 foram repetidos para cada uma das possíveis configurações geradas. Desta estrutura, cabe ressaltar a importância do salvamento do ponto de melhor desempenho verificado no treinamento, a fim de utilizar este caso como referência para as predições, otimizando os resultados destas. Também é fundamental a geração dos *logs* para acompanhamento das métricas de desempenho a cada ciclo de treinamento.

3.3 Evolução dos treinamentos

Para chegar ao modelo base deste trabalho, o modelo reproduzido da referência [30] foi avaliado de diversas formas. Inicialmente, haviam três blocos de convolução, *pooling* e *normalization*, com o quantitativo de filtros crescentes. O quantitativo de parâmetros treináveis era demasiado e o treinamento se estendia por diversas horas; os dados de entrada também eram brutos, não destacando quaisquer elementos.

Visto ser compatível com a literatura [12], a segmentação dos dados por limiarização e estratificação foi realizada para otimizar os dados de entrada. Assim, os modelos com muitos parâmetros treináveis apresentaram indícios de sobreajuste (*overfit*) e foi possível reduzir um trio de camadas. Ainda com o mesmo indício, os quantitativos de filtros foram reduzidos e os resultados começaram aproximar das expectativas – o limiar de 70% de acurácia foi estimado como ponto de partida, com base nos achados da pesquisa bibliográfica.

Por fim, o modelo base foi treinado para cada grupo de imagens associado a uma *task* e cada *slice* de interesse, igualmente espaçados em 5% dos níveis no eixo z da imagem, variando entre 40 e 75% (totalizando 8 níveis para cada grupo) – os arredondamentos fizeram com que as posições efetivas não fossem igualmente espaçadas. Este treinamento foi realizado tanto no computador local como no ambiente do Google Colab, a fim de verificar a consistência.

3.3.1 Data Augmentation

O processo de *data augmentation* no escopo deste trabalho foi realizado apenas por rotação. Considerando que o exame de ressonância magnética funcional é realizado com o paciente sempre na mesma posição em relação aos elementos de captura e que o volume de interesse é segmentado na reconstrução de imagem, o principal fator gerador de diferenciação seriam possíveis inclinações na cabeça do indivíduo. Desta forma, buscou-se uma compatibilidade com os processos físicos reais; a variabilidade gerada, por exemplo, por inversão das lateralidades foi desconsiderada, visto que os sinais de interesse podem ser localizados em hemisférios distintos do encéfalo.

O quantitativo de ângulos utilizados para o processo de rotação foi ajustado manualmente, pela limitação da memória disponível no sistema. Rotinas mais extensas geram mais *frames*, que neste escopo resulta em maiores volumes de extração de cada imagem. Os quantitativos de aumento variaram de seis a vinte vezes para os conjuntos distintos, enfatizando que este aumento é só realizado no bloco de treinamento, mantendo as

imagens de validação e predição exatamente com seus dados de extração originais.

3.3.2 Tunning

A possibilidade de reutilizar modelos previamente treinados em bases de dados distintas (*transfer learning*) é bastante comum [26]. Neste cenário, o *fine tuning* consiste no retreinamento de camadas de saída, por exemplo, para adaptar-se a base de dados atual. Outra abordagem busca a repetição do treinamento para o mesmo conjunto de dados em uso, pela alteração de hiperparâmetros da rede, sendo esta a forma executada neste estudo.

A *API Keras* contém uma ferramenta própria, chamada *keras-tuner*, onde os parâmetros a serem variados e seus valores possíveis são definidos – tipo de ativação, quantitativo de filtros e tamanho de *kernel*, taxa de aprendizagem e quantidade de camadas, por exemplo. Esta ferramenta, contudo, tende a escolher os parâmetros de forma aleatória, em um número máximo de execuções, salvando apenas os dados de predição para o modelo que tenha melhores métricas no treinamento.

A fim de validar as alterações causadas pela variação dos parâmetros selecionados (filtros, *kernel* e *dropout*), a repetição do treinamento foi realizada em *loop* para que todos os cenários fossem contemplados. Visto ser uma rotina bastante extensa, apenas cinco conjuntos de *task/slice* foram selecionados para esta avaliação, sendo preferencialmente aqueles que apresentaram destaque nas métricas de predição nas etapas anteriores.

3.3.3 n-fold Cross Validation

Kohavi [15] descreveu o processo de validação cruzada (*cross validation*) como subdivisões mutuamente exclusivas do conjunto de dados, onde uma destas divisões é deixada fora do bloco de treinamento e as predições são testadas nela, processo chamado de *k-fold*. Quando os conjuntos se tornam unitários, a predição para cada teste resulta em acurácia 0 ou 1, neste caso a validação é chamada de *n-fold* ou validação cruzada completa, pois são realizados n testes de acordo com o quantitativo de imagens. O objetivo da validação é estimar a confiança nas métricas de predição geradas.

Das 100 imagens disponíveis (50 de diagnóstico e 50 de controle), foram separadas 20 imagens para validação – 10 de cada grupo. Um par de imagens (uma de diagnóstico e uma de controle) foi destinada para predição e as outras 78 para o treinamento. Foi executado um *loop* em que todas as imagens do conjunto de treinamento foram utilizadas

para predição – cada iteração um par de imagens foi definida como predição. Para essa versão do algoritmo, o número de ciclos de treinamento (*epochs*) foi reduzido de 25 para 20, a fim de viabilizar a execução no ambiente do Google Colab.

O autor descreveu ainda a possibilidade de se obter 0% de acurácia nas validações do tipo *n-fold*, por instabilidades geradas no processo. A fim de evitar os valores nulos, este trabalho gerou as métricas de desempenho pela soma dos casos de sucesso e falha em uma única matriz de saída. A variância é medida com base neste resultados compilados.

3.3.4 Comparativo com imagens anatômicas

Com o objetivo de avaliar a possibilidade de que abordagem semelhante para os dados de ressonância magnética funcional e anatômica possam gerar métricas comparáveis, um modelo com *tuning* foi treinado com os volumes de entrada dados pela extração de todos os *slices* na faixa de 40 a 75% da altura z destas imagens.

Neste caso, a limiarização e estratificação não são realizadas – apenas a normalização – visto que não se busca um sinal de ativação, mas relações de proporcionalidade entre dimensões das estruturas encefálicas representadas ali. A hipótese de *tuning* foi considerada pela potencial variação no quantitativo de dados apresentados ao modelo em relação aos tensores criados com a sequência temporal das ressonâncias funcionais, visto que aqui não são eliminados quaisquer dados.

3.4 Avaliação de desempenho

3.4.1 Matriz de confusão

O primeiro passo na avaliação de métricas de desempenho de predição é a criação de uma tabela, denominada matriz de confusão. Ela apresenta nas colunas as classes reais e nas linhas as classes determinadas pelo modelo de predição [11]. O resultado é uma matriz nxn e, quando o número de classes é diferente de 2, também pode ser transformada em n matrizes 2x2, a fim de estimar os dados para cada classe as métricas distintamente. A métrica geral é uma médias das métricas para as classes individuais.

No objeto deste estudo, as classes reais são diagnóstico (D) e controle (C), enquanto as classes de predição são identificadas como predição de diagnóstico (PD) e predição de controle (PC). A Tabela 3.2 apresenta o formato da matriz de confusão para este cenário. Os valores *true* são os acertos de predição e os erros são representados como *false*; já o

diagnóstico é o caso positivo do nosso interesse (*positive*) e o controle é o caso negativo (*negative*).

Tabela 3.2. Matriz de confusão para avaliação binária em diagnóstico ou controle.
C: Controle; D: Diagnóstico;
PC: Predição como controle; PD: Predição como diagnóstico;
TN: *True negative*, acerto em controle; FN: *False negative*, erro em controle;
TP: *True positive*, acerto em diagnóstico; FP: *False positive*, erro em diagnóstico

	C	D
PC	TN	FN
PD	FP	TP

.

3.4.2 Acurácia

O quantitativo de acertos é medido pela acurácia (*accuracy*): quantas predições corretas do total de predições realizadas [11]. O cálculo da acurácia é dado por

$$Acc = \frac{TP + TN}{D + C},$$

no qual TP e TN representam o número de verdadeiros e falsos positivos respectivamente e a soma $D + C$ representa o número total de imagens (diagnóstico e de controle).

Neste caso, a soma das predições corretas em diagnóstico e controle e o total de casos avaliados nas mesmas classes. Oportunamente, um cenário com número idêntico de casos em cada conjunto de dados permite uma avaliação mais equilibrada das métricas – já que eventualmente o modelo pode acertar mais em uma classe que noutra.

3.4.3 Precisão

O grau de variação de resultados em relação ao diagnóstico é medido pela precisão (*precision*): Do total de afirmações positivas (presença do transtorno), quantas realmente acertaram [11]. A equação

$$Prec = \frac{TP}{TP + FP} = \frac{TP}{PD}$$

apresenta o cálculo da precisão, onde TP e FP configuram o número de verdadeiros e falsos positivos respectivamente, que somam o total de predições feitas como diagnóstico.

3.4.4 Sensibilidade

A quantificação dos acertos em diagnóstico, isoladamente, é medida diretamente pela razão entre o número de acertos na presença de transtorno (verdadeiros positivos) e o total de casos reais de diagnóstico. A equação

$$Sens = \frac{TP}{TP + FN} = \frac{TP}{D},$$

demonstra o cálculo da sensibilidade (*sensitivity/recall*) [11].

3.4.5 Especificidade

A capacidade de prever a exclusão do diagnóstico, neste caso, é dado pela especificidade (*specificity*) [11], que é calculada por

$$Spec = \frac{TN}{TN + FP} = \frac{TN}{C},$$

e aponta quantos acertos foram feitos dentre os casos efetivamente de controle.

3.4.6 F1 score

F1 score, *F-score* ou *F-measure*, a depender das referências, mede a confiabilidade da afirmação do diagnóstico. É dada pela média harmônica entre precisão e sensibilidade [11]. O equacionamento é dado por

$$F1 = \frac{2}{\frac{1}{Prec} + \frac{1}{Sens}} = \frac{2.Prec.Sens}{Prec + Sens}.$$

3.4.7 Função loss

Do inglês, o verbete *loss* significa perda. Entende-se que o modelo deve minimizar as suas perdas durante o treinamento, isto é, buscar o ponto ótimo de desempenho. "O propósito das funções *loss* é computar o quanto o modelo precisa buscar minimizar durante o treinamento" [7] (em tradução direta).

Há diversas formas de quantificar *losses*: de forma probabilística ou por regressão, por exemplo. Neste desenvolvimento, a classe de entropia cruzada binária (*binary cross-entropy*) – uma das formas probabilísticas de estima – foi utilizada. Há um equivalente

para cenários multiclasse, *categorical cross-entropy*, que não é conveniente neste escopo mas pode ser aplicado em avaliações de múltiplos diagnósticos.

Neste trabalho, a função *loss* é observada apenas nos registros de treinamento, para avaliação inicial das condições de aprendizado dos modelos gerados. Espera-se que haja um decaimento dos valores gerados conforme a evolução do treinamento. Um valor de *loss* sem decaimento ao longo dos ciclos de treinamento (*epochs*) é associado a falta de aprendizagem do modelo em relação ao banco de dados utilizado.

4 Resultados e discussões

4.1 Quantitativos de esforço computacional

Uma limitação das análises por modelos de inteligência artificial – em destaque as redes convolucionais – está na necessidade de se analisar um volume de dados extremamente grande. Para imagens estáticas, bidimensionais (2D), usualmente o aprendizado é feito com base em milhares de entradas (ou mesmo milhões, a depender da rede). No caso de imagens médicas, há proporcionalmente muito menos imagens disponíveis em bancos de dados com acesso público e muito mais dados em cada imagem.

A Tabela 4.1 apresenta o volume de informações com o qual foram alimentados os modelos produzidos neste trabalho, com as informações com e sem *data augmentation* em colunas distintas – não nesta ordem, considerando que cada *voxel* (unidade da representação espacial de uma imagem digital) é representado por 4 *bytes*. O total de *frames* (quantidade de capturas no tempo) varia para cada rotina, como se pode observar na referida tabela.

Tabela 4.1. Volumes de dados para cada conjunto de treinamento (*tasks*). Considerando o tamanho do *slice*, o quantitativo de *slices* e *frames*, o número de imagens disponíveis para o treinamento e validação, são estimados os quantitativos de dados, em *bytes*, para cada execução relacionada às *tasks*.

Bloco (*task*)	Tamanho do *Slice*	Frames	Número de imagens	Dados base (MB)	Fator de aumento	Dados aumentados (GB)
BART	64x64	267	98	429	10	3,13
BHT	64x64	79	100	129	20	1,76
Gate	64x64	112	324	595	6	2,84
PAM-Enc	64x64	242	90	357	10	2,60
PAM-Ret	64x64	268	90	395	10	2,88
Rest	64x64	152	118	294	10	2,15
Rest(a)	64x64	150	178	437	10	3,19
Rest(b)	64x64	240	68	267	10	1,95
SCAP	64x64	291	100	477	10	3,48
Stops.	64x64	208	100	341	10	2,49
Task sw.	64x64	208	100	341	10	2,49
Anat.	256x256	60*	98	1541	-	-

**Slices* compondo o volume de entrada para este caso.

O total de execuções utilizadas para análise (completadas com sucesso) foi de 850; já o total de tempo dessas execuções somou mais de 220 horas. Para este cálculo, foram utilizados apenas os *logs* gerados ao final das rotinas, portanto, eventuais interrupções no processamento – como na inatividade do Google Colab – não têm horas computadas.

4.2 Destaques de desempenho em classificação de imagens quanto ao diagnóstico

Nas versões preliminares, poucos casos se destacaram em métricas de predição. No conjunto de dados *Rest*(b), o treinamento no *slice* 16 – nível 50% – acertou todos os casos de predição para diagnóstico, mas apenas 1/3 dos casos de controle, sendo ainda o melhor resultado de todos. Tal fator foi crucial na decisão por alterações no modelo de base para evolução dos treinamentos. Nas seções a seguir são apresentados exemplos dos resultados considerados de destaque, por métricas do desempenho ou por representarem casos particulares dos treinamentos.

4.2.1 Modelo base – com sinal de alta ativação segmentado

Utilizando o *F1 score* como referência, por representar uma proporção baseada nas predições acertadas de diagnóstico e controle, os casos se destacam neste cenário por obterem esta métrica acima de 75% são apresentados na Tabela 4.2, onde aparecem, em ordem, acurácia, precisão, sensibilidade, especificidade e *f1 score*.

Tabela 4.2. Casos de destaque em métricas para o modelo base – *F1 score* acima de 0,75. Reportadas as métricas de acurária, precisão, sensibilidade, especificidade e *F1-score* para as relativas *tasks* e *slices*.

Task	*Slice*	Acc.	Prec.	Sens.	Spec.	F1
Task switch	20	0,85	0,77	1,00	0,70	0,87
Stopsignal	19	0,85	0,82	0,90	0,80	0,86
Task switch	19	0,85	0,89	0,80	0,90	0,84
SCAP	19	0,80	0,80	0,80	0,80	0,80
Task switch	17	0,80	0,80	0,80	0,80	0,80
Rest	19	0,80	0,88	0,70	0,90	0,78
SCAP	15	0,80	0,88	0,70	0,90	0,78
BHT	24	0,75	0,73	0,80	0,70	0,76

As Figuras 4.1 e 4.2 apresentam a evolução dos treinamentos para o melhor desempenho (*F1 score*) e pior desempenho nesta execução, respectivamente. Os casos são: *Task switch, slice* 20, como melhor e PAM-Ret, *slice* 19, como pior – por ter errado todos do grupo controle e um do grupo diagnóstico, gerando métricas incalculáveis em divisão por zero.

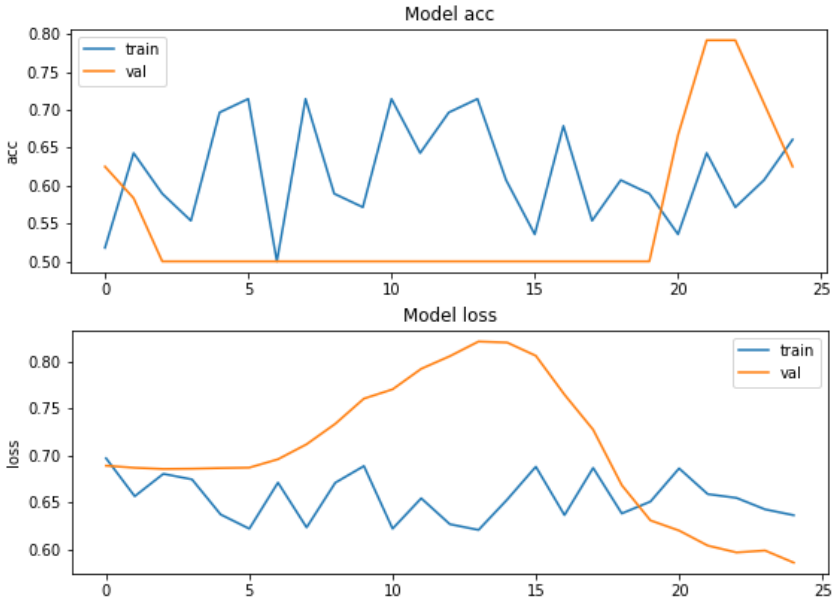

Figura 4.1. Gráfico com evolução do treinamento ao longo de 25 épocas do melhor desempenho no modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. Neste cenário, a acurácia não apresentou tendência clara de crescimento no treinamento, mas teve grande salto na validação. A função *loss* oscila no treimaneto, mas tem grande queda na validação à partir da *epoch* 15.

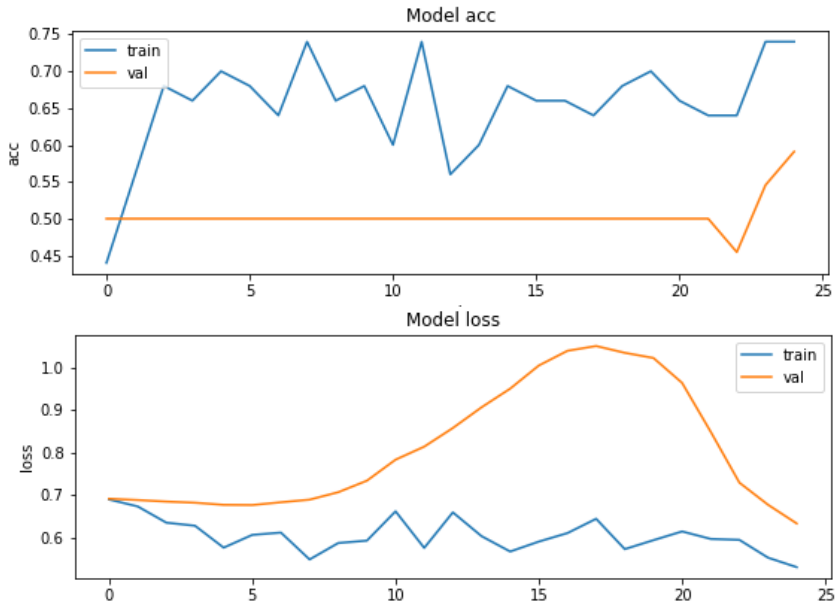

Figura 4.2. Gráfico com a evolução do treinamento ao longo de 25 épocas do pior desempenho no modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. Apesar de crescer a acurácia do treinamento, a validação aponta a falta de aprendizagem real. A função aponta *loss* crescente na validação, não reduzindo sensivelmente de um valor inicial em qualquer ponto.

Na observação dos gráficos, nota-se que no melhor caso, existe um ponto de máximo na acurácia da validação enquanto a função *loss* decresce para o cenário de melhor desempenho. Este é potencialmente o ponto de salvamento do modelo que será utilizado na predição.

Já no cenário de pior desempenho desta rotina, apesar de os valores do treinamento se apresentarem não muito diferentes do outro, os valores para a validação se mantém estáveis na acurácia da validação por longo período – indicando pouca ou nenhuma variação no aprendizado – enquanto a função *loss* apenas se destaca por apresentar valor elevado, também corroborando com a baixa aprendizagem para aquele conjunto de dados.

A característica de salvar apenas o melhor modelo é fundamental para o bom desempenho na tarefa de predição, visto que a aleatoriedade do modelo não necessariamente leva a uma evolução progressiva ao longo das *epochs* de treinamento. Todos os casos de destaque positivo apresentam como característica comum neste cenário – modelo base – picos superiores de acurácia e inferiores na função *loss* e nenhum deles apresentou tendência de crescimento regular. Os demais gráficos de treinamento para os modelos destacados podem ser encontrados no Anexo 6.2.1. As métricas de predição para todas as *tasks* e todos os *slices* segmentados estão disponíveis no Anexo 6.3.1.

4.2.2 Modelo com data augmentation por rotação

Executadas as rotinas com o aumento da variabilidade de dados de treinamento, as métricas destas foram extraídas e os destaques são apresentados na Tabela 4.3, as colunas contemplam as métricas de acurácia, precisão, sensibilidade, especificidade e *f1 score*, em ordem.

Tabela 4.3. Casos de destaque em métricas para o modelo com *data augmentation* – *F1 score* acima de 0,75. Reportadas as métricas de acurária, precisão, sensibilidade, especificidade e *F1-score* para as relativas *tasks* e *slices*.

Task	*Slice*	Acc.	Prec.	Sens.	Spec.	F1
Task switch	20	0,85	0,77	1,00	0,70	0,87
Task switch	17	0,80	0,71	1,00	0,60	0,83
Task switch	22	0,80	0,71	1,00	0,60	0,83
SCAP	14	0,85	1,00	0,70	1,00	0,82
Task switch	19	0,80	0,80	0,80	0,80	0,80
BHT	24	0,80	0,80	0,80	0,80	0,80
BHT	17	0,75	0,67	1,00	0,50	0,80
BHT	19	0,75	0,69	0,90	0,60	0,78
Stopsignal	15	0,80	0,88	0,70	0,80	0,78

As Figuras 4.3 e 4.4 apresentam a evolução dos treinamentos para o melhor desempenho (*F1 score*) e pior desempenho nesta execução, respectivamente. Os casos são: *Task switch, slice* 20, como melhor e *Stopsignal, slice* 14, como pior – por ter errado todos do grupo controle, gerando métricas incalculáveis em divisão por zero.

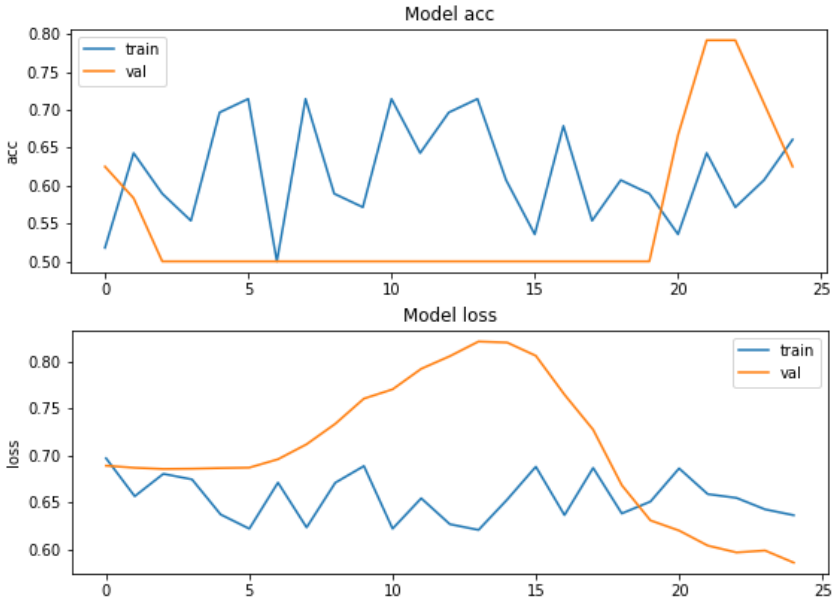

Figura 4.3. Gráfico com a evolução do treinamento ao longo de 25 épocas do melhor desempenho no modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. Observa-se principalmente a tendência de aumento na acurácia e queda na função *loss* na validação.

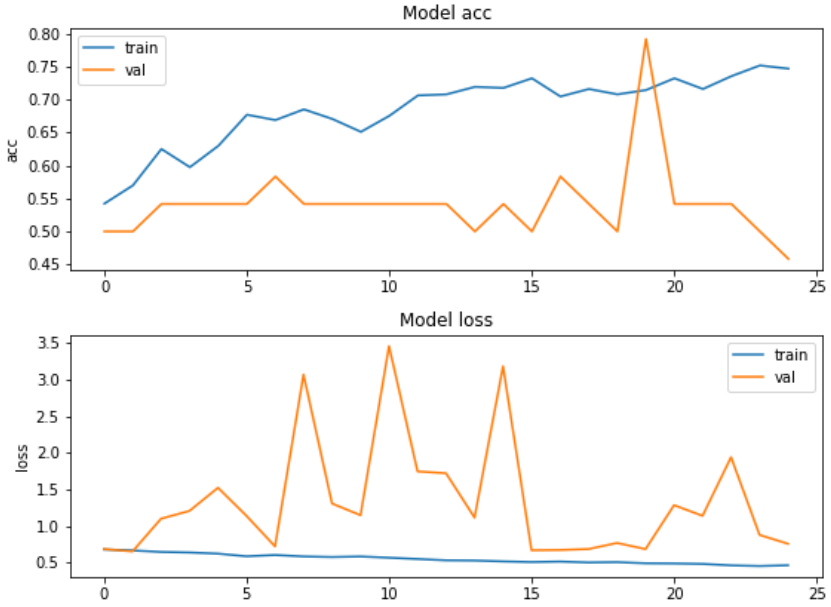

Figura 4.4. Gráfico com a evolução do treinamento ao longo de 25 épocas do pior desempenho no modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. Destaca-se a instabilidade da validação, apesar de o treimanento tentar inferir uma aprendizagem com acurácia crescente. A função *loss* inicia e se mantém em valores altos, oscilando mais para cima do que para baixo.

O cenário do treinamento para o melhor desempenho com *data augmentation* é similar ao caso anterior no modelo base. Já para o desempenho inferior, apesar de no treinamento a acurácia apresentar uma tendência crescente e a função *loss* apresentar estabilidade, a validação difere em muito deste comportamento. Tais fatores podem ser indícios de *overfit* e não de um aprendizado real, corroborado pelos dados de predição.

Os demais gráficos de treinamento para as rotinas destacadas nesta seção são encontradas no Anexo 6.2.2, enquanto as métricas de todas execuções relacionadas às diversas *tasks* e *slices* estão presentes no Anexo 6.3.2.

4.2.3 Tuning de casos selecionados

Do modelo base foram selecionados, dentre os casos de destaque, cinco pares *task* e *slice*: *Task switch*, 20; *Stopsignal*, 19; SCAP, 15; *Rest*, 19 e BHT, 24. A escolha foi baseada na acurácia e na precisão, para aqueles que já haviam sido segmentados pelo *F1 score*, por este motivo a rotina SCAP utiliza o *slice* 15 e não o 19. Também foi priorizado utilizar casos com diferentes *tasks*, sem utilizar mais de um *slice* da mesma imagem. Os casos em que o *tuning* superou as métricas do modelo base são apresentados na Tabela

4.4, com apresentação das configurações do modelo e de acurácia, precisão, sensibilidade, especificidade e *f1 score*, em ordem.

Tabela 4.4. Casos de destaque em métricas para *tuning – F1 score* acima do modelo base. Reportadas as métricas de acurária, precisão, sensibilidade, especificidade e *F1-score* para as *tasks* e *slices* selecionados e suas variações em quantitativo de filtros, tamanho de *kernel* e *dropout* implementados.

Task	*Slice*	*Filters*	*Kernel*	*Dropout*	Acc.	Prec.	Sens.	Spec.	F1
Rest	19	64	3	0,2	0,85	0,82	0,90	0,80	0,86
Stopsignal	19	128	3	0,2	0,85	0,82	0,90	0,80	0,86
Rest	19	16	5	0,2	0,85	0,89	0,80	0,90	0,84
BHT	24	16	3	0,3	0,80	0,80	0,80	0,80	0,80
BHT	24	32	5	0,3	0,80	0,80	0,80	0,80	0,80
Rest	19	64	5	0,2	0,80	0,80	0,80	0,80	0,80
Rest	19	64	5	0,3	0,80	0,80	0,80	0,80	0,80

Gráficos de treinamento e validação para estes casos de destaque são apresentados no Anexo 6.2.3. Como nos cenários anteriores, não houve tendência visível de crescimento, mas grandes oscilações – principalmente no treinamento – os resultados de predição para todas as configurações para os pares de *task* e *slice* selecionados são apresentados no Anexo 6.3.3.

4.2.4 n-fold de casos selecionados

Nos mesmos pares de *task/slice* utilizados para avaliação de desempenho através de *tuning*, foram realizadas as rotinas de verificação cruzada do tipo *n-fold*. A Tabela 4.5 apresenta os resultados para todas as execuções.

Tabela 4.5. Casos selecionados para o *n-fold*: resultados de métricas de desempenho. Reportadas as métricas de acurária, precisão, sensibilidade, especificidade e *F1-score* para as *tasks* e *slices* selecionados

Task	*Slice*	Acc.	Prec.	Sens.	Spec.	F1
Task switch	20	0,73	0,71	0,75	0,70	0,73
Stopsignal	19	0,70	0,67	0,80	0,60	0,73
Rest	19	0,68	0,63	0,85	0,50	0,72
BHT	24	0,71	0,70	0,75	0,68	0,72
SCAP	15	0,63	0,60	0,73	0,53	0,66

Não são incluídos aqui os gráficos de treinamento e validação, visto terem sido realizadas inúmeras rotinas com as n rotações realizadas na validação cruzada. A apresentação gráfica das métricas é apresentada no Anexo 6.3.4. A avaliação sobre a variância apresentada pelos modelos com o resultado desta rotina será realizadas na Seção 4.3.3.

4.2.5 Modelo para imagens anatômicas

Para as imagens anatômicas, apenas um modelo dentre os gerados pela rotina de *tuning* é destacado dos demais. O modelo com 32 e 64 filtros, respectivamente, na camada de convolução, com *kernel* de tamanho 3 e *dropout* de 0,3 apresentou: (a) 60% de acurácia; (b) 75% de precisão; (c) 90% de especificidade; (d) 30% de sensibilidade e (e) 43% no *F1 score*. Nota-se que o modelo teve predição melhor para casos de controle do que para diagnóstico, mas ainda assim o quantitativo de erro é bastante grande. A Figura 4.5 apresenta a evolução dos treinamentos para o caso destacado.

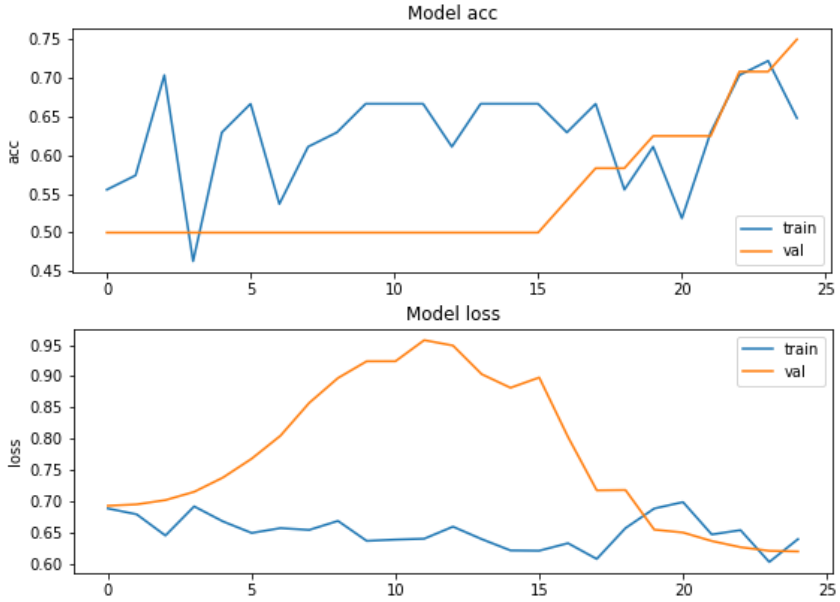

Figura 4.5. Gráfico com a evolução do treinamento ao longo de 25 épocas do melhor desempenho no modelo com *tuning* para imagem anatômica. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. Nota-se que há uma tendência crescente na acurácia e queda em *loss* no processo de validação, apesar de não ser observado no treinamento.

Se apresenta aqui uma tendência de crescimento na acurácia e queda na função *loss* para os dados de validação. Pode-se conceber como hipótese que um número maior de *epochs* gere uma tendência de melhora nos resultados, visto que a oscilação pode ser considerada ainda como um *underfit* – modelo subajustado, havendo a possibilidade de

se aprender mais dos dados fornecidos no treinamento. Contudo, os valores para esta rotina não estão aquém dos desempenhos de treinamento e validação de outros modelos realizados em imagens funcionais, mas as métricas de predição estão. Outros dois casos de desempenho próximo são apresentados no Anexo 6.2.4, enquanto as métricas de predição para todos os modelos gerados no *tuning* para estas entradas se apresenta no Anexo 6.3.5.

4.3 Estatísticas nos resultados dos treinamentos

4.3.1 Com relação às tasks

Um total de cinquenta e seis modelos de treinamento baseado nas imagens funcionais, com e sem *data augmentation*, apresentou *F1 score* acima de 70%. As quatro rotinas de execução que mais se destacaram positivamente foram *rest*, *breath hold*(BHT), *task switch* e SCAP, respectivamente.

Trinta e três modelos tiveram desempenho como destaques negativos, ou por zerarem os acertos em uma das classes – diagnóstico ou controle – ou por apresentarem *F1 score* menor do que 1/3. Os quatro maiores quantitativos foram para *rest*, *PAM-Enc*, *PAM-Ret* e *BART*, respectivamente. A Tabela 4.6 apresenta a distribuição entre as rotinas avaliadas para estes quantitativos positivos, negativos e a razão de casos positivos com negativos.

Tabela 4.6. Distribuição dos modelos com desempenho de destaque agrupado por *task*: quantitativo absoluto e percentual. São entendidas como preferenciais as *tasks* que apresentam mais destaques positivos e menos negativos.

Task	Absoluto positivo	Percentual positivo	Absoluto negativo	Percentual negativo	Razão Pos./Neg.
BART	1	1,8%	4	12,1%	0,25
BHT	12	21,4%	1	3,0%	12
Gate	0	0%	1	3,0%	0
PAM-Enc	0	0%	8	24,2%	0
PAM-Ret	1	1,8%	5	15,2%	0,2
Rest	18	32,1%	11	33,3%	1,64
SCAP	9	16,1%	1	3,0%	9
Stopsignal	5	8,9%	2	6,1%	2,5
Task switch	10	17,9%	0	0%	*
TOTAL	56	100%	33	99,9%	-

* Não houveram casos no destaque negativo, portanto a relação não pode ser estabelecida, já que tende ao infinito.

4.3.2 Com relação aos slices

Dos quantitativos de modelos com os destaques positivos – com *F1 score* maior que 0,7 – ou negativos – *F1 score* menor que 0,33 ou erros de todos os casos em uma classe – foram verificados a quais regiões estes *slices* pertencem. Como as reconstruções de imagem segmentam o eixo z (profundidade) com eventual diferença no quantitativo de camadas, preferiu-se representar a posição dos elementos utilizados na análise como uma relação proporcional desta altura. Assim, a Tabela 4.7 apresenta a distribuição dos casos para estes níveis, avaliando números absolutos e percentuais dos resultados. A Figura 4.6 apresenta a distribuição das métricas para o exemplo de *task switch* em todos os *slices* analisados – é possível observar a tendência de maiores métricas em um certo *range* de *slices*. Outras métricas estão disponíveis no Anexo 6.3, distribuídos por modelos e *tasks*.

Tabela 4.7. Distribuição dos modelos com desempenho de destaque agrupado por *slices*: quantitativo absoluto e percentual. São entendidos como preferenciais os *slices* que apresentam mais destaques positivos e menos negativos, quando possível.

Nível do eixo z (%)	Absoluto positivo	Percentual positivo	Absoluto negativo	Percentual negativo	Razão Pos./Neg.
40	6	10,7%	8	24,2%	0,75
45	9	16,1%	4	12,1%	2,25
50	9	16,1%	4	12,1%	2,25
55	9	16,1%	4	12,1%	2,25
60	8	14,3%	4	12,1%	2
65	5	8,9%	3	9,1%	1,67
70	4	7,1%	3	9,1%	1,33
75	6	10,7%	3	9,1%	2
TOTAL	56	100%	33	99,9%	-

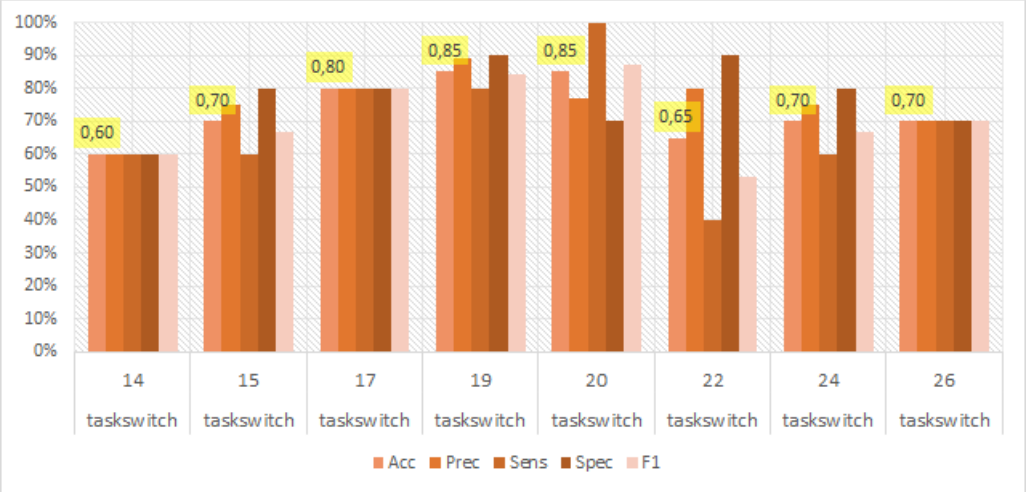

Figura 4.6. Distribuição das métricas de desempenho associadas aos *slices* da rotina *task switch* no modelo base, destacando a acurácia em amarelo em cada *slice*. O eixo vertical apresenta a porcentagem obtida na métrica, enquanto o eixo horizontal apresenta as *tasks* e *slices* relacionadas. Para cada conjunto são apresentadas, respectivamente, acurácia, precisão, sensibilidade, especificidade e *F1-score*.

4.3.3 Com relação ao n-fold

Considerando que as predições de um classificador não devem ser muito diferenciadas por se excluir apenas um elemento do conjunto de dados do treinamento, a variância deve se aproximar de

$$Acc_{cv}.(1 - Acc_{cv})/n$$

[15]. Acc_{cv} é a acurácia estimada para o processo de validação cruzada e n é o total de segmentações do conjunto de dados de treinamento. Neste cenário, foram realizados 40 rotinas, com um caso de diagnóstico e outro de controle no bloco segmentado, portanto $n = 40$, para as cinco rotinas avaliadas. O resultado é apresentado na Tabela 4.8, com um comparativo entre a acurácia obtida no treinamento padrão e a obtida no processo de *n-fold* e a variância calculada para cada um dos casos. A Figura 4.7 apresenta as demais métricas calculadas para os modelos de treinamento com validação cruzada.

Tabela 4.8. Comparativo entre acurácia no treinamento padrão e no *n-fold* para casos selecionados e variância estimada. Acc - acurária do modelo base; Acc_{cv} - acurácia do modelo de validação cruzada.

task, slice	Acc	Acc_{cv}	Variância
task switch, 20	0,85	0,73	0,50%
BHT, 24	0,75	0,71	0,51%
stopsignal, 19	0,85	0,70	0,53%
rest, 19	0,80	0,68	0,55%
SCAP, 15	0,80	0,63	0,59%

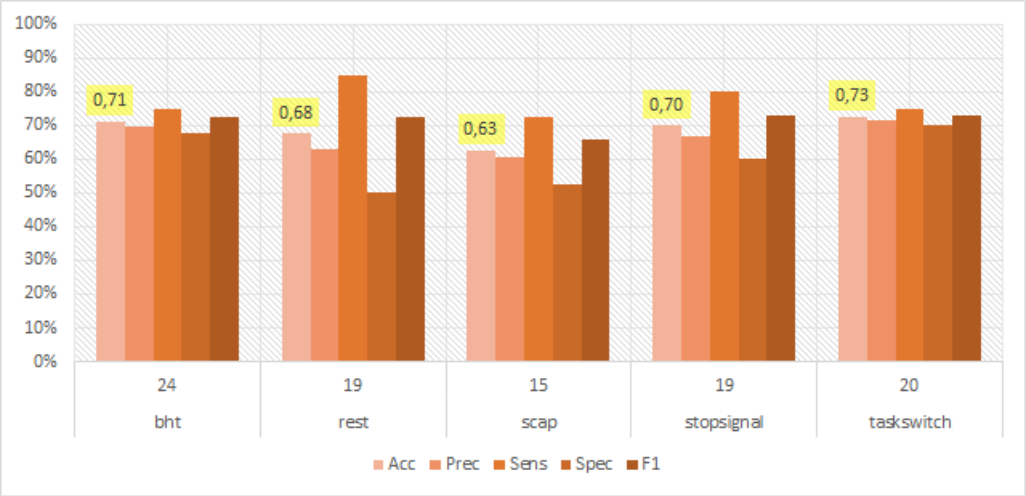

Figura 4.7. Distribuição das métricas de desempenho da técnica de validação cruzada, associadas aos modelos selecionados, destacando a acurácia em amarelo em cada modelo. O eixo vertical apresenta a porcentagem obtida na métrica, enquanto o eixo horizontal apresenta as *tasks* e *slices* relacionadas. Para cada conjunto são apresentadas, respectivamente, acurácia, precisão, sensibilidade, especificidade e *F1-score*.

5 Considerações finais

Na avaliação de imagens de ressonância magnética funcional com uso de redes convolucionais para classificação quanto ao diagnóstico em esquizofrenia, tendo sido realizado não menos do que 176 treinamentos diferentes (11 grupos de *tasks* com 8 diferentes *slices*, com e sem *data augmentation*), foram observadas métricas de predição consideradas suficientes para justificar uma continuação da abordagem proposta. A expectativa inicial dos pesquisadores era de 70% em acurácia – tendo sido também considerado o *F1 score*, a fim de observar tendência clara de não aleatoriedade. A consistência dos resultados é observada no quantitativo de modelos treinados que superaram esta estimativa, no caso 56 dos 176 treinamentos, sendo quase 32% do total.

A segmentação de níveis de ativação nos exames de ressonância magnética funcional, para servir de entrada para as redes convolucionais apresentou potencial de classificação quanto a diagnóstico e exclusão. Com a técnica de limiarização foi possível diminuir a quantidade de informação em cada *slices*, reduzindo o tempo de treinamento e os parâmetros treináveis dos modelos. Realizando isto foi reduzido então o custo, e ainda se obteve resultados expressivos.

O procedimento de *data augmentation*, quando tomado o limite inferior de *F1 score* em 0,7, resultou na melhora significativa de onze treinamentos em relação ao modelo base – para aqueles que tinham métricas abaixo deste limiar – e ainda melhorou os resultados para outros cinco que já apareciam entre os destaques. O aspecto, contudo, não foi sempre positivo, visto que em outros dezesseis modelos o treinamento com *data augmentation* resultou em métricas inferiores ao modelo base. Entende-se desta avaliação que o aumento de dados realizado não é fator decisivo no desempenho, mas que é conveniente que seja realizado para fins de investigação. Mesmo com o aumento dos dados, o total de exemplos para o treinamento não chegou a mil, o que é considerado número baixo para o treinamento de redes complexas como as convolucionais. Maiores aumentos não foram realizados pela limitação dos sistemas computacionais disponíveis.

A técnica de *tuning* proporcionou um refino nos modelos. Alguns apresentaram melhora de desempenho, como é o caso do *slice* 19 da *task rest*, que no modelo base de 32

e 64 filtros na convolução, *kernel* igual a 3 e 20% de *dropout* se destacou com 80% de acurácia e 78% de *F1 score*. Após o *tuning*, o mesmo conjunto aparece quatro vezes entre os sete melhores valores de *F1 score* (Tabela 4.4). Apesar de ter havido destaque com 16/32 e 128/256 filtros, a maior parte dos destaques foram modelos com 32 e 64 filtros na primeira camada. Houve mais destaque de modelos com um *kernel size* de 5 e 0.2 de *dropout*. Tais resultados colocam esta técnica como uma das abordagens adequadas para otimização dos modelos em redes convolucionais, visto que os resultados podem ser tanto modelos que demandam maior esforço computacional, mas geram métricas superiores, como modelos significativamente menores e de menor custo computacional com métricas equivalentes.

O processo de *n-fold* não apresentou resultados positivos nos treinamentos, os quais seriam dados por métricas equivalentes aos treinamentos gerais. A baixa quantidade de exemplos faz com que o *folding* – retirar imagens do conjunto de treinamento – e a aleatoriedade da inicialização do modelo possam reduzir significativamente seu desempenho. Vale ressaltar que os resultados não foram todos ruins, já que quatro dos cinco modelos testados exibiram valores de *F1 score* maiores que 70%. Não foi possível executar o processo com outros *slices* e *tasks* pelo tempo disponível para realização do trabalho e poder computacional disponível, visto que, uma rotina com um *slice* de uma *task* demorou cerca de 6 horas de execução, mesmo tendo um número de *epochs* reduzido de 25 para 20. Este último fator, inclusive, pode ser ajustado em casos de uso desta técnica, permitindo que um modelo de melhor desempenho possa ser gerado em iterações mais tardias.

Foi observado que, para se conseguir boas predições, não é determinante que o modelo apresente uma curva crescente de treinamento e validação, normalmente associadas a uma evolução da aprendizagem. De forma geral, nos processos de treinamento e validação, embora a acurácia no gráfico tenha variado bastante ao longo das 25 épocas, em algum momento, o modelo definiu pesos que determinaram acurácia razoável e salvou este ponto. Ao se fazer a predição com o melhor modelo, salvo ali, alcançou-se resultados de até 85% de acurácia, validando o processo.

É possível notar que algumas *tasks* evidenciaram desempenho superior às demais classificações, como é o caso do *task switch* e *stopsignal*, que alcançaram, em alguns cenários, um valor de sensibilidade de 100% e 90% respectivamente, além de acurácia de 85% (Tabela 4.2). Outro caso é o do *rest* que, no processo de *tuning*, mostrou os mesmos valores de sensibilidade e de acurácia equivalentes aos do *stopsignal* (Tabela 4.4). Analisando os 56 testes que se destacaram por apresentarem *F1 score* acima de 70%, 18 foram da *task rest* e 12 da BHT (Tabela 4.6).

Entende-se, portanto, que há um potencial maior no diagnóstico quando determinadas

rotinas forem realizadas como exame complementar, se houver o desenvolvimento de uma ferramenta de auxílio a diagnóstico baseada em técnicas aqui empregadas. Os casos positivos representados por *rest*, BHT, *task switch* e *SCAP* somados representam 88% dos casos destaque. A relação entre casos positivos e negativos aponta, ainda, a *task switch*, BHT e SCAP como mais relevantes para ampliar as investigações, visto que a *task rest* teve muitos mais destaques negativos.

Mais de 62% dos casos positivos estão compreendidos em segmentações realizadas entre 40% e 60% dos planos axiais das imagens. Ainda que alturas semelhantes também tenham tido os maiores quantitativos de índices de baixo desempenho, estas estão desatreladas às *tasks* que geraram as métricas positivas. Isto mostra que determinadas rotinas de testes ativam áreas diferentes do cérebro e que a informação não está exatamente nessa porção em todos os casos. Apesar das limitações de quantidade de imagens e de poder computacional este trabalho mostrou resultados equiparáveis à literatura.

Hipóteses para trabalhos futuros

Num cenário contemporâneo de uso de inteligência artificial, os processos de explicação (*explainable artificial intelligence – X-AI*) são, além de incentivados, necessários. Cabe trazer os processo de X-AI para as análises aqui realizadas, a fim de apontar os fatores que se tornam decisivos na classificação das imagens. Este aspecto pode também contribuir, por apontar as regiões de interesse, na observação clínica das imagens. O modelo de X-AI a ser implementado deve apontar seções do tensor que possam destacar a região espacial e temporal tomados como principais para a classificação. Desta forma, pode haver o cruzamento de informação entre o estímulo e a resposta apresentada pelo participante.

As abordagens aplicadas neste trabalho podem ser reproduzidas em conjuntos de imagens de outros diagnósticos de duas maneiras: de forma binária (diagnóstico *versus* controle) ou multiclasse (vários diagnósticos e controle), avaliando de forma mais abrangente o método proposto. Para tal, é demandado o uso de recursos computacionais mais avançados – CPU, GPU e memória – ou de maior tempo de execução. Cabe a tentativa de entrada com a imagem não segmentada ou com menores estágios de pré-processamento, se houver poder computacional para tal. Com uma demanda menor, a extração prévia de características ou sinais de interesse pode resultar em processamentos mais leves e direcionados – como a segmentação de regiões corticais específicas, reduzindo o volume observado.

Pode-se variar o método de pré-processamento, como outros níveis de segmentação

– utilizando níveis baixos e médios de intensidade, por exemplo – ou aumentar ainda mais o volume de dados com o *data augmentation*. Quanto aos dados de entrada, pode-se explorar cortes em outros planos. Também é possível realizar a técnica de *tuning* com mais parâmetros a serem variados e, com um maior poder computacional, elevar a quantidade de épocas do treinamento. Ampliar a quantidade de imagens exemplo para o treinamento também é fundamental, permitindo, entre outros fatores, explorar a técnica *n-fold* com menor variância.

Considera-se fundamental a implementação de rotina de salvamento de contexto para execuções em servidores remotos como, no caso apresentado, o uso do Google Colab. Tal rotina permite a continuação dos trabalhos em um estágio avançado, não sendo perdidas tantas horas de processamento como nos casos realizados neste trabalho. Espera-se que, desta forma, possam ser realizados treinamentos mais extensos em conjuntos semelhantes, utilizando características de aprendizagem crescente típicas das arquiteturas de redes convolucionais.

A fim de produzir uma ferramenta de auxílio a diagnóstico completa, cabe ressaltar que é preciso otimizar a interação para profissionais de outras áreas – principalmente médicos. Deve-se facilitar a entrada dos dados de imagem bruta e a saída com destaque para a classificação e resultados de X-AI, fomentando o trabalho de avaliação do resultado pelo profissional especialista. Isto evidencia a função da ferramenta como acessório e não como elemento definitivo, reconhecendo-se que o desenvolvimento é limitado pelas características dos dados utilizados no treinamento. Até que haja dados suficientes para a criação de ferramentas de aplicação global, é necessário destacar tais características.

Referências Bibliográficas

[1] Anthony O Ahmed, Peter F Buckley, and Mona Hanna. Neuroimaging schizophrenia: a picture is worth a thousand words, but is it saying anything important? *Current psychiatry reports*, 15:1–11, 2013.

[2] CJ Aine, Henry Jeremy Bockholt, Juan R Bustillo, José M Cañive, Arvind Caprihan, Charles Gasparovic, Faith M Hanlon, Jon M Houck, Rex E Jung, John Lauriello, et al. Multimodal neuroimaging in schizophrenia: description and dissemination. *Neuroinformatics*, 15(4):343–364, 2017.

[3] Muhammad Qutayba Almerie, Hassan Alkhateeb, Adib Essali, Hosam E Matar, and Emtithal Rezk. Cessation of medication for people with schizophrenia already stable on chlorpromazine. *Cochrane Database of Systematic Reviews*, 2007.

[4] DS American Psychiatric Association, American Psychiatric Association, et al. *Diagnostic and statistical manual of mental disorders: DSM-5*, volume 5. American psychiatric association Washington, DC, 2013.

[5] Peter F Buckley. Neuroimaging of schizophrenia: structural abnormalities and pathophysiological implications. *Neuropsychiatric disease and treatment*, 1(3):193–204, 2005.

[6] S. Y. Chaganti, I. Nanda, K. R. Pandi, T.G.N.R.S.N. Prudhvith, and N. Kumar. Image classification using svm and cnn. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–5, 2020.

[7] Francois Chollet et al. Keras. *GitHub*, 2015.

[8] Brasil. Ministério da Saúde. *PORTARIA Nº 364, DE 9 DE ABRIL DE 2013.* Ministério da Saúde, Brasília, 2012.

[9] P. Dalgalarrondo. *Psicopatologia e semiologia dos transtornos mentais.* Artmed Editora, 2018.

[10] P. Falkai, A. Schmitt, and N. Andreasen. Forty years of structural brain imaging in mental disorders: is it clinically useful or not? *Dialogues in clinical neuroscience*, 2018.

[11] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[12] M. Filippi. *fMRI Techniques and Protocols*. Neuromethods. Humana Press, 2009.

[13] Aurélien Géron. *Hands-on machine learning with scikit-learn and tensorflow: Concepts*. O'Reilly Media, 2017.

[14] E. Johnstone, C. D. Frith, T.J. Crow, J. Husband, and L. Kreel. Cerebral ventricular size and cognitive impairment in chronic schizophrenia. *The Lancet*, 308(7992):924–926, 1976.

[15] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[16] M. Medina, D. Lee, D. M. Garza, E. L. Goldwaser, T. T. Truong, A. Apraku, J. Cosgrove, and J. J. Cooper. Neuroimaging education in psychiatry residency training: needs assessment. *Academic Psychiatry*, 44(3):311–315, 2020.

[17] C. D. Mellon and L. D. Clark. A developmental plasticity model for phenotypic variation in major psychiatric disorders. *Perspectives in Biology and Medicine*, 34(1):35–43, 1990.

[18] G. Orru, W. Pettersson-Yeo, A. F. Marquand, G. Sartori, and A; Mechelli. Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. *Neuroscience & Biobehavioral Reviews*, 36(4):1140–1152, 2012.

[19] R. Phadnis, J. Mishra, and S. Bendale. Objects talk - object detection and pattern tracking using tensorflow. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1216–1219, 2018.

[20] Russell A Poldrack, Eliza Congdon, William Triplett, KJ Gorgolewski, KH Karlsgodt, JA Mumford, FW Sabb, NB Freimer, ED London, TD Cannon, et al. A phenome-wide examination of neural and cognitive function. *Scientific data*, 3(1):1–12, 2016.

[21] Russell A Poldrack, Eliza Congdon, William Triplett, KJ Gorgolewski, KH Karlsgodt, JA Mumford, FW Sabb, NB Freimer, ED London, TD Cannon, et al. A phenome-wide examination of neural and cognitive function. *Scientific data*, 3(1):1–12, 2016.

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[23] Conduct Science. Balloon analog risk test, 2022. Acesso em: 31 jan. 2023.

[24] K. Sitek. Can computers use brain scans to diagnose psychiatric disorders? *Science in the News*, 2016.

[25] F. Sultana, A. Sufian, and P. Dutta. Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 122–129. IEEE, 2018.

[26] J. M. Valverde, V. Imani, A. Abdollahzadeh, R; De Feo, M. Prakash, R. Ciszek, and J. Tohka. Transfer learning in magnetic resonance brain imaging: A systematic review. *Journal of imaging*, 7(4):66, 2021.

[27] Schnack HG; Nieuwenhuis M; van Haren NE; Abramovic L; Scheewe TW; Brouwer RM; Hulshoff Pol HE and Kahn RS. Can structural mri aid in clinical classification? a machine learning study in two independent samples of patients with schizophrenia, bipolar disorder and healthy subjects. *NeuroImage*, 2014.

[28] WHO. *International statistical classification of diseases and related health problems (11th ed.)*. World Health Organization, Geneva, Switzerland, 2019.

[29] X. Zhan and R. Yu. A window into the brain: advances in psychiatric fmri. *BioMed research international*, 2015.

[30] Hasib Zunair. 3d image classification from ct scans. *Keras Documentation*, 2020.

6 Anexos

6.1 Anexo I - Exemplos de visualização das imagens utilizadas nos treinamentos

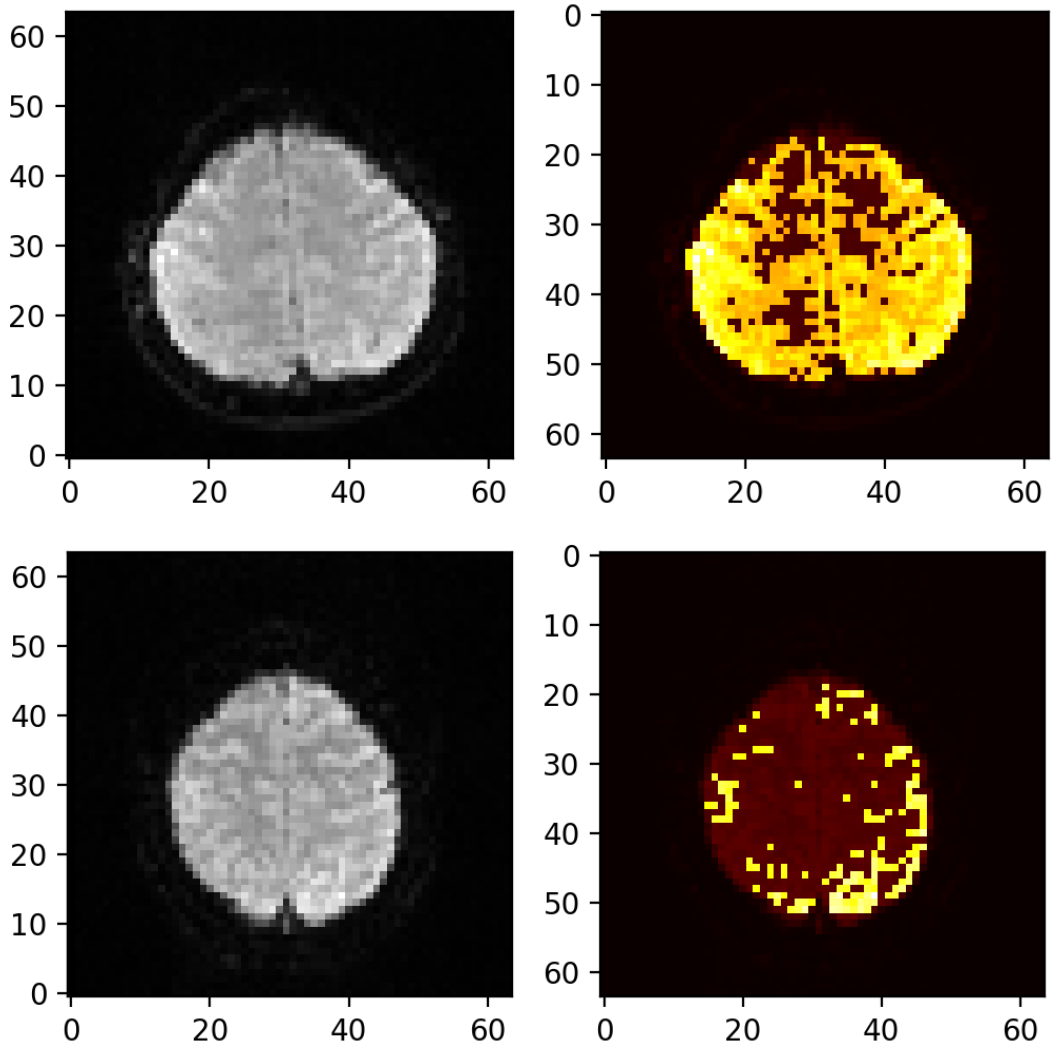

Figura 6.1. *Task* BHT, *slice* 24, exemplo de segmentação. Acima, diagnóstico; abaixo, controle. (E) Imagem original em escala de cinza; (D) Imagem em mapa de calor com destaque ao sinal segmentado. Os pontos amarelos representam o sinal relativo a altos níveis de oxigenação sanguínea. Os eixos representam os *pixels* da imagem recortada.

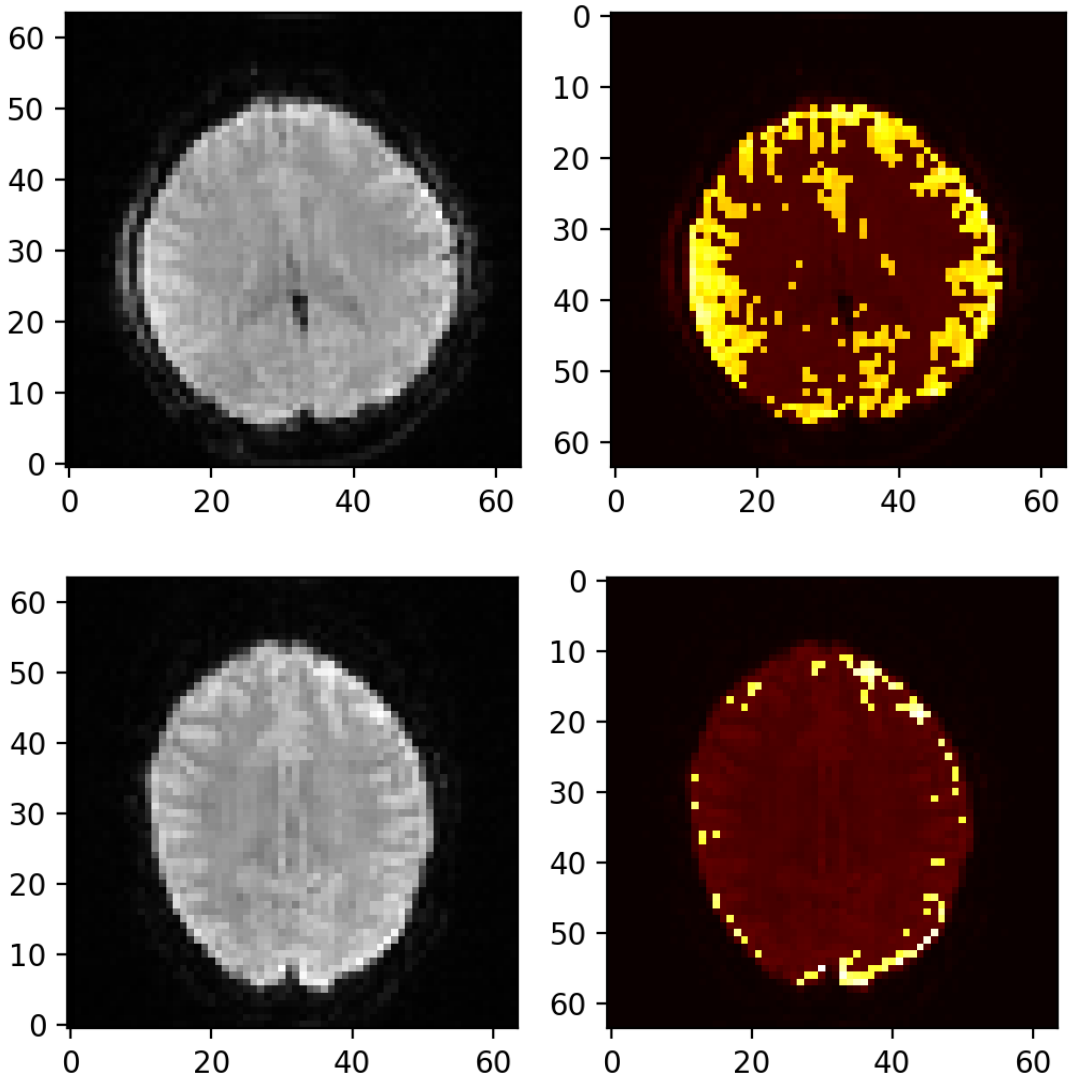

Figura 6.2. *Task Rest*, *slice* 19, exemplo de segmentação. Acima, diagnóstico; abaixo, controle. (E) Imagem original em escala de cinza; (D) Imagem em mapa de calor com destaque ao sinal segmentado. Os pontos amarelos representam o sinal relativo a altos níveis de oxigenação sanguínea. Os eixos representam os *pixels* da imagem recortada.

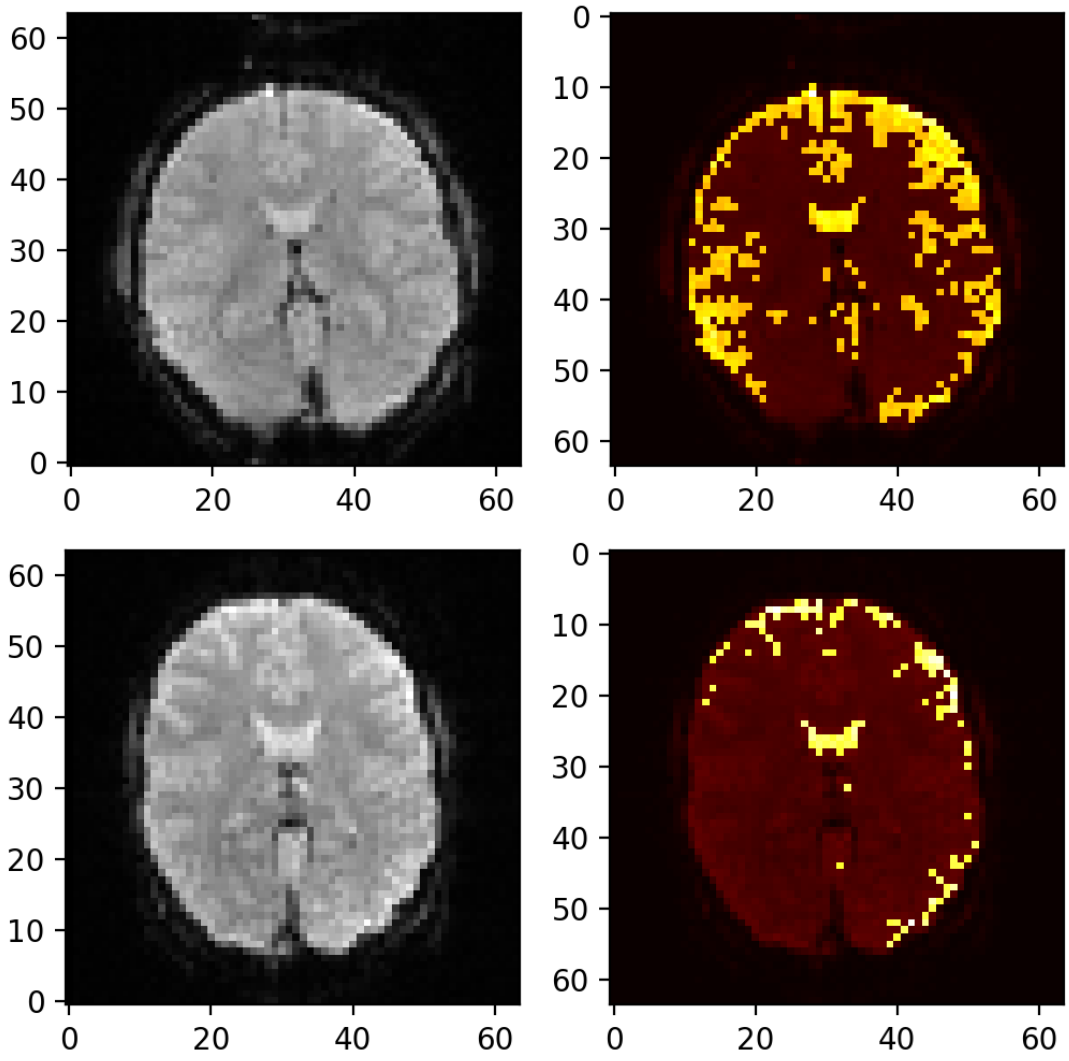

Figura 6.3. *Task* SCAP, *slice* 15, exemplo de segmentação. Acima, diagnóstico; abaixo, controle. (E) Imagem original em escala de cinza; (D) Imagem em mapa de calor com destaque ao sinal segmentado. Os pontos amarelos representam o sinal relativo a altos níveis de oxigenação sanguínea. Os eixos representam os *pixels* da imagem recortada.

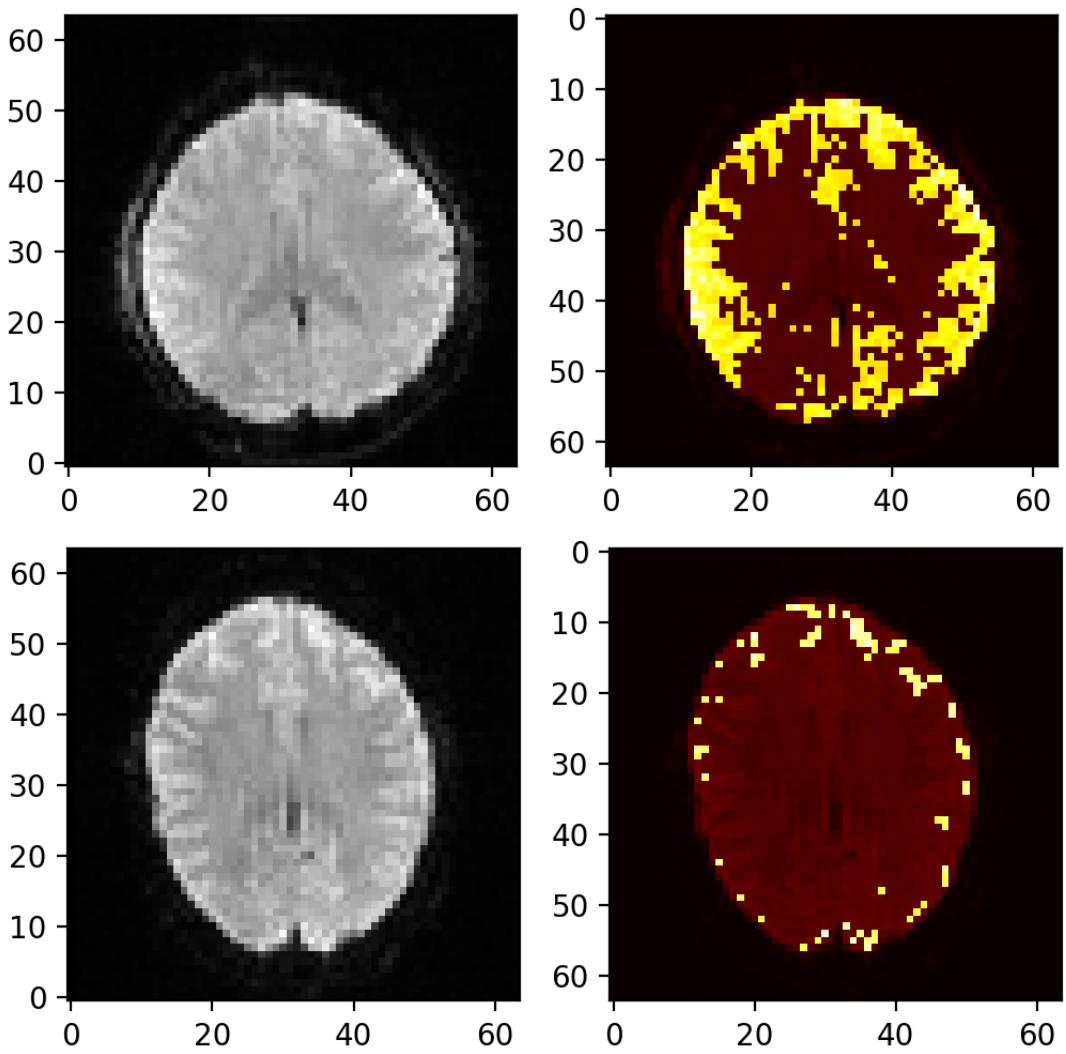

Figura 6.4. *Task Stopsignal*, *slice* 19, exemplo de segmentação. Acima, diagnóstico; abaixo, controle. (E) Imagem original em escala de cinza; (D) Imagem em mapa de calor com destaque ao sinal segmentado. Os pontos amarelos representam o sinal relativo a altos níveis de oxigenação sanguínea. Os eixos representam os *pixels* da imagem recortada.

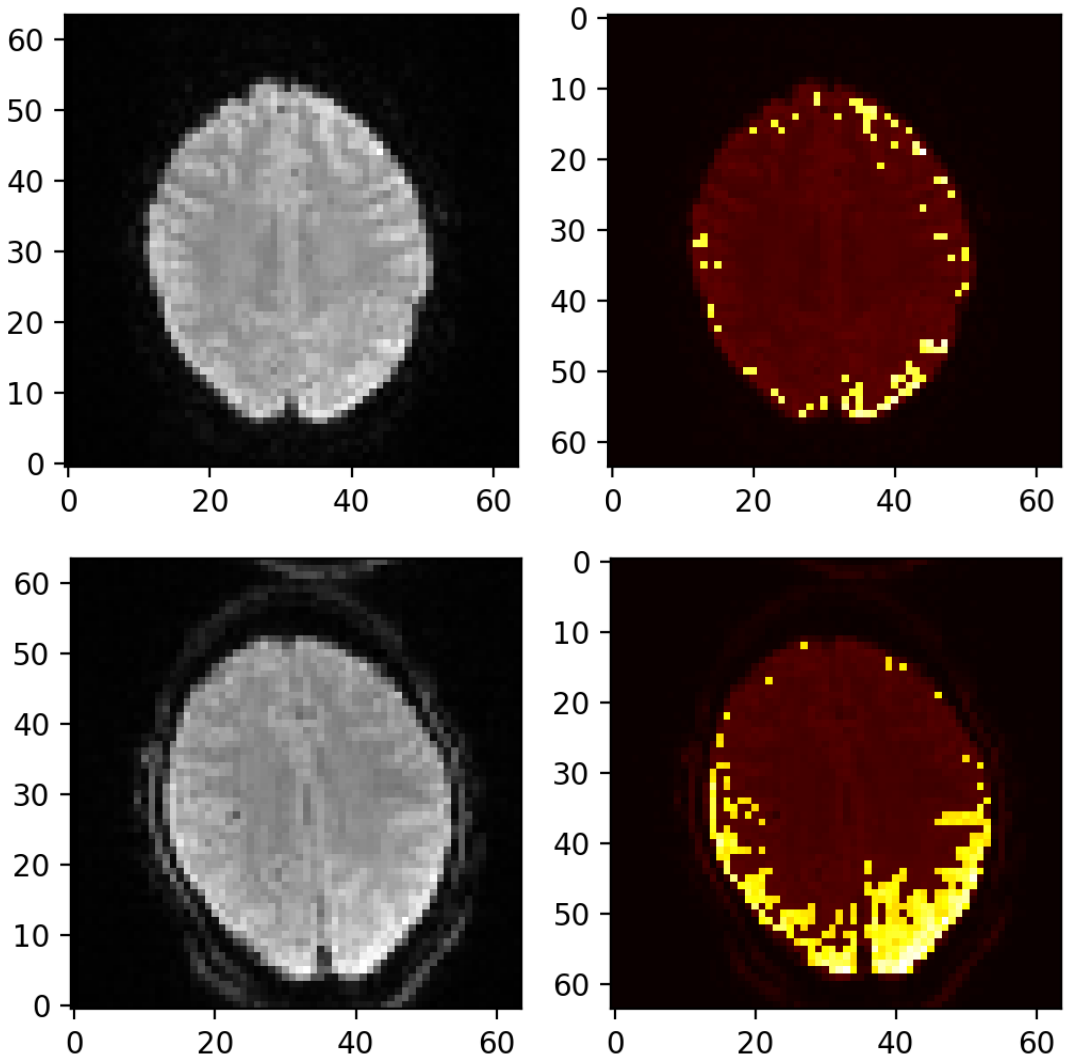

Figura 6.5. *Task switch*, *slice* 20, exemplo de segmentação. Acima, diagnóstico; abaixo, controle. (E) Imagem original em escala de cinza; (D) Imagem em mapa de calor com destaque ao sinal segmentado. Os pontos amarelos representam o sinal relativo a altos níveis de oxigenação sanguínea. Os eixos representam os *pixels* da imagem recortada.

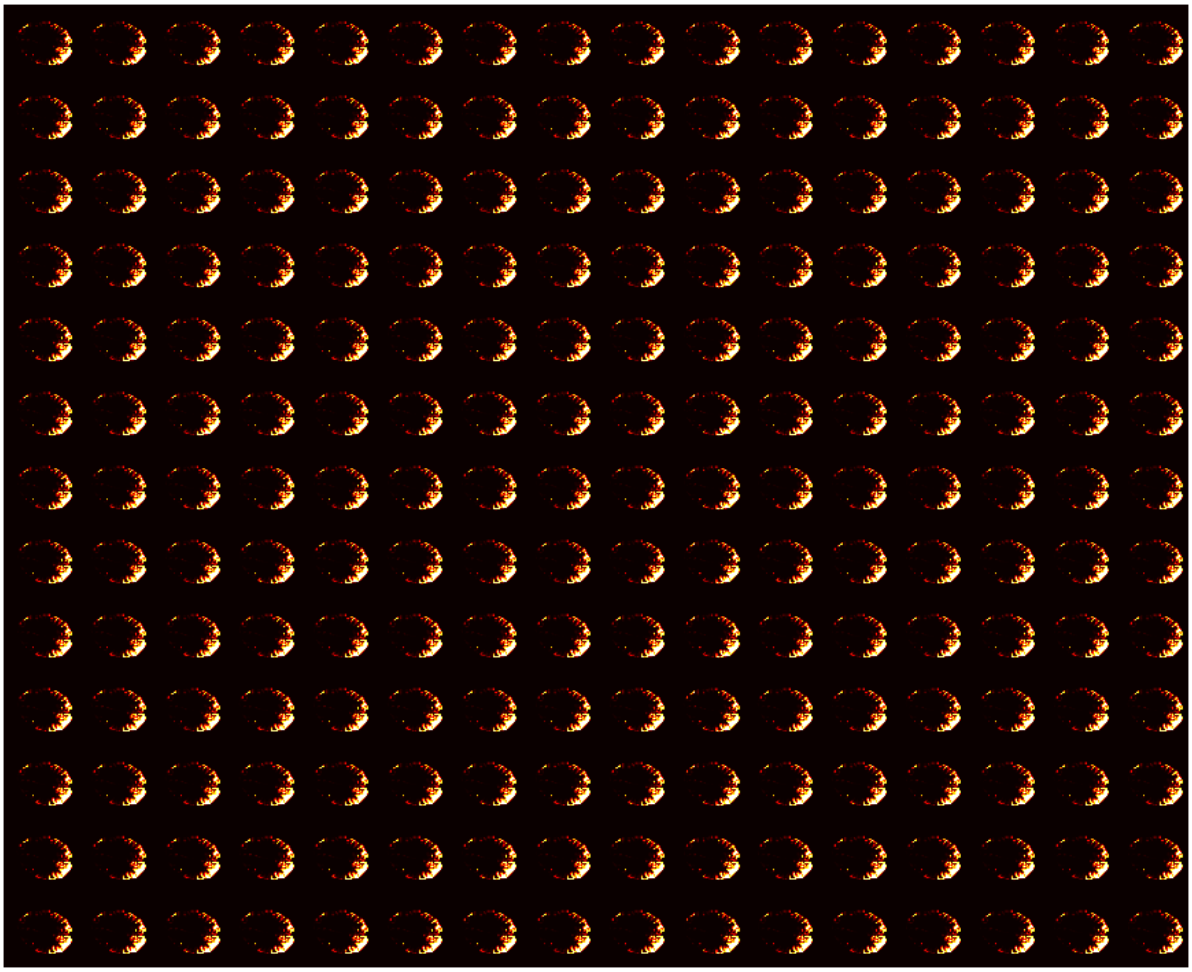

Figura 6.6. Exemplo de montagem de tensor: 208 frames para *task switch, slice* 20. Da esquerda para direita, de cima para baixo, cada elemento temporal utilizado na composição do volume de dados de entrada. Caso com rotação aleatória realizada para o conjunto de treinamento.

6.2 Anexo II - Métricas de treinamento e validação para *slices* selecionados por destaque em desempenho

6.2.1 Modelo base

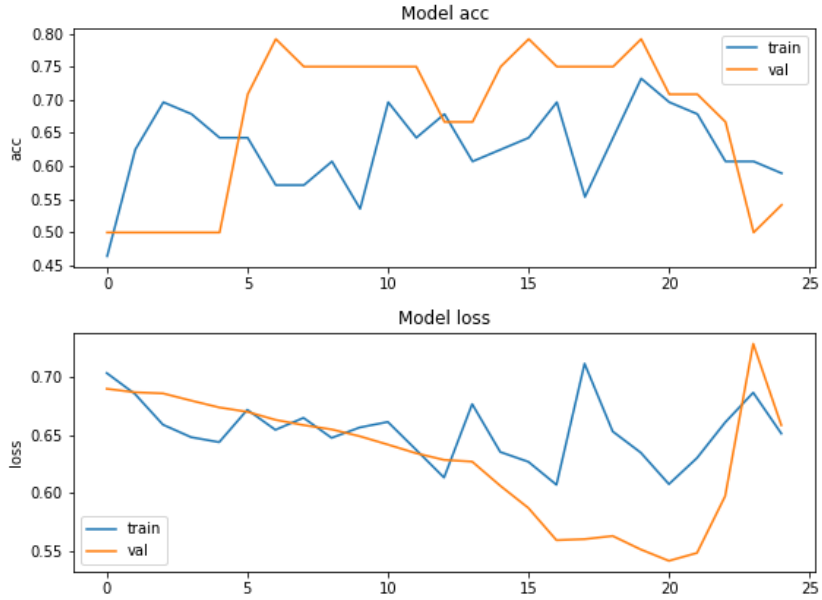

Figura 6.7. Treinamento e validação para *BART, slice* 24, modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

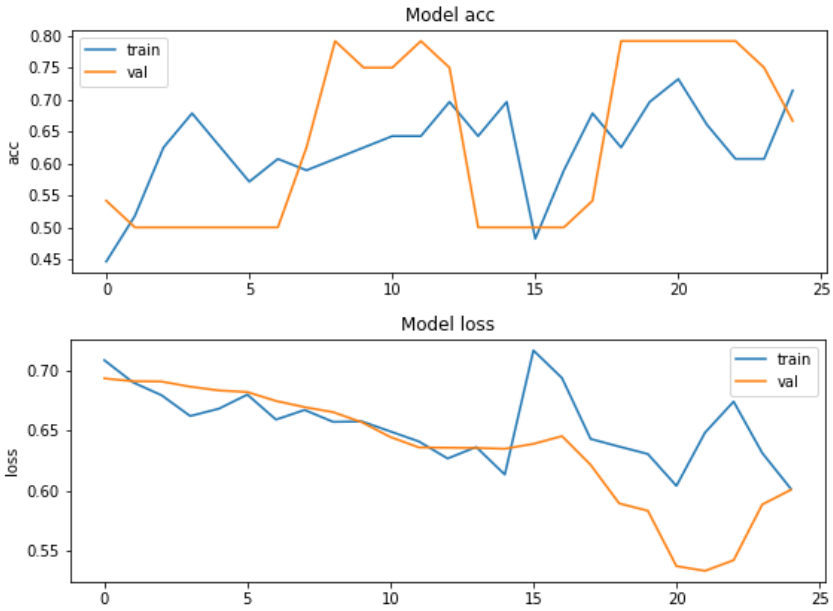

Figura 6.8. Treinamento e validação para *Rest, slice* 19, modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

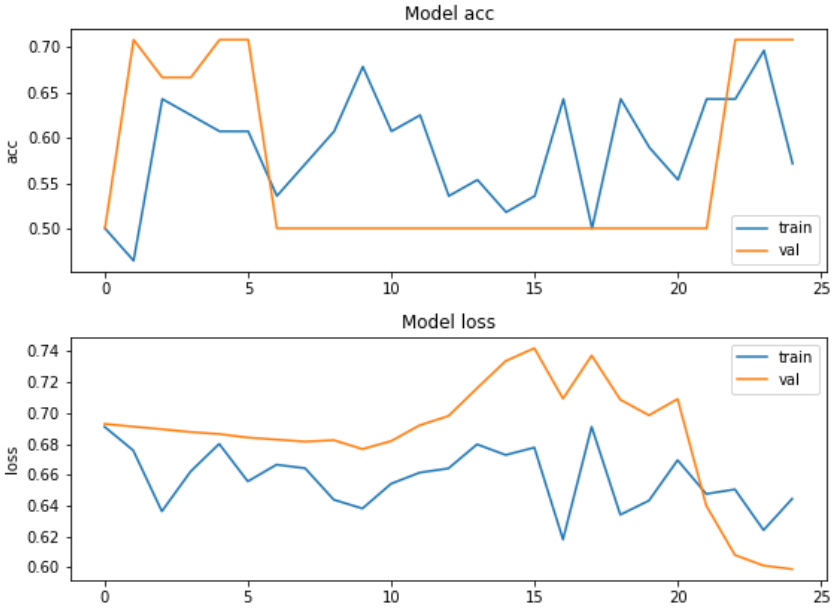

Figura 6.9. Treinamento e validação para *SCAP, slice* 15, modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

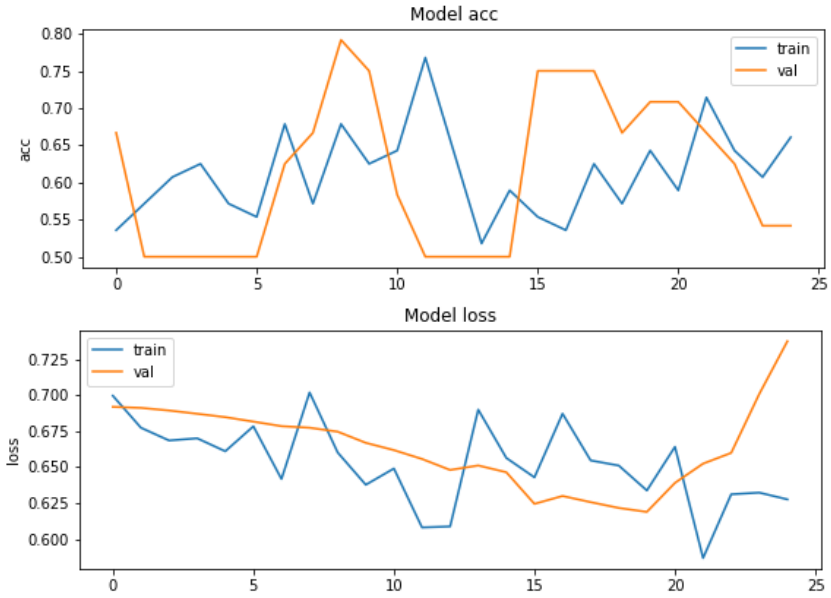

Figura 6.10. Treinamento e validação para *Stopsignal, slice* 19, modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

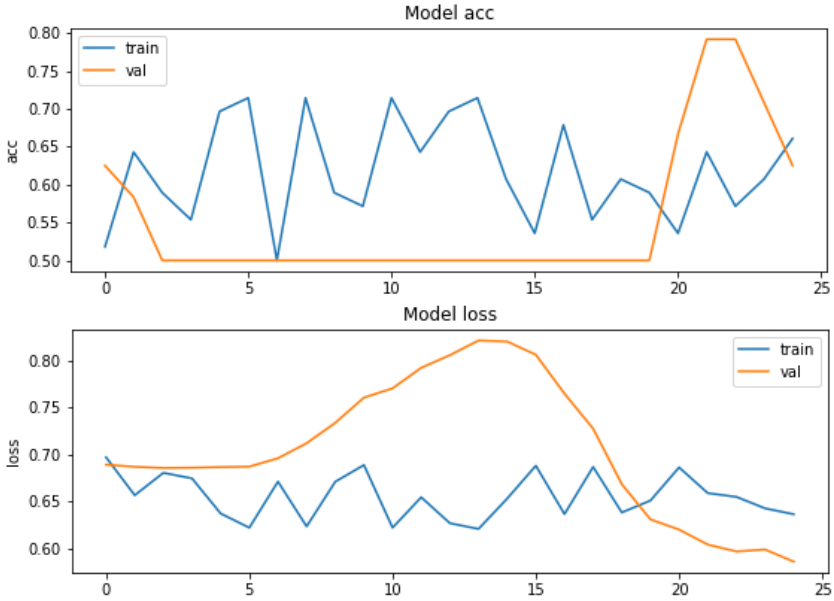

Figura 6.11. Treinamento e validação para *Task switch, slice* 20, modelo base. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

6.2.2 Modelo com data augmentation

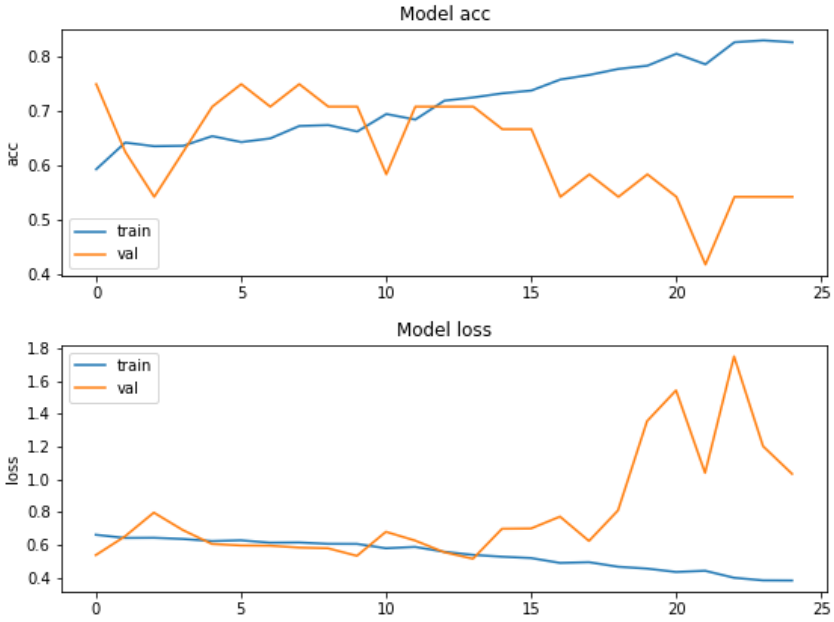

Figura 6.12. Treinamento e validação para *BART, slice* 24, modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

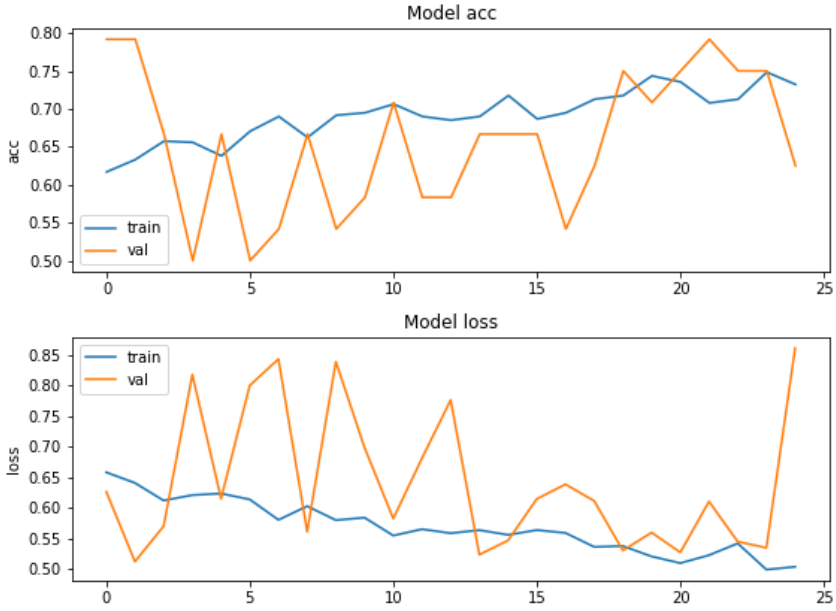

Figura 6.13. Treinamento e validação para *Rest, slice* 19, modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

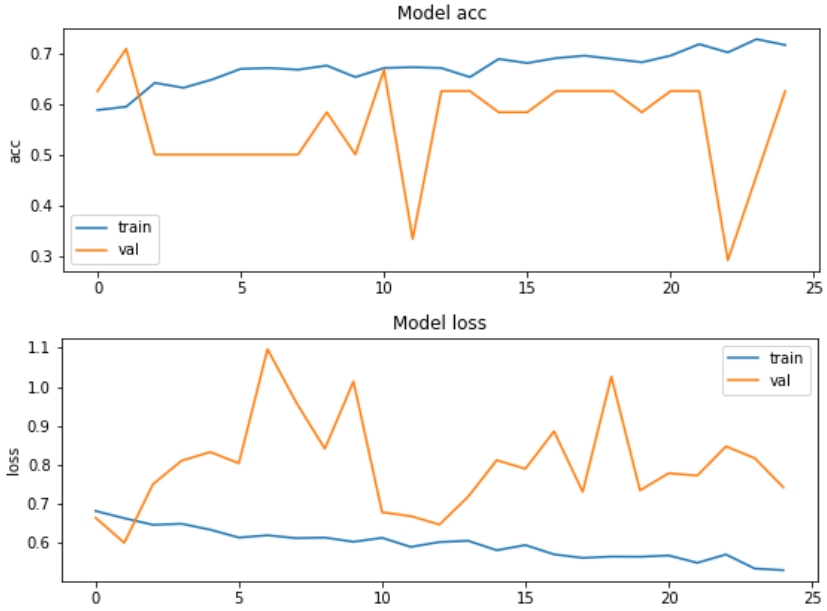

Figura 6.14. Treinamento e validação para *SCAP, slice* 15, modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

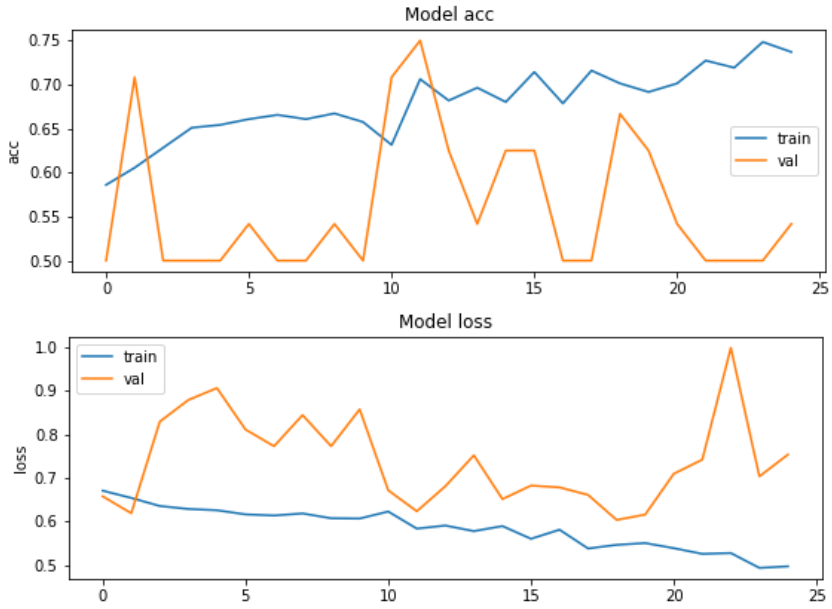

Figura 6.15. Treinamento e validação para *Stopsignal, slice* 19, modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

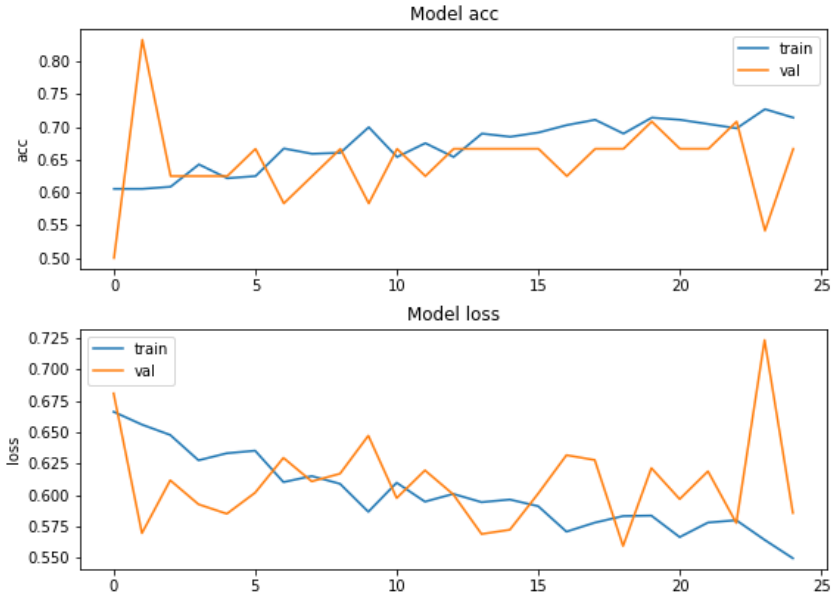

Figura 6.16. Treinamento e validação para *Task switch, slice* 20, modelo com *data augmentation*. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

6.2.3 Modelos de tuning

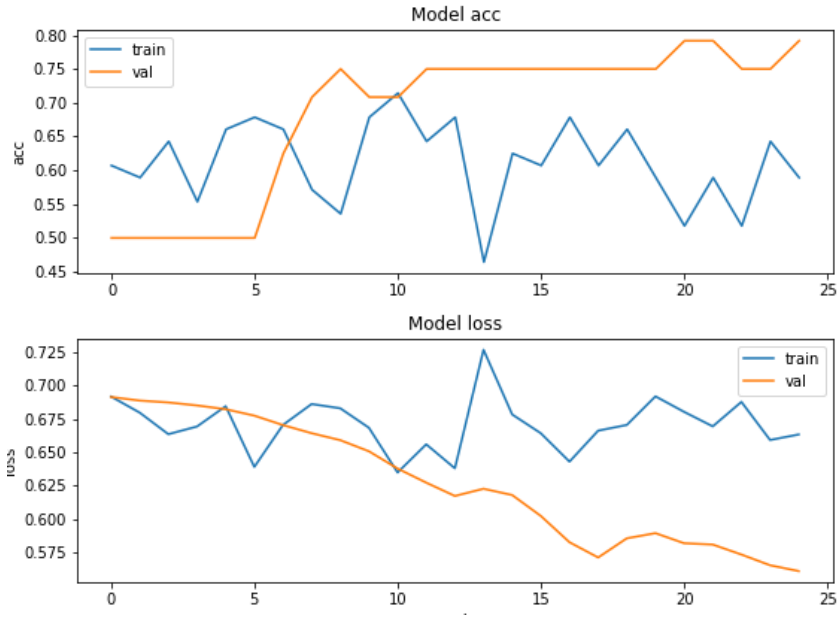

Figura 6.17. Treinamento e validação para *BART, slice* 24, 16/32 filtros, *kernel size* 3, *dropout* 0.3: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

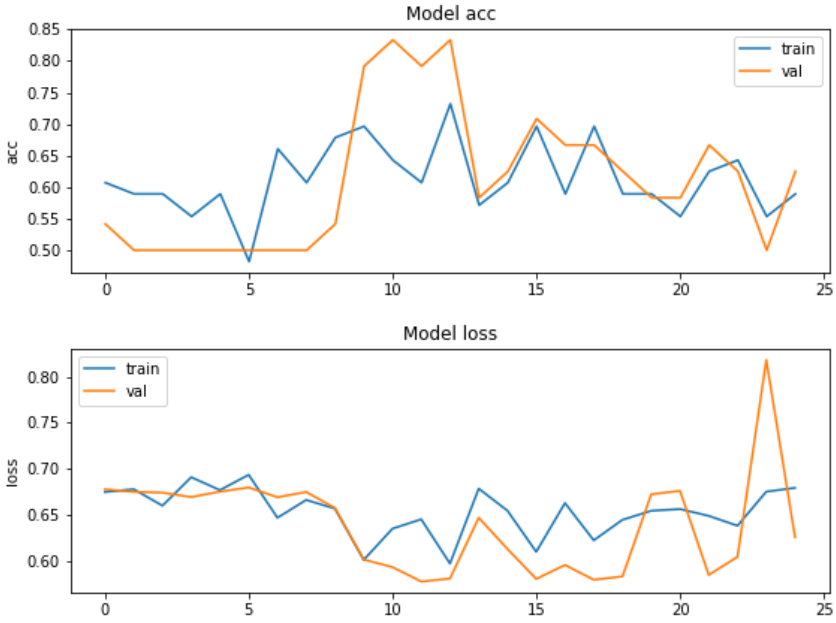

Figura 6.18. Treinamento e validação para *BART, slice* 24, 32/64 filtros, *kernel size* 5, *dropout* 0.3: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

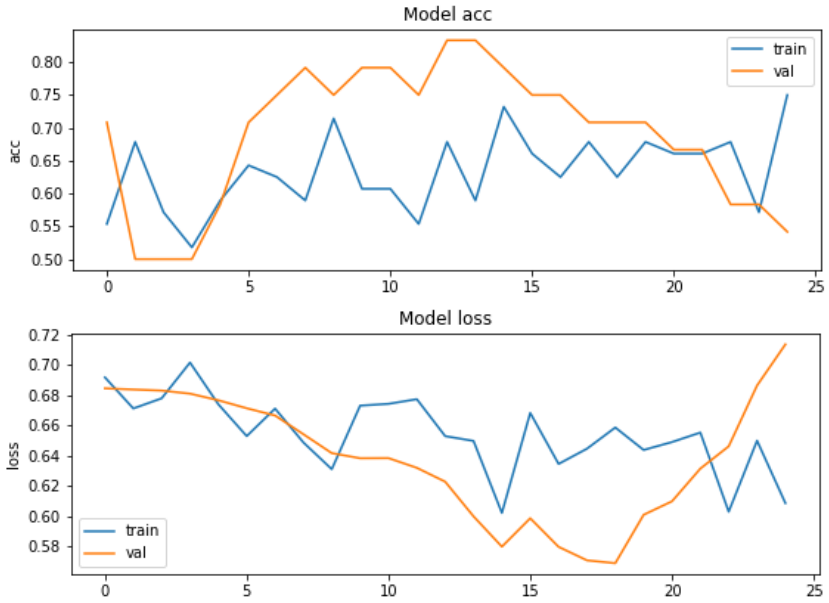

Figura 6.19. Treinamento e validação para *Rest, slice* 19, 16/32 filtros, *kernel size* 5, *dropout* 0.2: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

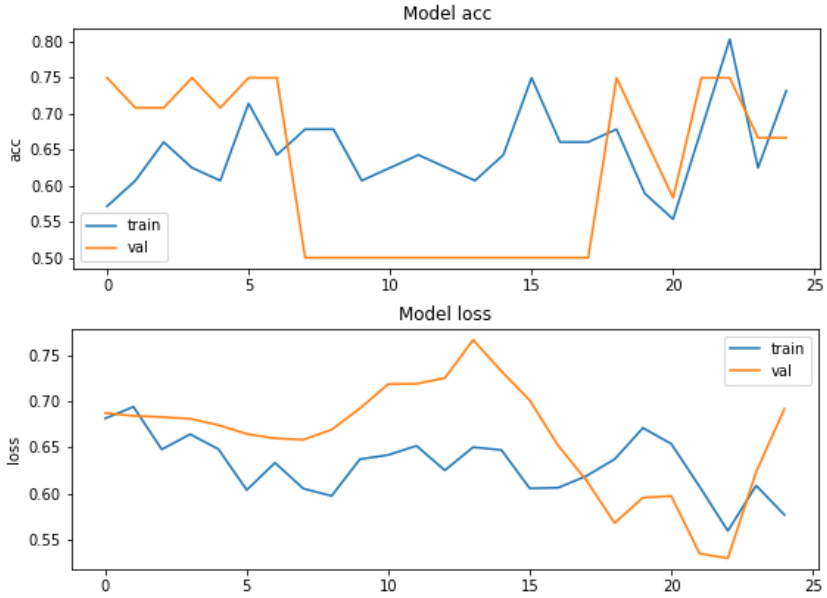

Figura 6.20. Treinamento e validação para *Rest, slice* 19, 64/128 filtros, *kernel size* 3, *dropout* 0.2: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

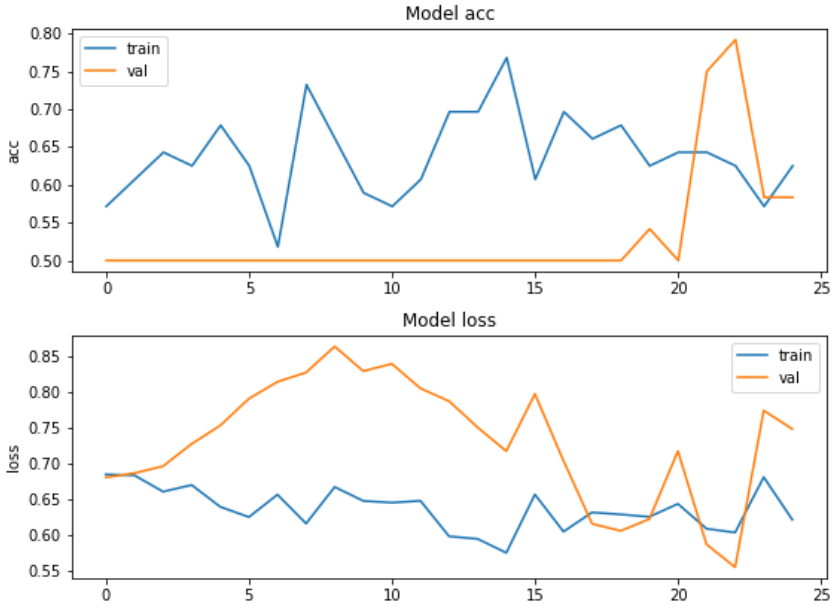

Figura 6.21. Treinamento e validação para *Rest, slice* 19, 64/128 filtros, *kernel size* 5, *dropout* 0.2: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

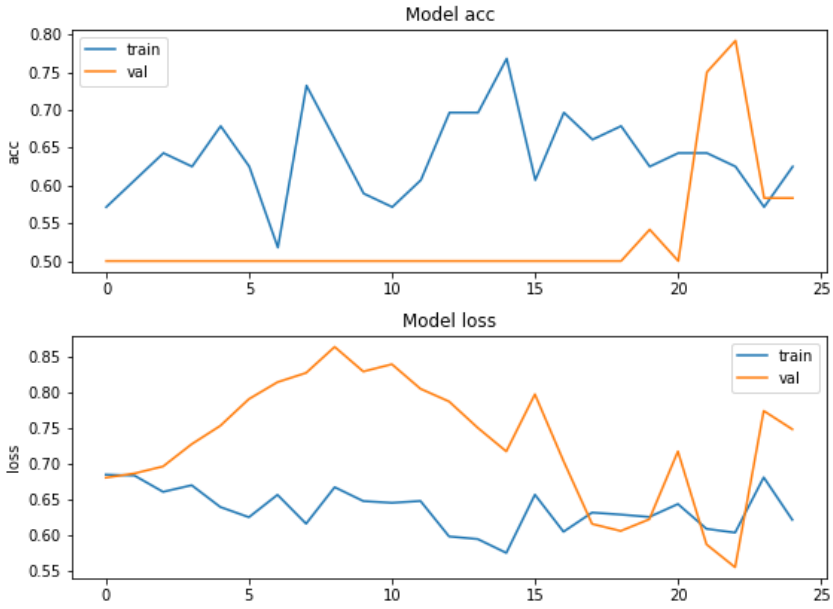

Figura 6.22. Treinamento e validação para *Rest, slice* 19, 64/128 filtros, *kernel size* 5, *dropout* 0.3: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

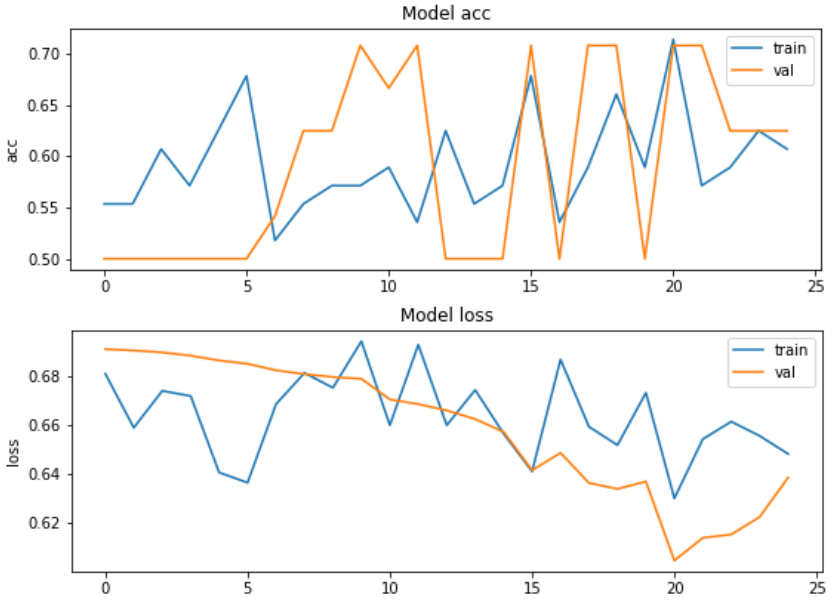

Figura 6.23. Treinamento e validação para *SCAP, slice* 19, 32/64 filtros, *kernel size* 3, *dropout* 0.2: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

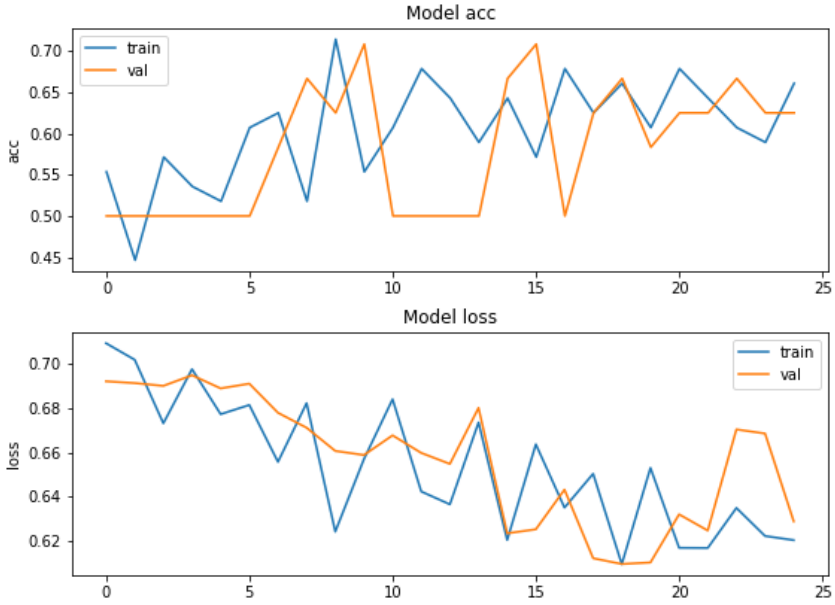

Figura 6.24. Treinamento e validação para *SCAP, slice* 19, 128/256 filtros, *kernel size* 3, *dropout* 0.2: *tuning* supera modelo base em predição. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

6.2.4 Modelos de tuning para imagens anatômicas

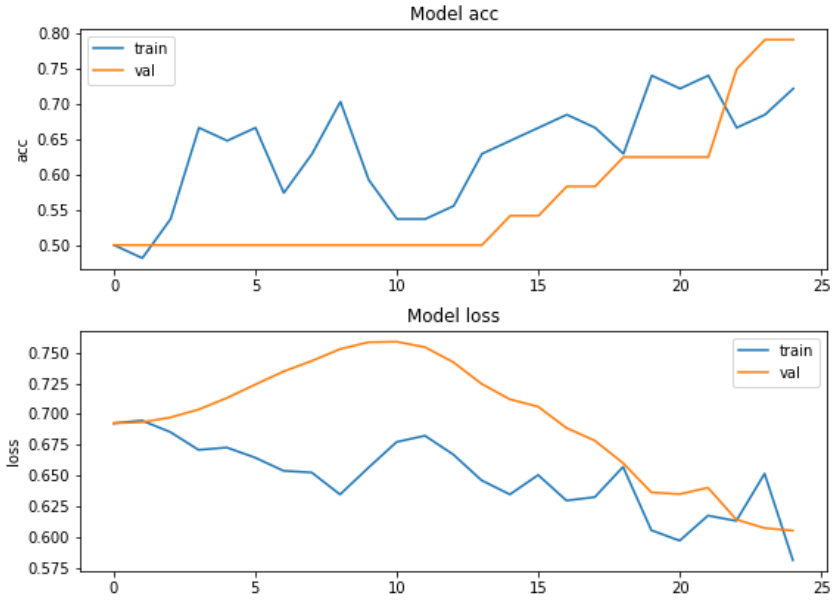

Figura 6.25. Treinamento e validação para imagem anatômica, 16/32 filtros, *kernel size* 3, *dropout* 0.2. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

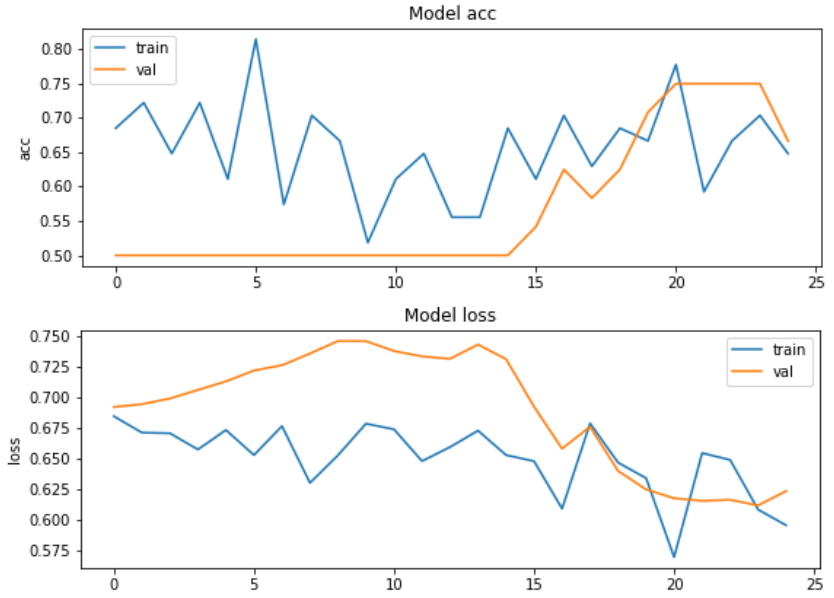

Figura 6.26. Treinamento e validação para imagem anatômica, 16/32 filtros, *kernel size* 3, *dropout* 0.3. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

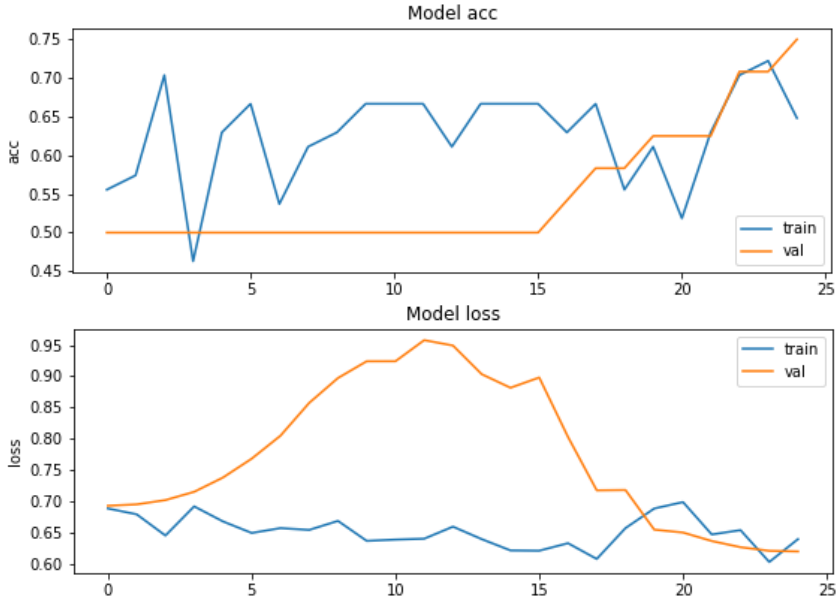

Figura 6.27. Treinamento e validação para imagem anatômica, 32/64 filtros, *kernel size* 3, *dropout* 0.3. Acima, acurácia; abaixo: *loss*. Em azul, métricas de treinamento; em amarelo, métricas de validação. O eixo vertical apresenta o valor da função em relação às *epochs* no eixo horizontal. O modelo utilizado nas predições é baseado no ponto em que houve maior valor de acurácia e menor valor de *loss* durante o processo de validação.

6.3 Anexo III - Gráficos das métricas de predição de todos os modelos avaliados

6.3.1 Modelo base

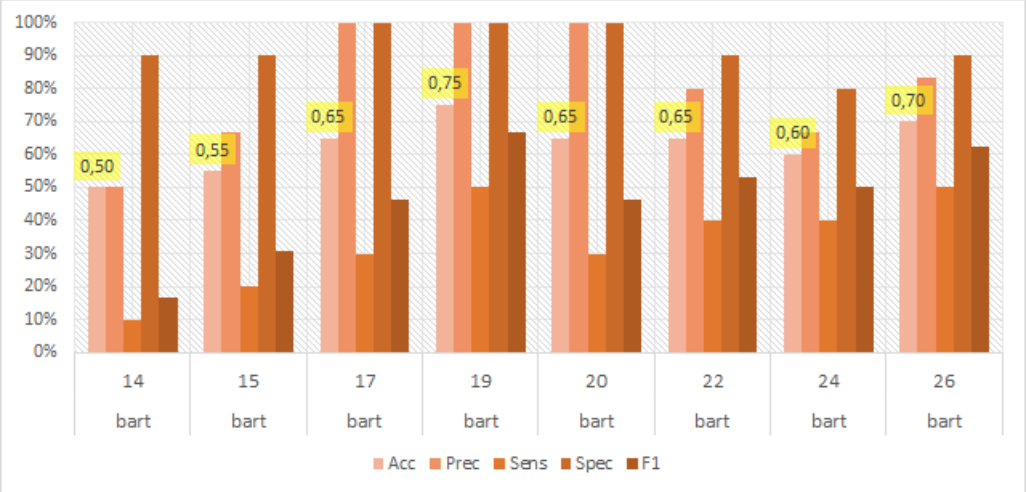

Figura 6.28. Métricas de predição para *BART* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

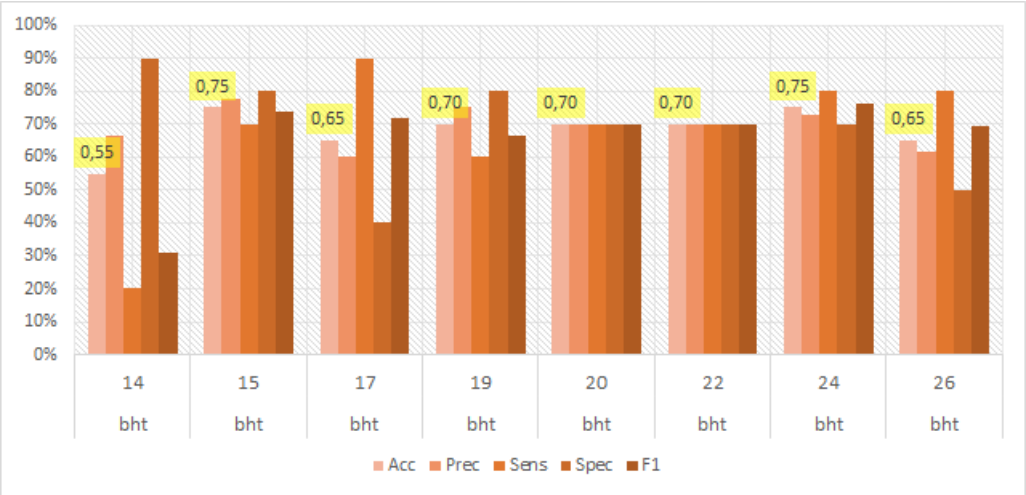

Figura 6.29. Métricas de predição para *BHT* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

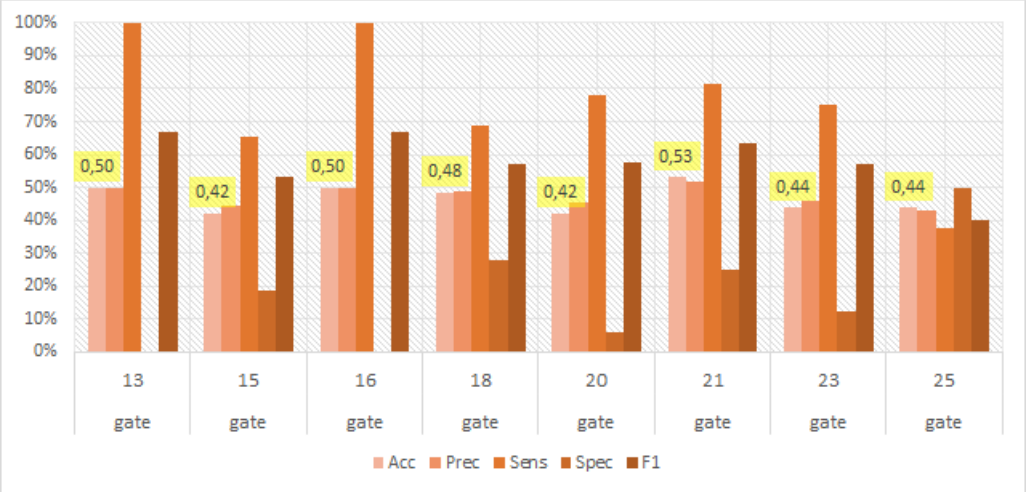

Figura 6.30. Métricas de predição para *Gate* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

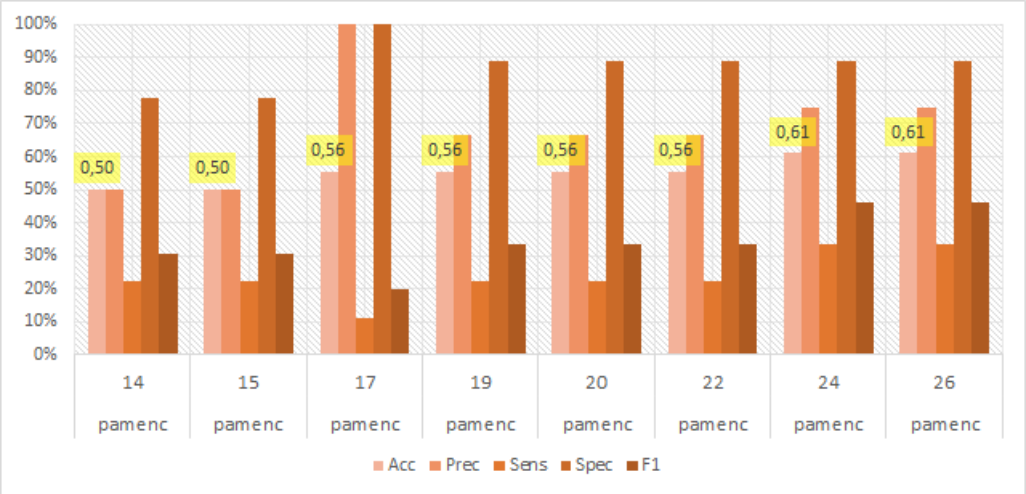

Figura 6.31. Métricas de predição para *PAM-Enc* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

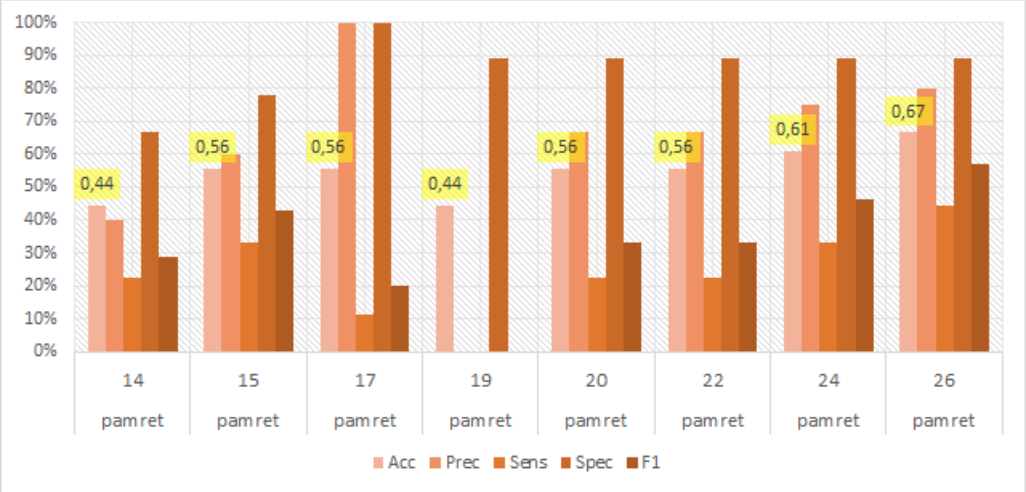

Figura 6.32. Métricas de predição para *PAM-Ret* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

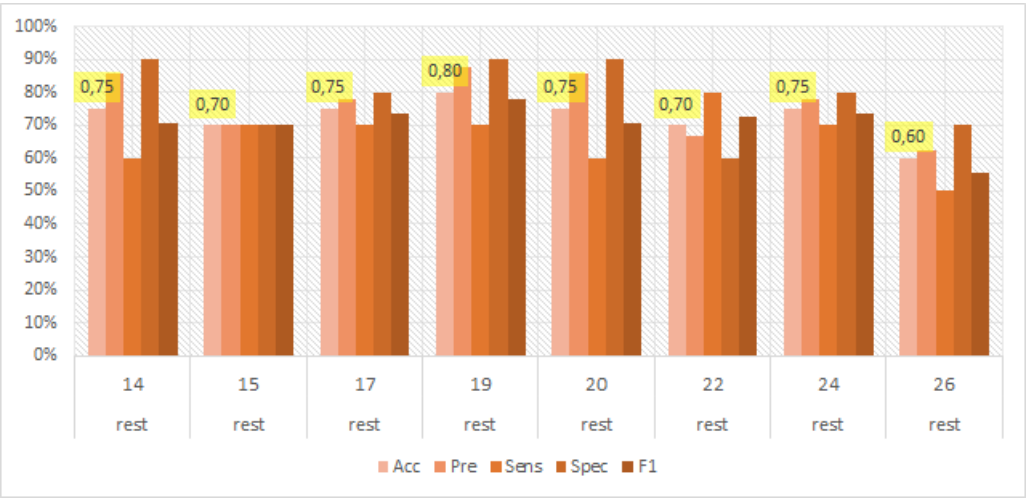

Figura 6.33. Métricas de predição para *Rest* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

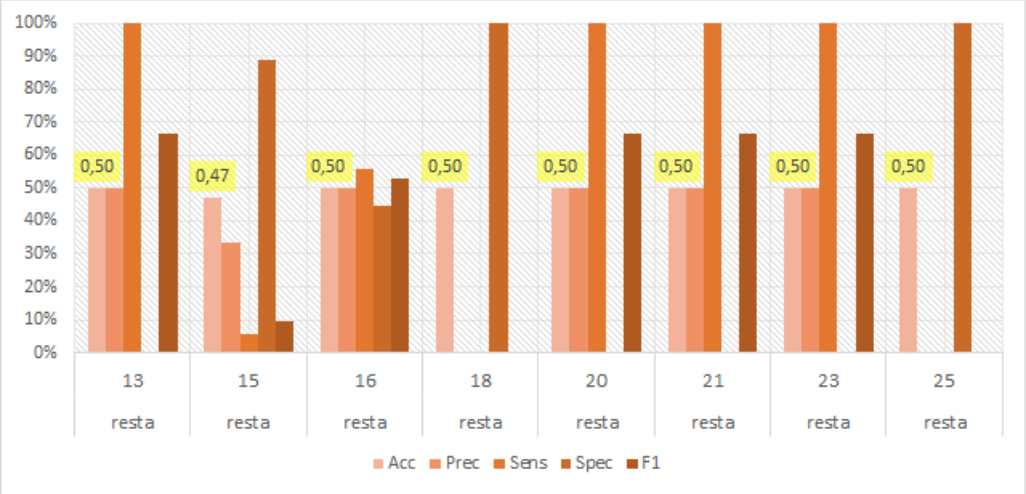

Figura 6.34. Métricas de predição para *Rest*(a) por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

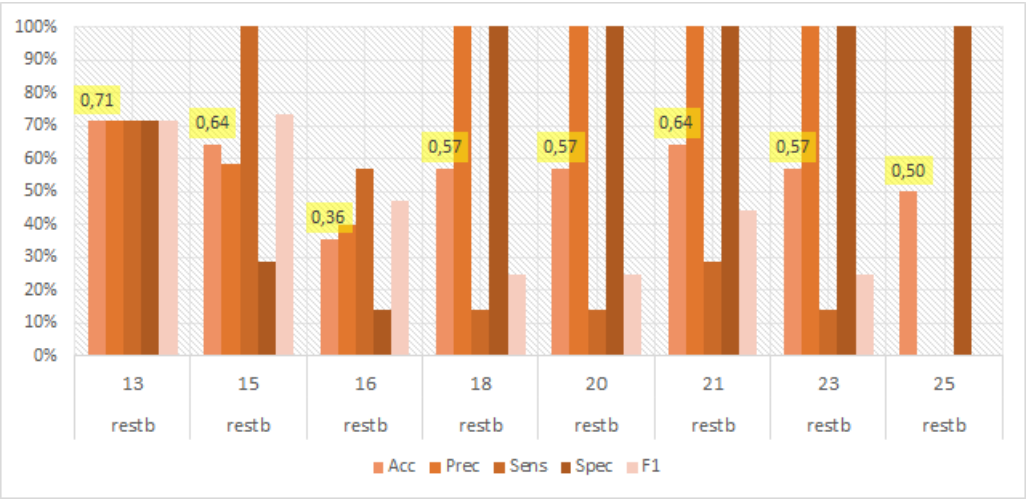

Figura 6.35. Métricas de predição para *Rest*(b) por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

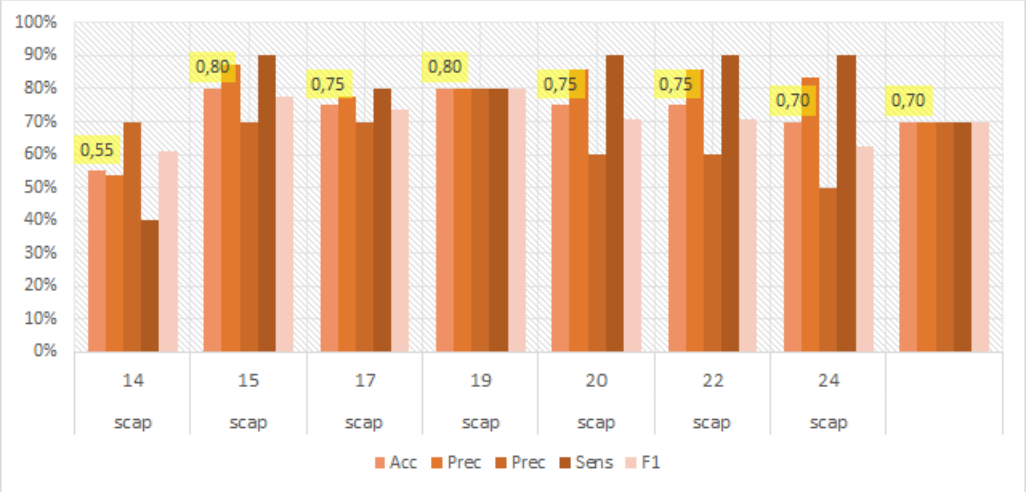

Figura 6.36. Métricas de predição para *SCAP* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

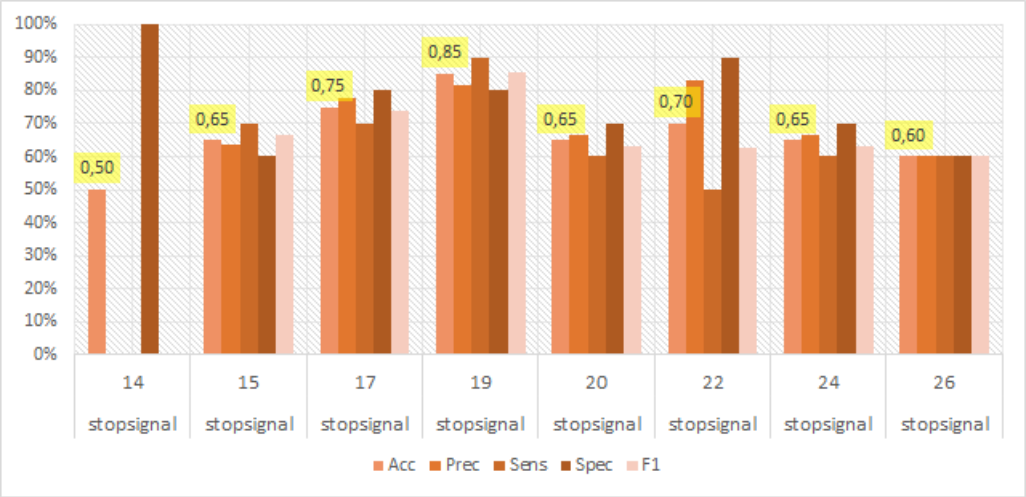

Figura 6.37. Métricas de predição para *Stopsignal* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

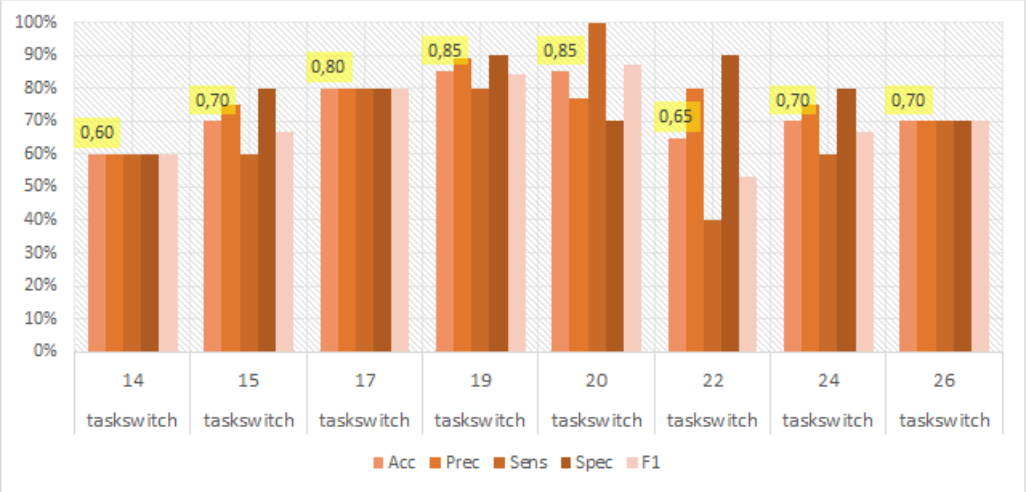

Figura 6.38. Métricas de predição para *Task switch* por *slices* no modelo base. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

6.3.2 Modelo com data augmentation

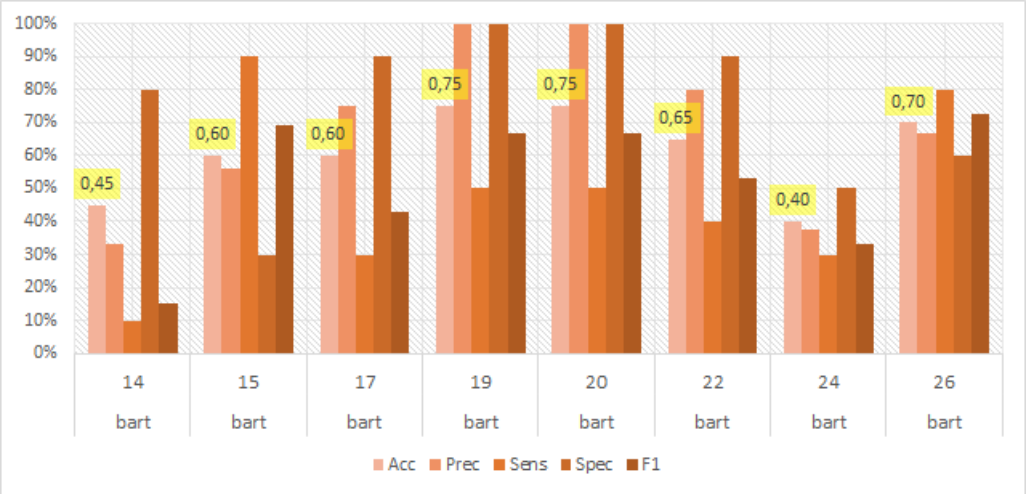

Figura 6.39. Métricas de predição para *BART* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

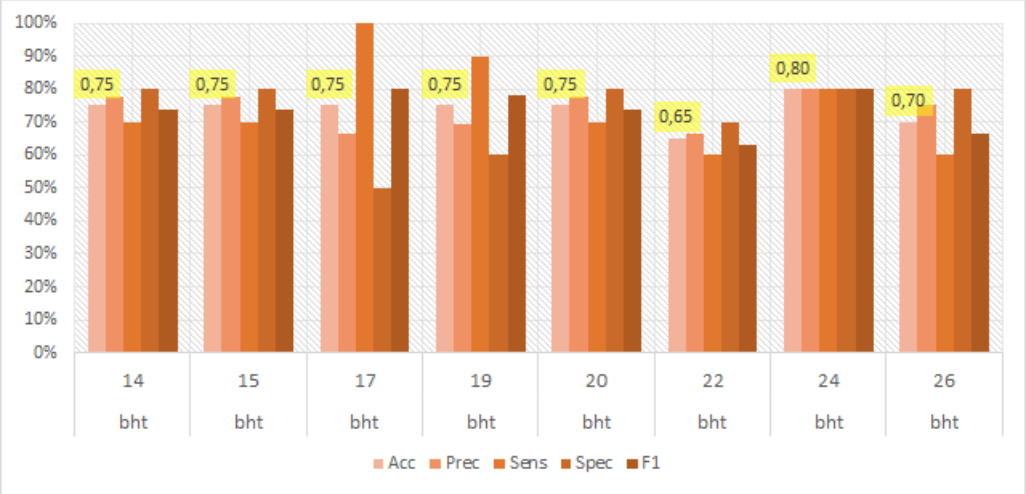

Figura 6.40. Métricas de predição para *BHT* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

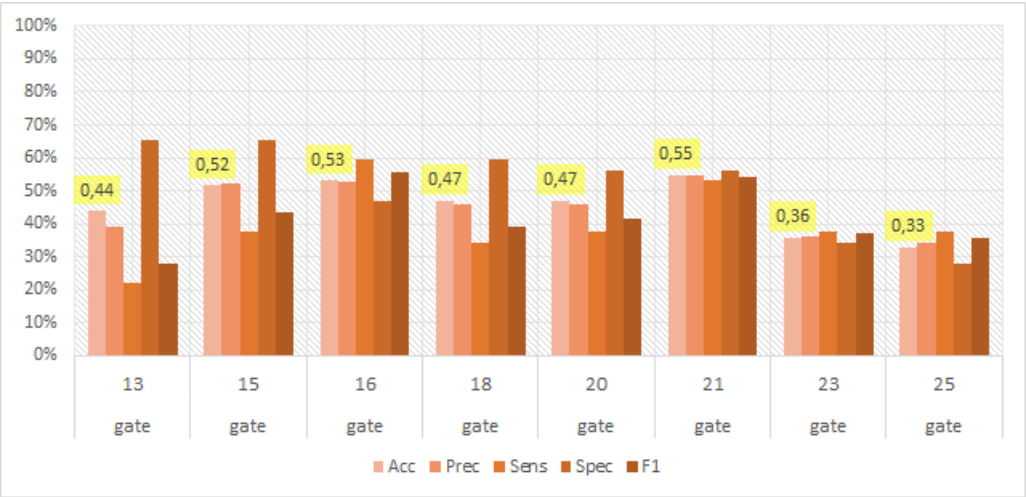

Figura 6.41. Métricas de predição para *Gate* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

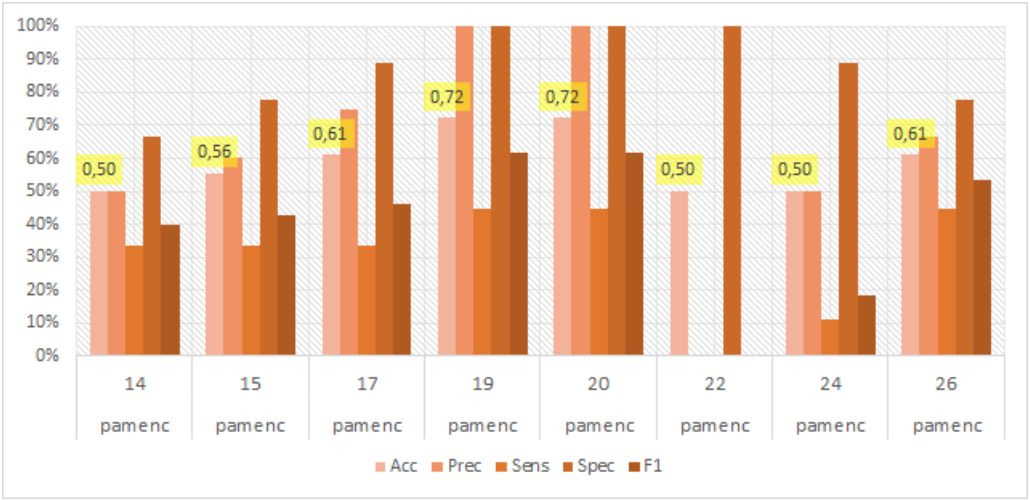

Figura 6.42. Métricas de predição para *PAM-Enc* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

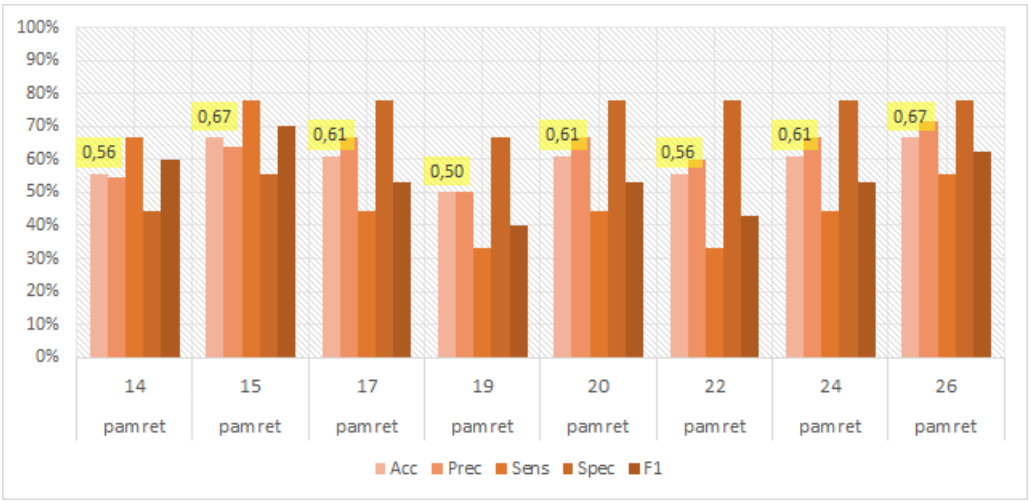

Figura 6.43. Métricas de predição para *PAM-Ret* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

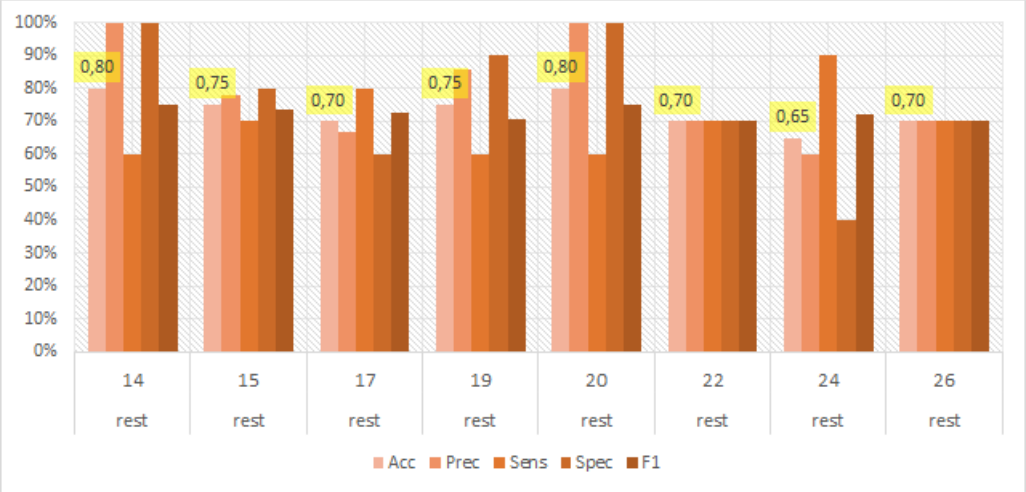

Figura 6.44. Métricas de predição para *Rest* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

Figura 6.45. Métricas de predição para *Rest*(a) por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

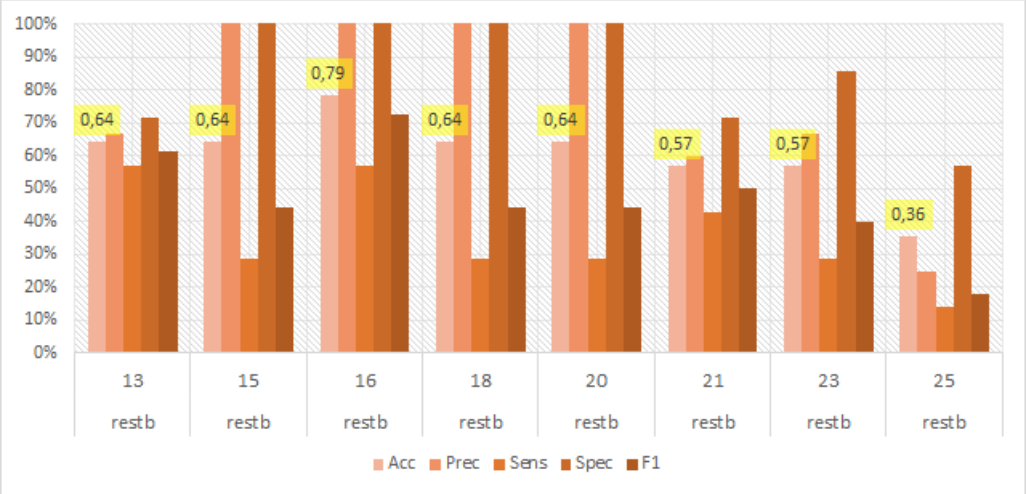

Figura 6.46. Métricas de predição para *Rest*(b) por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

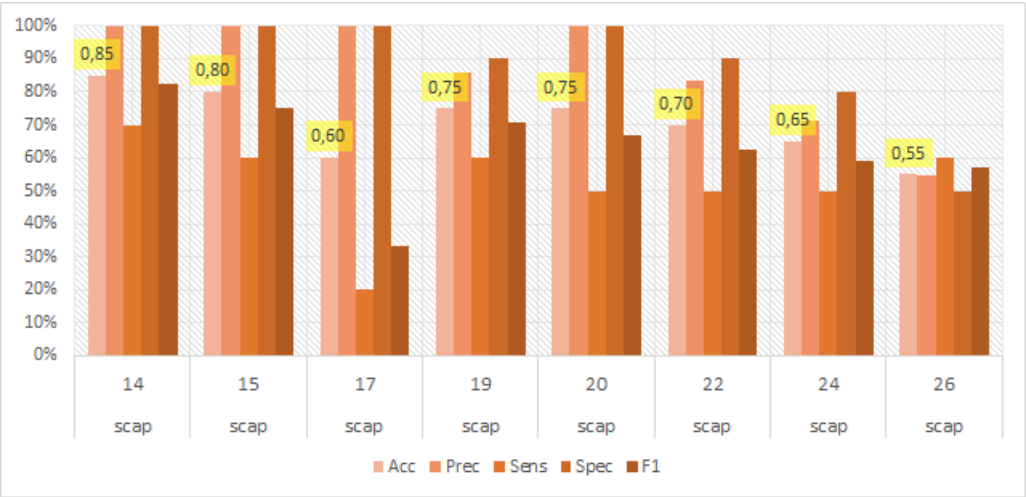

Figura 6.47. Métricas de predição para *SCAP* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

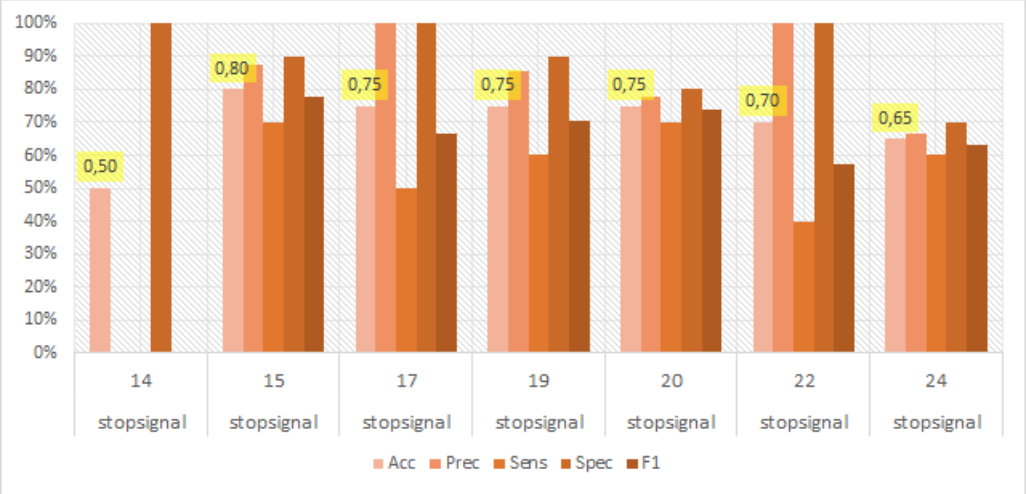

Figura 6.48. Métricas de predição para *Stopsignal* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

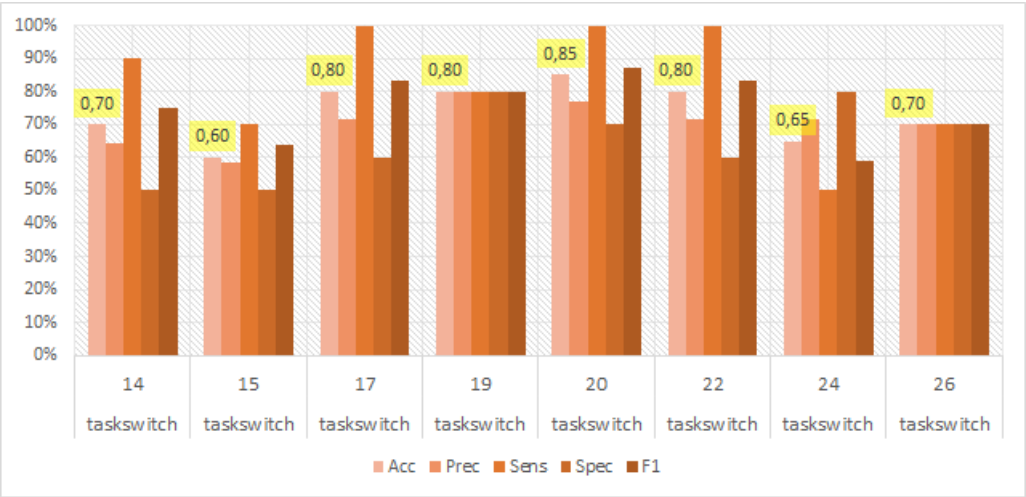

Figura 6.49. Métricas de predição para *Task switch* por *slices* no modelo com *data augmentation*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

6.3.3 Modelos de tuning

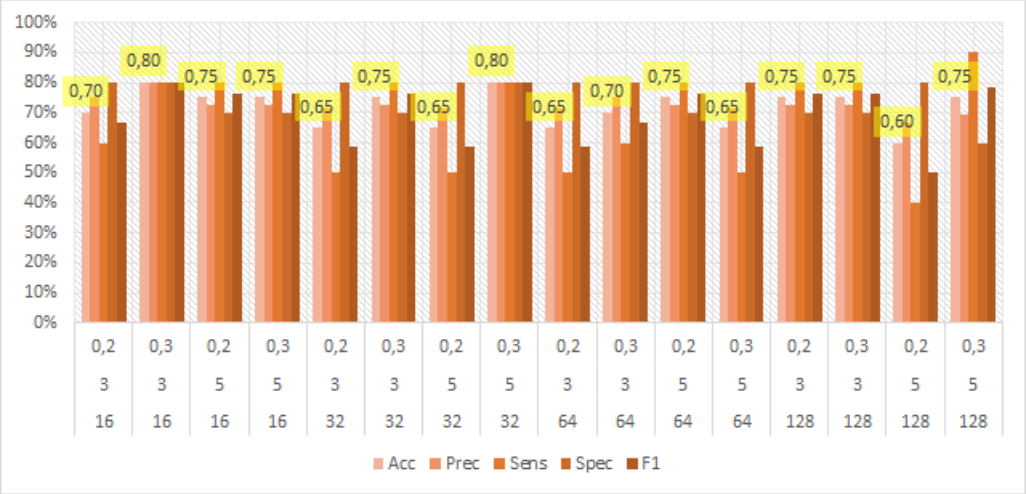

Figura 6.50. Métricas de predição para *BHT, slice* 24 nos modelos de *tunning* – *dropout, kernel size, filters*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

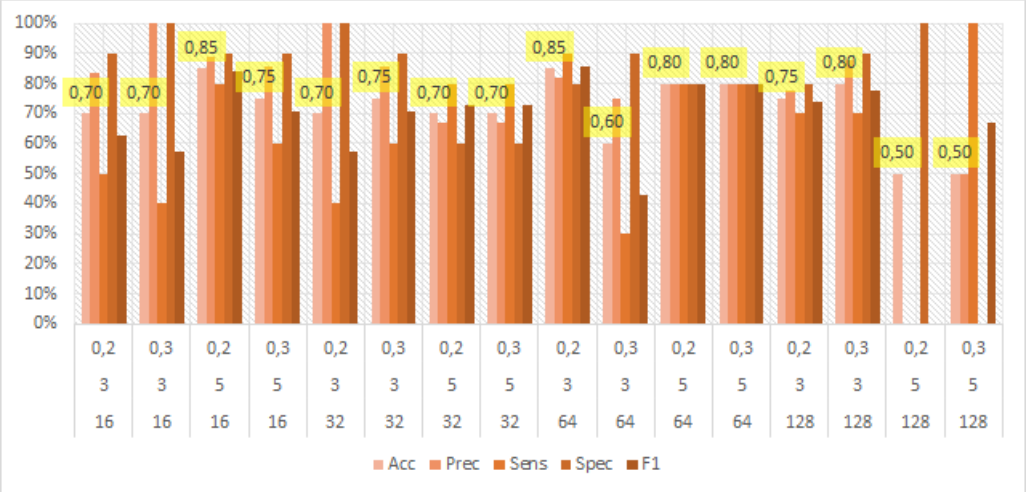

Figura 6.51. Métricas de predição para *Rest, slice* 19 nos modelos de *tunning* – *dropout, kernel size, filters*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

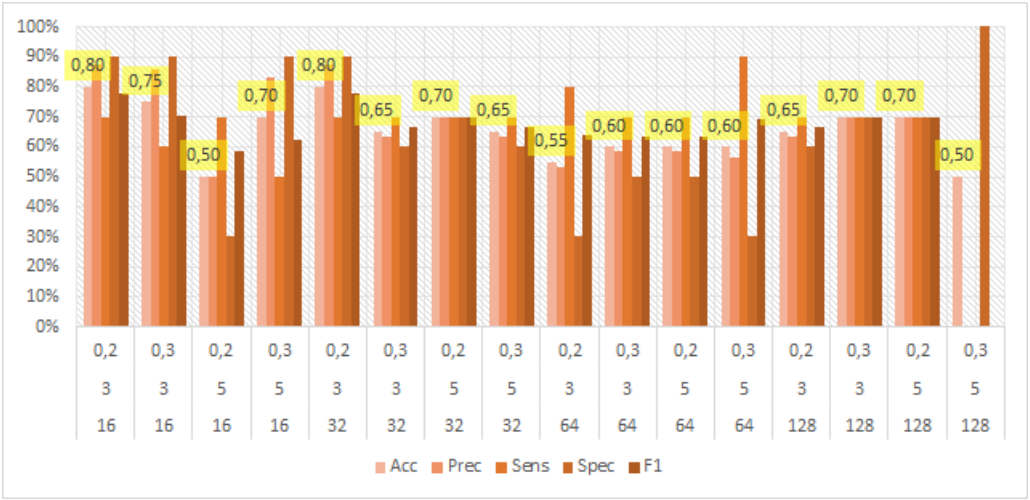

Figura 6.52. Métricas de predição para *SCAP, slice* 15 nos modelos de *tunning* – *dropout, kernel size, filters*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

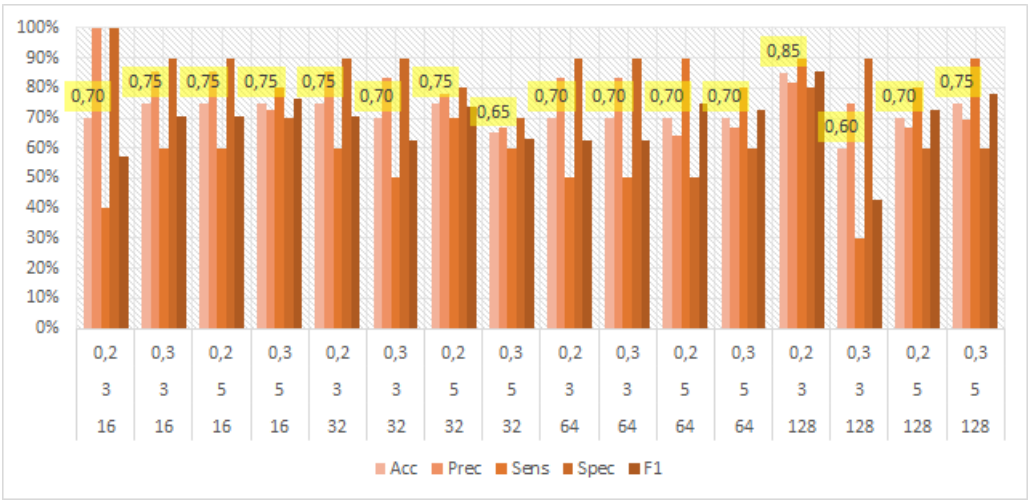

Figura 6.53. Métricas de predição para *Stopsignal, slice* 19 nos modelos de *tunning* – *dropout, kernel size, filters*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

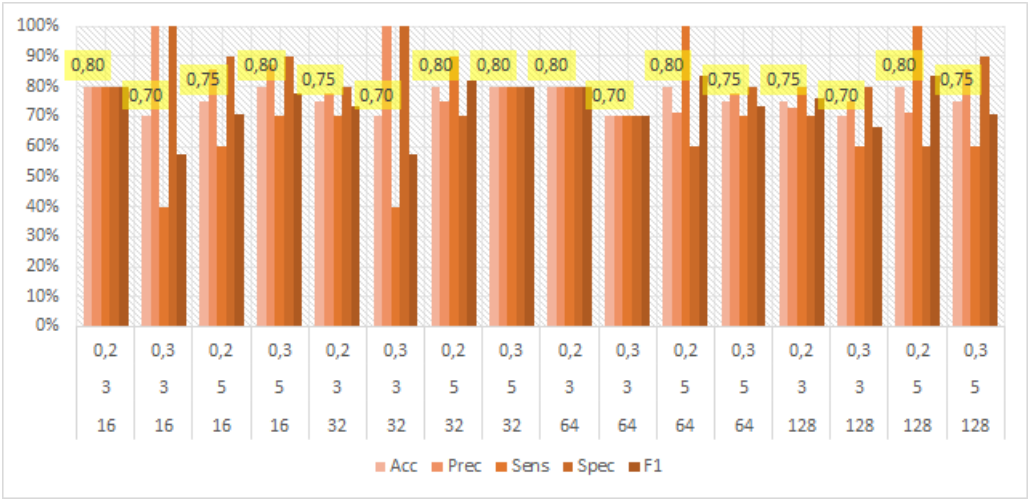

Figura 6.54. Métricas de predição para *Task switch, slice* 20 nos modelos de *tunning – dropout, kernel size, filters*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

6.3.4 Modelos com n-fold

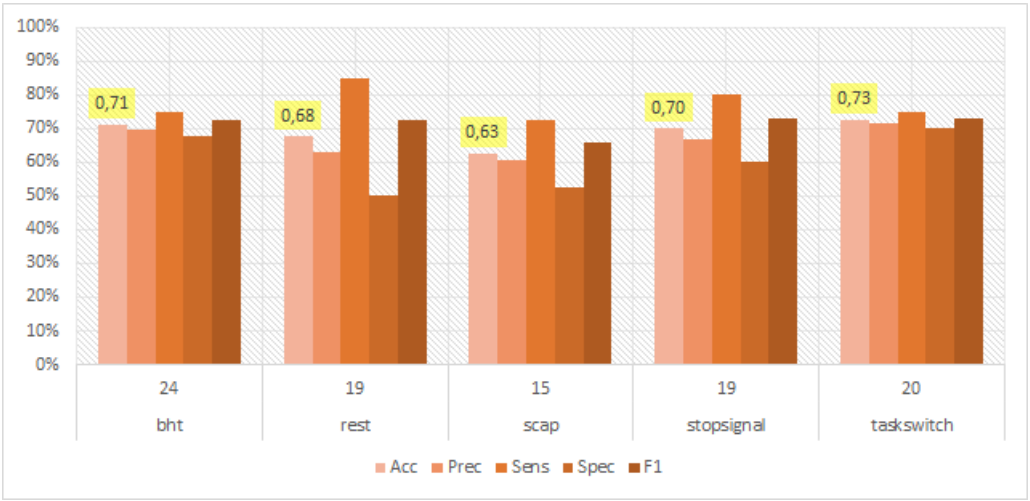

Figura 6.55. Métricas de predição para *n-fold*: BHT(*slice* 24), Rest(*slice* 19), SCAP(*slice* 15), Stopsignal(*slice* 19) e Task switch(*slice* 20). Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.

6.3.5 Modelos de tuning para imagens anatômicas

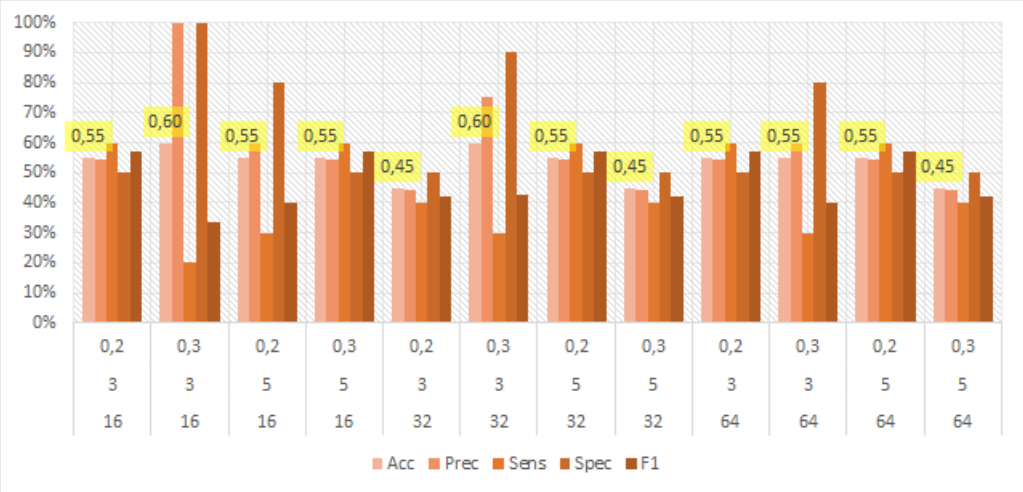

Figura 6.56. Métricas de predição para *tuning* em imagens anatômicas – *dropout, kernel size, filters*. Acurácia (Acc), precisão (Prec), sensibilidade (Sens), especificidade (Spec) e *F1 score* para cada conjunto. Destaque para acurácia – em amarelo. No eixo vertical, porcentagem obtida na métrica, no eixo horizontal destaque para *slice* e *task* associados.