



Rafael Duarte de Sousa

Identificação do Infarto do Miocárdio em Sinais Eletrocardiográficos a partir de Métodos de Análise Não Linear e Redes Neurais Convolucionais

João Pessoa – PB

Março de 2024



Rafael Duarte de Sousa

Identificação do Infarto do Miocárdio em Sinais Eletrocardiográficos a partir de Métodos de Análise Não Linear e Redes Neurais Convolucionais

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Instituto Federal da Paraíba, como requisito necessário à obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento de Sinais.

Carlos Danilo Miranda Regis, Prof. Dr.

Orientador

Vinícius Jefferson Dias Vieira, Prof. Dr.

Coorientador

João Pessoa - PB, Março de 2024

© Rafael Duarte de Sousa – rafaelds57@gmail.com

Dados Internacionais de Catalogação na Publicação – CIP
Biblioteca Nilo Peçanha – IFPB, *campus* João Pessoa

S725i	Sousa, Rafael Duarte de. Identificação do infarto do miocárdio em sinais eletrocardiográficos a partir de métodos de análise não linear e redes neurais convolucionais / Rafael Duarte de Sousa. – 2024. 202 f. : il. Dissertação (Mestrado em Engenharia Elétrica) – Instituto Federal de Educação da Paraíba – IFPB / Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), 2024. Orientação : Prof. Dr. Carlos Danilo Miranda Regis. Coorientação : Prof. Dr. Vinicius Jefferson Dias Vieira. 1. Infarto do miocárdio. 2. Vetocardiograma (VCG). 3. Gráfico de recorrência. 4. Redes neurais convolucionais. I. Título . CDU 616.12-073.7:004.032.26 (043)
-------	---



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU*

MESTRADO EM ENGENHARIA ELÉTRICA

RAFAEL DUARTE DE SOUSA

**IDENTIFICAÇÃO DO INFARTO DO MIOCARDIO EM SINAIS ELETROCARDIOGRÁFICOS A PARTIR DE
MÉTODOS DE ANÁLISE NÃO LINEAR E REDES NEURAIS CONVOLUCIONAIS**

Dissertação apresentada como requisito para obtenção do título de Mestre em Engenharia Elétrica pelo Programa de Pós-Graduação em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB - Campus João Pessoa.

Aprovado em 12 de março de 2024.

Membros da Banca Examinadora:

Dr. Carlos Danilo Miranda Regis

IFPB – PPGEE

Dr. Vinicius Jefferson Dias Vieira

Dra. Silvana Luciene do Nascimento Cunha Costa

IFPB – PPGEE

Dr. Leonardo Vidal Batista

UFPB

João Pessoa/2024

Documento assinado eletronicamente por:

- **Carlos Danilo Miranda Regis**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/03/2024 14:34:44.
- **Vinicius Jefferson Dias Vieira**, PROFESSOR DE ENSINO SUPERIOR NA ÁREA DE ORIENTAÇÃO EDUCACIONAL, em 12/03/2024 20:26:58.
- **Silvana Luciene do Nascimento Cunha Costa**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 13/03/2024 11:13:46.
- **Leonardo Vidal Batista**, PROFESSOR DE ENSINO SUPERIOR NA ÁREA DE ORIENTAÇÃO EDUCACIONAL, em 13/03/2024 16:22:39.

Este documento foi emitido pelo SUAP em 08/03/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 544012
Verificador: e24401ffad
Código de Autenticação:



Av. Primeiro de Maio, 720, Jaguaribe, JOAO PESSOA / PB, CEP 58015-435

<http://ifpb.edu.br> - (83) 3612-1200

AGRADECIMENTOS

- ▣ Em primeiro lugar, gostaria de agradecer a Deus por ter me dado todas as condições que me levaram à apresentação deste trabalho;
- ▣ Agradeço também à minha família, por sempre ter me incentivado a focar nos estudos, na pesquisa, por ter me ensinado que uma das maiores formas de se crescer como pessoa são os estudos e por sempre ter me dado oportunidades de continuar crescendo;
- ▣ Agradeço à Júlia, minha namorada, pela paciência comigo durante esse período final em que estive bem ausente e por sempre entender essas ausências e me motivado;
- ▣ Também tenho que agradecer a todos os meus colegas do GPDS, destaco aqueles que se tornaram meus amigos: José Raimundo Barbosa, Ittalo Santos e Walisson Silva. Obrigado pelos conselhos relacionados tanto com pesquisas como com relação à carreira. Agradeço em especial a José, que sempre me ensinou sobre diversos tópicos e tecnologias relacionadas à computação, participou comigo em vários trabalhos e ajudou a abrir muitas portas para mim;
- ▣ A meus orientadores Danilo Regis e Vinícius Jefferson. Ao primeiro por ter me guiado em toda a jornada de pesquisa durante o ensino superior, pelo suporte dado durante essa pesquisa com equipamentos e conhecimento, e pela paciência com os alunos durante as disciplinas; ao segundo por me apresentar aos gráficos de recorrência e pela incríveis aulas e proveitosas discussões que tivemos sobre espaço de fase e gráficos de recorrência. Agradeço muito aos dois também pelas correções e ponderações durante todo o trabalho e pela paciência nessa etapa final;
- ▣ Ao professor Washington Costa, membro da Banca durante a qualificação, pelas excelentes aulas durante a minha graduação, por compartilhar conhecimentos e sugestões para a melhoria deste trabalho.
- ▣ À professora Silvana Cunha, membro da Banca, por aceitar avaliar este trabalho e também pelas maravilhosas aulas de Circuitos Elétricos I e aulas práticas de Princípios de Comunicações ministradas por ela durante a minha graduação.

-
- ▣ Ao professor Leonardo Vidal, membro da Banca, por aceitar fazer parte desta banca e pelas correções, sugestões e conhecimentos compartilhados direta e indiretamente que ajudaram na melhoria deste trabalho;
 - ▣ Aos colegas de mestrado Maísa Lauriane, Leonardo Marçal, Smith Lima, Gabriel Gutierrez e Felipe Ribeiro por estarem comigo durante as disciplinas e pelos trabalhos em grupo que fizemos, onde sempre foi muito tranquilo e agradável trabalhar com vocês. Destaco aqui o agradecimento a Felipe por ajudar na correção e oferecer sugestões para a melhoria do texto deste trabalho;
 - ▣ Agradeço a todos os amigos e colegas que fiz no IFPB durante todo esse tempo. Obrigado por todas as conversas na cantina, nas lanchonetes, *online*, na praia e também por todas as horas que passamos juntos estudando para provas ou fazendo trabalhos. Deixo um obrigado especial para Kerven, Danilo Domingos, o "pessoal do Discord", o "foro de Jaguaribe" e "Esses Daí";
 - ▣ Por fim, devo agradecer também aos professores que tive durante toda a minha vida. Da educação infantil e do ensino fundamental merecem destaque as professoras Sheila Serápio e Luzinete Nóbrega por terem me ensinado a ler, o professor Josean Freire e a professora Isabela Melo por estimularem meu interesse pela matemática e leitura. Agradeço também aos excelentes professores que tive no IFPB, além dos que já foram citados nestes agradecimentos. Dentre eles destaco: Aniceto Duarte, Ilton Barbacena, Marcílio Onofre, Márcio Diniz, Adolfo Wagner, Alfredo Gomes, Emmanuel Dupouy, Jefferson Costa, Laurivan Diniz, Leonardo Moraes, Leônidas Francisco, Lincoln Machado, Paulo Henrique da Fonseca, Rossana Moreno e Suzete Correia. Obrigado pela dedicação em ensinar as turmas que fiz parte, digo com toda a certeza que ela foi essencial para despertar o interesse dos alunos pelas suas disciplinas.

RESUMO

Em 2019, o infarto do miocárdio (IM) com acidentes vasculares cerebrais representaram cerca de 85% das mortes por doenças cardiovasculares, a maior causa de mortes globalmente nesse ano. Dentre os sinais obtidos por meio de exames para a classificação do IM está o eletrocardiograma (ECG), sinal elétrico que registra a atividade elétrica cardíaca, onde sua modalidade mais comum possui 12 derivações. Além dele, o sinal de vetocardiografia, o vetocardiograma (VCG), que representa a atividade elétrica do coração em três dimensões por meio de três derivações ortogonais, também pode ser utilizado para a identificação de infartos, sendo possível reconstruí-lo a partir do ECG de 12 derivações. Para cada derivação desses sinais é possível extrair características não lineares, as quais podem detectar padrões que não estão evidenciados no domínio do tempo e que são sensíveis a alterações nos sinais. Uma forma de fazer essa extração é por meio da reconstrução do espaço de fase (EF) desse sinal. Apesar dos EF poderem contar com um alto número de dimensões, sua visualização é limitada pela visão humana. Por isso, os gráficos de recorrência (RP, *Recurrence Plot*), uma representação bidimensional independente do seu número de dimensões e baseada na quantidade de recorrências na trajetória do EF, foram propostos. Parâmetros podem ser extraídos de imagens dessa representação e de EF bidimensionais, sendo eles já utilizados para a identificação de arritmias e infartos. Apesar do EF bidimensional não ter necessariamente a dimensão ideal para representar adequadamente as características do sinal, eles possuem menor custo computacional associado em comparação com os RP, já que não é necessário determinar a dimensão de imersão ideal. Em razão desses dois tipos de representação serem bidimensionais, eles podem ser tratados como imagens e serem classificados por redes neurais convolucionais (CNN, *Convolutional Neural Network*). Com base nisso, o presente trabalho propõe a utilização de EFs bidimensionais e RPs, obtidos a partir de derivações vetocardiográficas reconstruídas com base em sinais de ECG, como entradas para CNNs com as arquiteturas MobileNetV2, ResNet-50, ResNet-101 e DenseNet201. Com isso, pretende-se avaliar qual tipo de imagem, entre EF e RP, e qual dessas arquiteturas têm uma melhor desempenho para a identificação do IM, além de analisar as diferenças de desempenho entre os dois tipos de imagem e ponderar seu custo computacional. Essa avaliação mostrou que a combinação de imagens de gráficos de recorrência com a arquitetura DenseNet201 levou ao melhor desempenho, sendo obtidas métricas de classificação para o

conjunto de teste de 0,8833 (acurácia), 0,7851 (sensibilidade), 0,9317 (especificidade), 0,8502 (precisão), 0,8164 (escore F1) e 0,9397 (área abaixo da curva ROC, AUC). Comparando as métricas obtidas para o conjunto de validação para essa combinação com as obtidas para a arquitetura DenseNet201 e EF, foi observado que a diferença entre valores das métricas foi inferior a 0,025 para todas elas, ao mesmo tempo, em que a diferença de tempo de processamento foi de, em média para cada sinal, de 6,52 ms.

Palavras-Chave: Infarto do Miocárdio; Vetocardiograma; Gráfico de Recorrência; Redes Neurais Convolucionais.

ABSTRACT

In 2019, myocardial infarction (MI) and stroke accounted for about 85% of deaths from cardiovascular diseases, which were the main reason of death globally that year. Among the signals obtained through exams to classify MI is the electrocardiogram (ECG), an electrical signal that records cardiac electrical activity, where its most common modality has 12 leads. In addition, the vectorcardiography signal, the vectorcardiogram (VCG), which represents the electrical activity of the heart in three dimensions through three orthogonal leads, can also be used to identify infarction and can be reconstructed from the 12 lead ECG. For each lead of these signals, it is possible to extract non-linear characteristics that can detect patterns that are not evident in the time domain and are sensitive to changes in the signals. One way to do this extraction is by reconstructing signal phase space (PS). Although PSs can be highly dimensional, human vision limits their visualization. Therefore, a two-dimensional representation independent of its number of dimensions and based on the number of recurrences in the PS trajectory called recurrence plots (RP) was proposed. Parameters can be extracted from images of this representation and two-dimensional PS, already used to identify arrhythmias and infarction. Although two-dimensional PS does not necessarily have the ideal dimension to adequately represent the signal characteristics, they have a lower associated computational cost when compared to RP, as it is not necessary to determine the ideal embedding dimension. Since these two types of representation are two-dimensional, they can be treated as images and classified by convolutional neural networks (CNN). Based on this, the present work proposes using two-dimensional PSs and RPs, obtained from vectorcardiographic leads reconstructed from ECG signals, as inputs for CNNs with the MobileNetV2, Resnet-50, Resnet-101, and DenseNet201 architectures. With this, we intend to evaluate which type of image, between PS and RP, and which of these architectures have better performance for identifying MI, in addition to analyzing the differences in performance between the two types of image and weighing their computational cost. This evaluation showed that combining RP images with the DenseNet201 architecture led to the best performance, obtaining classification metrics for the test set of 0.8833 (accuracy), 0.7851 (sensitivity), 0.9317 (specificity), 0.8502 (precision), 0.8164 (F1 score), and 0.9397 (area under the ROC curve, AUC). Comparing the metrics obtained for the validation set for this combination with those obtained for the DenseNet201 architecture and PS, the difference

between the metrics values was less than 0.025 for all of them, at the same time that the difference in processing time was, on average for each signal, 6.52 ms.

Key-Words: Myocardial Infarction; Vectorcardiogram; Recurrence Plot; Convolutional Neural Networks.

LISTA DE SIGLAS E ABREVIATURAS

ACC – Acurácia

AUC – *Area under the ROC curve* (Área Abaixo da curva ROC)

CNN – *Convolutional neural network* (Rede Neural Convolucional)

DAC – Doença arterial coronariana

DT – *Decision Tree* (Árvore de Decisão)

ECG – Eletrocardiograma

EF – Espaço de fase

FNN – *False Nearest Neighbors* Falsos Vizinhos mais Próximos

GPDS – Grupo de Processamento Digital de Sinais

IAM – Infarto agudo do miocárdio

IAMCSST – Infarto agudo do miocárdio com supra de ST

IAMSSST – Infarto agudo do miocárdio sem supra de ST

IFPB – Instituto Federal da Paraíba

IM – Infarto do Miocárdio

KNN – *K-Nearest Neighbors* (K-Vizinhos mais próximos)

MI – *Mutual information* (Informação mútua)

MLP – *Multi-layer perceptron* (Perceptron multicamadas)

ms – Unidade de medida milissegundo

MSE – *Mean squared error* (Erro quadrático médio)

PNN – *Probabilistic Neural Network* (Rede Neural Probabilística)

PTB – *Physikalisch-Technische Bundesanstalt* (Instituto nacional de metrologia da Alemanha)

REF – Reconstrução do espaço de fase

ReLU – *Rectified Linear Unit* (Unidade linear retificada)

RNA – Rede neural artificial

ROC – *Receiver operating characteristic* (Característica de operação do receptor)

RP – *Recurrence plot* (Gráfico de recorrência)

VCG – Vetocardiograma

s – Unidade de medida segundo

SVM – *Support Vector Machine* (Máquina de Vetores de Suporte)

LISTA DE SÍMBOLOS

% – Porcentagem

m – Dimensão de imersão de um espaço de fase

τ – Letra grega minúscula Tau, tempo de atraso utilizado para reconstrução do espaço de fase

d – Razão da quantidade de blocos visitados pelo espaço de fase pela quantidade total de blocos

d_0 – Valor de d usado como limiar para a classificação

T – Quantidade de estados num espaço de fase

N – Quantidade de pontos numa série temporal

X – Espaço de fase reconstruído

X_i – Estado i do espaço de fase

x_n – n -ésimo ponto para a série temporal de uma variável

$x(t)$ – Sinal original utilizado na reconstrução do espaço de fase bidimensional

q – Evento para o qual se quer determinar sua probabilidade de ocorrência

$P(q)$ – Probabilidade de ocorrência de um evento q

$h(q)$ – Quantidade de informação associada a um evento q

Q – Variável aleatória associada ao evento q

\in – Operador de pertinência "Pertence"

$H(Q)$ – Entropia para Q

$H(Q|r = g_k)$ – Entropia condicional de Q dado que a variável $r = g_k$

$H(Q|R)$ – Entropia condicional de Q dado R

$I(Q; R)$ – Informação que se tem de Q quando se conhece a variável R

Θ – Letra grega Teta representando a função degrau unitário

ϵ – Letra grega minúscula Épsilon representando o raio de vizinhança

$R_{i,j}^{m,\epsilon}$ – Recorrência para estados i e j , dimensão de imersão m e raio de vizinhança ϵ .

$f(x)$ – Valor de saída de determinada função

in_j – Entrada para determinado neurônio j

NI – Número de entradas

a_i – Entrada i

W_{ji} – Conexão entre entrada i com neurônio j

NH – Número de neurônios da camada oculta

b_j – Saída do neurônio j

v_{kj} – Conexão do neurônio j com o neurônio k

z_n – Saída obtida no neurônio n

c_n – Saída esperada para o neurônio n

e_n – Erro obtido para o neurônio n

N – Total de neurônios na saída

η – Letra grega minúscula Eta

∂ – Operador diferencial Del

$U(i, j)$ – Valor de entrada bidimensional para os índices i e j

K – *Kernel* bidimensional para os índices i e j

S – Resultado da convolução entre U e K

P_{NORM} – Peso para a classe "Normal"

Q_{NORM} – Quantidade de amostras para a classe "Normal"

P_{MI} – Peso para a classe "Infarto do Miocárdio"

Q_{MI} – Quantidade de amostras para a classe "Infarto do Miocárdio"

LISTA DE FIGURAS

1.1	Sinais de ECG para diferentes classes e seus gráficos de recorrência sem limiarização correspondentes.	9
2.1	Representação da Doença Arterial Coronariana.	14
2.2	Representação de um espasmo coronariano.	15
2.3	Sistema de condução dos impulsos cardíacos.	16
2.4	ECG de 12 derivações.	17
2.5	Sinal básico de eletrocardiograma e suas ondas características.	18
2.6	Fases de evolução de um Infarto Agudo do Miocárdio com Supra ST.	20
2.7	Exemplo de sinal de vetocardiograma.	21
2.8	Exemplo de atrator caótico tridimensional de Lorenz.	24
2.9	Esquerda: Espaço de fase reconstruído para um sinal de ECG de um paciente saudável. Direita: Espaço de fase reconstruído para um sinal de ECG de paciente com fibrilação ventricular.	25
2.10	Curva de informação mútua e reconstrução do espaço de fases de uma reação química oscilante com τ obtido a partir do primeiro mínimo local dessa curva.	27
2.11	Representação de dois estados, X_i e X_j , considerados recorrentes por estarem dentro do raio de vizinhança ϵ em uma trajetória de espaço de fase arbitrário.	29
2.12	Exemplo de gráfico de recorrência de uma série temporal de variabilidade de frequência cardíaca para um sinal de um recém-nascido prematuro.	30
2.13	Sinal de ECG saudável e seu Gráfico de recorrência obtido sem limiarização.	30
2.14	Ilustração de rede neural do tipo MLP com duas camadas ocultas.	31
2.15	Representação de um neurônio considerando seu viés, suas conexões de entrada, função de entrada, de ativação, sua saída e suas conexões de saída.	32

2.16	Rede neural MLP com entradas a_i , pesos da camada de entrada para a camada oculta em um vetor W , neurônios na camada oculta b_j , pesos da camada oculta para a camada de saída em um vetor V e neurônio z_1 na camada de saída. . . .	33
2.17	Hierarquização dos padrões espaciais a partir da imagem de um gato para a categorização dessa entrada.	36
2.18	Comparação da quantidade de interações entre neurônios com o uso ou não de conexões esparsa.	38
2.19	Comparação do campo receptivo de uma unidade em camada mais profunda com o de uma em camada mais rasa.	39
2.20	Uso de elemento central de um núcleo com três elementos em regiões de entrada e seu uso em RNA tradicional.	39
2.21	Representação da redução da sensibilidade às pequenas translações pelo <i>pooling</i>	40
2.22	Exemplo da presença de invariância à rotação devido ao estágio de <i>pooling</i>	41
2.23	Extração de características e redução de dimensionalidade por meio das camadas convolucionais e de <i>pooling</i> , respectivamente.	41
2.24	Representação do <i>transfer learning</i> com treinamento apenas para as camadas finais.	43
3.1	Fluxograma das atividades propostas para o desenvolvimento do presente trabalho.	46
3.2	Divisão dos sinais do PTB-XL em suas superclasses e subclasses.	47
3.3	Distribuição dos valores de atraso para cada derivação vetocardiográfica.	51
3.4	Exemplo de imagem resultante da junção dos espaços de fase de derivações vetocardiográficas nos canais do sistema RGB.	52
3.5	Exemplo de imagem resultante da junção dos gráficos recorrência de derivações vetocardiográficas nos canais do sistema RGB.	53
4.1	Fluxograma para as análises realizadas com resultados no presente capítulo.	60
4.2	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com EF em cada época na arquitetura MobileNetV2.	63
4.3	Evolução das métricas para os conjuntos de treino e validação com EF em cada época na arquitetura MobileNetV2.	64
4.4	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-50.	66

4.5	Evolução das métricas para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-50.	67
4.6	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-101.	68
4.7	Evolução das métricas para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-101.	69
4.8	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com EF em cada época na arquitetura DenseNet201.	70
4.9	Evolução das métricas para os conjuntos de treino e validação com EF em cada época na arquitetura DenseNet201.	71
4.10	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com RP em cada época na arquitetura MobileNetV2.	74
4.11	Evolução das métricas para os conjuntos de treino e validação com RP em cada época na arquitetura MobileNetV2.	75
4.12	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-50.	76
4.13	Evolução das métricas para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-50.	77
4.14	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-101.	78
4.15	Evolução das métricas para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-101.	79
4.16	Evolução da métrica <i>loss</i> para os conjuntos de treino e validação com RP em cada época na arquitetura DenseNet201.	80
4.17	Evolução das métricas para os conjuntos de treino e validação com RP em cada época na arquitetura DenseNet201.	81
4.18	Matriz de confusão obtida para os sinais do conjunto de teste.	83
4.19	Curvas para a métrica <i>loss</i> obtidas para a arquitetura VGG8 utilizando gráficos de recorrência considerando o conjunto de validação.	87
B.1	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura MobileNetV2.	111

B.2	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura MobileNetV2.	112
B.3	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura MobileNetV2.	113
B.4	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura MobileNetV2.	114
B.5	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura MobileNetV2.	115
B.6	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura MobileNetV2.	116
B.7	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura MobileNetV2.	117
B.8	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura MobileNetV2.	118
B.9	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-50.	119
B.10	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-50.	120
B.11	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-50.	121
B.12	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-50.	122
B.13	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-50.	123

B.14	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-50.	124
B.15	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura ResNet-50.	125
B.16	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura ResNet-50.	126
B.17	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-101.	127
B.18	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-101.	128
B.19	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-101.	129
B.20	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-101.	130
B.21	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-101.	131
B.22	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-101.	132
B.23	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura ResNet-101.	133
B.24	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura ResNet-101.	134
B.25	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura DenseNet201.	135

B.26	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura DenseNet201.	136
B.27	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura DenseNet201.	137
B.28	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura DenseNet201.	138
B.29	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura DenseNet201.	139
B.30	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura DenseNet201.	140
B.31	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura DenseNet201.	141
B.32	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura DenseNet201.	142
B.33	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura MobileNetV2.	143
B.34	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura MobileNetV2.	144
B.35	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura MobileNetV2.	145
B.36	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura MobileNetV2.	146
B.37	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura MobileNetV2.	147

B.38	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura MobileNetV2.	148
B.39	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura MobileNetV2.	149
B.40	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura MobileNetV2.	150
B.41	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-50.	151
B.42	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-50.	152
B.43	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura ResNet-50.	153
B.44	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura ResNet-50.	154
B.45	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-50.	155
B.46	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-50.	156
B.47	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-50.	157
B.48	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-50.	158
B.49	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-101.	159

B.50	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-101.	160
B.51	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura ResNet-101.	161
B.52	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura ResNet-101.	162
B.53	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-101.	163
B.54	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-101.	164
B.55	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-101.	165
B.56	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-101.	166
B.57	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura DenseNet201.	167
B.58	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura DenseNet201.	168
B.59	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura DenseNet201.	169
B.60	Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura DenseNet201.	170
B.61	Evolução por época da métrica <i>loss</i> para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura DenseNet201.	171

B.62 Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura DenseNet201. 172

B.63 Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura DenseNet201. 173

B.64 Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura DenseNet201. 174

LISTA DE TABELAS

1.1	Estado da arte.	11
2.1	Coefficientes para as derivações do ECG utilizadas reconstrução do VCG.	22
3.1	Relação do número de ECGs por superclasses e subclasses.	48
3.2	Quantidade de sinais a ser utilizada no trabalho para os conjuntos de treino, validação e teste.	49
3.3	Quantidade de ciclos para os conjuntos de treino, validação e teste.	50
3.4	Quantidade de sinais, compostos por um ciclo cardíaco, baseado na sua derivação vetocardiográfica e na dimensão de imersão ideal.	52
3.5	Quantidade de ciclos para os conjuntos de treino e validação, divididos para a otimização dos hiperparâmetros e treinamento do modelo, além do conjunto de teste.	54
3.6	Valores de hiperparâmetros otimizados para cada arquitetura treinada com imagens de espaço de fase bidimensional.	58
3.7	Valores de hiperparâmetros otimizados para cada arquitetura treinada com gráficos de recorrência.	59
4.1	Métricas ao utilizar a inicialização com pesos aleatórios.	62
4.2	Métricas ao utilizar <i>transfer learning</i>	62
4.3	Métricas obtidas para o conjunto de validação nos cinco modelos treinados com EF e arquitetura MobileNetV2.	63
4.4	Métricas obtidas para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura ResNet-50.	65
4.5	Métricas obtidas para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura ResNet-101.	66

4.6	Métricas obtidas para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura DenseNet201.	69
4.7	Valores médios para as 5 rodadas das métricas de acordo com cada valor de percentual utilizado como raio de vizinhança.	72
4.8	Métricas para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura MobileNet-V2.	73
4.9	Métricas para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura ResNet-50.	74
4.10	Métricas para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura ResNet-101.	76
4.11	Métricas para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura DenseNet201.	79
4.12	Métricas considerando o conjunto de teste para o modelo 1 da arquitetura DenseNet201 cujo treinamento foi feito com imagens de gráficos de recorrência.	82
4.13	Métricas e suas diferenças para cada tipo de sinal utilizado.	85
4.14	Valores de <i>Dropout</i> , regularização L2 e taxa de aprendizagem determinados na otimização de hiperparâmetros para a arquitetura VGG8 utilizando gráficos de recorrência.	86
A.1	Quantidade de ciclos para cada valor de quantidade de amostras atrasadas para a derivação X.	107
A.2	Quantidade de ciclos para cada valor de quantidade de amostras atrasadas para a derivação Y.	108
A.3	Quantidade de ciclos para cada valor de quantidade de amostras atrasadas para a derivação Z.	109

SUMÁRIO

1	Introdução	1
1.1	Motivação e Definição do Problema	1
1.2	Objetivos	4
1.2.1	Objetivo geral	4
1.2.2	Objetivos específicos	4
1.3	Revisão da literatura	5
1.4	Estrutura do Documento	12
2	Fundamentação Teórica	13
2.1	Infarto Agudo do Miocárdio	13
2.2	Sinais Elétricos Cardíacos	15
2.2.1	Eletrocardiograma	15
2.2.2	Vetocardiograma	19
2.3	Métodos de análise não linear	23
2.3.1	Reconstrução do espaço de fase	23
2.3.2	Gráficos de recorrência	28
2.4	Redes Neurais	30
3	Descrição da Proposta	44
3.1	Aplicabilidade	44
3.2	Metodologia	45
3.2.1	Banco de Dados	45
3.2.2	Processamento	49
3.2.3	Redes Neurais Convolucionais	53

4	Resultados	60
4.1	Avaliação do uso da técnica de <i>transfer learning</i> com imagens de espaço de fase	61
4.2	Treinamento com imagens de espaço de fase	62
4.2.1	MobileNetV2	62
4.2.2	ResNet-50	65
4.2.3	ResNet-101	66
4.2.4	DenseNet201	68
4.2.5	Comparação entre arquiteturas	70
4.3	Avaliação do raio de vizinhança ideal	72
4.4	Treinamento com gráficos de recorrência	73
4.4.1	MobileNetV2	73
4.4.2	ResNet-50	74
4.4.3	ResNet-101	76
4.4.4	DenseNet201	78
4.4.5	Comparação entre arquiteturas	80
4.5	Avaliação com conjunto de teste	82
4.6	Discussão	84
5	Considerações Finais	91
5.1	Contribuições da pesquisa	92
5.2	Sugestões para trabalhos futuros	93
5.3	Trabalhos Publicados	94
	Referências Bibliográficas	104
	APÊNDICES	104
A	Quantidade de amostras atrasadas por derivação vetocardiográfica	106
B	Evolução das métricas para os demais modelos	110
B.1	Treinamento com imagens de espaço de fase	110
B.1.1	MobileNetV2	110
B.1.2	ResNet-50	119
B.1.3	ResNet-101	127
B.1.4	DenseNet201	135

B.2	Treinamento com gráficos de recorrência	143
B.2.1	MobileNetV2	143
B.2.2	ResNet-50	151
B.2.3	ResNet-101	159
B.2.4	DenseNet201	167

1.1 – Motivação e Definição do Problema

É estimado que 17,9 milhões de pessoas morreram de doenças cardiovasculares em 2019. De acordo com a Organização Mundial de Saúde, esse número correspondeu a 32% das mortes e representou a maior causa de mortes globalmente. Dentre essas doenças se destaca o infarto do miocárdio, o qual, com acidentes vasculares cerebrais, causou cerca de 85% dessas mortes. Além disso, mais de 75% das mortes por doenças cardiovasculares ocorreram em países de média e baixa renda devido às possibilidades reduzidas de diagnóstico, aumentando as chances de ele ocorrer tardiamente, comprometendo o prognóstico (Organization, 2021; Brasil, 2017).

O diagnóstico de um infarto do miocárdio pode ser feito a partir de exames como angiografias, análise de biomarcadores cardíacos, como troponinas, em conjunto com sintomas, evidências de anormalidades na movimentação da parede cardíaca (Vafaie, 2016); ou pela eletrocardiografia: um exame rápido, barato e não invasivo (de Cardiologia, 2003). Nesse exame, eletrodos para registrar as variações dos potenciais elétricos cardíacos por um intervalo de tempo são dispostos sobre a pele do paciente, onde esse registro é chamado de eletrocardiograma (ECG). Por sua conveniência, conjuntos desses exames são utilizados em estudos acerca da condução elétrica cardíaca e na prática clínica.

O ECG mais comum tem 12 derivações, o qual permite a análise da atividade elétrica cardíaca por meio de diferentes perspectivas, auxiliando, dessa forma, a identificação do infarto agudo do miocárdio (IAM), arritmias e demais distúrbios que afetam a condução elétrica (de Cardiologia, 2003).

Essa análise em um ECG é, no geral, feita de forma qualitativa e para vários batimentos cardíacos, estando, portanto, dependente da experiência do examinador e sujeita a erros da sua parte (Hagiwara *et al.*, 2018). Ademais, certos subtipos de IAM não apresentam as alterações

marcantes que caracterizam um infarto no sinal de ECG, o que dificulta o diagnóstico com base apenas no ECG, além de esses casos estarem relacionados a um pior prognóstico (Pollack *et al.*, 2020).

De forma complementar ao ECG, o vetocardiograma (VCG) oferece uma representação vetorial dos potenciais elétricos cardíacos baseados em vetores ortogonais, sendo eles as derivações vetocardiográficas. Por meio delas, é possível visualizar as atividades elétricas do coração nas três dimensões do espaço euclidiano, tornando também possível a extração de parâmetros quantitativos como ângulos espaciais que podem ser utilizados como critérios de diagnóstico. O VCG pode ser obtido pelo Método de Frank (Frank, 1956), considerado seu padrão ouro. Sua obtenção, no entanto, não é clinicamente prática devido à necessidade de *hardware*, treinamento e posicionamento de eletrodos específico, que não favorece seu uso em pacientes na posição supina. A fim de contornar essas dificuldades, foram desenvolvidos métodos alternativos para a obtenção do VCG a partir do ECG de 12 derivações, os quais possuem resultados extremamente satisfatórios, de forma que ele possa ser facilmente aplicado na prática clínica (Kors *et al.*, 1990; Cortez & Schlegel, 2010).

Entre os métodos propostos para a obtenção do VCG está o método da Regressão de Kors (Kors *et al.*, 1990). Por meio dele é possível obter as três derivações do VCG a partir de uma combinação linear de 8 das 12 derivações de um ECG, sendo obtido um erro mínimo entre os parâmetros calculados por VCG obtido por esse método e os parâmetros calculados para um sinal obtido pelo método de Frank (Cortez & Schlegel, 2010).

Em razão dos sinais das derivações de ECG e de VCG naturalmente não possuírem comportamento perfeitamente periódico e variável com o tempo (Sridhar *et al.*, 2021), além da baixa amplitude dessas variações (Mathunjwa *et al.*, 2021), a extração de características não lineares desses sinais é eficiente em detectar essas mudanças e comportamentos não evidentes no domínio do tempo (Acharya *et al.*, 2016). Um caminho para isso é por meio da Reconstrução do Espaço de Fase (REF) por meio do método do tempo de atraso apresentado por Takens (Takens, 1981, 1983). Com ele, são obtidas informações acerca do comportamento caótico do sistema analisado, as quais são mais sensíveis a alterações morfológicas no sinal. Em razão disso, essa técnica já vem sendo empregada para a detecção de arritmias a partir de sinais de ECG, como mostrado por Amann *et al.* (2006) e Roopaei *et al.* (2010), além de isquemia e infarto do miocárdio, como já apresentado por Zimmerman *et al.* (2003) e Costa *et al.* (2018), respectivamente.

A visualização do espaço de fase (EF) de um sinal possui limitações, pois ele geralmente possui um número de dimensões superior ao limite de três dimensões do olho humano (Vieira, 2014; Mathunjwa *et al.*, 2021). Assim, ela é limitada à tridimensional ou em diferentes planos. Para contornar essa limitação, Eckmann *et al.* (1995) apresentou os gráficos de recorrência, uma forma de representar bidimensionalmente um EF m -dimensional com base na quantidade de retornos, isto é, recorrências, que ocorrem na trajetória do EF. Dessa forma, independentemente da dimensão de imersão (m), será obtida uma imagem contendo informações baseadas na dinâmica não linear e que podem representar os padrões intrínsecos à presença da patologia nesses sinais.

Esses padrões podem ser, então, determinados ou aproximados com base na análise das interações entre os parâmetros desses dados por meio de técnicas de aprendizado de máquina. Com isso, é possível gerar um modelo para a predição de outras variáveis ou a classificação de novos conjuntos de dados. Por esse motivo, essas técnicas vêm sendo utilizadas na área médica para a identificação de diferentes patologias como câncer (Daoud & Mayo, 2019), COVID-19 (Narin *et al.*, 2021), arritmias (Mathunjwa *et al.*, 2022; Isin & Ozdalili, 2017), e infarto (Hammad *et al.*, 2021; Khan & Pachori, 2021); devido a sua capacidade de auxiliar na tomada de decisões de forma rápida e precisa (Jiang *et al.*, 2017).

Para sinais cardíacos, isso pode ser feito a partir dos dados relacionados aos sinais como parâmetros extraídos a partir de sinais de ECG (Dokur & Ölmez, 2001), VCG (Dehnavi *et al.*, 2011), da REF (Costa *et al.*, 2018) ou a partir de imagens de uma determinada derivação de ECG (Acharya *et al.*, 2017), de um plano bidimensional de EF, de um EF com dimensão 2 (Rubin *et al.*, 2017), ou de um gráfico de recorrência (Mathunjwa *et al.*, 2022). Para esse último, no entanto, é necessário ser obtida a dimensão de imersão, podendo isso ser feito por meio da análise progressiva de diferentes valores de dimensão de imersão associado ao Método dos Falsos Vizinhos (Vieira, 2014), tendo ele um custo computacional associado à determinação do valor adequado, que pode não representar um ganho significativo na capacidade de detecção de patologias.

Considerando os sinais cardíacos representados como imagens, como o espaço de fase bidimensional e os gráficos de recorrência, sua classificação se dá, preferencialmente, por meio de redes neurais convolucionais, do inglês *Convolutional Neural Networks* (CNN) por elas serem capazes de realizar a extração automática de características e terem um foco no reconhecimento de padrões e formas em duas dimensões, com baixa variância em relação a

distorções na imagem como a sua translação ou seu escalonamento (Haykin, 2008). Duas CNNs foram utilizadas de forma complementar como classificadores por Mathunjwa *et al.* (2022) para realizar a identificação de diferentes tipos de arritmias a partir de gráficos de recorrências em sinais de ECG. Já Acharya *et al.* (2017) utilizou sinais de ECG como entradas em CNN para a identificação do infarto do miocárdio. Apesar disso, não foi observado na literatura a associação de gráficos de recorrência gerados a partir de derivações de VCG com CNNs.

Nesse contexto, além da possibilidade de redução do tempo de processamento caso a dimensão de imersão igual a 2 seja suficiente para o diagnóstico do infarto, o presente trabalho propõe a análise de diferentes classificadores baseados em CNNs, testando diferentes arquiteturas delas.

O primeiro conjunto desses classificadores é treinado com imagens de espaços de fase de derivações vetocardiográficas com dimensão de imersão igual a 2. Já o segundo conjunto será treinado com gráficos de recorrências a fim de determinar, considerando o custo computacional envolvido e os resultados obtidos, qual dos dois métodos possui maior viabilidade de ser associado à prática clínica para a identificação de infarto do miocárdio.

1.2 – Objetivos

1.2.1 – Objetivo geral

O presente trabalho procura comparar o desempenho de diferentes arquiteturas de redes neurais convolucionais treinadas a partir de imagens de espaço de fase bidimensional e gráficos de recorrência. Com essa comparação para os mesmos sinais, espera-se determinar qual arquitetura e qual tipo de imagem possui um melhor custo-benefício para a identificação do infarto do miocárdio.

1.2.2 – Objetivos específicos

- ▣ Determinar, dentre os sinais de ECG de pacientes normais e infartados da base de dados escolhida, quais serão utilizados;
- ▣ Reconstruir o VCG pelo Método de Kors com os sinais a serem utilizados;
- ▣ Reconstruir o espaço de fase bidimensional das três derivações de VCG;

- ▣ Determinar um valor para dimensão de imersão mais adequado para o conjunto de sinais a ser utilizado;
- ▣ Obter o gráfico de recorrência para as três derivações de VCG;
- ▣ Realizar o treinamento e teste de diferentes arquiteturas de redes neurais convolucionais para a classificação dos dois tipos de imagens em análise;
- ▣ Identificar qual dos tipos de imagens leva a um melhor custo-benefício para a identificação do infarto considerando as métricas de classificação e custo computacional envolvido no processamento de sinais.

1.3 – Revisão da literatura

O uso de métodos para análise não linear para identificar cardiopatias têm ocorrido desde a década de 2000, especialmente para detecção de arritmias cardíacas como a fibrilação ventricular e de taquicardia ventricular em sinais de ECG, além da sua diferenciação de sinais normais (Amann *et al.*, 2006; Roopaei *et al.*, 2010).

Em Amann *et al.* (2006), foi utilizado o método do tempo de atraso para reconstruir o espaço de fase bidimensional a partir de uma derivação eletrocardiográfica, onde o tempo de atraso definido foi de 0,5s. Para esse trabalho foram utilizados 48 arquivos com 2 canais de 1805s por arquivo da base BIH-MIT, 35 arquivos com 1 canal por arquivo de 508s da base CU e 40 arquivos com 2 canais de 1800s da base AHA (Amann *et al.*, 2006). Deles, segmentos de 8s foram obtidos. A imagem do espaço de fase obtida foi então dividida em uma malha de 40×40 blocos quadrados, sendo obtido o parâmetro d , o qual corresponde à razão da quantidade de blocos visitados pelo espaço de fase pelo total de blocos. Caso d seja acima de algum limiar d_0 , nesse caso definido por meio de testes como $d_0 = 0,15$, esse sinal pertence à classe "Fibrilação Ventricular", caso contrário é um sinal da classe "Normal". Destaca-se que não há especificação da quantidade de arquivos para cada classe.

Esse método, em comparação com outros métodos para identificação de fibrilação ventricular citados nesse mesmo trabalho, foi o mais rápido e que possuiu a melhor sensibilidade ao considerar qualquer valor de especificidade por meio da curva Característica de Operação do Receptor (ROC, *Receiver Operating Characteristic*). Considerando todas as bases de dados utilizadas nesse trabalho foram obtidos valores de acurácia, sensibilidade e especificidade

de 96,2%, 79% e 97%, respectivamente. A especificidade, nesse caso, foi considerada mais importante, pois é priorizada a obtenção de um baixo número de falsos positivos no lugar de falsos negativos a fim de evitar a desfibrilação desnecessária, o que poderia levar a uma parada cardíaca.

Com base no método apresentado por Amann *et al.* (2006), Roopaei *et al.* (2010) também utilizou espaços de fase bidimensionais e propôs parâmetros adicionais para serem usados em conjunto com o parâmetro d para a diferenciação de sinais das classes "Fibrilação Ventricular" e "Taquicardia Ventricular", os quais são semelhantes, e sinais das classes "Normal" e "Fibrilação Ventricular".

No total, 140 sinais foram utilizados, onde eles são provenientes de duas bases: a base de arritmias MIT-BIH, que contém ECGs de 49 sujeitos, sendo 19 da classe "Normal", 20 da classe "Taquicardia Ventricular" e 20 da classe "Fibrilação ventricular"; e base da unidade coronariana da Enfermaria real de Edimburgo, contendo 81 sujeitos, sendo 50 da classe "Normal", 50 de "Taquicardia Ventricular" e 40 de "Fibrilação Ventricular".

Para a classificação desses sinais, dois espaços de fase para cada sinal são reconstruídos por meio do método do tempo de atraso, sendo eles tratados como duas imagens. Delas, parâmetros para avaliar a autossimilaridade entre as trajetórias nessas imagens são obtidos para, então, classificar o sinal com base nela, já que um sinal rítmico e regular possui uma alta autossimilaridade.

Para quantificar a autossimilaridade por meio de dois espaços de fase, Roopaei *et al.* (2010) propôs três novos métodos. São eles: o método da diferença, onde o mesmo índice d utilizado por Amann *et al.* (2006) é calculado para cada um dos espaços de fase, sendo a diferença absoluta entre eles o resultado desse método; o método da similaridade, no qual os espaços de fase reconstruídos são divididos em blocos, é feita a operação lógica "E" entre os blocos correspondentes, sendo os blocos visitados pelo espaço de fase no resultado da operação "E" contados; e o método da contagem ponderada de blocos, o qual é semelhante ao método da diferença, mas blocos visitados pelo espaço de fase cujos vizinhos também tenham sido visitados recebem peso dois, sendo ao fim determinada a diferença entre os índices d calculados para os dois espaços de fase reconstruídos.

Visando avaliar esses métodos, foi determinado para cada um deles um valor de limiar para a classificação binária entre cada par para as classes "Fibrilação Ventricular", "Taquicardia Ventricular" e "Normal", sendo esse limiar determinado para o conjunto de treino e sua capacidade

de generalização verificada por meio de um conjunto de teste, sendo ele o limiar que levou ao maior valor de área sob a curva ROC. Para esses casos, o tempo de atraso utilizado para a reconstrução dos espaços de fase também foi determinado empiricamente em uma faixa de testes para esse tempo entre 0,01s e 0,09s. Para a discriminação de sinais entre classes "Fibrilação Ventricular" e "Taquicardia Ventricular", e entre as classes "Fibrilação Ventricular" e "Normal", as acurácias de teste foram de 89,80% e 94,12% considerando o método da similaridade, respectivamente. Já para a diferenciação entre as classes "Taquicardia Ventricular" e "Normal" o método da diferença possuiu um melhor desempenho, levando a uma acurácia de 91,51%.

Além da detecção de arritmias, doenças como a Doença arterial Coronariana e Infarto do Miocárdio podem ser detectados por métodos de análise não linear. A partir dos métodos apresentados por Amann *et al.* (2006) e Roopaei *et al.* (2010), Costa *et al.* (2018) utilizou redes neurais artificiais do tipo MLP com diferentes combinações de parâmetros baseados nos parâmetros apresentados nesses trabalhos para a classificação binária, além de diferentes tamanhos de bloco terem sido utilizados.

Nesse trabalho, dados de 153 sujeitos, os quais podem possuir 1 ou mais sinais contendo as três derivações vetocardiográficas de Frank, provenientes da base de dados PTB (*Physikalisch-Technische Bundesanstalt*, Instituto Nacional de Metrologia da Alemanha) (Bousseljot *et al.*, 1995), foram classificados nas classes "Normal" versus "Infarto do miocárdio" (Grupo 1), "Normal" versus "Infarto do miocárdio na parede anterior" (Grupo 2), "Normal" versus "Infarto do miocárdio na parede inferolateral" (Grupo 3) e "Normal" versus "Infarto do miocárdio na parede inferior" (Grupo 4). Os grupos, possuem respectivamente 120, 120, 46 e 70 sinais, sendo metade deles da classe normal e a metade correspondentes ao tipo de infarto a ser identificado. Ademais, entre os sinais de cada grupo, 1/3 deles foi destinado ao conjunto de teste e o restante destinado ao treinamento dos modelos.

Para isso, foi reconstruído o espaço de fase bidimensional de cada uma dessas derivações, onde seus parâmetros foram extraídos considerando os tamanhos de bloco 5×5 , 10×10 , 20×20 e 40×40 . Combinações desses parâmetros foram utilizadas como entradas em redes neurais, sendo possível, para a classificação em "Normal" versus "Infarto do Miocárdio", obter uma sensibilidade, especificidade e acurácia de 92%, 96% e 94%, respectivamente; para as classes "Normal" versus "Infarto do Miocárdio na Parede Anterior" as métricas tiveram, na mesma ordem, valores de 91,16%, 100% e 95,83%; para a parede inferolateral, 90%, 90% e 90%; e, para a parede inferior, 92,3%, 53,84% e 73,07%.

Além disso, visando diferenciar sinais normais, de Doença Arterial Coronariana (DAC) e Infarto do Miocárdio, Hagiwara & Faust (2017) utilizou, entre métodos de análise não linear, parâmetros de análise de quantificação de recorrência obtidos a partir de gráficos de recorrência provenientes de batimentos cardíacos de ECG, onde 10.546 pertencem à classe "Normal" e provêm de 52 sujeitos; 40.182 são da classe "infarto do miocárdio" (148 sujeitos) e 41.545 são da classe "DAC", sendo eles de 7 sujeitos.

Esses parâmetros foram então ranqueados com base em um Teste t , sendo o critério de ordenação o valor t . Com base nisso, foi possível mostrar que esses parâmetros são capazes de serem combinados para a criação de um modelo para classificar sinais dessas classes. Além disso, considerando os gráficos de recorrência, gráficos de bi-espectro, gráficos para o cumulante e gráficos de violino para as métricas: entropia de permutação, dimensão fractal, entropia *fuzzy* e comprimento médio da linha diagonal para cada classe, é possível observar diferenças de forma qualitativa entre elas (Hagiwara & Faust, 2017).

A discriminação de sinais da classe "Infarto do miocárdio" de sinais da classe "Normal" com base em parâmetros não lineares também foi feita por Sridhar *et al.* (2021). Nesse trabalho, 200 sinais da derivação eletrocardiográfica II, sendo 52 saudáveis e 148 infartados, também provenientes da base de dados PTB. Segmentos de 2 segundos desses sinais foram obtidos, totalizando 16.826 segmentos normais e 3.796 segmentos de infarto, tendo eles suas características extraídas por métodos de análise não linear.

Entre esses métodos estão a Análise de Quantificação de Recorrência, o Bi-espectro, Entropia Aproximada, Entropia de Permutação, Análise de Flutuação Destendenciada, Dimensão Fractal, Maior expoente de Lyapunov, Entropia Amostral, Entropia de Rényi, entre outros (Sridhar *et al.*, 2021), totalizando um total de 45 características extraídas por esses métodos. Por meio de um teste t foi possível determinar quais características seriam consideradas significativas para a classificação e, portanto, utilizadas como entrada nos classificadores, o que ocorreu para 14 dessas 45 características.

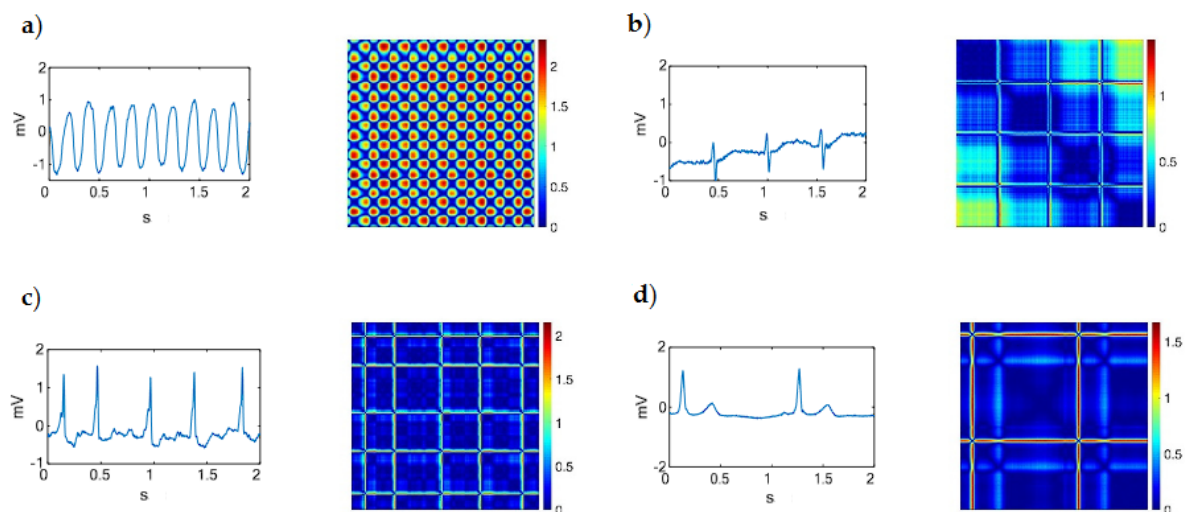
Quatro classificadores foram treinados e testados usando uma validação cruzada de dez vezes (*ten-fold cross-validation*). Foram eles: K-Vizinhos mais próximos (KNN, *K-nearest neighbors*), Árvore de Decisão (DT, *Decision Tree*), Máquina de vetores de suporte (SVM, *Support Vector Machine*) e Rede Neural Probabilística (PNN, *Probabilistic Neural Network*). Deles, o melhor desempenho obtido foi para o SVM, tendo obtido valores de acurácia, sensibilidade e especificidade de 97,96%, 98,89% e 93,81%, respectivamente. Entre as limitações

apresentadas por esse trabalho está a extração e seleção manual de características, o que requer algum custo de tempo para a computação de todas elas (Sridhar *et al.*, 2021).

A fim de evitar a extração e seleção manual de características, uma rede neural convolucional pode ser utilizada com as imagens de espaços de fase reconstruídos com duas dimensões ou gráficos de recorrência. (Mathunjwa *et al.*, 2022) utilizou gráficos de recorrência como entradas em redes neurais convolucionais para diferenciar quatro classes de arritmias (fibrilação ventricular, fibrilação atrial, contração atrial prematura e contração ventricular prematura), ruídos e sinais de pacientes saudáveis.

Para isso, foram utilizados 128 sinais das classes acima, onde seus tempos de duração poderiam ser 8 minutos, 30 minutos ou 10 horas. Para cada sinal foram obtidos segmentos de 2 segundos, sendo cada segmento convertido para um gráfico de recorrência sem limiarização. Exemplos desse gráfico podem observados na Figura 1.1.

Figura 1.1 – Sinais de ECG para diferentes classes e seus gráficos de recorrência sem limiarização correspondentes: a) Fibrilação ventricular; b) Ruído; c) Fibrilação Atrial; d) Normal.



Fonte: Adaptado de (Mathunjwa *et al.*, 2022).

A classificação desses sinais foi feita em dois estágios, cada um composto por uma rede neural convolucional, onde a primeira era responsável por diferenciar sinais de fibrilação ventricular de ruídos e dos sinais das demais classes, já que tanto a fibrilação ventricular quanto o ruído levam ao desaparecimento da onda R (Mathunjwa *et al.*, 2022). Os sinais das demais classes atuam como entradas em um segundo classificador a fim de determinar a qual das demais

classes (fibrilação atrial, contração atrial prematura, contração ventricular prematura ou normal) eles pertencem.

Os dois classificadores utilizados possuíam a arquitetura de rede neural convolucional ResNet, onde para o primeiro estágio foi utilizada a ResNet-18 enquanto para o segundo estágio foi utilizada a ResNet-50, sendo elas inicializadas com pesos aleatórios e treinadas apenas com o conjunto de dados desse trabalho. Além disso, foi utilizada uma validação cruzada de cinco vezes (*five-fold cross-validation*). Por fim, considerando os dois modelos operando em conjunto, as métricas de acurácia, sensibilidade, especificidade retornadas para o conjunto de teste foram respectivamente 94,85%, $94,44 \pm 2,94\%$ e $94,96 \pm 7,31\%$. Um resumo dos trabalhos descritos acima é apresentado na Tabela 1.1.

É notado que os trabalhos de Amann *et al.* (2006), Roopaei *et al.* (2010) e Costa *et al.* (2018) necessitam de uma etapa para extração dos parâmetros de espaço de fase. De forma análoga, Hagiwara & Faust (2017) e Sridhar *et al.* (2021) também necessitam dessa etapa para parâmetros relacionados aos gráficos de recorrência, de forma que, para ambos os casos, é necessário um tempo maior de processamento devido a essas etapas.

Além disso, os métodos empregados por Roopaei *et al.* (2010) e Costa *et al.* (2018) requerem que dois espaços de fase sejam reconstruídos para cada sinal, o que também é responsável por um acréscimo nesse tempo.

Essa etapa de processamento para extração de parâmetros das imagens de espaços de fase reconstruídos e gráficos de recorrência para a sua classificação pode ser substituída pelo emprego de CNNs, as quais realizam essa extração de forma inerente ao seu funcionamento, por meio de suas camadas convolucionais. Apesar de ter empregado essa ferramenta, Mathunjwa *et al.* (2022) utilizou poucos sinais para a obtenção das amostras utilizadas no treinamento e teste dessa CNN, de forma que sua capacidade de generalização pode estar comprometida.

Isso pode ser atribuído ao fato de que com dezenas de milhares de sinais usados no treinamento, eles eram provenientes de apenas 128 exames, onde cada trecho com 2 segundos era considerado um sinal. Nenhum procedimento de separação adequada desses sinais, considerando os diferentes conjuntos, foi descrito. Dessa forma, há o risco da ocorrência de *data leakage* (vazamento de dados).

Neste cenário de *data leakage*, a informação utilizada no treinamento também está sendo utilizada na avaliação do modelo, o que leva a uma superestimação das suas capacidades (JM *et al.*, 2018; Tampu *et al.*, 2022), algo reportado por Yagis *et al.* (2021), no qual foram comparados

Tabela 1.1 – Estado da arte: Trabalhos para classificação de sinais cardíacos que utilizam métodos de análise não linear, base de dados utilizadas, metodologia e resultados alcançados.

Autoria	Metodologia	Resultados Alcançados
Amann <i>et al.</i> (2006)	<p>Utilizou uma base com 48 arquivos com 2 canais contendo 1805s, 35 arquivos com 1 canal de 508s e 40 arquivos com 2 canais de 1800s, obtendo segmentos de 8s deles.</p> <p>Espaço de fase bidimensional de uma derivação eletrocardiográfica para detectar fibrilação ventricular; tempo de atraso fixo de 0,5s; imagens divididas em malha de blocos quadrados; razão de blocos visitados pelo espaço de fase limiar de classificação.</p>	<p>Acurácia: 96,2%; Sensibilidade: 79,0%; Especificidade: 97,0%.</p>
Roopaei <i>et al.</i> (2010)	<p>Base de 140 sinais, sendo 50 da classe "Normal", 50 "Taquicardia Ventricular" e 40 "Fibrilação Ventricular".</p> <p>Discriminação (Fibrilação ventricular x Taquicardia ventricular) e identificação de arritmias (Normal x Fibrilação ventricular e Normal x Taquicardia ventricular) por meio da comparação da autossimilaridade das trajetórias de dois espaços de fase bidimensionais de uma derivação eletrocardiográfica; comparação baseada em parâmetros obtidos após a divisão em malha de blocos; para par de classe se considerou o método de comparação com melhor desempenho.</p>	<p>Discriminação de arritmias: acurácia de 89,80%; Identificação de Fibrilação ventricular: acurácia de 94,12%; Identificação de Taquicardia ventricular: acurácia de 91,51%.</p>
Costa <i>et al.</i> (2018)	<p>Base de obtida a partir da base PTB com sinais de 153 sujeitos, onde o grupo 1 possui 120 sinais, sendo 60 "Normal" e 60 "Infarto do Miocárdio".</p> <p>Identificação de infarto em diferentes paredes por rede neural MLP em diferentes paredes a partir de combinações dos parâmetros utilizados por Roopaei <i>et al.</i> (2010) para espaço de fase bidimensional, sendo suas malhas determinadas para diferentes tamanhos de bloco.</p>	<p>Classificação x Infarto do Miocárdio: Acurácia: 94%; Sensibilidade: 92%; Especificidade: 96%.</p>
Hagiwara & Faust (2017)	<p>Diferenciação entre Doença Arterial Coronariana e Infarto do Miocárdio com parâmetros de análise de quantificação de recorrência rankeados por teste t considerando sinais de uma derivação provenientes de 207 sujeitos.</p> <p>Também foi feita uma comparação qualitativa dos gráficos de recorrência, gráficos de bi-espectro, para o cumulante e gráficos de violino para entropia de permutação, dimensão fractal, entropia <i>fuzzy</i> e comprimento médio da linha diagonal.</p>	<p>Por meio da análise qualitativa das figuras geradas foi possível diferenciar as três classes.</p>
Sridhar <i>et al.</i> (2021)	<p>Classificação utilizando 52 sinais da classe "Normal" e 148 da classe "Infarto do Miocárdio" da base PTB; cada sinal foi dividido em segmentos de 2s, sendo feita a extração de 45 características (análise de quantificação de recorrência, bi-espectro, entropia aproximada, etc.); as 14 mais significantes foram usadas como entrada em classificadores (KNN, DT, SVM, PNN).</p>	<p>Para a SVM, a qual possuiu melhor desempenho: Acurácia: 97,96%; Sensibilidade: 98,89%; Especificidade: 93,81%.</p>
Mathunjwa <i>et al.</i> (2022)	<p>Classificação em dois estágios, sendo no primeiro por uma ResNet-18 entre as classes "Fibrilação Ventricular" e "Ruído", e no segundo entre as classes "Normal" e três tipos de arritmia por uma ResNet-50; foram utilizados 128 sinais, sendo obtidos segmentos de 2s, dos quais se construíram seus gráficos de recorrência não limiarizados.</p>	<p>Para os dois modelos operando em conjunto: Acurácia: 94,85%, Sensibilidade: 94,44±2,94% Especificidade: 94,96±7,31%.</p>

modelos treinados com dados separados adequada e inadequadamente, havendo diferenças de acurácias para o conjunto de teste de até 54,83% entre eles, onde para todas elas estavam acima de 90% quando separadas inadequadamente e não passavam de 71%, caso contrário.

Esse problema pode ocorrer em sinais biomédicos, por exemplo, quando dados de um mesmo paciente estão nos conjuntos de treinamento e teste (Samala *et al.*, 2021). No caso, de Mathunjwa *et al.* (2022), caso não tenha ocorrido uma separação adequada, sinais de ciclos próximos de um mesmo exame, os quais são similares, podem estar sendo utilizados no treinamento e teste. Nessa situação, gráficos de recorrência semelhantes serão obtidos, o que facilitaria a inferência deles, caso algum gráfico de recorrência similar deste mesmo sinal tenha sido apresentado ao modelo durante seu treinamento, devido à alta correlação entre sinais de um mesmo paciente. A consequência disso é um aumento das classificações corretas e, em razão disso, das métricas obtidas para o conjunto de teste nesse trabalho.

1.4 – Estrutura do Documento

Inicialmente, no Capítulo 1, são expostas informações sobre a incidência de doenças cardiovasculares e infarto do miocárdio, além de metodologias de diagnóstico, destacando o uso do sinal cardíaco ECG, além de métodos de processamento que podem ser aplicados ao ECG, além de possibilidades de associação desse processamento para o diagnóstico automático do infarto. Em seguida, se apresentam os objetivos gerais e específicos e a presente estrutura do documento para todo o trabalho. Por fim, uma revisão da literatura é apresentada.

No Capítulo 2, são apresentadas as características e epidemiologia do infarto agudo do miocárdio. Nesse mesmo capítulo, são então descritos os sinais elétricos cardíacos de ECG e VCG e os impactos do infarto nesses sinais. Também são expostos métodos de análise não linear de um sinal, particularmente as representações em espaço de fase e gráficos de recorrência. Em seguida, são apresentadas as redes neurais.

No Capítulo 3, é inicialmente mostrada a proposta do trabalho e cenários de aplicação dela. Ademais, é apresentada a metodologia, o que inclui o banco de dados a ser utilizado nesse trabalho, os processamentos realizados nos sinais eletrocardiográficos, a elaboração e treinamento dos modelos de redes neurais convolucionais e as métricas utilizadas para avaliar seu desempenho.

No Capítulo 4 são expostos os resultados da pesquisa, particularmente em relação às métricas para avaliação dos modelos de classificação obtidos e a discussão delas.

Por fim, no Capítulo 5, são sumarizados os resultados obtidos com este trabalho e são mostradas propostas para a sua melhoria e continuação.

FUNDAMENTAÇÃO TEÓRICA

2.1 – Infarto Agudo do Miocárdio

O infarto agudo do miocárdio se destaca entre as doenças cardiovasculares, doenças que afetam o coração e o suprimento de sangue pelo sistema vascular para órgãos vitais, devido a sua alta taxa de mortalidade, a qual ocorre para 33% dos pacientes em um intervalo de 30 dias, sendo metade delas antes da chegada ao hospital (Gaziano *et al.*, 2006).

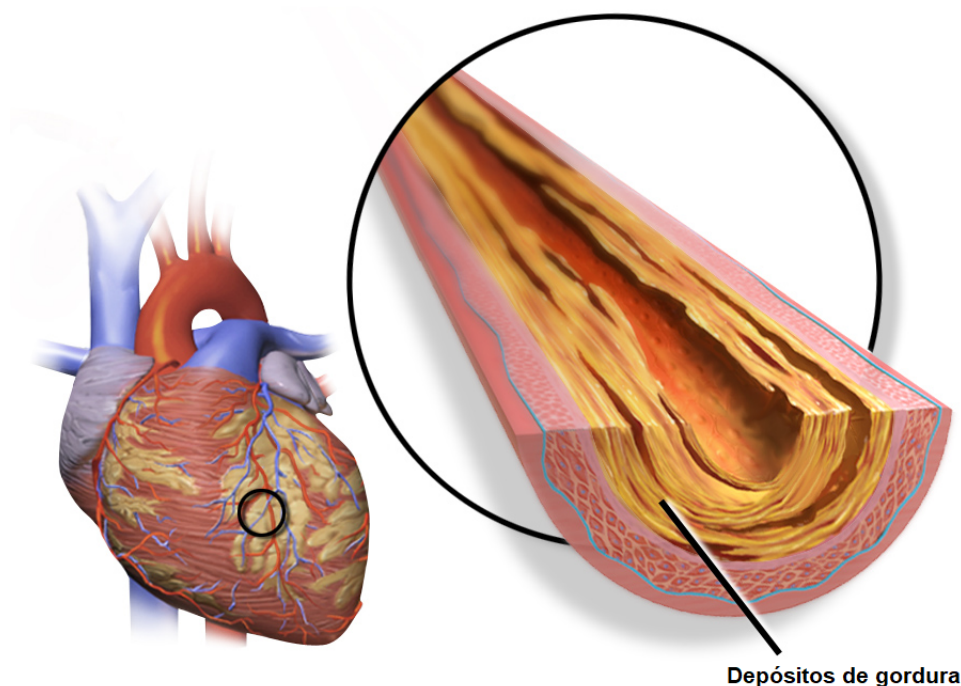
As doenças cardiovasculares, durante o século XX, passaram a representar de 10% para 30% das mortes ao redor do mundo, das quais 80% ocorreram em países subdesenvolvidos ou emergentes, havendo uma tendência de aumento das mortes a partir da metade do século XX, enquanto, antes, elas se concentravam em países desenvolvidos (White & Chew, 2008). Esse fenômeno de transição epidemiológica se deu pelo aumento dos fatores de riscos para essas doenças, como obesidade e diabetes, como consequência das transformações econômicas e demográficas deste século, e de hábitos alimentares, como o aumento do consumo de ácidos graxos saturados (Gaziano *et al.*, 2006; White & Chew, 2008; Lima *et al.*, 2000).

O IAM ocorre devido a uma obstrução, seja ela total ou parcial, de uma artéria coronária, responsável pelo fluxo sanguíneo que supre o oxigênio e nutrientes necessários para o músculo cardíaco. Com a interrupção de, em média, de 2 a 4 horas, e falta desses componentes necessários para seu funcionamento, há a necrose do músculo cardíaco (Gaziano *et al.*, 2006; Thygesen *et al.*, 2007), o que prejudica sua capacidade de contração, reduzindo, assim, o volume de sangue ejetado a cada batimento e as capacidades físicas do indivíduo. Todo esse processo também possui consequências na atividade elétrica do coração, sendo esses efeitos mostrados na Seção 2.2.

Cerca de 70% dos casos fatais de IAM ocorrem devido às obstruções nas artérias coronárias, causadas por placas de gordura em seu interior, sendo essa condição chamada de

Doença Arterial Coronariana e representada na Figura 2.1. Essas placas formam então coágulos que impedem a circulação de sangue para o músculo cardíaco. Além disso, o IAM pode ser causado por espasmos coronarianos, representados na Figura 2.2, onde há uma compressão excessiva das paredes das artérias do coração, o que causa o bloqueio do fluxo sanguíneo (Yasue *et al.*, 2008).

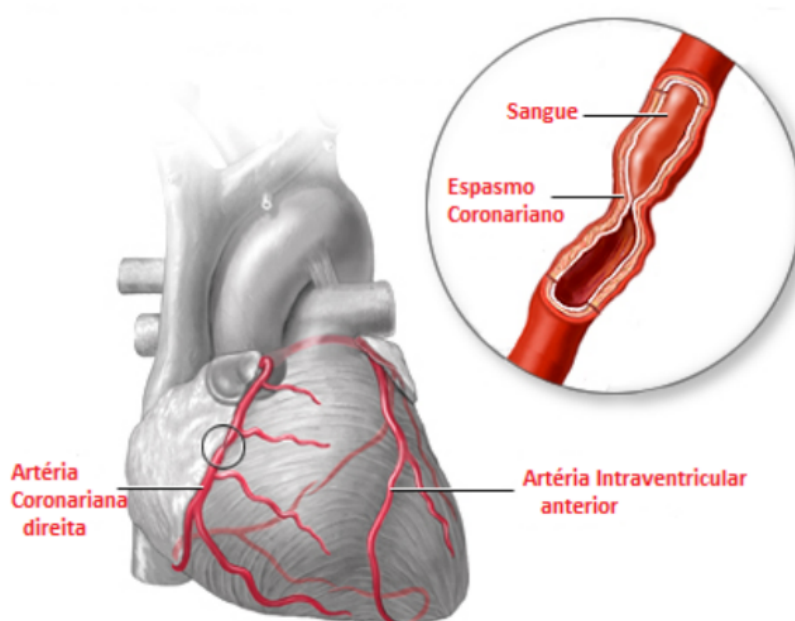
Figura 2.1 – Representação da Doença Arterial Coronariana.



Fonte: Adaptada de (Blaus, 2014).

Os sintomas mais usuais para um IAM são dor intensa no tórax que pode se propagar para os ombros, braço e mandíbula; náuseas, vômito, tontura, desfalecimento, sensação de compressão no peito e ansiedade. Em certos casos esses sintomas podem ser mascarados, especialmente para indivíduos que fazem parte de ao menos um dos seguintes grupos: sexo feminino, idosos, diabéticos, com insuficiência cardíaca ou com marca-passo (Andrade *et al.*, 2009).

Figura 2.2 – Representação de um espasmo coronariano.



Fonte: Adaptada de (Costa, 2017).

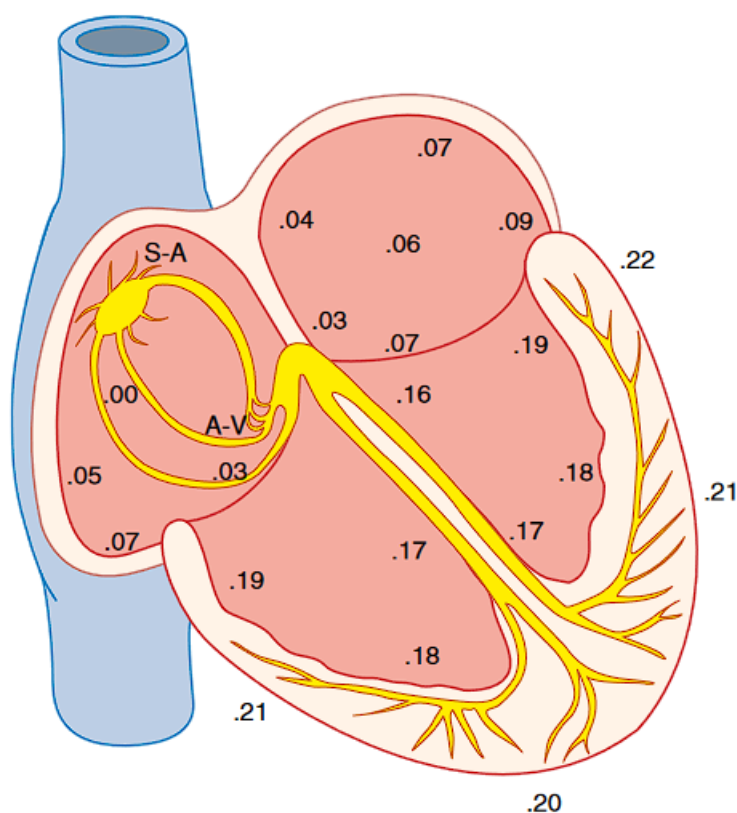
2.2 – Sinais Elétricos Cardíacos

2.2.1 – Eletrocardiograma

A atividade cardíaca e fenômenos relacionados a ela podem ser registrados por meio do exame de eletrocardiografia, cujo resultado é um eletrocardiograma, um registro dos potenciais de campo elétrico gerados pelo coração. Isso é possível, pois ele possui um sistema responsável pela geração e condução de impulsos que geram uma contração rítmica do músculo cardíaco (Guyton & John, 2011), mostrado na Figura 2.3. A geração desses impulsos ocorre em uma região com células especializadas no átrio (câmara cardíaca superior) direito, onde essas células possuem a capacidade de autoexcitação. Os impulsos são então propagados para o nodo atrioventricular, responsável por atrasar esses impulsos para que os átrios se contraíam antes dos ventrículos, câmaras cardíacas inferiores responsáveis pela ejeção de sangue para fora do coração. Os impulsos então se propagam para o feixe de His, os quais se dividem nos ramos esquerdo e direito, havendo então a propagação rápida desses impulsos e a contração síncrona dos ventrículos (Guyton & John, 2011).

Em um ECG, graças à corrente elétrica que se propaga do coração para a superfície do corpo (Guyton & John, 2011), os potenciais de campo elétrico, os quais estão associados

Figura 2.3 – Em amarelo: sistema de condução dos impulsos cardíacos. Nele, os impulsos são gerados nos nodos sinoatriais (S-A), propagando-se das câmaras superiores para as inferiores. Os números representam o intervalo para chegada do impulso no local em frações de um segundo a partir do impulso inicial.



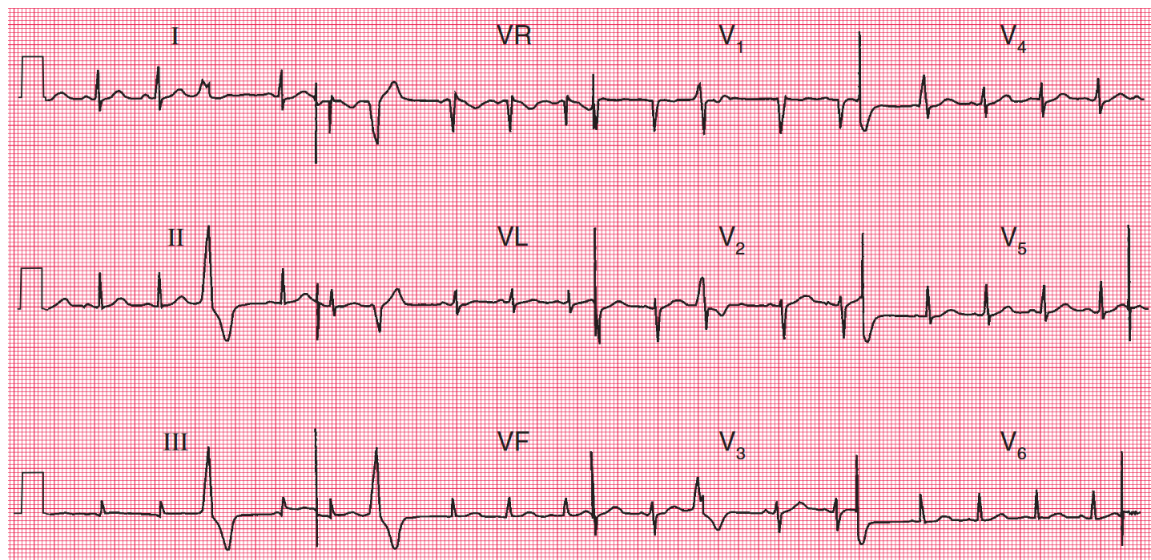
Fonte: (Guyton & John, 2011).

às despolarizações e repolarizações que ocorrem no ciclo cardíaco, são obtidos por meio de eletrodos dispostos sobre o tronco e membros do paciente, amplificados e registrados em uma fita de papel ou digitalmente. Por ser um método não invasivo, de rápida aplicação e baixo custo, tanto de uso quanto dos equipamentos necessários, que também são de fácil transporte e manuseio, o ECG se tornou um exame de rotina em clínicas e consultórios para a avaliação da atividade cardíaca e diagnóstico de doenças cardiovasculares (de Cardiologia, 2003).

O sinal obtido por um par de eletrodos sobre o corpo do paciente corresponde a uma perspectiva de visualização dos potenciais elétricos, sendo cada perspectiva chamada de derivação eletrocardiográfica. Normalmente, exames de ECG voltados para diagnóstico ou pesquisas possuem doze derivações (Moffa, 2001), como mostrado na Figura 2.4. Seis delas, denominadas DI, DII, DIII, aVR, aVL e aVF; são chamadas de derivações periféricas já que, considerando o plano frontal do coração como referência, elas representam o deslocamento vetorial das diferenças de potencial, sendo esse deslocamento para cima, baixo, direita ou

esquerda. As derivações, V1, V2, V3, V4, V5 e V6; são chamadas de derivações precordiais e representam o deslocamento para frente e para trás, dando uma ideia de profundidade.

Figura 2.4 – ECG de 12 derivações.

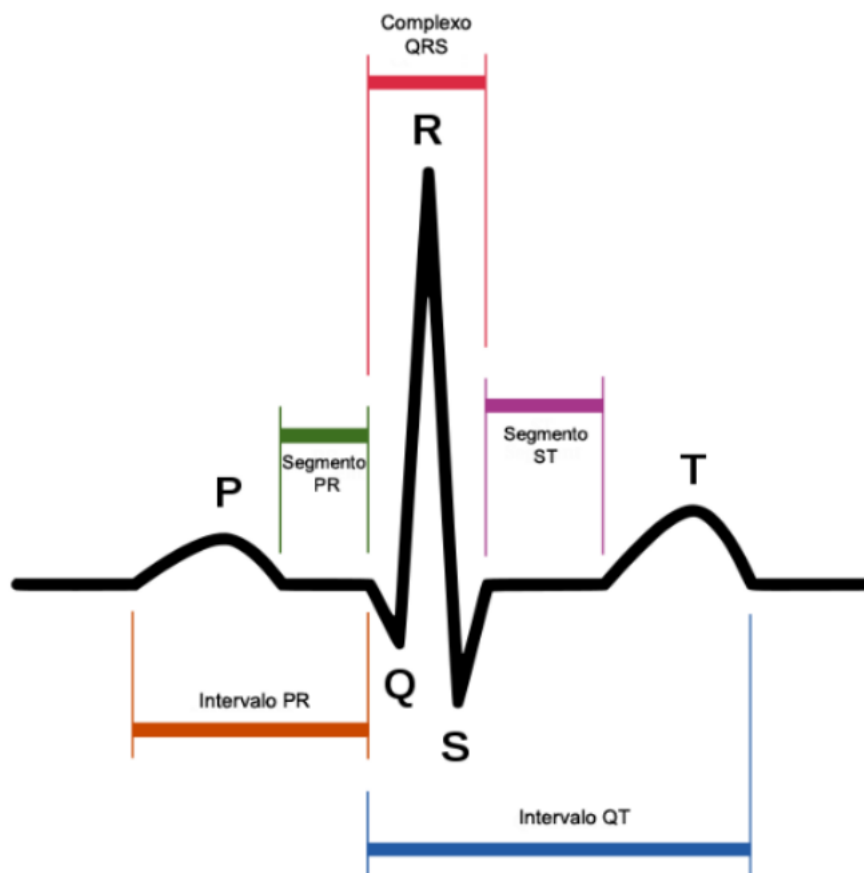


Fonte: (Hampton, 2014).

Cada derivação de ECG possui componentes que representam os fenômenos do ciclo cardíaco que podem ser associados à função dos pontos no coração em que eles se originam ou ocorrem. Dessa forma, eles podem ajudar na identificação da região afetada por alguma anormalidade. Essas componentes são mostrados na Figura 2.5. Nela, a linha horizontal é chamada de linha de base ou linha isoeletrica. Os pontos fora dessa linha correspondem aos eventos do ciclo cardíaco e são, da esquerda para a direita, correspondentes à onda P, que representa a despolarização dos átrios, as câmaras superiores do coração, que ocorre devido a estímulos gerados no nodo sinoatrial; o intervalo PR, o qual é o tempo até que esses estímulos cheguem aos ventrículos; o complexo QRS, correspondente à despolarização dos ventrículos, o que causa sua contração; o segmento ST, intervalo para a repolarização ventricular; e a onda T, quando essa repolarização ocorre (Pastore *et al.*, 2016).

Por fim, a partir da análise de todos esses eventos por diferentes perspectivas pelas derivações de ECG e associando eles com os sintomas reportados e histórico clínico do paciente é possível diagnosticar diferentes patologias cardíacas (Hampton, 2014), como isquemias, infartos, arritmias, distúrbios de condução, inflamações, entre outras (Sukienik, 2015).

Figura 2.5 – Sinal básico de eletrocardiograma e suas ondas características.



Fonte: (HacksCooking, 2013).

Impactos do IAM no sinal de ECG

De acordo com Sukienik (2015), as características da obstrução determinam, além da evolução da doença, os impactos do infarto no sinal eletrocardiográfico. Uma oclusão total, a qual representa cerca de um terço dos casos, resulta em um Infarto Agudo do Miocárdio com Supra de ST (IAMCSST), o qual normalmente evolui para um IAM com onda Q patológica. Um critério para considerá-la patológica é sua amplitude ser maior ou igual a um quarto do complexo QRS em no mínimo duas derivações e sua duração ser maior que 0,04 segundos (Pastore *et al.*, 2016). Além disso, um dos fenômenos associados a infartos é o supra desnível no segmento ST em ao menos duas derivações contíguas, isso é, de perspectivas adjacentes (Bueno, 2011).

Já se a oclusão, independente de ser parcial ou total, possuir boa circulação colateral, isso é, por caminhos alternativos, é provável ocorrer um Infarto Agudo do Miocárdio Sem Supra de ST (IAMSSST), o qual evolui para um IAM sem onda Q patológica. Nesse segundo caso, pode haver uma depressão do segmento ST ou ainda, em torno de 60% dos casos, uma inversão

da onda T. Entre 5% e 10% desses casos o ECG pode permanecer inalterado em relação a exames anteriores. Ademais, um IAMCSST pode evoluir para um IAMSSST e vice versa.

Ainda segundo Sukienik (2015), para os casos de IAMCSST em que não há a intervenção para restauração da circulação, a evolução da região infartada ocorre em uma sequência de quatro fases, estando elas apresentadas na Figura 2.6:

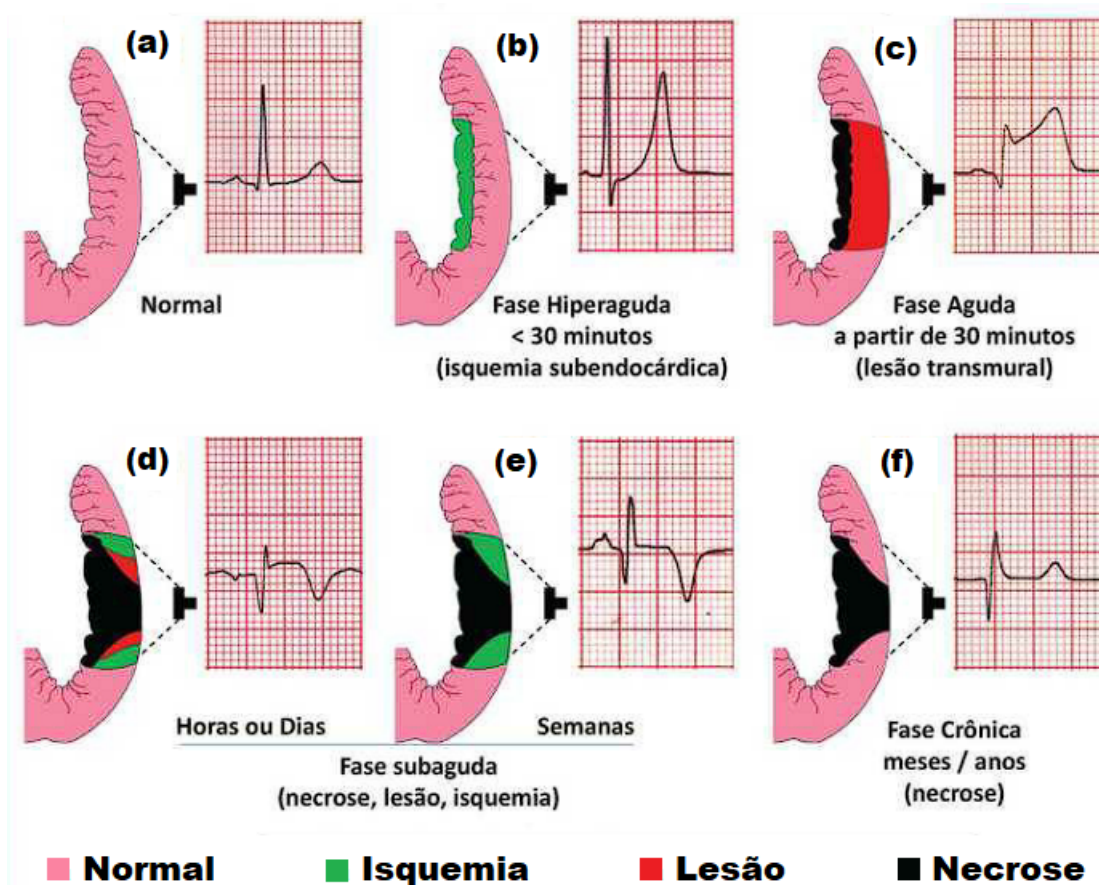
- ▣ Fase hiperaguda: ocorre por 15 a 30 minutos após a obstrução. A onda T é elevada, simétrica e possui base larga. Por essa fase ser nos momentos iniciais e durar pouco tempo, ela é raramente observada em ECGs da prática clínica. Ela é ilustrada na Figura 2.6(b).
- ▣ Fase aguda: com a persistência da oclusão, ocorrem lesões ainda reversíveis, assim como o início de necrose, a qual resulta na aparição da onda Q patológica. É apresentada na Figura 2.6(c).
- ▣ Fase subaguda ou recente: inicia-se após horas ou dias a partir do início do IM, como mostrado na Figura 2.6(d), sendo caracterizada por ondas T negativas em derivações que possuíam elevação no segmento ST. A onda Q patológica torna-se mais aparente, aparecendo também ondas T negativas onde havia elevação do segmento ST. Após algum tempo, o supra desnível ST sofre uma redução até seu desaparecimento em até 4 semanas, como pode ser observado na Figura 2.6(e).
- ▣ Fase crônica ou antiga: apresentada na Figura 2.6(e). Nela predomina a necrose, a qual se reflete na onda Q anormal. Além disso, o segmento ST passa a estar na linha de base, podendo haver a normalização ou a permanência do estado anormal da onda T.

2.2.2 – Vetocardiograma

Uma forma alternativa para a representação das atividades elétricas cardíacas é por meio do VCG, o qual é composto por três derivações, chamadas de derivações bipolares. Elas representam a variação dos potenciais elétricos cardíacos com base nos três vetores ortogonais V_x , V_y e V_z . Cada par dessas derivações define um plano no espaço-tempo que corresponde aos planos cardíacos, ou seja: aos planos sagital, frontal e transversal (Yang *et al.*, 2013).

Além disso, as três derivações em conjunto permitem que o ciclo cardíaco seja visto de forma tridimensional, como apresentado na Figura 2.7, sendo esse sinal o VCG, mostrado na Figura 2.7. Nele são apresentados *loops* que correspondem às ondas que compõem um sinal de

Figura 2.6 – Fases de evolução de um Infarto Agudo do Miocárdio com Supra ST.



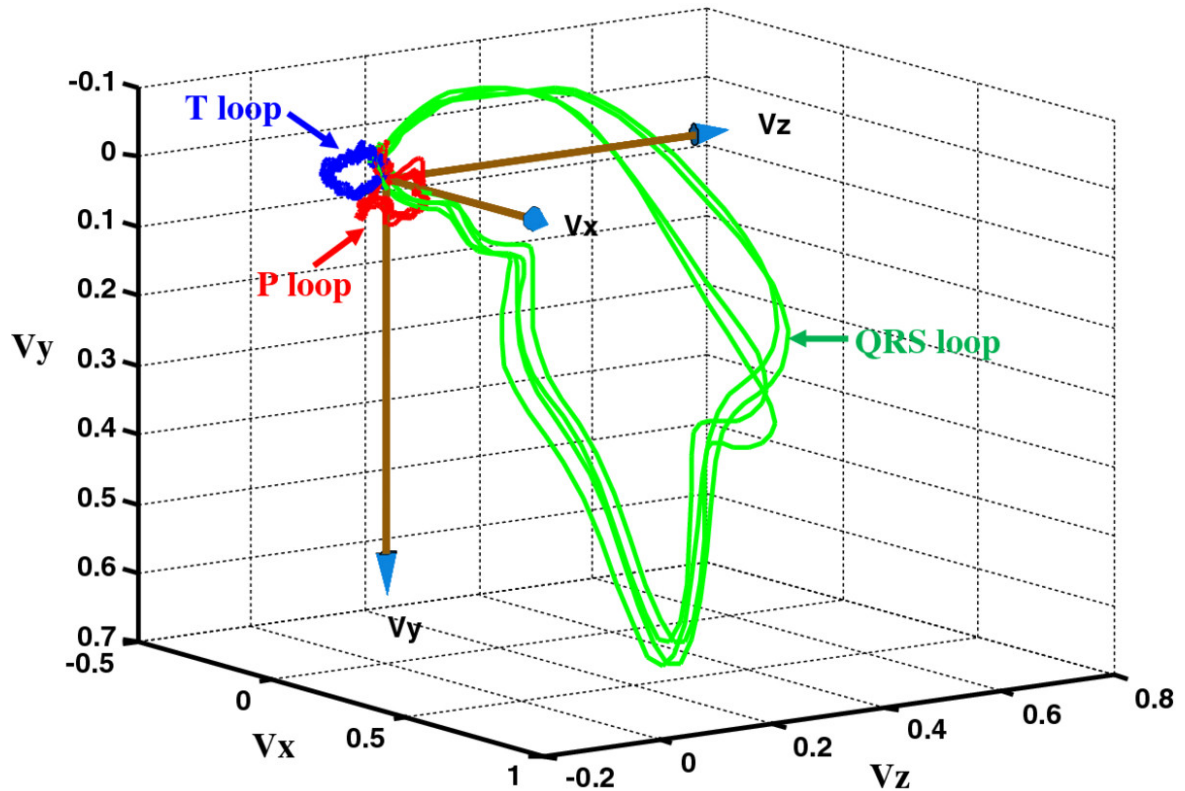
Adaptado de: (Sukienik, 2015).

ECG, onde o *loop P* representa a despolarização atrial, o *loop QRS* representa a despolarização ventricular e o *loop T* representa a sua repolarização. Considerando eles, é possível diagnosticar doenças com base em alterações morfológicas ou com base em suas áreas (Ginefra *et al.*, 2006).

O método considerado padrão ouro para sua obtenção é o VCG ortogonal corrigido de Frank (Frank, 1956), isso se deu, pois ele passou por aprimoramentos para a redução de erros resultantes das variações no posicionamento dos eletrodos e das diferenças anatômicas entre os pacientes. Mesmo sendo padrão ouro, seu uso não foi adotado, pois requeria eletrodos e um equipamento diferenciado. Além disso, devido aos arranjos requeridos para sua obtenção, os quais envolvem eletrodos nas costas e no pescoço, não era possível registrá-lo com o paciente na posição supina sem causar desconforto no paciente (Macfarlane & Edenbrandt, 1991).

Assim, para aproveitar as vantagens oferecidas pelo VCG de Frank, mas mitigando essas desvantagens, foram desenvolvidos métodos para a reconstrução desse VCG a partir do ECG de 12 derivações de forma digital. Tornando possível então que o VCG seja obtido apenas

Figura 2.7 – Exemplo de sinal de vetocardiograma. Em vermelho: *Loop P*; em verde: *Loop QRS*; em azul: *Loop T*.



Fonte: (Yang *et al.*, 2012).

com os equipamentos e técnicas para obtenção de um ECG, sem requerer treinamento adicional os profissionais que fazem o exame (Kors *et al.*, 1990).

Reconstrução do VCG a partir do ECG de 12 derivações

Três métodos se destacam para a reconstrução de forma digital do VCG a partir do ECG são eles: o método Quasi-Ortogonal de Kors; o método da Matriz Inversa de Dower e o método da Regressão de Kors, sendo eles descritos por Kors *et al.* (1990). Eles se baseiam na obtenção de uma matriz de regressão a qual é multiplicada com uma matriz com as derivações do ECG para ser obtida uma matriz com as três derivações vetocardiográficas. Esse processo é exemplificado pela Equação 2.1.

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} = \begin{bmatrix} a_{1,1} & \dots & a_{1,12} \\ a_{2,1} & \dots & a_{2,12} \\ a_{3,1} & \dots & a_{3,12} \end{bmatrix} \cdot \begin{bmatrix} DI \\ \vdots \\ V6 \end{bmatrix} \quad (2.1)$$

O método Quasi-Ortogonal de Kors é o que possui a premissa mais simples: as três derivações eletrocardiográficas que mais se aproximam das derivações vetocardiográficas são multiplicadas por um fator de escalonamento, de forma que as derivações de ECG DII, V2 e V6 levam à reconstrução das derivações V_x , V_y e V_z , respectivamente (Kors *et al.*, 1990). Considerando isso e a matriz de coeficientes para a multiplicação do ECG, apenas os coeficientes relacionados à multiplicação dessas derivações não serão zerados, enquanto os demais corresponderão ao fator de escalonamento.

O método da Matriz Inversa de Dower foi baseado em um método elaborado por Dower *et al.* (1980) para a obtenção das 12 derivações de ECG a partir das três derivações do vetocardiograma de Frank. Posteriormente, Edenbrandt & Pahlm (1988) utilizou a inversa da matriz com 24 coeficientes para se obter as 8 derivações independentes de ECG (sem as derivações DIII, aVR, aVL, aVF), de forma que se utilizam 8 das 12 derivações para a reconstrução do VCG.

Já o método da Regressão de Kors consiste no aprendizado supervisionado de uma matriz de regressão com base em ECGs e VCGs capturados simultaneamente para reduzir a soma dos erros ao quadrado entre os valores das derivações de Frank e as derivações reconstruídas. Para isso foram usadas 8 das 12 derivações de ECG, são elas: DI, DII, V1, V2, V3, V4, V5 e V6.

Após uma comparação dos parâmetros quantitativos, como ângulos de pico e de ângulos médios entre *loops* QRS e T, de sinais de VCGs reconstruídos com os obtidos pelo método de Frank, o método da Regressão de Kors resultou em uma menor diferença nesses parâmetros em relação aos demais métodos descritos acima (Cortez & Schlegel, 2010).

Por fim, a Tabela 2.1 apresenta os coeficientes para as matrizes de reconstrução para cada um dos métodos de reconstrução do VCG citados anteriormente, de acordo com Cortez *et al.* (2014), destacando que foram utilizadas as 12 derivações de ECG para a reconstrução pelo método de Dower.

Tabela 2.1 – Coeficientes para as derivações do ECG utilizadas reconstrução do VCG.

Método	Derivação do VCG	I	II	III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6
Regressão Linear de Kors	X	0.380	-0.070	0.000	0.000	0.000	0.000	-0.130	0.050	-0.010	0.140	0.060	0.540
	Y	-0.070	0.930	0.000	0.000	0.000	0.000	0.060	-0.020	-0.050	0.060	-0.170	0.130
	Z	0.110	-0.230	0.000	0.000	0.000	0.000	-0.430	-0.060	-0.140	-0.200	-0.110	0.310
Kors Quasi-Ortogonal	X	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
	Y	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Z	0.000	0.000	0.000	0.000	0.000	0.000	-0.500	0.000	0.000	0.000	0.000	0.000
Matriz Inversa de Dower	X	0.632	0.235	-0.397	-0.434	0.515	-0.081	-0.515	0.044	0.882	1.212	1.125	0.831
	Y	-0.235	1.066	1.301	-0.415	-0.768	1.184	0.157	0.164	0.098	0.127	0.127	0.076
	Z	0.059	-0.132	-0.191	0.037	0.125	-0.162	-0.917	-0.139	-1.277	-0.601	-0.086	0.230

2.3 – Métodos de análise não linear

2.3.1 – Reconstrução do espaço de fase

Um sistema, de acordo com a definição dada por Ljung & Glad (1994) é, de forma abrangente, um componente ou um conjunto deles cujas propriedades e comportamento se deseja estudar, seja por meio de experimentos ou com o uso de modelos. Dentre os modelos matemáticos elaborados estão os modelos dinâmicos, nos quais as variáveis se relacionam entre si, mas também dependem de seus valores anteriores. Em razão disso, Ljung & Glad (1994) define um sistema dinâmico como um sistema que pode ser descrito por meio de equações diferenciais ou de diferenças.

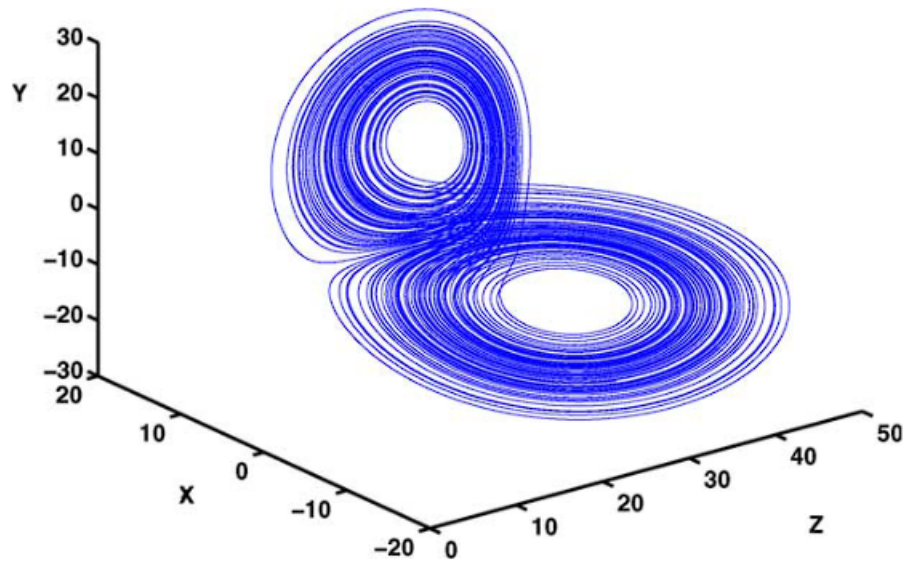
A dimensão de um sistema dinâmico se dá conforme a quantidade de variáveis de estado, ou seja, as variáveis que permitem a descrição do sistema de acordo com sua variação no tempo, para sistemas descritos por equações diferenciais ordinárias. Uma forma de investigação do comportamento desses sistemas se dá por meio do seu Espaço de Fase (Strogatz, 1994; Vieira, 2014).

O Espaço de Fase de um sistema, é um espaço que contém todos os seus estados possíveis, de forma que cada ponto corresponde a uma solução dele de acordo com sua evolução no tempo, ou seja, um estado; e cada eixo representa uma variável de estado. O conjunto de pontos forma a trajetória do espaço de fase, ou atrator. Por meio do comportamento do atrator pode-se avaliar o comportamento do sistema, como o quão caótica é a sua dinâmica, além de ser possível evidenciar comportamentos determinísticos em sistemas com dinâmicas aparentemente caóticas.

Um exemplo de atrator é o Atrator de Lorenz, apresentado na Figura 2.8. Ele foi descrito pela primeira vez em 1963, tendo sido obtido a partir da análise de um sistema descrito por três equações diferenciais de um modelo para convecção atmosférica e se destacou por ser altamente sensível às suas condições iniciais, uma característica do caos, tendo, ao mesmo tempo, uma regularidade (ordem) na forma que as curvas se dispõem (Stewart, 2000).

Considerando o sinal de uma derivação cardíaca como a saída de um sistema, o espaço de fase se mostrou uma ferramenta útil para a discriminação de sinais eletrocardiográficos de pacientes patológicos, por esses possuírem, de acordo com Govindan *et al.* (1998), comportamento mais caótico que sinais de pacientes saudáveis.

Figura 2.8 – Exemplo de atrator caótico tridimensional de Lorenz.



Fonte: Yang *et al.* (2006).

As equações diferenciais do sistema de geração e condução da atividade elétrica cardíaca, necessárias para a obtenção do EF pela sua forma usual, no entanto, não são conhecidas. Assim, são necessários métodos alternativos para a obtenção do EF para esses sinais. Um desses métodos é o Método do Tempo de Atraso, descrito por (Takens, 1981, 1983), o qual permite a reconstrução do espaço de fase m -dimensional original a partir de uma série temporal de uma única variável, como uma derivação de ECG ou VCG. Caso a dimensão do espaço de fase reconstruído seja alta o suficiente, a Reconstrução do Espaço de Fase (REF) é semelhante ao EF original (Zimmerman *et al.*, 2003).

Essa REF composta por T estados em uma série temporal de única variável com N pontos é representada por:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_T \end{bmatrix} = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & \vdots & \vdots \\ x_T & x_{T+\tau} & \cdots & x_N \end{bmatrix} \quad (2.2)$$

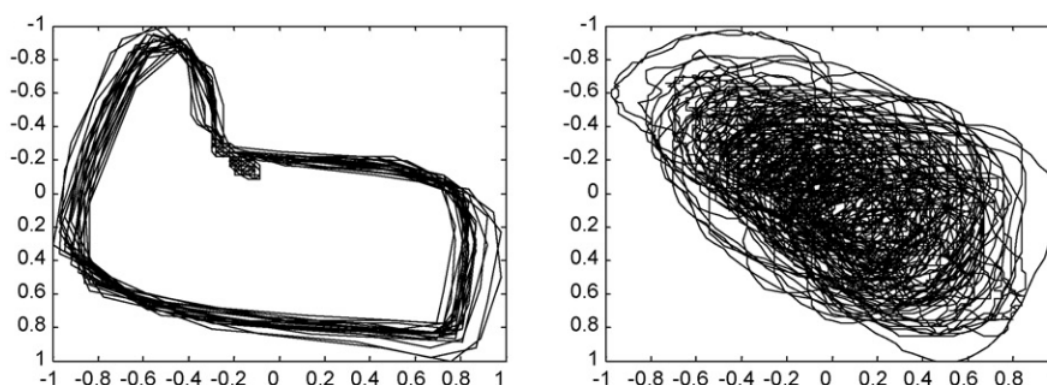
Na Equação 2.2, X é o espaço de fase reconstruído para m dimensões, x_n é o n -ésimo ponto da série temporal de uma variável, N é a quantidade de amostras da série, τ é o tempo de atraso utilizado para a reconstrução e m é a dimensão do espaço de fase, sendo geradas $m - 1$ versões atrasadas em um tempo múltiplo inteiro de τ (Zimmerman *et al.*, 2003).

Com base na Equação 2.2, a quantidade de estados, ou seja, a quantidade de pontos no espaço de fase (T) é limitada por N e pela dimensão da REF, isso é, sua dimensão de imersão. Sendo ela dada pela Equação 2.3:

$$T = N - (m - 1) \cdot \tau. \quad (2.3)$$

Para um REF bidimensional (dimensão de imersão igual a 2) de uma série temporal obtido pelo método do tempo do atraso, é possível obter a imagem dessa reconstrução realizando o traçado do sinal original $x(t)$ no eixo das abcissas e a sua versão atrasada no eixo das ordenadas, como feito por Amann *et al.* (2006). Nesse caso, foi obtida a REF bidimensional para uma derivação de ECG. Para esses sinais, caso eles sejam provenientes de um paciente saudável, eles possuem comportamento menos caótico, o que é representado no REF pela concentração do traçado em uma região definida. Já caso o paciente possua uma patologia, um comportamento mais caótico é observado, o que leva o traçado a ocupar uma região maior e mais irregular, como mostrado na Figura 2.9.

Figura 2.9 – Esquerda: Espaço de fase reconstruído para um sinal de ECG de um paciente saudável. Direita: Espaço de fase reconstruído para um sinal de ECG de paciente com fibrilação ventricular.



Fonte: Adaptada de Roopaei *et al.* (2010).

Determinação do Tempo de Atraso

O valor de tempo de atraso escolhido para ser utilizado na Equação 2.2 tem influência na correlação entre as amostras do sinal original e das suas $m - 1$ versões atrasadas, devendo o tempo de atraso ser escolhido de forma que a informação redundante entre o sinal original e sua versão atrasada seja minimizada (Vieira, 2014). Uma forma de fazer isso é com base na Informação Mútua (MI, do inglês, *Mutual Information*), um conceito da Teoria da Informação.

Para se obter a MI é necessário se conhecer a Quantidade de Informação de um Evento. Esse conceito é baseado na ideia de que um evento q com baixa probabilidade de ocorrência, $p(q)$, possui mais informação associada, $h(q)$, dada em *bits* e apresentada na Equação 2.4, que um evento que possui uma maior probabilidade (MacKay & Mac Kay, 2003).

$$h(q) = \log_2 \frac{1}{P(q)} \quad (2.4)$$

Considerando então uma variável aleatória Q , ao calcular a Quantidade de Informação para a distribuição de probabilidade dos eventos dela, se tem a sua entropia ($H(Q)$), dada pela Equação 2.5:

$$H(Q) \equiv \sum_{q \in A_Q} P(q) \log \frac{1}{P(q)}. \quad (2.5)$$

Onde é convencionado que $0 \log \frac{1}{0} \equiv 0$ caso $P(q) = 0$, já que $\lim_{\theta \rightarrow 0^+} \theta \log \frac{1}{\theta} = 0$.

Além da entropia, se tem a entropia condicional, a qual, para uma variável Q dado R , é a entropia da distribuição $P(q|r = g_k)$ e se dá pela Equação 2.6:

$$H(Q|r = g_k) \equiv \sum_{q \in A_Q} P(q|r = g_k) \log \frac{1}{P(q|r = g_k)}. \quad (2.6)$$

Pela média considerando r da entropia condicional de Q dado r , é obtida a entropia condicional de Q dado R . Sendo ela a medida da incerteza média em Q quando se conhece r , a expressão para sua expressão é apresentada na Equação 2.7 (MacKay & Mac Kay, 2003).

$$H(Q|R) \equiv \sum_{qr \in A_Q A_R} P(q, r) \log \frac{1}{P(q|r)} \quad (2.7)$$

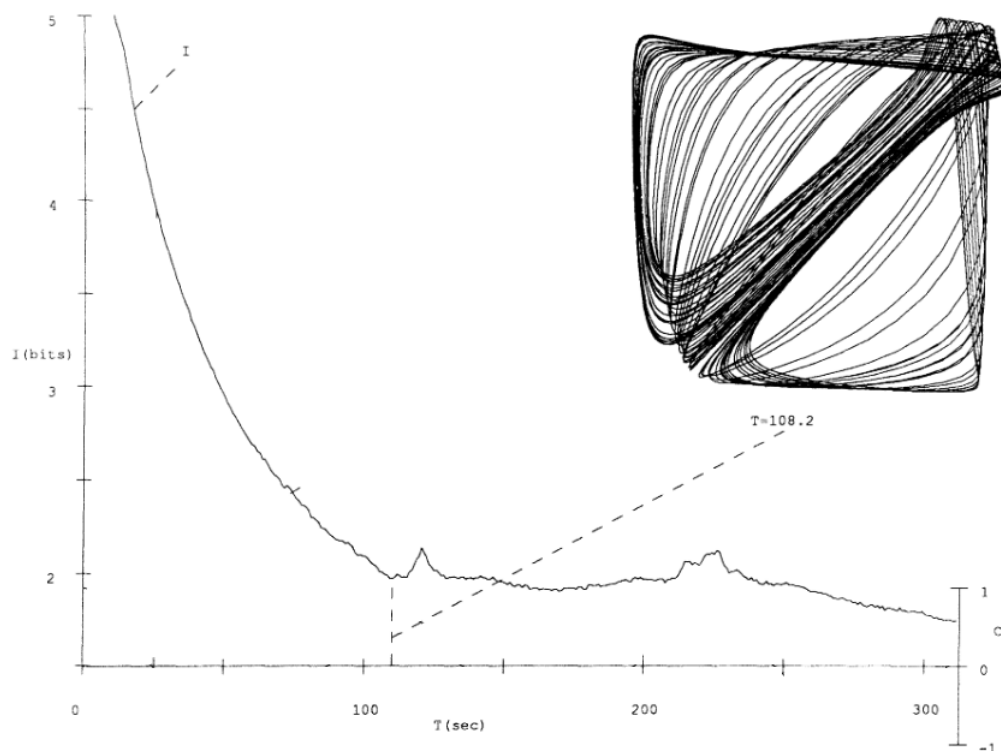
Com base na equações 2.5, 2.6 e 2.7, a MI (I) que representa a quantidade de informação que se tem de Q quando a variável R é conhecida é dada por:

$$I(Q; R) \equiv H(Q) - H(Q|R) = H(R) - H(R|Q). \quad (2.8)$$

Com base na medida da informação mútua entre a série temporal correspondente ao sinal original e sua versão atrasada por um tempo τ , é possível se obter uma estimativa da curva de informação mútua a partir do cálculo dessa medida para diferentes tempos de atrasos. Com isso, o tempo de atraso ideal é determinado como sendo o tempo para o qual ocorre o primeiro

mínimo local na curva de informação mútua, como exemplificado na Figura 2.10 (Fraser & Swinney, 1986).

Figura 2.10 – Curva de informação mútua e reconstrução do espaço de fases de uma reação química oscilante com τ obtido a partir do primeiro mínimo local dessa curva.



Fonte: Adaptada de Fraser & Swinney (1986).

Alguns valores de tempo de atraso já foram analisados em pesquisas com sinais de ECG. Para o trabalho de Roopaei *et al.* (2010), eles estão entre 0,01s e 0,08s, onde foram testadas combinações para a comparação entre dois espaços de fase bidimensionais com o objetivo de detectar arritmias, sendo escolhidos valores nessa faixa para a comparação. Já para o trabalho de Amann *et al.* (2006) foi definido um valor de 0,5s a fim de se definir uma métrica quantitativa para atuar como limiar para a classificação de sinais com fibrilação ventricular. Por fim, Costa *et al.* (2018) utilizou valores de tempo de atraso baseados nos valores utilizados por Roopaei *et al.* (2010) para a identificação de infarto do miocárdio a partir de derivações vetocardiográficas.

Determinação da Dimensão de Imersão

De acordo com a teoria exposta em Takens (1981) e Takens (1983), a dimensão de imersão para a REF, dada por m na Equação 2.2, deve satisfazer a relação $m \geq 2d + 1$. Na qual d é a dimensão fractal do espaço de fase, nem sempre conhecida para determinados sistemas.

Uma forma de contornar esse problema se dá pela aplicação do método dos Falsos Vizinhos mais Próximos (FNN, do inglês, *False Nearest Neighbors*). Esse método parte da premissa de que pontos no REF que são vizinhos verdadeiros em um EF continuam próximos mesmo que a dimensão de reconstrução aumente, enquanto os falsos vizinhos se afastam, já que aparentavam estar próximos apenas pelo fato do atrator ser uma projeção de uma dimensão maior em uma menor (Chen *et al.*, 2021). Assim, o valor da dimensão de imersão é incrementado de forma gradativa e é avaliado o percentual de pontos no REF obtido que se afastaram ou a dimensão de imersão é dada pelo menor valor de m para o qual se obtém o menor percentual de falsos vizinhos (Vieira, 2014).

2.3.2 – Gráficos de recorrência

A visualização de espaços de fases é limitada pela visão do olho humano, isso é, em três dimensões. Esses espaços, no entanto, podem na realidade ter 150 dimensões, como os apresentados no trabalho de Chen *et al.* (2021), de forma que métodos alternativos para a análise qualitativa desses EF dentro dos limites do olho humano foram desenvolvidos.

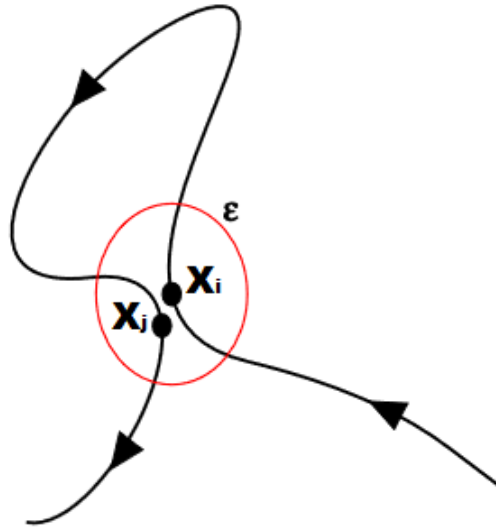
Um desses métodos é conhecido como gráfico de recorrência (RP, do inglês *Recurrence Plot*). Os RPs tiveram seu conceito apresentado por Eckmann *et al.* (1995) e permitem que se tenha, em duas dimensões, uma ideia do comportamento de um EF com base na sua recursão, isso é, onde suas trajetórias retornam a um ponto anterior.

Considerando dois estados X_i e X_j de um espaço de fase com T estados e dimensão de imersão m , há recorrência (R) entre esses dois estados caso a distância entre eles por alguma norma, geralmente a norma euclidiana, for inferior a um limiar chamado de raio de vizinhança (ϵ). Para representar a presença ou ausência de recorrência, é utilizada a função Degrau unitário(Θ), de forma que caso a distância seja inferior ao limiar, será retornado 1, representando uma recorrência. Toda essa lógica é dada pela Equação 2.9, podendo ser ilustrada conforme apresentado na Figura 2.11 (Vieira, 2014; Mathunjwa *et al.*, 2022).

$$R_{i,j}^{m,\epsilon} = \Theta(\epsilon - \|X_i - X_j\|), \quad x_i \in R^m; \quad i, j = 1, 2, \dots, T. \quad (2.9)$$

Tendo em conta que o RP é uma matriz quadrada de ordem T onde T é a quantidade de estados do EF, com o cálculo das normas entre os elementos é obtida uma matriz chamada de Matriz Distância, a qual pode sofrer escalonamento. O raio de vizinhança associado com a função degrau atuam como uma forma de binarização, de forma que, caso haja recorrência, um

Figura 2.11 – Representação de dois estados, X_i e X_j , considerados recorrentes por estarem dentro do raio de vizinhança ϵ em uma trajetória de espaço de fase arbitrário.



Fonte: Adaptada de Vieira (2014).

ponto preto será marcado no RP para o índice correspondente a essa recorrência, e caso contrário, haverá um ponto branco (Vieira, 2014). Um exemplo de gráfico de recorrência gerado dessa forma é mostrado na Figura 2.12.

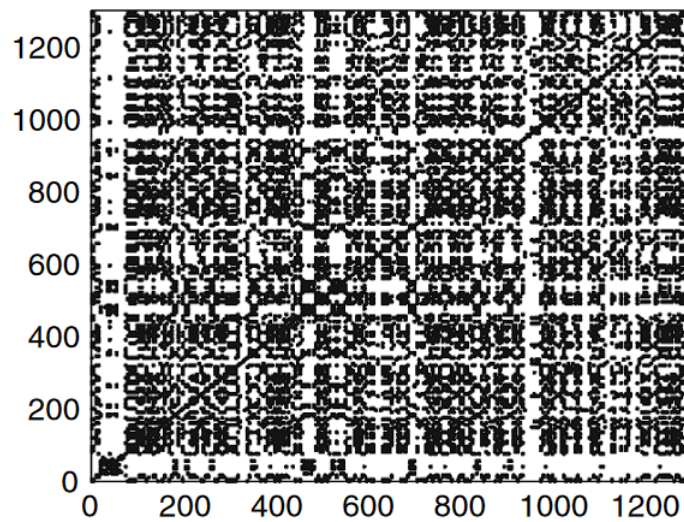
A determinação de recorrências é dependente do raio de vizinhança escolhido. Caso ele seja um valor alto, várias falsas recorrências podem ser detectadas e, caso seu valor seja baixo, poucas ocorrências serão detectadas. Alguns métodos, no entanto, são sugeridos para a obtenção desse valor, sendo eles enumerados por Vieira (2014).

O primeiro deles consiste em analisar o máximo diâmetro do espaço de fase e utilizar um valor de raio de vizinhança correspondente a um pequeno percentual desse diâmetro. O segundo consiste em avaliar a taxa de recorrência e utilizar um valor de ϵ que garanta um pequeno valor mínimo de taxa de recorrência. Por fim, também pode se analisar o ruído em séries temporais, quando ele é conhecido, e determinar o raio de vizinhança como ao menos cinco vezes o valor do desvio padrão desse ruído (Vieira, 2014).

Uma definição alternativa àquela dada em Equação 2.9 utiliza apenas as normas entre os estados, sendo as distâncias representadas por meio de cores, sem a necessidade de limiarização pela função degrau considerando o raio de vizinhança (Mathunjwa *et al.*, 2022). Sendo isso mostrado pela Equação 2.10.

$$R_{i,j}^m = ||X_i - X_j||, \quad x_i \in R^m; \quad i, j = 1, 2, \dots, T. \quad (2.10)$$

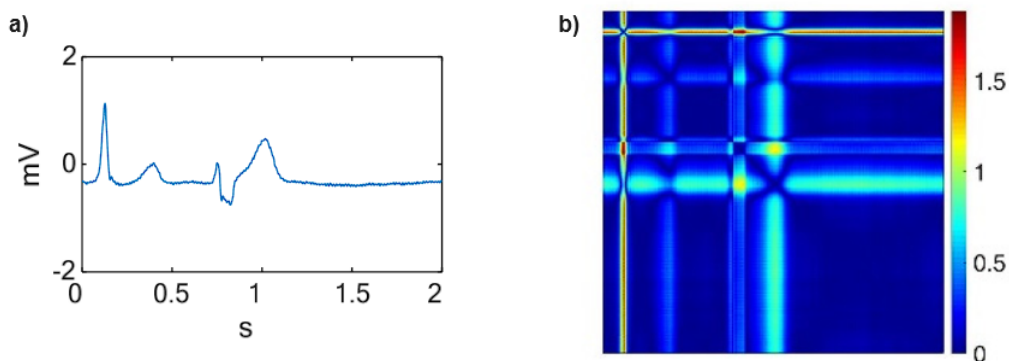
Figura 2.12 – Exemplo de gráfico de recorrência de uma série temporal de variabilidade de frequência cardíaca para um sinal de um recém-nascido prematuro, para o qual se teve REF obtido para $m = 3$, $\tau = 3$ e $\epsilon = 8$.



Fonte: Adaptada de Santos *et al.* (2014).

A matriz distância obtida por esse método é convertida em uma matriz composta por elementos com três valores entre 0 e 1, os quais representam os valores dos três canais RGB, sendo finalmente obtida a versão modificada do gráfico de recorrência (Mathunjwa *et al.*, 2022). Um exemplo dessa versão para um sinal de ECG considerado normal é apresentado na Figura 2.13.

Figura 2.13 – a) Sinal de ECG saudável; b) Gráfico de recorrência para esse sinal obtido sem limiarização.



Fonte: Adaptada de Mathunjwa *et al.* (2022).

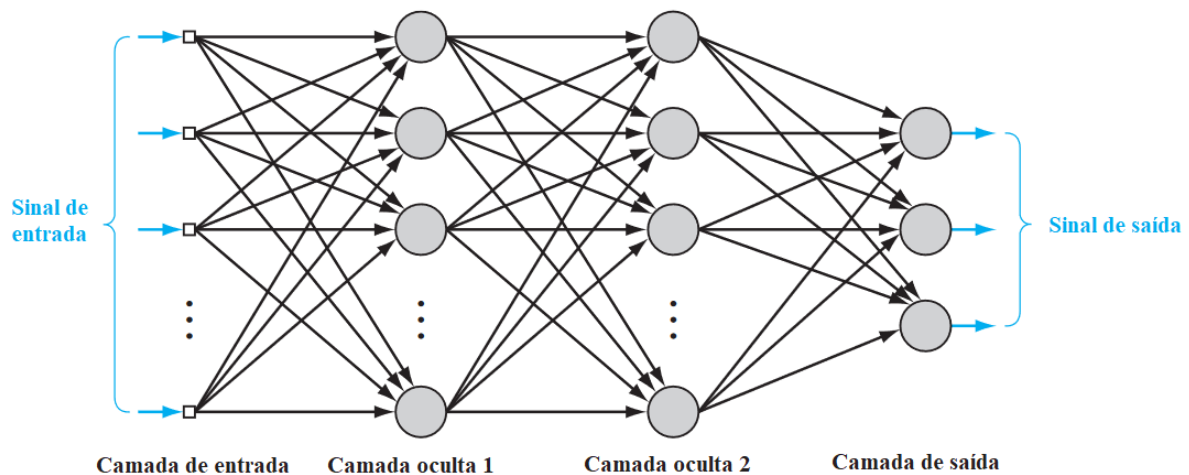
2.4 – Redes Neurais

De acordo com Haykin *et al.* (2009), uma Rede Neural Artificial (RNA) é uma máquina criada a fim de modelar como o cérebro realiza uma tarefa ou função de interesse. Uma classe de RNA são as redes *Perceptron* Multicamadas, em inglês *Multilayer Perceptron* (MLP), as quais a

partir de um conjunto de entradas e de um conjunto de unidades unitárias altamente conectadas entre si chamadas de *perceptron*, baseadas em neurônios, retornam um vetor de saída com base nos resultados das operações realizadas por eles. Cada conexão possui um peso próprio e os pesos de cada conexão são modificados durante a etapa de treinamento com base na comparação da saída obtida com a saída desejada, a qual já é conhecida, e na propagação desse sinal de erro. Nesse caso, é dito que o aprendizado é supervisionado.

Uma rede MLP possui a topologia exemplificada na Figura 2.14 e que se dá em três camadas ou mais camadas. Nela, os neurônios de uma camada, representados por círculos, se conectam com todos da camada seguinte, onde cada seta representa essa conexão. A primeira camada é a camada de entrada, na qual os *perceptrons* recebem um vetor contendo as diferentes informações utilizadas como entrada pela rede para determinação da sua saída, apresentada como um vetor na camada de saída. Entre elas estão as camadas ocultas, as quais fazem o mapeamento de forma não linear das entradas para saídas por meio de uma função criada pela superposição de diversas funções de transferência não lineares mais simples (Gardner & Dorling, 1998).

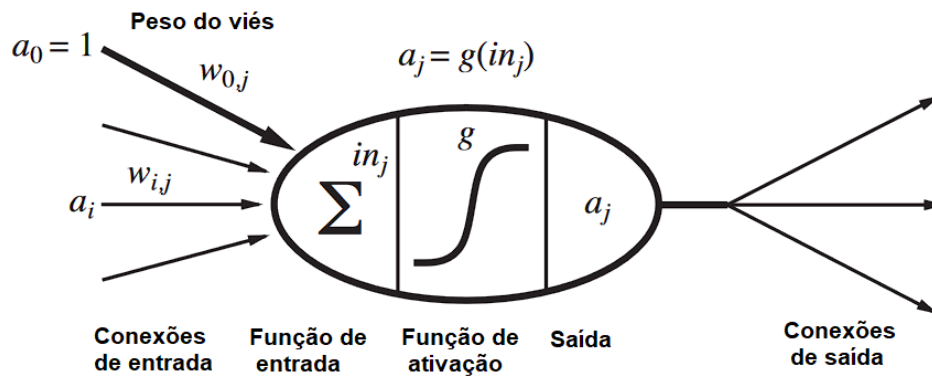
Figura 2.14 – Ilustração de rede neural do tipo MLP com duas camadas ocultas.



Fonte: Adaptado de (Haykin *et al.*, 2009).

Cada neurônio recebe como entrada a soma ponderada das entradas da rede ou das saídas dos neurônios anteriores de acordo com o peso associado para cada conexão. Além disso, o resultado dessa soma pode ser mapeado para um intervalo pela função de ativação do neurônio, o que oferece um nível maior de abstração para as entradas do neurônio, além de, como será mostrado mais adiante, influenciar na complexidade da etapa de treinamento. Cada neurônio também possui associado a si um viés (ou *bias*), isso é, um valor de erro sistemático adicionado à soma ponderada. Todo esse sistema é apresentado na Figura 2.15

Figura 2.15 – Representação de um neurônio considerando seu viés, suas conexões de entrada, função de entrada, de ativação, sua saída e suas conexões de saída.



Fonte: Adaptado de (Haykin *et al.*, 2009).

Dentre as funções de ativação para as camadas ocultas se destaca a função ReLU (no inglês: Rectified Linear Unit, Unidade Linear Retificada), dada pela Equação 2.11 (Haykin *et al.*, 2009). Isso se dá pelo fato de suas derivadas, as quais devem ser utilizadas durante o treinamento, terem uma computação simples e terem valores maiores que as de uma função sigmoide (Ramachandran *et al.*, 2017).

$$f(x) = \max(0, x) \quad (2.11)$$

No contexto das camadas de saída em RNAs utilizadas para classificação binária, destaca-se a função sigmoide, a qual também pode ser usada nas camadas ocultas, dada pela Equação 2.12 (Haykin *et al.*, 2009).

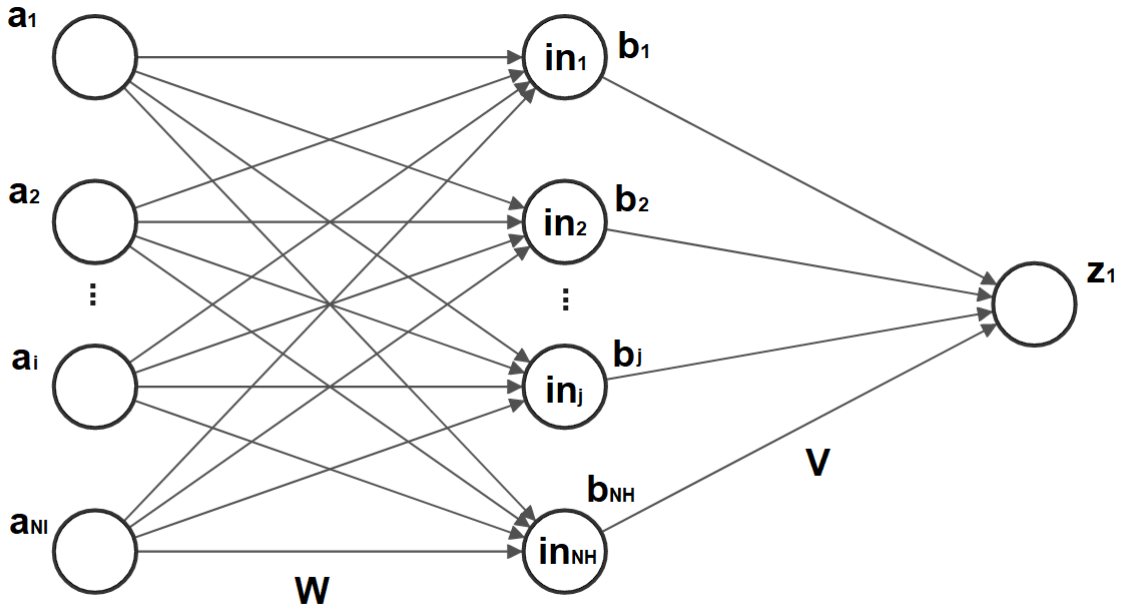
$$f(x) = \frac{1}{1 + \exp(-sx)}, \quad s = 1 \quad (2.12)$$

Os pesos, inicializados com pequenos valores aleatórios, são ajustados durante essa etapa de treinamento por meio do algoritmo de retro-propagação ou, no inglês *back-propagation*, sendo essa uma das abordagens mais utilizadas para esse ajuste.

Para explicar esse algoritmo, é considerada uma MLP, mostrada Figura 2.16, focada em regressão, de forma que ela possuirá apenas um neurônio na camada de saída; a função de ativação na sua camada oculta será a função sigmoide e sua camada de saída não possuirá função de ativação, isso é, já retornando o resultado da soma ponderada nas entradas dos neurônios. Além disso, para simplificação da análise, a rede não possuirá vieses e conterá apenas uma

camada oculta. Dois sinais percorrem essa rede: os sinais de função, que correspondem à propagação direta, e os sinais de erro, os quais correspondem a retro-propagação.

Figura 2.16 – Rede neural MLP com entradas a_i , pesos da camada de entrada para a camada oculta em um vetor W , neurônios na camada oculta b_j , pesos da camada oculta para a camada de saída em um vetor V e neurônio z_1 na camada de saída.



Fonte: Elaborada pelo autor.

No sentido direto, cada neurônio tem em sua entrada o resultado da soma ponderada dos elementos da camada anterior com relação aos pesos de conexão. Para a camada oculta isso é dado pela Equação 2.13:

$$in_j = \sum_{i=0}^{NI} w_{ji} \cdot a_i, \quad j = 0, 1, \dots, NH. \quad (2.13)$$

Sendo a função de ativação sigmoide, a saída do neurônio b_j é,

$$b_j = \frac{1}{1 + \exp(-s \cdot in_j)}, \quad s = 1. \quad (2.14)$$

A derivada de b_j com relação a in_j é dada por $b_j(1 - b_j)$. Da mesma forma, as saídas b_j junto com as conexões v_{kj} formam a soma ponderada que atua como entrada no neurônio da camada de saída, sendo essa soma representada pela Equação 2.15 e correspondente, nesse caso em que não há função de ativação, às saídas da rede.

$$z_1 = \sum_{j=0}^{NH} v_{kj} \cdot b_j, \quad k = 1. \quad (2.15)$$

No aprendizado supervisionado as saídas obtidas em z_n para cada entrada, são comparadas com as respectivas saídas esperadas, chamadas aqui de c_n , sendo obtidos erros e_n para cada um deles, como apresentado na Equação 2.16:

$$e_n = c_n - z_n \quad (2.16)$$

Os erros obtidos para cada entrada inserida na rede são então quantificados em uma função custo, a qual será posteriormente minimizada. Uma função custo simples utilizada em redes neurais para regressão é a MSE (*Mean Squared Error*, em português: Erro Quadrático Médio) (Zhang *et al.*, 2019), dada para esse problema pela Equação 2.17:

$$MSE = E(\text{época}) = \frac{1}{2 \cdot N} \sum_{n=1}^N e_n^2, \quad (2.17)$$

para o qual n representa uma determinada entrada apresentada à rede, época à iteração atual do algoritmo de ajuste de pesos. O fator 2 é adicionado a essa equação a fim de simplificar o resultado final da expressão para ajuste de pesos da rede.

Uma forma de minimizar a função custo, melhorando o desempenho da RNA para o conjunto de treinamento, se dá por meio do método do gradiente descendente. Ele consiste em realizar o ajuste dos pesos das conexões entre neurônios de acordo com a direção oposta à do vetor gradiente da função custo com relação aos pesos das conexões. Para a RNA de exemplo esse ajuste é dado pela Equação 2.18 e Equação 2.19:

$$v_{1j}(\text{época} + 1) = v_{1j}(\text{época}) - \eta \frac{\partial E(\text{época})}{\partial v_{1j}} \quad (2.18)$$

$$w_{ji}(\text{época} + 1) = w_{ji}(n) - \eta \frac{\partial E(\text{época})}{\partial w_{ji}} \quad (2.19)$$

O η é um fator chamado de Taxa de Aprendizagem, o qual determina a intensidade do ajuste dos pesos. Por meio do cálculo das derivadas parciais com base na regra da cadeia são obtidas as expressões para ajustes dos pesos dadas pelas equações 2.18 e 2.19, que passam a ser as expressões das equações 2.20 e 2.21, respectivamente:

$$v_{1j}(\text{época} + 1) = v_{1j}(\text{época}) + \eta e_n(\text{época}) b_j(\text{época}) \quad (2.20)$$

$$w_{ji}(\text{época} + 1) = w_{ji}(\text{época}) + \eta e_n(\text{época}) v_{1j}(\text{época}) b_j(\text{época}) (1 - (b_j(\text{época}))) a_i(\text{época}) \quad (2.21)$$

Nessa etapa, o sinal de erro é obtido na camada de saída e propagado dela para a camada de entrada, como mostrado nas equações 2.20 e 2.21. Ele ocorre a partir de um número definido de iterações, chamadas de épocas, ou até o erro atingir um valor suficientemente pequeno (Gardner & Dorling, 1998; Haykin *et al.*, 2009).

É importante destacar que, caso a rede apresentada possuísse mais camadas, a lógica apresentada acima para determinação dos novos pesos também seria válida, bastando apenas determinar a relação entre a função custo e os pesos da camada a ser ajustada pela regra da cadeia, a qual também pode ser usada para a determinação do ajuste dos vieses.

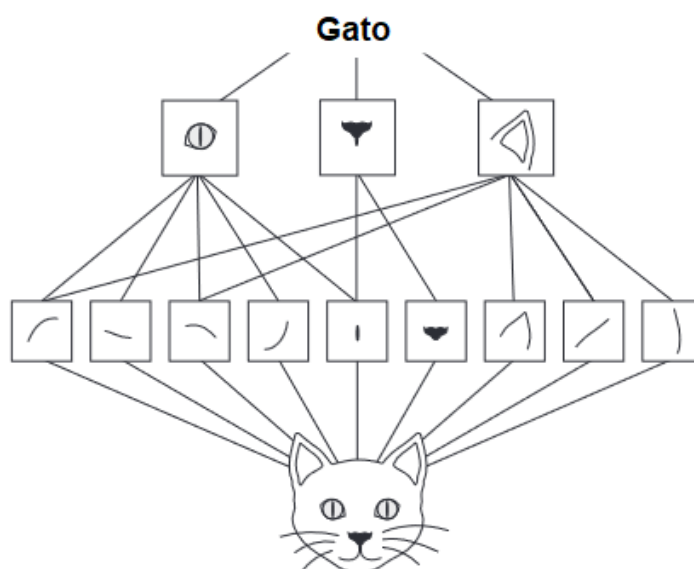
Inicialmente, as redes MLP utilizadas possuíam poucas camadas ocultas, sendo chamadas de redes rasas, e limitadas a uma ou duas camadas ocultas, em função do poder computacional e métodos de treinamento limitados. Com o aumento do primeiro e melhorias nos métodos de treinamento que ocorreram na primeira década do século XXI, o treinamento de redes com múltiplas camadas, chamadas de Redes Neurais Profundas, tornou-se possível (Aggarwal *et al.*, 2018). Um dos tipos mais populares de redes neurais profundas é Rede Neural Convolutiva (CNN, do inglês *Convolutional Neural Network*), a qual se destaca no reconhecimento de padrões em imagens (Albawi *et al.*, 2017).

As CNN são uma arquitetura específica de RNAs que são especializadas em processar dados que possuem uma topologia em grade como uma série temporal, valores obtidos em um intervalo regular distribuídos no tempo, ou imagens, valores de *pixel* distribuídos em duas dimensões (Goodfellow *et al.*, 2016).

Seu uso é motivado devido a sua característica de pouca variância à distorções como escalonamento na forma ou translação (Haykin, 2008; O'Shea & Nash, 2015). Dessa forma, padrões, como rostos, por exemplo, não precisam ser espacialmente dependentes, podendo ser identificados independentemente da sua posição em uma imagem (Albawi *et al.*, 2017). Além disso, elas são capazes de hierarquizar os padrões espaciais de forma que bordas, por exemplo,

se combinam em camadas mais profundas para a construção e reconhecimento de formas mais complexas, como um rosto (Chollet, 2021). Um exemplo disso é apresentado na Figura 2.17.

Figura 2.17 – Hierarquização dos padrões espaciais a partir da imagem de um gato para a categorização dessa entrada.



Fonte: Adaptada de Chollet (2021).

Ademais, outra motivação para seu uso para a classificação de imagens no lugar de RNAs do tipo MLP comuns se dá pelas limitações das RNAs quanto a complexidade computacional, especialmente devido à quantidade de parâmetros na primeira camada. Isso é exemplificado para uma rede composta por um neurônio para a classificação de imagens RGB com resolução 32×32 , onde essa rede não é viável para classificação de imagens, já seriam necessárias $32 \times 32 \times 3$ conexões. Além do aumento do custo computacional de processamento e tempo, o aumento da quantidade de conexões e neurônios para lidar com essa quantidade de parâmetros de entrada diminui a capacidade de generalização do modelo (O'Shea & Nash, 2015).

Algumas técnicas foram então desenvolvidas para aumentar a eficiência e reduzir o número de pesos diferentes necessários, sendo elas utilizadas antes da etapa de classificação, funcionando como uma etapa de extração de características ao mesmo tempo em que reduzem a resolução das imagens. Isso se dá por meio de neurônios que recebem como entradas apenas as saídas de uma região da camada anterior, o que representa a extração de características locais. Além disso, a posição relativa entre as características possui maior importância que a posição absoluta delas (Haykin, 2008).

Considerando uma entrada bidimensional $U(i, j)$, a qual pode ser uma imagem, e um núcleo (*kernel*, em inglês) K também em duas dimensões, a convolução é definida pela Equação 2.22:

$$S(i, j) = (U * K)(i, j) = \sum_m \sum_n U(m, n)K(i - m, j - n), \quad (2.22)$$

Em que o resultado dessa convolução é chamado de *feature map*. Considerando a propriedade comutativa dessa operação, a Equação 2.22 pode ser reescrita na forma da Equação 2.23:

$$S(i, j) = (K * U)(i, j) = \sum_m \sum_n U(i - m, j - n)K(m, n). \quad (2.23)$$

Em implementações de CNNs, é utilizada uma função semelhante à apresentada na Equação 2.23, mas sem o espelhamento do núcleo K , a qual é chamada de *cross-relation*. Além disso, a convolução discreta pode ser vista como uma multiplicação de matrizes, na qual uma matriz possui vários elementos semelhantes (Goodfellow *et al.*, 2016).

De acordo com Goodfellow *et al.* (2016), a convolução engloba três ideias úteis para um sistema de aprendizado de máquina.

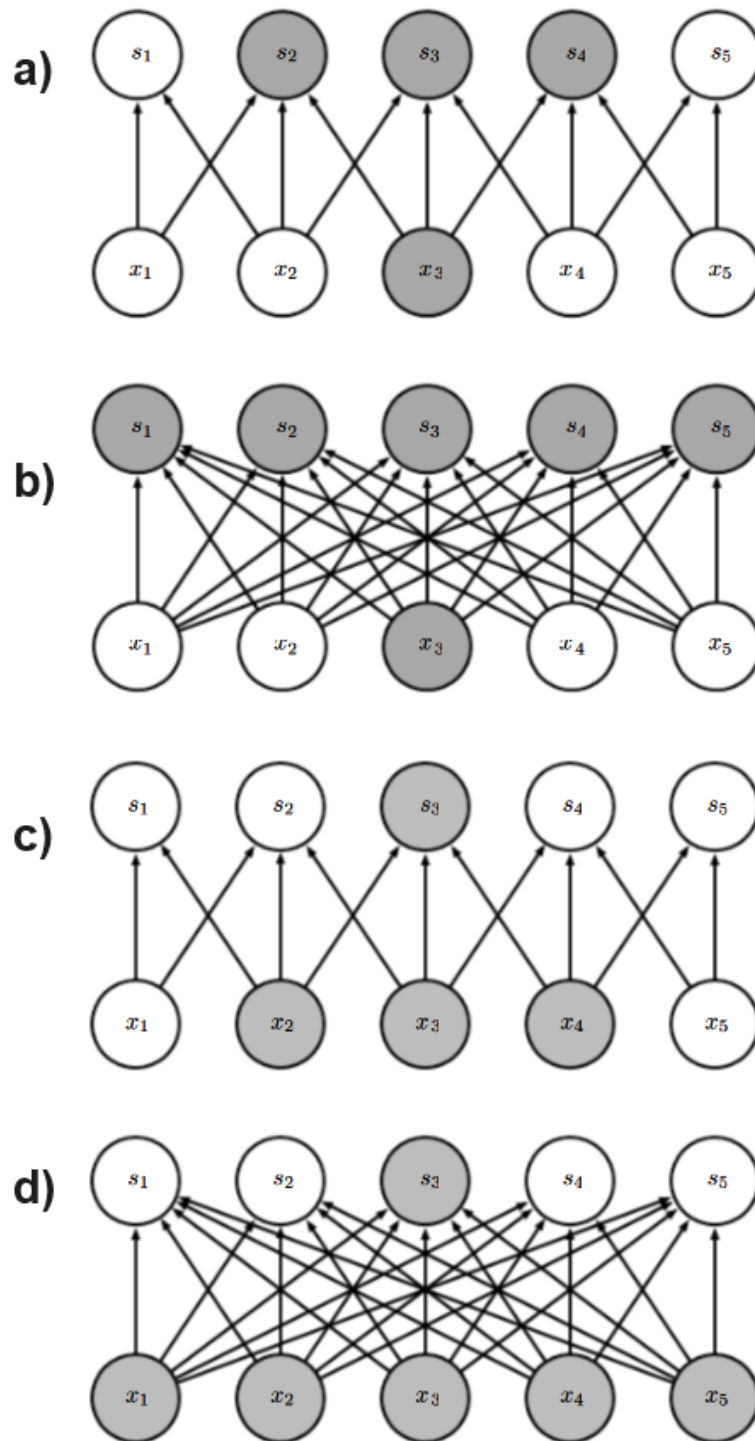
A primeira delas são as interações esparsas. Essa característica, ao contrário de uma rede neural tradicional para a qual há uma multiplicação de matrizes de parâmetros que relacionam as interações de cada entrada com cada saída de um perceptron, se dá devido ao fato do núcleo ser menor que a entrada. Com isso, ao processar uma imagem, por exemplo, é feita uma seleção de pequenas características como bordas utilizando núcleos da ordem de centenas de *pixels*, de forma que menos parâmetros são armazenados em comparação com a quantidade que seria armazenada em uma RNA tradicional, reduzindo a quantidade de operações para computar a saída (Goodfellow *et al.*, 2016).

Essa diferença entre a quantidade de interações é ilustrada na Figura 2.18.

Além disso, as unidades em camadas mais profundas podem interagir com uma grande porção das entradas, de forma que várias interações mais complexas são criadas a partir de interações mais simples e esparsas (Goodfellow *et al.*, 2016), como ilustrado na Figura 2.19.

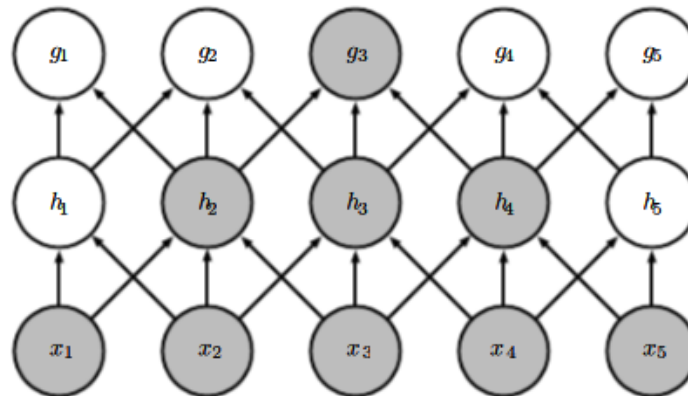
A segunda ideia útil implementada em uma CNN é o compartilhamento de parâmetros, o qual consiste em utilizar um mesmo parâmetro para mais de uma função em um modelo. Isso significa que o valor de um peso aplicado a região uma entrada está atrelado ao peso aplicado em outra região dessa mesma entrada, de forma que ao invés de otimizar vários conjuntos de

Figura 2.18 – Comparação da quantidade de interações entre neurônios com o uso ou não de conexões esparsas: a) Conexão esparsa, na qual uma unidade de entrada, x_3 , interage apenas com três unidades de saída (s_2, s_3, s_4) em razão do seu núcleo ter lado 3; b) Saídas computadas a partir de uma multiplicação de matrizes, de forma que todas as saídas são afetadas por uma única entrada x_3 ; c) Conexão esparsa na qual uma unidade de saída (s_3) é afetada por três unidades de entrada (x_2, x_3, x_4), as quais são chamadas de campo receptivo; d) Quando a saída s_3 é computada a partir de multiplicação de matrizes todas as entradas afetam essa saída.



Fonte: Adaptada de Goodfellow *et al.* (2016).

Figura 2.19 – O campo receptivo de uma unidade g_3 em uma camada mais profunda é composto por um número maior de unidades que um campo em uma camada mais rasa, como a unidade h_2 .

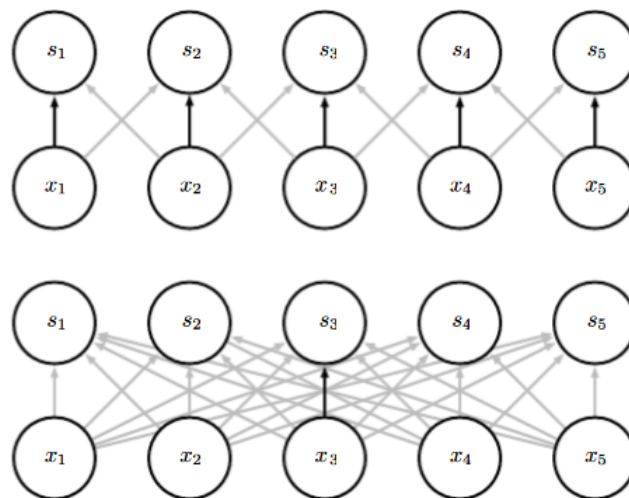


Fonte: Goodfellow *et al.* (2016).

parâmetros para cada região, é possível otimizar apenas um, reduzindo assim os requisitos de memória (Goodfellow *et al.*, 2016).

Uma comparação entre dois modelos, no qual um usa o compartilhamento de parâmetros e outro não, é mostrada na Figura 2.20. Esse compartilhamento de pesos leva à terceira ideia útil em uma CNN: a equivariância em relação à translação, ou seja, uma mesma saída é obtida caso seja aplicado um deslocamento na entrada seguido de uma convolução ou caso a convolução seja aplicada à entrada e seu resultado seja deslocado.

Figura 2.20 – Acima: as setas pretas indicam o uso do elemento central de um núcleo com 3 elementos em todas as regiões de entrada. Abaixo: O uso desse elemento central em um modelo de RNA tradicional sem compartilhamento de parâmetros implica em um único uso desse elemento.



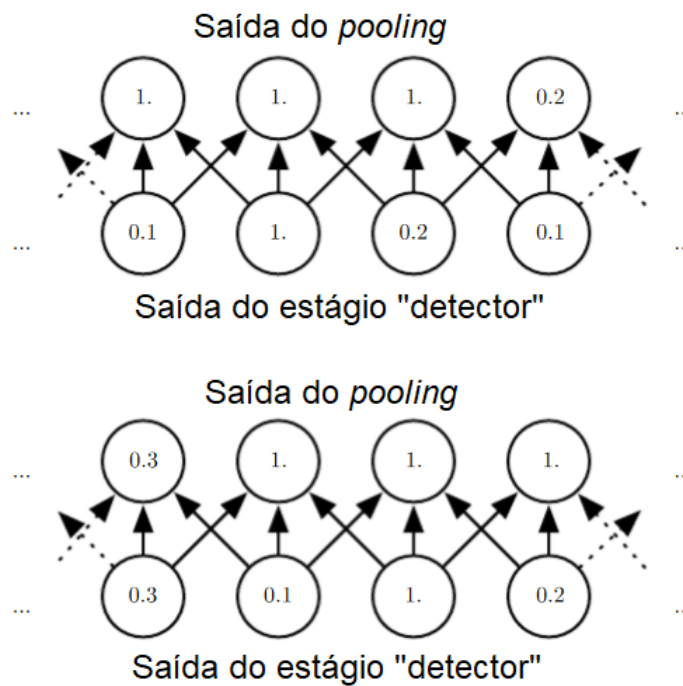
Fonte: Goodfellow *et al.* (2016).

Após o primeiro estágio, no qual várias convoluções são aplicadas em paralelo a fim de se obter uma resposta linear, há um estágio para o qual cada uma dessas respostas passa por uma

função de ativação com objetivo de adicionar não linearidades, sendo esse estágio chamado de "detector".

Em um último estágio, é utilizada uma função de *pooling*, a qual é responsável por substituir um conjunto de saídas de uma região por uma sumarização estatística delas, como o valor máximo ou médio desse conjunto. Com isso, a representação se torna menos sensível a translações em menor escala na entrada, como pode ser visto na Figura 2.21, além de atuar como uma forma de subamostragem, já uma região de *pooling* com largura de K *pixels* oferece em torno de K vezes menos entradas para processar na camada seguinte, reduzindo mais ainda a quantidade de memória necessária para armazenamento dos parâmetros.

Figura 2.21 – Representação da redução da sensibilidade às pequenas translações pelo *pooling*. Acima: Representação do *pooling* com largura de três *pixels* por valor máximo das saídas do estágio detector. Abaixo: Mesma representação considerando um deslocamento para a direita em um *pixel*, de forma que os valores de todos os neurônios do estágio detector mudaram, mas apenas metade dos valores da saída de *pooling* mudaram.

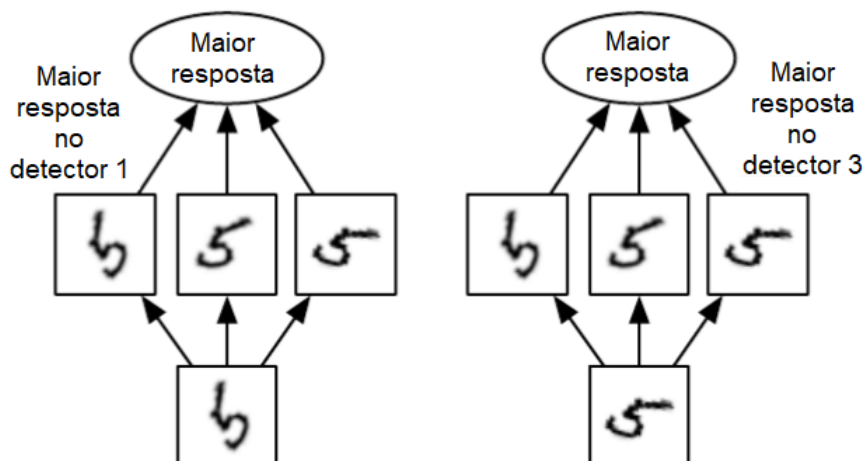


Fonte: Adaptada de Goodfellow *et al.* (2016).

Ademais, quando uma unidade do estágio de *pooling* atua sobre o resultado de convoluções com parâmetros diferentes, há a possibilidade do aprendizado da invariância de outras transformações na entrada, como rotações, por exemplo (Goodfellow *et al.*, 2016). Esse efeito é apresentado na Figura 2.22.

Como apresentado na Figura 2.23, os *feature maps* são então utilizados como entrada para uma região chamada de camada completamente conectada, a qual é responsável por classificar o sinal inserido como entrada da CNN (O'Shea & Nash, 2015; Vargas *et al.*, 2016).

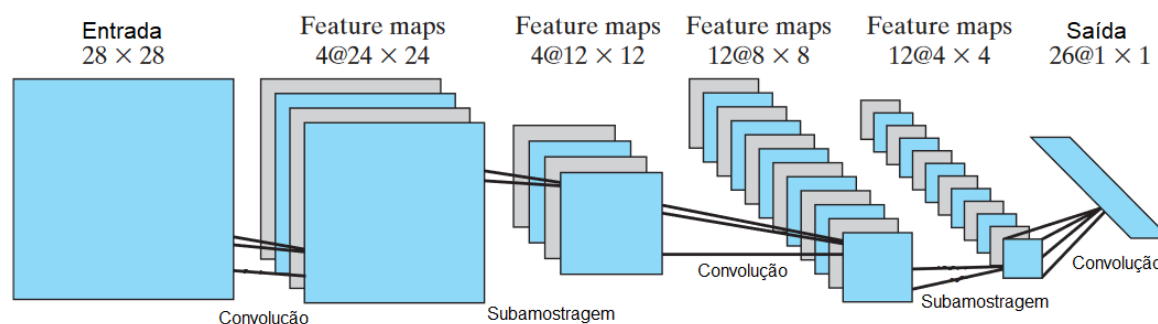
Figura 2.22 – Dois exemplos diferentes de três filtros treinados para a detecção do dígito 5, onde cada um é treinado para lidar com uma variação de rotação desse dígito. Tanto para o exemplo da esquerda quanto para o da direita, haverá a ativação na unidade de detector correspondente, sendo essa ativação refletida na unidade de *pooling* máximo.



Fonte: Adaptada de Goodfellow *et al.* (2016).

Com relação ao seu treinamento, apesar de diversos métodos terem sido propostos, o principal método utilizado hoje é a *back-propagation* (Goodfellow *et al.*, 2016).

Figura 2.23 – Extração de características e redução de dimensionalidade por meio das camadas convolucionais e de *pooling*, respectivamente.



Fonte: Adaptada de Haykin (2008).

Foi observado que as redes neurais convolucionais treinadas com imagens possuem a tendência de terem suas primeiras camadas com características mais gerais, comuns independentemente do tipo de imagem utilizado, podendo ser aplicadas em diferentes conjuntos de dados e tarefas, enquanto as últimas camadas são específicas, sendo essa transição da especialidade das camadas gradual à medida que se aproxima da última camada (Yosinski *et al.*, 2014).

Com base nisso, o aprendizado de um modelo, chamado de modelo de base, treinado com um conjunto de dados maior pode ser reaproveitado por meio do uso das camadas de

extração de características que são mais gerais, ou seja, sua transferência, para outro modelo. Por isso, esse processo é chamado de *transfer learning*, ou transferência de aprendizado (Yosinski *et al.*, 2014).

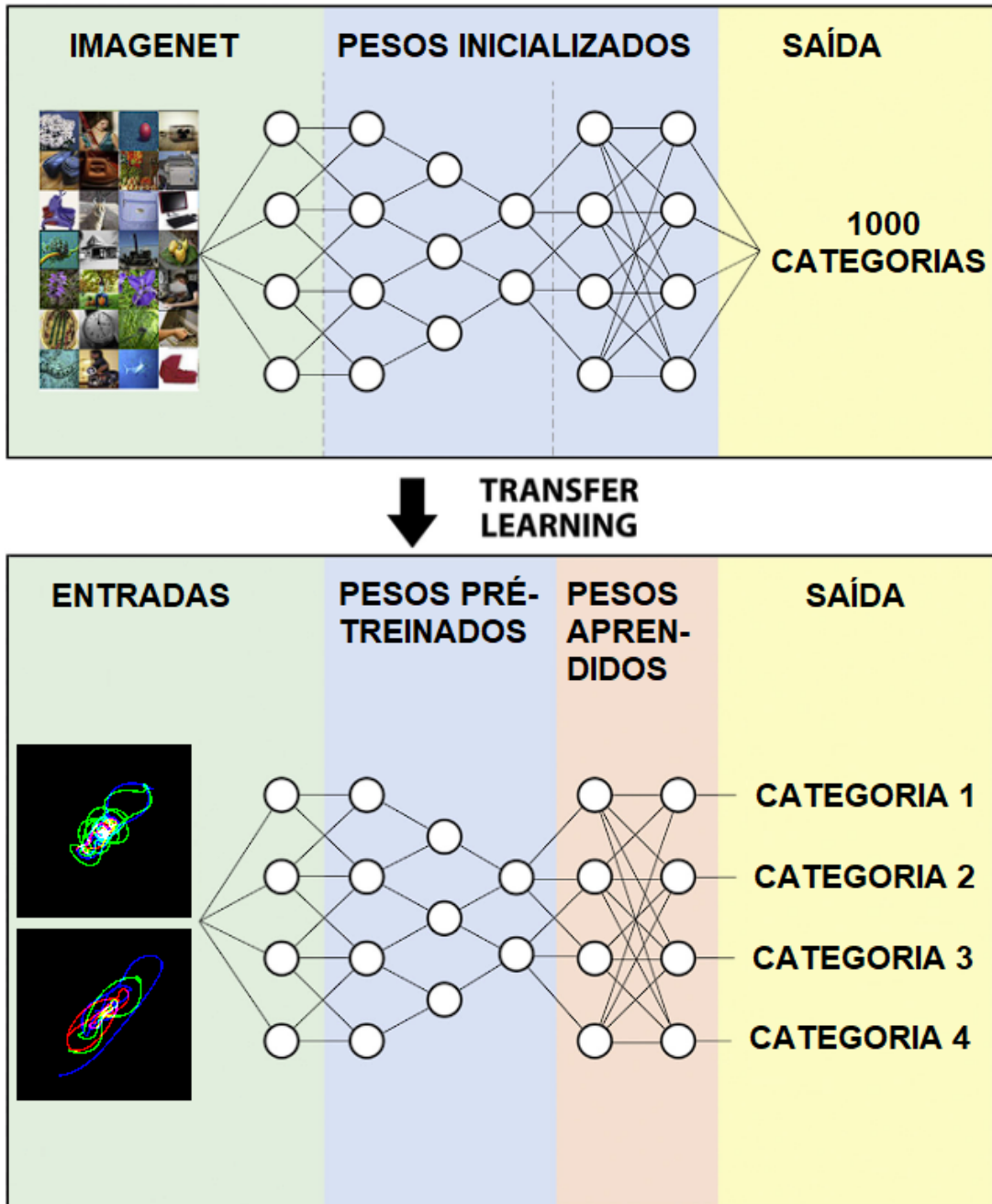
Entre os modelos-base usualmente disponíveis estão a MobileNet-V2, ResNet-50, ResNet-101 e DenseNet-201 (MathWorks, 2022; Chollet *et al.*, 2015), as quais são treinadas com a base de dados ImageNet, que contém mais de 14 milhões de imagens classificadas em 21841 categorias chamadas de *synset* (conjunto de sinônimos). O treinamento desses modelos, no entanto, é feito considerando 1000 categorias de objetos correspondentes a 1000 *synsets* (Nielsen, 2018; Lab, 2020).

O processo de *transfer learning*, apresentado na Figura 2.24 é particularmente útil para treinar CNNs com um conjunto de dados menor, pois os pesos do modelo de base são usados como ponto de partida no lugar de pesos aleatórios, o que reduz o tempo e o custo computacional do treinamento (Ilakiyaselvan *et al.*, 2020; Kermany *et al.*, 2018; Yosinski *et al.*, 2014).

Por fim, Yosinski *et al.* (2014) mostrou que a inicialização com pesos transferidos leva a um melhor desempenho que uma inicialização com pesos aleatórios, além de aumentar a capacidade de generalização após o ajuste para um novo conjunto de dados.

Baseado na viabilidade do uso de imagens de espaço de fases e gráficos de recorrência para a classificação de cardiopatias e das fragilidades nos métodos citadas na Seção 1.3, deve-se avaliar se o emprego de CNNs para classificação é uma alternativa viável para a identificação de infarto a partir de imagens de espaço de fase reconstruído ou gráficos de recorrência, além de determinar qual desses dois tipos de imagem leva a um melhor desempenho nessa identificação.

Figura 2.24 – Representação do *transfer learning* na qual os pesos obtidos para o treinamento de uma rede com o banco de dados Imagenet são reaproveitados para uma rede com imagens de outras categorias, sendo apenas as camadas finais retreinadas.



Fonte: Adaptado de (Kermany *et al.*, 2018).

DESCRIÇÃO DA PROPOSTA

A fim de fazer uso dos métodos de análise não linear, mas evitando a dependência da extração manual associada com aplicação de métodos estatísticos para a seleção dos parâmetros mais relevantes, o presente trabalho tem como proposta utilizar dois tipos de entradas: espaços de fase bidimensionais e gráficos de recorrência. Ambos podem ser tratados como imagem, provenientes de sinais vetocardiográficos como entradas para redes neurais convolucionais, o que permite automatizar a classificação binária desses sinais em "Normal" ou "Infarto do miocárdio". Além disso, este trabalho tem como proposta avaliar qual desses dois tipos de entrada e qual arquitetura de CNN entre o conjunto de arquiteturas testadas que leva a uma melhor desempenho na classificação.

3.1 – Aplicabilidade

Uma possível aplicação para o presente trabalho é sua utilização como parte de um sistema de apoio ao diagnóstico, o qual é composto por quatro processos principais: o pré-processamento dos sinais de ECG, o qual consiste em: uma etapa de filtragem, conversão de ECG para VCG, seleção de diferentes períodos do sinal de VCG, reconstrução do espaço de fase bidimensional e gráfico de recorrência para as derivações de VCG; a extração de características, a seleção das características com maior significância e a sua classificação. Com sistemas desse tipo, busca-se contribuir com uma segunda opinião aos profissionais de diagnóstico, o que pode levar à redução do tempo para detecção de doenças e em possibilitar um tratamento mais rápido para elas. Ademais, esses sistemas também podem ser utilizados como uma forma de treinamento da interpretação de sinais eletrocardiográficos para novos profissionais (Hagiwara *et al.*, 2018).

Um sistema com aplicação semelhante a deste trabalho é proposto por Mathunjwa *et al.* (2022). Nele, um dispositivo de captura de ECG envia os dados para um telefone celular, o qual possui um aplicativo responsável por realizar a segmentação dos sinais, reconstrução dos

gráficos de recorrência e inferência desses sinais por meio de duas CNNs treinadas. A adaptação desse sistema para uma aplicação final envolvendo o presente trabalho é dependente, no entanto, da capacidade de memória e processamento do dispositivo a ser utilizado.

Outra contribuição deste trabalho é a determinação do tipo de entrada mais adequado para a identificação do infarto do miocárdio, entre o espaço de fase reconstruído com imersão 2 e o gráfico de recorrência de um espaço de fase m -dimensional. Com isso, é possível determinar se há a possibilidade de redução do processamento necessário para a geração das entradas das CNNs, caso o primeiro tipo de entrada possua um desempenho melhor ou com custo benefício melhor em relação ao segundo tipo de entrada. Por fim, a arquitetura de CNN mais adequada para essa tarefa, entre as avaliadas, será determinada.

3.2 – Metodologia

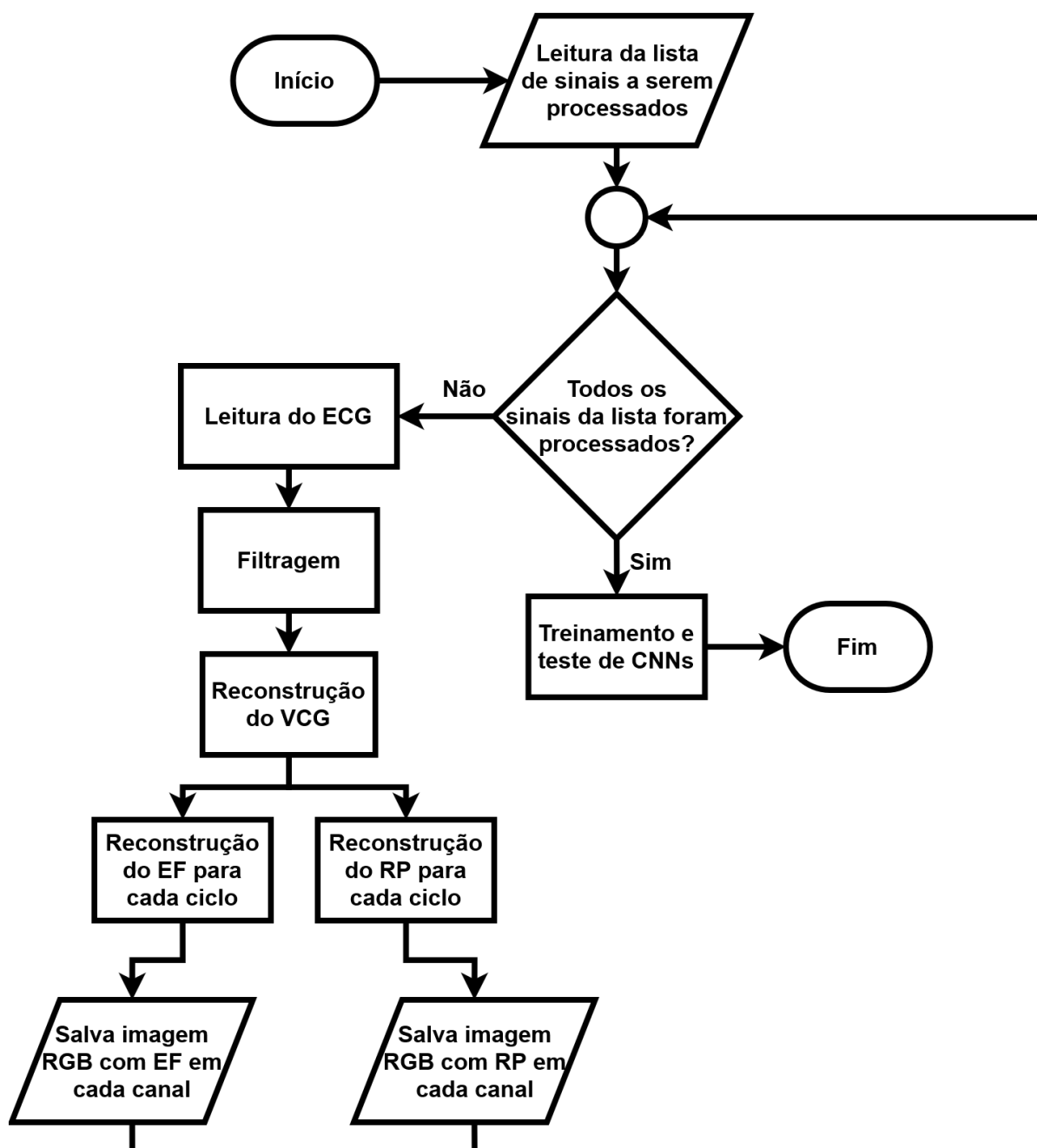
Para o desenvolvimento da proposta, três etapas foram consideradas, são elas: a aquisição dos sinais de ECG de uma base de dados, o seu processamento, no qual serão reconstruídas as derivações vetocardiográficas e reconstruídos seus espaços de fase bidimensionais e gráficos de recorrência; e o treinamento e teste das redes neurais convolucionais para a classificação desses sinais.

Essas três etapas são detalhadas nas subseções a seguir. Nelas são apresentados o banco de dados utilizado e o critério de utilização de seus sinais de ECG, o processamento realizado, incluindo a definição dos parâmetros como tempo de atraso e parâmetros de geração das imagens desses espaços de fase e gráficos de recorrência utilizados e, por fim, as arquiteturas de CNNs utilizadas, a forma de treinamento empregada e a forma de definição para o melhor desempenho entre as arquiteturas. Um fluxograma dessas atividades é apresentado na Figura 3.1, sendo suas etapas descritas nas subseções a seguir.

3.2.1 – Banco de Dados

A base de dados utilizada nesse trabalho é a base pública apresentada pelo PTB (Instituto Nacional de Metrologia da Alemanha) em 2020, chamada de PTB-XL (Wagner *et al.*, 2020). Esse base é a sucessora da base PTB (Bousseljot *et al.*, 1995), a qual já é amplamente utilizada em trabalhos relacionados ao processamento de sinais cardíacos e identificação de infarto, como nos trabalhos de Costa *et al.* (2018) e Sridhar *et al.* (2021).

Figura 3.1 – Fluxograma das atividades propostas para o desenvolvimento do presente trabalho.



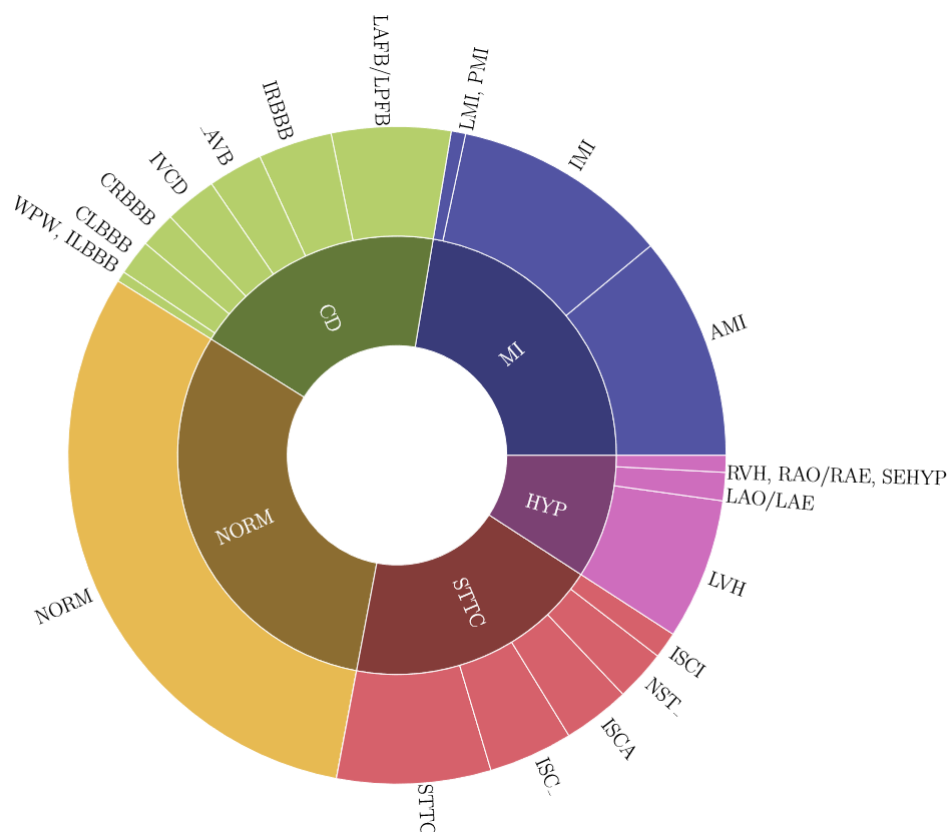
Fonte: Autoral.

A vantagem oferecida pelo PTB-XL em relação ao PTB é a sua maior quantidade de registros, havendo 21837 deles, os quais foram coletados de 18885 pacientes (9820 homens e 9065 mulheres) com idades entre 0 e 95 anos, sendo a mediana de idade 62 anos e a amplitude interquartil de 22 anos. A coleta dos registros ocorreu entre outubro de 1989 e junho de 1996.

Cada registro corresponde a um ECG de 12 derivações com duração de 10 segundos, estando associado a ele uma ou mais superclasses (Normal, Infarto do Miocárdio, Distúrbio de Condução, Hipertrofia e Alteração do Segmento ST/T), havendo para cada superclasse uma ou

mais subclasses associadas a ele. A relação do número total de ECGs no PTB-XL para cada superclasse e subclasse está disponível na Tabela 3.1, podendo também ser vista na Figura 3.2. Além disso, metadados clínicos como idade, sexo, altura, peso, data de aquisição do sinal, entre outras também podem estar disponíveis (Wagner *et al.*, 2020).

Figura 3.2 – Divisão dos sinais do PTB-XL em suas superclasses e subclasses.



Fonte: (Strodthoff *et al.*, 2020).

O processo de aquisição desses sinais se deu pela realização dos exames para a captura do ECG de 12 derivações com um eletrodo de referência no braço direito, onde a frequência de amostragem original era de 400 Hz, sendo os metadados correspondentes ao sinal cadastrados por um profissional de enfermagem. Em seguida, cada registro recebeu uma anotação, a qual pode ter sido gerada por um cardiologista que interpretou o sinal (67,13% dos casos) ou por interpretação automática (31,2% dos casos). Para esses casos, 4,45% foram validados por um cardiologista humano e 26,75% possuem informação incompleta quanto a essa validação. Os demais 1,67% não receberam qualquer anotação inicial. As anotações são então convertidas para um conjunto padronizado de informações acerca do sinal, as quais incluem a probabilidade de cada diagnóstico dado. Em seguida, um cardiologista independente cria uma anotação ou edita a anotação feita anteriormente, sendo eventuais divergências apresentadas nessa anotação. Por

Tabela 3.1 – Relação do número de ECGs por superclasses e subclasses.

*Os acrônimos são baseados nos nomes das superclasses/subclasses na língua inglesa.

		Acrônimo (*)	Descrição	Número de ECGs
Superclasse		NORM	ECG Normal	9528
		CD	Distúrbio de condução	5772
		MI	Infarto do Miocárdio	6866
		HYP	Hipertrofia	2819
		STTC	Alterações ST/T	5788
Subclasse	NORM	NORM	ECG Normal	9528
	CD	LAFB/LPFB	Bloqueio Fascicular Anterior/Posterior Esquerdo	1803
		IRBBB	Bloqueio de Ramo Direito Incompleto	1118
		ILBBB	Bloqueio de Ramo Esquerdo Incompleto	77
		CLBBB	Bloqueio de Ramo Esquerdo Completo	536
		CRBBB	Bloqueio de Ramo Direito Completo	542
		_AVB	Bloqueio Atrioventricular	827
		IVCB	Bloqueio Intraventricular Não Específico	789
		WPW	Síndrome de Wolff-Parkinson-White	80
	HYP	LVH	Hipertrofia do Ventrículo Esquerdo	2137
		RHV	Hipertrofia do Ventrículo Direito	126
		LAO/LAE	Sobrecarga/Aumento do Átrio Esquerdo	427
		RAO/RAE	Sobrecarga/Aumento do Átrio Direito	99
		SEHYP	Hipertrofia Septal	30
	MI	AMI	Infarto do Miocárdio Anterior	3387
		IMI	Infarto do Miocárdio Inferior	3281
		LMI	Infarto do Miocárdio Lateral	201
		PMI	Infarto do Miocárdio Posterior	17
	STTC	ISCA	Isquemia em Derivações Anteriores	1016
		ISCI	Isquemia em Derivações Inferiores	398
		ISC_	Isquemia Não Específica	1275
		STTC	Alterações ST-T	2329
		NST_	Alterações ST Não Específicas	770

fim, outra anotação, relativa às características do sinal, é feita manualmente por um especialista técnico, o qual pode reportar eventuais problemas na captura desses sinais (Wagner *et al.*, 2020).

Após sua aquisição, os arquivos contendo os sinais são convertidos do formato original, o qual é proprietário, para um formato digital com precisão de 16 bits e resolução de $1\mu\text{V}/\text{LSB}$ e passam por uma sobreamostragem e subamostragem para as frequências de 500 Hz e 100 Hz. Além disso, para a publicação desses dados, metadados utilizados para identificação dos pacientes foram removidos, sendo a data de nascimento substituída pela idade no momento da captura do exame, nomes substituídos por identificadores únicos, além da substituição dos nomes dos hospitais e profissionais de saúde envolvidos (Wagner *et al.*, 2020).

Para o presente trabalho, são selecionados os sinais das superclasses "NORM" e "MI", correspondentes às superclasses "Normal" e "Infarto do miocárdio". Onde desses, são utilizados apenas os que passaram por uma validação por um humano e que não possuíam problemas em seus eletrodos. Com isso, 11325 sinais serão utilizados, sendo 7.840 obtidos de pacientes saudáveis e 3.485 de pacientes com infarto nas paredes anterior, lateral, inferior ou posterior,

havendo a possibilidade de ocorrência desse infarto com outras patologias. Destaca-se que a versão dos sinais utilizada foi a versão com frequência de amostragem de 500 Hz devido à redução de erros que ocorrem pela interação desses sinais na versão subamostrada com as bibliotecas utilizadas na etapa de processamento.

Visando o treinamento da CNN, esse conjunto de sinais é dividido na proporção de 70:15:15 em conjuntos de treino, validação e teste, sendo para cada um desses conjuntos gerado um arquivo contendo os nomes dos sinais pertencentes a ele. A quantidade de sinais para cada um deles é apresentada na Tabela 3.2.

Tabela 3.2 – Quantidade de sinais a ser utilizada no trabalho para os conjuntos de treino, validação e teste.

	Normal (quantidade de sinais)	Infarto do Miocárdio (quantidade de sinais)
Treino	5.496	2.431
Validação	1.179	520
Teste	1.165	534
Total	7.840	3.485

3.2.2 – Processamento

O processamento dos sinais eletrocardiográficos é realizado com o uso de uma biblioteca desenvolvida para essa finalidade pelo Grupo de Processamento Digital de Sinais (GPDS) do Instituto Federal da Paraíba (IFPB), a qual contém funções para a leitura de ECGs provenientes da base de dados PTB-XL, filtragem desses sinais, para a reconstrução do VCG por meio dos métodos de Kors (quasi-ortogonal e regressão linear) e Dower, para a reconstrução de espaços de fase bidimensionais a partir de séries temporais e para o armazenamento das imagens desses espaços de fase.

De posse dos nomes dos sinais a serem utilizados, os quais foram obtidos dos arquivos para os conjuntos de treino, validação e teste; cada sinal de ECG foi lido e passou por uma etapa de filtragem. O filtro utilizado, definido pela biblioteca BioSPPy (Carreiras *et al.*, 2015), é um FIR passa faixa de ordem igual a 30% da frequência de amostragem do sinal e com frequências de corte iguais a 3 e 45 Hz, o qual é utilizado para realçar os picos R do sinal. Com isso, é facilitada a obtenção e o armazenamento dos índices correspondentes a esses pontos no sinal de ECG na série temporal, algo também feito por uma função da biblioteca BioSPPy.

Tabela 3.3 – Quantidade de ciclos para os conjuntos de treino, validação e teste.

	Normal (quantidade de ciclos)	Infarto do Miocárdio (quantidade de ciclos)
Treino	44.728	22.397
Validação	9.585	4.843
Teste	9.685	4.780
Total	63.998	32.020

As três derivações vetocardiográficas são então reconstruídas pelo Método da Regressão Linear de Kors e, a partir dos pontos R, os sinais dessas três derivações são divididos em ciclos, onde cada ciclo é delimitado por dois pontos R.

Com essas divisões em ciclos, a quantidade de sinais aumenta, pois cada ciclo corresponde agora a um sinal, para os conjuntos de treinamento, validação e teste. Sendo isso mostrado na Tabela 3.3. Além disso, em função da semelhança dos sinais de ciclos diferentes, mas que são provenientes de um mesmo exame, foi garantido que ciclos de um mesmo sinal estarão apenas no mesmo conjunto (treino, validação ou teste) do qual o sinal foi lido, a fim de, mais adiante, verificar corretamente a capacidade de generalização dos modelos treinados.

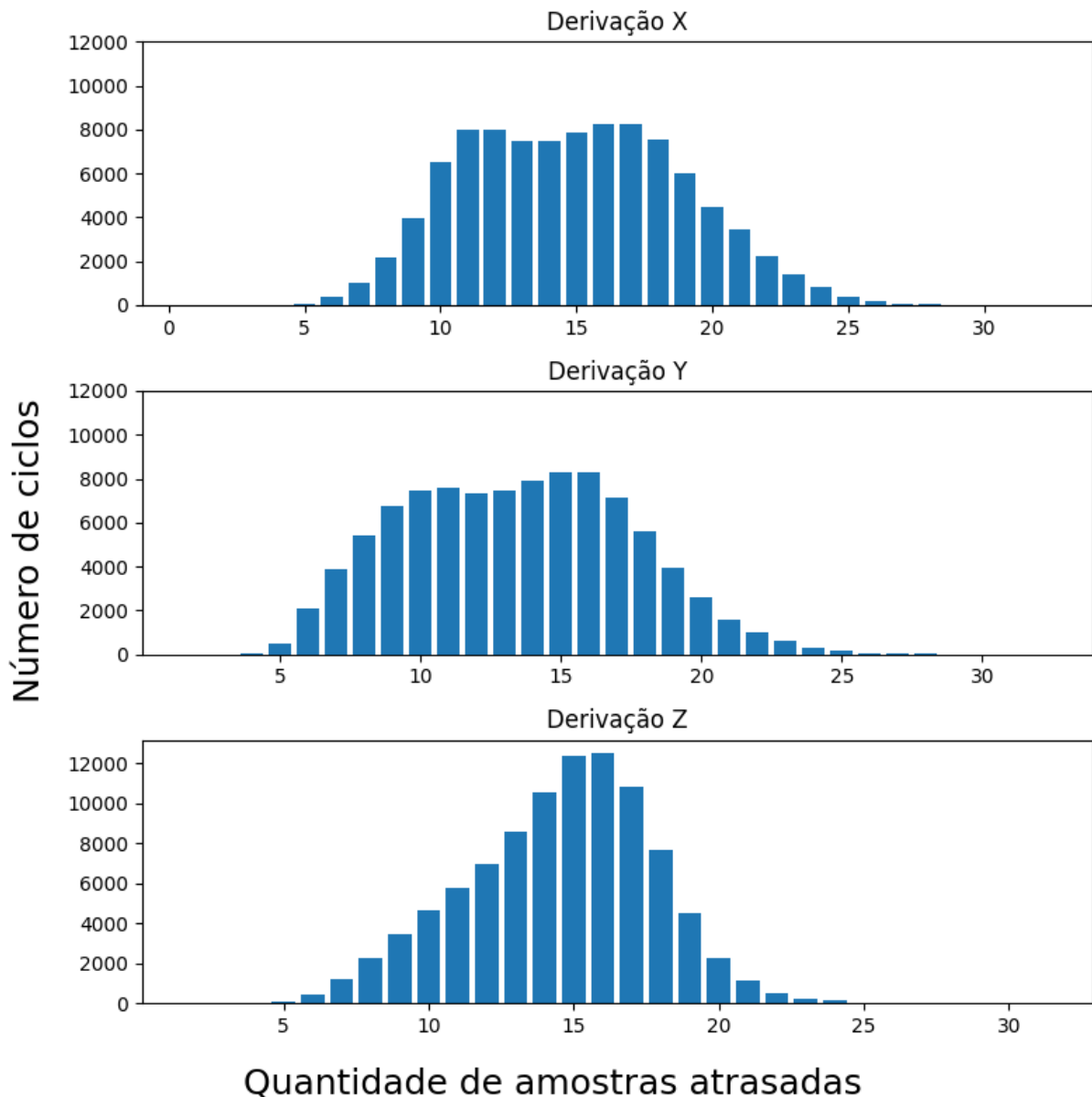
Para esses ciclos, agora sendo considerados como um sinal, foram feitos os processamentos para reconstrução do espaço de fase bidimensional e dos gráficos de recorrência.

Espaço de fase bidimensional reconstruído

A reconstrução do espaço de fase bidimensional pelo Método do Tempo de Atraso de Takens, descrita por Takens (1981, 1983), é realizada para cada sinal de cada derivação vetocardiográfica, na qual o tempo de atraso utilizado foi determinado com base no primeiro ponto mínimo da curva de informação mútua para o sinal do ciclo dessa derivação. Com base nisso, a distribuição dos valores de atraso, dados em número de amostras, estão dispostos na Figura 3.3.

Considerando cada espaço de fase reconstruído de uma derivação, foi gerada uma figura com valores de *pixels* binarizados contendo apenas a curva correspondente ao espaço de fase. Os limites dos eixos invisíveis das abcissas e ordenadas foram definidos no intervalo $[-1; 2]$. Esse intervalo foi escolhido por satisfazer os limites de valores desses eixos para a maioria das imagens. Além disso, as figuras foram geradas com resolução 224×224 , em razão de essa ser uma resolução utilizada no treinamento com o banco de dados ImageNet de modelos CNNs como a MobileNetV2, ResNet-50, ResNet-101 e DenseNet201 (Chollet *et al.*, 2015).

Figura 3.3 – Distribuição dos valores de atraso para cada derivação vetocardiográfica.



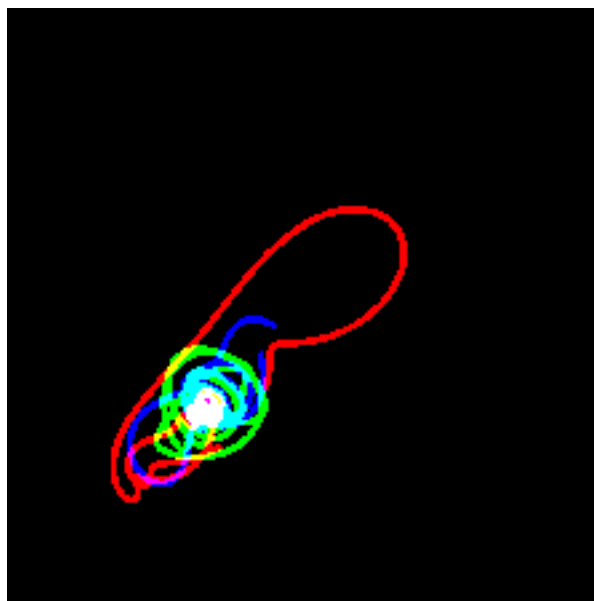
Fonte: Autoral.

De forma que fosse gerada uma única imagem correspondente aos EF para as três derivações vetocardiográficas de um paciente, cada um dos EF foi salvo em um canal de cor do sistema de cores RGB, sendo a derivação X salva no canal B, a derivação Y no canal G e a derivação Z no canal R. Um exemplo dessa imagem resultante pode ser visto na Figura 3.4.

Gráficos de recorrência

De maneira similar ao que foi feito para a reconstrução do espaço de fase, considerando cada ciclo como um sinal, utilizando o mesmo tempo de atraso da reconstrução do espaço de fase, foi determinada para cada ciclo de cada derivação vetocardiográfica a dimensão de imersão

Figura 3.4 – Exemplo de imagem resultante, após redimensionamento para melhor visualização, da junção dos espaços de fase de derivações vetocardiográficas nos canais do sistema RGB.



Fonte: Autoral.

mínima conforme o método dos falsos vizinhos mais próximos cujo algoritmo foi descrito por (Kennel *et al.*, 1992) e implementado pela biblioteca *Teaspoon* (Myers *et al.*, 2020).

Os valores da dimensão de imersão obtidos para cada derivação vetocardiográfica, considerando todos os ciclos, estão apresentados na Tabela 3.4. Dela, destaca-se a presença de poucos valores para a dimensão de imersão ideal como 4, podendo eles serem considerados *outliers* (inglês para pontos discrepantes) em razão de representarem uma amostra de menos de 0,01% da população para cada derivação vetocardiográfica (Osborne & Overbay, 2019). Sua natureza, todavia, não será investigada neste trabalho.

Tabela 3.4 – Quantidade de sinais, compostos por um ciclo cardíaco, baseado na sua derivação vetocardiográfica e na dimensão de imersão ideal.

Derivação vetocardiográfica	Dimensão de imersão ideal		
	2	3	4
X	15.470	80.538	9
Y	10.210	85.801	6
Z	17.426	78.587	4

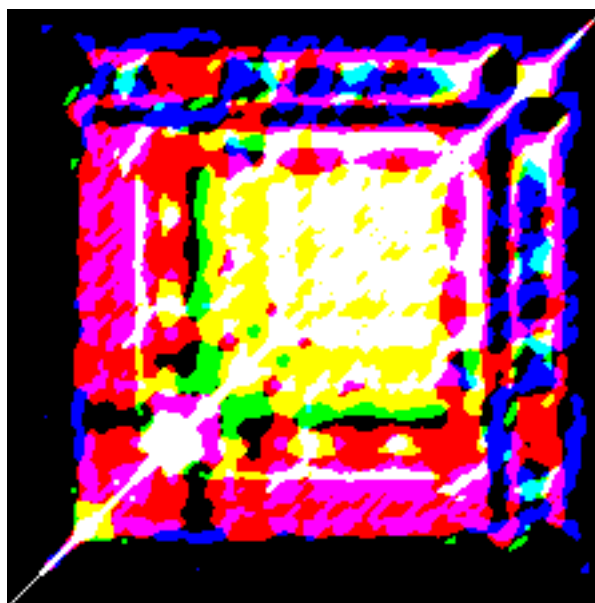
Para determinação do raio de vizinhança, se utilizou o critério de distância, onde o raio de vizinhança é um percentual do valor de distância máxima do espaço de fase. Como, usualmente, não se utilizam valores maiores que 10% dessa distância máxima (Zbilut & Webber Jr, 1992), foram gerados 10 conjuntos gráficos de recorrência utilizando a biblioteca

pyts (Faouzi & Janati, 2020) para todos os sinais, onde cada conjunto têm seus gráficos construídos com valores de percentual inteiros variando de 1% a 10%.

A determinação do valor adequado de raio de vizinhança é feita por comparação do desempenho de 10 classificadores treinados com mesmos hiperparâmetros e arquitetura, onde cada um deles foi treinado e testado 5 vezes com um conjunto de gráficos de recorrência gerado com cada um dos raios de vizinhança, onde o valor de raio de vizinhança escolhido foi o valor de 7% da distância máxima por ele gerar o modelo com menor métrica de erro (*loss*) em média para os 5 testes, sendo esse processo mais detalhado na Subsubseção 3.2.3.

Por fim, da mesma forma que para o espaço de fase bidimensional, o gráfico de recorrência de cada derivação foi salvo em um canal de cor do sistema RGB e tendo resolução 224×224 . Um exemplo de imagem obtida para os gráficos de recorrência é apresentada na Figura 3.5.

Figura 3.5 – Exemplo de imagem resultante, após redimensionamento para melhor visualização, da junção dos gráficos de recorrência de derivações vetocardiográficas nos canais do sistema RGB.



Fonte: Autoral.

3.2.3 – Redes Neurais Convolucionais

Para a classificação dos sinais de espaço de fase bidimensional e de gráficos de recorrência, foram utilizadas as seguintes arquiteturas de CNN: MobileNetV2, ResNet-50, ResNet-101, e DenseNet201 (Sandler *et al.*, 2018; He *et al.*, 2016; Huang *et al.*, 2017). Toda essa etapa foi realizada com as bibliotecas TensorFlow e Keras, tomando modelos de tutoriais apresentados em sua documentação como ponto de partida, (TensorFlow, 2022b). É importante

destacar que os treinamentos foram realizados utilizando uma GPU *Nvidia GeForce RTX 2060 Super*, a qual conta com 8GB de memória gráfica GDDR6.

Para todas as etapas a seguir foram carregados os sinais presentes na base de dados exposta na Tabela 3.5, na qual, foi utilizada uma parcela de 10% do conjunto de treino e 10% do conjunto de validação dela para a etapa de otimização dos hiperparâmetros descrita a seguir, sendo os demais sinais utilizados nos treinamentos dos modelos. É importante destacar que os conjuntos separados contêm sempre os mesmos sinais para todos os modelos utilizados.

Tabela 3.5 – Quantidade de ciclos para os conjuntos de treino e validação, divididos para a otimização dos hiperparâmetros e treinamento do modelo, além do conjunto de teste.

		Normal (quantidade de ciclos)	Infarto do Miocárdio (quantidade de ciclos)
Treino	Otimização de Hiperparâmetros	4.487	2.225
	Modelo	40.241	20.172
Validação	Otimização de Hiperparâmetros	960	482
	Modelo	8.625	4.361
Teste		9.685	4.780
Total		63.998	32.020

Considerando esses conjuntos, o tamanho de lote utilizado foi de 16, sendo esse valor escolhido com base nas limitações de memória do *hardware* utilizado para o treinamento dos modelos.

Em todas as arquiteturas testadas foi definida uma camada anterior ao modelo base, chamada de camada de pré-processamento, a qual é responsável por escalonar os *pixels* das imagens para a faixa de valores de intensidade esperada pelo modelo base. Essa camada é seguida pelo modelo base, do qual se utilizam apenas as camadas responsáveis pela extração de características, as quais retornam um conjunto de blocos quadrados de valores com um tamanho fixo para cada imagem, onde cada bloco é uma característica.

Para a camada de classificação, é adicionada uma camada de *pooling*, responsável por obter a média dos valores dos blocos quadrados, de forma que, para cada imagem, é obtido um vetor correspondente ao conjunto de todas as características. Esse vetor de características passa por uma camada para adição de *dropout*, descrito a seguir, a fim de aumentar a capacidade de generalização do modelo, atuando como entrada para um único neurônio de classificação com função de ativação sigmoide.

Ademais, para avaliação do uso do *transfer learning* e para a definição do valor otimizado de raio de vizinhança, descritos mais adiante, foram definidos valores-base para os hiperparâmetros do modelo. Para a taxa de aprendizagem foi definido um valor de $5 \cdot 10^{-4}$, sendo o algoritmo utilizado o Adam, já o *dropout* definido foi o de 20%.

As métricas a serem retornadas pela CNN ao fim de cada época de treinamento, na validação do modelo, foram a acurácia, a sensibilidade, a precisão, o escore F1, a área abaixo da curva ROC (AUC), a quantidade de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

Além disso, visando lidar com o desbalanceamento nas quantidades de sinais das classes "Normal" e "Infarto", foi utilizado um dos métodos sugeridos pela documentação do TensorFlow. Nele, pesos diferentes são atribuídos para cada classe, onde a classe com menor representação no conjunto de treinamento recebe um maior peso, de forma que uma maior importância será dada a essa classe, enquanto a classe com maior representação terá sua importância reduzida (TensorFlow, 2022a). Sendo Q_{NORM} a quantidade de sinais da classe "Normal" e Q_{MI} a quantidade de sinais da classe Infarto do Miocárdio, seguindo o método apresentado na documentação, os pesos das classes P_{NORM} e P_{MI} são dados, respectivamente, pela Equação 3.1 e Equação 3.2.

$$P_{NORM} = \frac{1}{Q_{NORM}} * \frac{Q_{NORM} + Q_{MI}}{2} \quad (3.1)$$

$$P_{MI} = \frac{1}{Q_{MI}} * \frac{Q_{NORM} + Q_{MI}}{2} \quad (3.2)$$

Para todos os modelos, foi utilizada como função custo a função Entropia Cruzada Binária (*Binary Cross-Entropy*), sendo feito um treinamento com duração máxima de 110 épocas. Ele, no entanto, pode ter um número menor de épocas devido ao uso de um critério de parada precoce (*Early Stopping*). Esse critério determina que o treinamento é parado caso não haja uma melhoria em uma certa métrica após uma quantidade definida de épocas. No caso desse trabalho, isso é: caso não haja qualquer redução no valor da métrica *loss* por 5 de épocas considerando o conjunto de validação.

Ao fim do treinamento, um gráfico correspondente à evolução do valor do custo e das métricas conforme as épocas para os conjuntos de treino e validação é salvo. Além disso, também é salvo o modelo após a restauração dos pesos que levaram ao menor valor da métrica

loss durante o treinamento. Por fim, a desempenho do modelo é novamente avaliada para todo o conjunto de validação considerando o modelo salvo, sendo retornadas as métricas nessa avaliação e gerada uma imagem correspondente à matriz de confusão para esse conjunto.

Avaliação do uso de *transfer learning*

Inicialmente se avaliou a possibilidade de utilização do *transfer learning* devido a um menor custo de memória associado a esse método de treinamento de CNNs, já que os pesos das camadas iniciais não necessitam ser treinados, reduzindo a quantidade de valores de parâmetros a serem armazenadas na memória. Por outro lado, esse método poderia não ser vantajoso em razão das imagens da base Imagenet (Lab, 2020), utilizadas para a obtenção dos pesos nos modelos-base, serem muito diferentes das imagens de espaço de fase e gráficos de recorrência.

Para essa avaliação foram treinados modelos utilizando a arquitetura MobileNetV2, em função do seu menor tempo de treinamento, com as características descritas anteriormente na Subseção 3.2.3 inicializados com pesos aleatórios por meio da inicialização uniforme de Xavier e inicializados com os pesos obtidos a partir do treinamento realizado com a base de dados ImageNet (Lab, 2020), o qual possui uma etapa de ajuste fino.

Nos dois casos, o algoritmo de otimização utilizado foi o Adam. Além disso, foi utilizada uma taxa de aprendizagem base de $5 \cdot 10^{-4}$, sendo ela reduzida para 50% de seu valor caso não haja uma redução no valor de *loss* de validação para 3 épocas consecutivas.

Para o cenário de *transfer learning*, um máximo de 50 das 110 épocas são utilizadas para uma etapa de ajuste fino após uma etapa de treinamento inicial de 60 épocas ou parada precoce. Nessa etapa o último 1/5 das camadas do modelo base são destravadas para serem retreinadas. Com isso, a extração de características dessas camadas, as quais são mais específicas, é adaptada para o conjunto de dados deste trabalho. Para isso, a taxa de aprendizagem é reduzida para 1/10 da taxa inicial. Além disso, seguindo a metodologia apresentada pela documentação, o algoritmo de otimização é mudado do Adam para o RMSprop (TensorFlow, 2022b), sendo o modelo novamente recompilado e treinado por mais 50 épocas ou até que atinja o mesmo critério de parada precoce.

Para a comparação, foram treinados 5 modelos, considerando os hiperparâmetros base descritos previamente e os conjuntos de treino e validação para seus testes. Os valores das métricas para o conjunto de validação foram então armazenados, considerando a época em que

houve o menor valor de *loss* para o conjunto de validação, sendo então obtida a média e mediana de todos esses valores.

Comparando as médias das métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e AUC considerando o uso ou não uso do *transfer learning*, foi observado que o uso dessa técnica estava sendo improdutivo. A razão disso é que, em média, o *loss* foi menor e as demais métricas foram maiores quando o modelo foi iniciado com pesos aleatórios, conforme apresentado em Capítulo 4 nas Tabelas 4.1 e 4.2. Com base nisso, o uso da técnica de *transfer learning* foi descartado para as imagens de espaço de fase reconstruído e também para os gráficos de recorrência.

Otimização dos hiperparâmetros

A otimização dos hiperparâmetros se deu pelo uso do Keras Tuner, uma biblioteca cujo objetivo é auxiliar a encontrar o melhor conjunto de deles para um modelo. Como, por exemplo, a quantidade de neurônios para uma determinada camada ou algoritmo de otimização dos pesos da rede (TensorFlow, 2023).

O algoritmo de otimização de hiperparâmetros utilizado foi a otimização bayesiana, a qual, ao contrário da busca aleatória e da busca em grade, considera o desempenho de iterações anteriores para o teste de novos conjuntos de hiperparâmetros, propondo combinações baseadas em combinações anteriores que se mostraram promissoras (Snoek *et al.*, 2012).

O conjunto a ser otimizado foi o valor de *dropout*, o valor da regularização L2 e a taxa de aprendizagem. O primeiro definido pelas amostras no intervalo $[0, 2; 0, 8]$ com passo 0,1; o segundo a ser definido na faixa $[1 \cdot 10^{-6}; 1 \cdot 10^{-2}]$ com amostragem do tipo logarítmica e o terceiro também com esse mesmo tipo de amostragem, mas no intervalo $[1 \cdot 10^{-4}; 1 \cdot 10^{-2}]$.

A otimização foi feita visando obter o conjunto de hiperparâmetros que levasse à minimização da métrica *loss* para o conjunto de validação, sendo feito com o parâmetro *max_trials* definido como 100, de forma que um máximo de 100 combinações de hiperparâmetros é testada ou até que o espaço de busca seja esgotado. Cada combinação é testada por até 20 épocas, onde se utilizaram os pesos para as classes definidos nas equações 3.1 e 3.2, além disso, é utilizado um critério para *early stopping*, onde o treinamento do modelo para essa combinação de parâmetros é interrompido caso não haja uma melhoria da métrica *loss* por até 5 épocas.

Esse processo de otimização de hiperparâmetros é feito considerando cada arquitetura e cada tipo de sinal, sendo os conjuntos de hiperparâmetros obtidos expostos nas subseções a seguir.

Espaço de fase bidimensional

Para o treinamento dos modelos utilizando as imagens de espaço de fase, eles foram compilados, sendo definidos nessa etapa, para cada um deles, o algoritmo de otimização e a taxa de aprendizagem definidos na função de otimização de hiperparâmetros para aquele modelo, onde esses hiperparâmetros são eles apresentados na Tabela 3.6. As demais características para o modelo foram iguais às do modelo utilizado para avaliação do *transfer learning*.

Tabela 3.6 – Valores de *Dropout*, regularização L2 e taxa de aprendizagem determinados na otimização de hiperparâmetros para cada arquitetura considerando imagens de espaço de fase bidimensional.

	Dropout	Regularização L2	Taxa de aprendizagem
MobileNetV2	0,5	$2,751 \cdot 10^{-4}$	0,01
ResNet-50	0,4	$1,775 \cdot 10^{-6}$	$1,274 \cdot 10^{-3}$
ResNet-101	0,5	$6,981 \cdot 10^{-5}$	$6,181 \cdot 10^{-4}$
DenseNet201	0,4	$1,237 \cdot 10^{-6}$	$1,984 \cdot 10^{-4}$

Com base nisso, foram treinados 5 modelos para cada uma das arquiteturas, sendo obtidas suas métricas, além dos gráficos para sua evolução. Considerando as métricas, foram obtidas suas médias e os desvios-padrão para o conjunto de validação, sendo esses resultados apresentados no Capítulo 4.

Nele, também são apresentadas as métricas e matriz de confusão considerando o conjunto de teste para a arquitetura e modelo com maior valor da métrica score F1 para o conjunto de validação.

Gráficos de recorrência

Um processo semelhante ao realizado para as imagens de espaço de fase reconstruído foi realizado para os gráficos de recorrência. Antes disso, todavia, houve uma avaliação para determinar o raio de vizinhança mais adequado para a classificação desses sinais. Para isso, considerando cada conjunto de sinais de gráficos de recorrência gerados, onde cada conjunto possui um raio de vizinhança diferente, e utilizando os conjuntos de treino e validação deles. Foram treinados 5 modelos com a arquitetura MobileNetV2, em razão do menor tempo de

treinamento necessário, a fim de, a partir da média do valor de *loss* para cada um deles, determinar o valor ideal desse parâmetro, sendo ele o que levasse à menor média dessa métrica.

Comparando os valores de média, foi observado que o raio de vizinhança otimizado foi para o valor de 7% da distância máxima do atrator, conforme apresentado na Tabela 4.7, apresentada na Seção 4.3. Com essa informação, utilizando apenas o conjunto de gráficos de recorrência obtidos com o raio de vizinhança otimizado, foi realizado um processo semelhante ao utilizado para as imagens de espaço de fase bidimensional.

As mesmas arquiteturas de CNN passam por uma etapa de otimização de seus hiperparâmetros para que, em seguida, sejam treinados 5 modelos para cada arquitetura, onde são salvas suas métricas para o conjunto de validação, gráficos com a evolução delas por época. Por fim, também são obtidas as métricas e matriz de confusão considerando o conjunto de teste para a arquitetura e modelo com maior valor de escore F1. Nesse cenário, os hiperparâmetros obtidos são apresentados na Tabela 3.7.

Tabela 3.7 – Valores de *Dropout*, regularização L2 e taxa de aprendizagem determinados na otimização de hiperparâmetros para cada arquitetura considerando gráficos de recorrência.

	Dropout	Regularização L2	Taxa de aprendizagem
MobileNetV2	0,6	$7,176 \cdot 10^{-06}$	$3,319 \cdot 10^{-3}$
ResNet-50	0,4	$1,539 \cdot 10^{-05}$	$6,486 \cdot 10^{-3}$
ResNet-101	0,2	$8,322 \cdot 10^{-3}$	$6,813 \cdot 10^{-3}$
DenseNet201	0,2	$3,690 \cdot 10^{-4}$	$6,335 \cdot 10^{-4}$

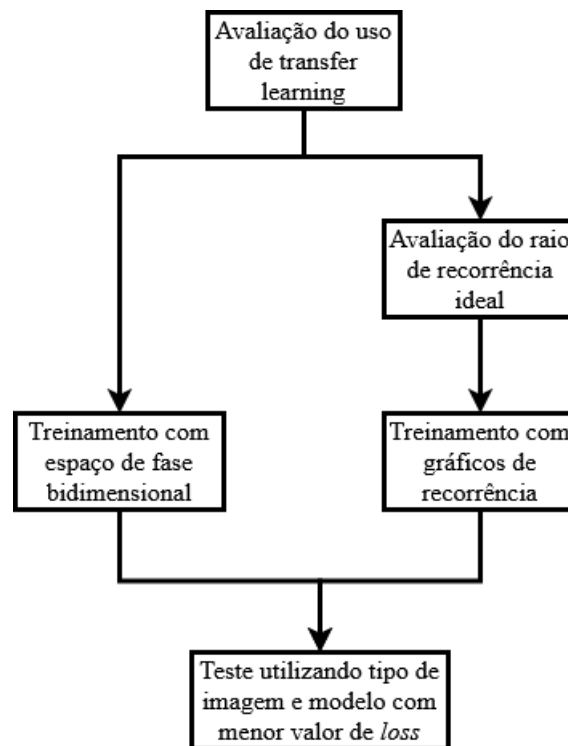
Avaliação com conjunto de teste

Por fim, com base no valor médio da métrica *loss*, obtido considerando o conjunto de validação, foi selecionado o tipo de sinal e arquitetura que levou ao menor valor dessa métrica. Com isso, se avaliou o desempenho do melhor modelo dessa arquitetura e tipo de sinal considerando o conjunto de teste, até então não utilizado, para a obtenção do conjunto de métricas do presente trabalho, além da respectiva matriz de confusão para esse teste.

Os resultados obtidos com base na metodologia descrita neste capítulo são apresentados como tabelas e gráficos de evolução de métricas a seguir, no Capítulo 5, seguidos também de sua discussão, na qual eles são comparados com os trabalhos citados na Tabela 1.1.

Neste capítulo estão apresentados os resultados das análises para o uso do *transfer learning*, dos treinamentos dos modelos com as arquiteturas MobileNetV2, ResNet-50, ResNet-101 e DenseNet201 utilizando imagens de espaço de fase bidimensional, da avaliação do raio de vizinhança ideal e dos treinamentos dos modelos com o uso gráficos de recorrência, conforme sintetizado na Figura 4.1. Ao final do capítulo, essas análises serão discutidas.

Figura 4.1 – Fluxograma para as análises realizadas com resultados no presente capítulo.



Fonte: Autoral.

Para a análise do uso do *transfer learning*, estão apresentados os valores das métricas de acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para as etapas de análise do uso da técnica de *transfer learning* considerando o conjunto de validação das imagens de espaço de fase bidimensional a fim de determinar em qual cenário a métrica *loss* é minimizada.

Em seguida, são apresentados os valores dessas métricas considerando as 5 rodadas de treinamento e teste com o conjunto de validação utilizando o espaço de fase bidimensional para cada arquitetura, além dos valores da média e desvio padrão dessas métricas. Ademais, para cada arquitetura, estão disponibilizados os gráficos de evolução das métricas nas etapas de treinamento e validação para a rodada com menor valor de *loss* considerando o conjunto de validação.

Após isso, estão dispostos os resultados para a avaliação do raio de vizinhança ideal, onde os valores médios de 5 rodadas das métricas para o conjunto de validação considerando cada valor de raio entre 1% a 10%, conforme descrito no Capítulo 3, são expostos, sendo o raio que leva a menor *loss* utilizado na etapa de treinamento e teste considerando gráficos de recorrência, descrita a seguir.

Da mesma forma que para os modelos treinados com imagens de espaço de fase bidimensional, também são mostradas as métricas para os 5 modelos treinados com gráficos de recorrência para cada arquitetura. Também serão exibidos os gráficos de evolução das métricas para o modelo com menor valor para a *loss* de validação de cada arquitetura.

Com base nos resultados obtidos para os dois tipos de sinais e para as 4 arquiteturas, é observada a arquitetura associada com o tipo de sinal que teve o menor valor para a métrica *loss*. Dessa associação, o modelo com menor valor para essa métrica será avaliado com o conjunto de teste para esse sinal que levou à minimização da *loss*. Serão então expostas suas métricas, além de sua matriz de confusão. Por fim, ao final deste capítulo, os resultados apresentados serão discutidos.

4.1 – Avaliação do uso da técnica de *transfer learning* com imagens de espaço de fase

A seguir, na Tabela 4.1 e na Tabela 4.2 são apresentados os 5 modelos treinados utilizando a inicialização com pesos aleatórios e utilizando a técnica de *transfer learning*, respectivamente. Para cada uma delas, estão as métricas obtidas durante o treinamento e o teste com conjunto de validação para os 5 modelos treinados, além das suas médias e desvios padrão para os valores desses modelos.

Como pode ser observado nas Tabelas 4.1 e 4.2, a métrica *loss* foi, em média, ligeiramente inferior para os modelos inicializados com pesos aleatórios, os quais são

Tabela 4.1 – Métricas de *loss*, acurácia, sensibilidade, precisão, escore F1, área abaixo da curva ROC do treino e teste dos modelos ao utilizar a inicialização com pesos aleatórios.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3290	0,8621	0,7627	0,8148	0,7879	0,9242
2	0,3240	0,8642	0,7585	0,8231	0,7895	0,9241
3	0,3273	0,8653	0,7200	0,8561	0,7822	0,9260
4	0,3334	0,8671	0,7216	0,8601	0,7848	0,9243
5	0,3395	0,8632	0,7283	0,8429	0,7814	0,9244
Média	0,3306	0,8644	0,7382	0,8394	0,7851	0,9246
Desvio Padrão	0,0054	0,0017	0,0185	0,0178	0,0031	0,0007

Tabela 4.2 – Métricas de *loss*, acurácia, sensibilidade, precisão, escore F1, área abaixo da curva ROC e duração em segundos do treino e teste dos modelos ao utilizar *transfer learning*.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3928	0,8265	0,6611	0,7881	0,7190	0,8873
2	0,3938	0,8266	0,6460	0,7991	0,7144	0,8860
3	0,3938	0,8260	0,6632	0,7852	0,7190	0,8869
4	0,3978	0,8287	0,6971	0,7708	0,7321	0,8897
5	0,3953	0,8278	0,6423	0,8056	0,7147	0,8854
Média	0,3947	0,8271	0,6619	0,7898	0,7199	0,8870
Desvio Padrão	0,0017	0,0010	0,0194	0,0120	0,0064	0,0015

inicializados conforme a inicialização uniforme de Xavier. Ao mesmo tempo, para esses modelos, as demais métricas possuíram valores em média superiores em comparação com os modelos treinados com *transfer learning*. O que indica que o treinamento de modelos inicializados com pesos aleatórios seja o mais indicado para essa aplicação.

4.2 – Treinamento com imagens de espaço de fase

Considerando a etapa de treinamento com imagens correspondentes à reconstrução do espaço de fase bidimensional, as subseções 4.2.1, 4.2.2, 4.2.3, 4.2.4 apresentam, respectivamente, os resultados para os modelos treinados com imagens de espaço de fase reconstruído com base nas arquiteturas MobileNetV2, ResNet-50, ResNet-101 e DenseNet201.

4.2.1 – MobileNetV2

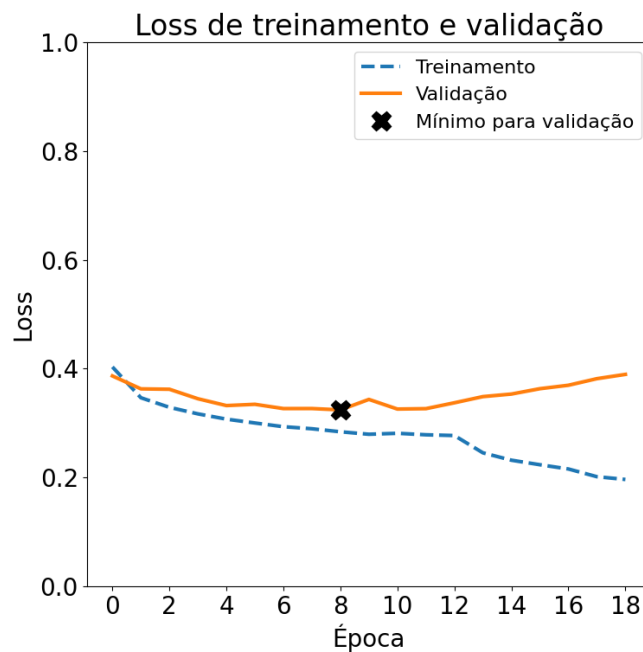
Para as cinco rodadas dos modelos treinados utilizando como modelo base a arquitetura MobileNet-V2, as métricas obtidas podem ser observadas na Tabela 4.3. Considerando a rodada número 1 na Tabela 4.3, para o qual foi obtido o menor valor para a métrica *loss*, a evolução

dessa métrica conforme as épocas do treinamento para os conjuntos de treino e validação pode ser observada na Figura 4.2.

Tabela 4.3 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura MobileNetV2.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3258	0,8635	0,7244	0,8471	0,7810	0,9260
2	0,3265	0,8629	0,7505	0,8253	0,7861	0,9243
3	0,3270	0,8632	0,7597	0,8196	0,7885	0,9233
4	0,3325	0,8605	0,7700	0,8059	0,7875	0,9242
5	0,3314	0,8602	0,7452	0,8217	0,7816	0,9241
Média	0,3286	0,8620	0,7500	0,8239	0,7850	0,9244
Desvio Padrão	0,0031	0,0016	0,0171	0,0149	0,0035	0,0010

Figura 4.2 – Evolução da métrica *loss* para os conjuntos de treino e validação com EF em cada época na arquitetura MobileNetV2.

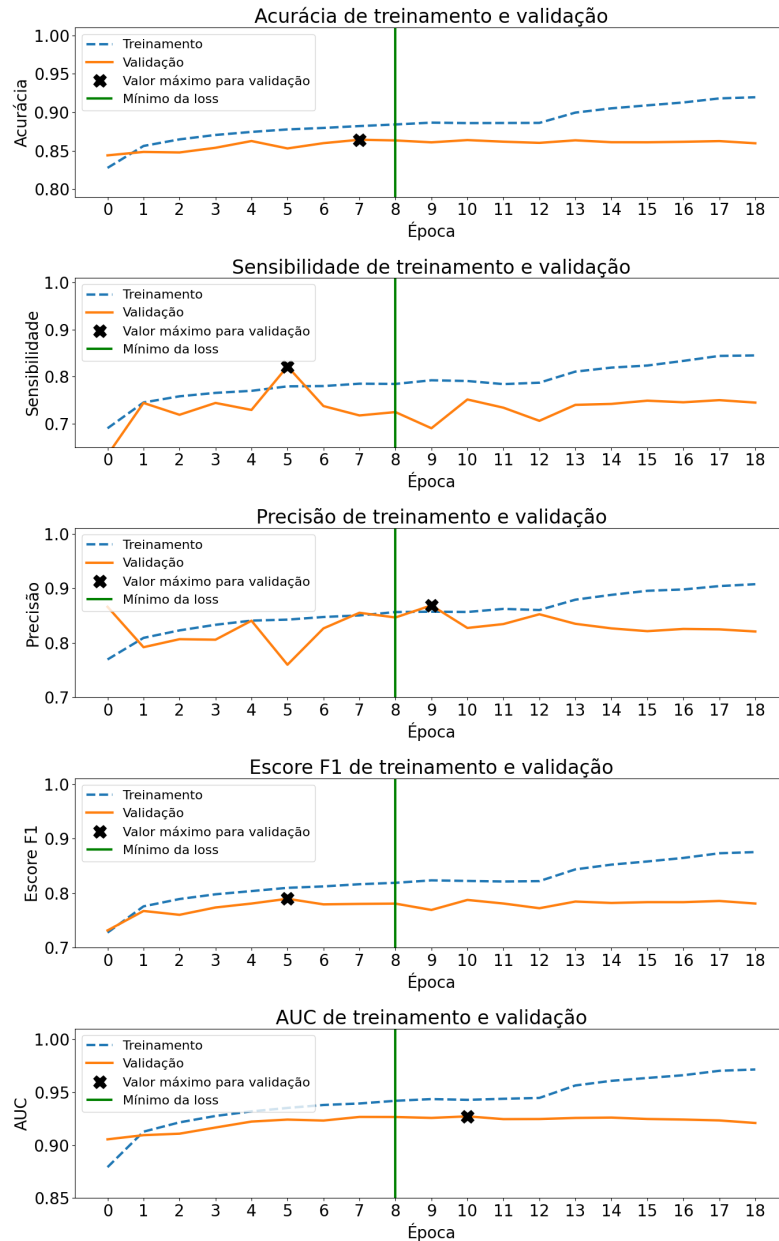


Fonte: Autoral.

Como se pode observar, a métrica atinge seu valor mínimo para o conjunto de validação na oitava época, passando então a crescer, especialmente a partir da época 12, indicando uma perda da capacidade de generalização do modelo. É importante lembrar que, devido ao critério de parada precoce, o modelo salvo para o qual foram obtidas as métricas da Tabela 4.3 utiliza os pesos da época 8, a qual teve o valor mínimo de *loss*, ocorrendo algo análogo para os demais modelos apresentados a seguir.

A evolução das demais métricas avaliadas também em função da sua época durante o treinamento e validação pode ser visualizada na Figura 4.3. Pode ser observado que a métrica acurácia para o conjunto de validação evoluiu até a época 7, se mantendo estável e tendo um leve decaimento ao final do treinamento.

Figura 4.3 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com EF em cada época na arquitetura MobileNetV2.



Fonte: Autoral.

Na Figura 4.3, a sensibilidade inicia com valores abaixo de 0,7, tendo seu pico na época 5, havendo então oscilações nos valores dessa métrica e uma estabilização em torno de 0,75. Já a precisão tem seu valor de pico na época 9, sendo ele levemente maior que o valor de precisão

inicial, havendo então um decaimento para valores em torno de 0,8. Com base nessas duas métricas, o escore F1 tem seu pico na época 5, havendo então um leve decaimento até o fim do treinamento. Por fim, já para a métrica AUC, seu pico ocorre na época 10, havendo então uma queda pequena queda para as épocas seguintes.

4.2.2 – ResNet-50

De forma similar ao apresentado para a arquitetura MobileNetV2, para a arquitetura ResNet-50, os valores para as métricas estão dispostos na Tabela 4.4. Nesse caso, a rodada número 4 foi a que possuiu o menor *loss*. A evolução da *loss* de treinamento e validação para ela em função das épocas pode ser observada na Figura 4.4.

Tabela 4.4 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura ResNet-50.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3400	0,8655	0,7271	0,8506	0,7840	0,9273
2	0,3385	0,8640	0,7546	0,8254	0,7885	0,9244
3	0,3368	0,8651	0,7636	0,8220	0,7917	0,9266
4	0,3304	0,8672	0,7434	0,8427	0,7900	0,9261
5	0,3347	0,8649	0,7386	0,8399	0,7860	0,9251
Média	0,3361	0,8653	0,7455	0,8361	0,7880	0,9259
Desvio Padrão	0,0037	0,0012	0,0141	0,0120	0,0031	0,0012

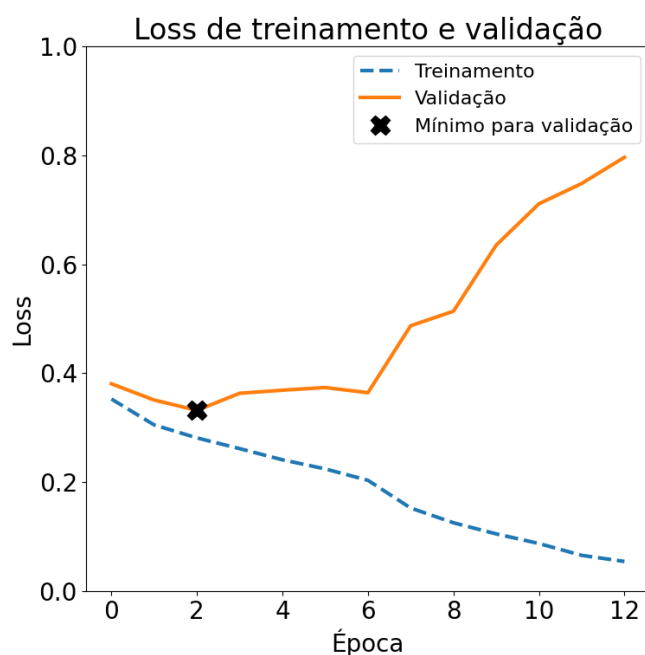
O valor mínimo de *loss* para o conjunto de validação esteve na época 2 do treinamento, havendo uma leve estabilização até a época 6. Após isso, a métrica passa a crescer rapidamente, indicando a perda da capacidade de generalização do modelo.

Conforme mostrado na Figura 4.5, a segunda época também é o ponto onde ocorrem os valores máximos de acurácia e escore F1, onde as duas métricas sofrem um leve decaimento.

No que diz respeito à sensibilidade, ela tem uma tendência de crescimento, tendo seu pico apenas na época 9. Essa tendência de crescimento dessa métrica somada com a queda que ocorre na precisão, cujo pico foi na época 0, mostra que ao decorrer do treinamento, o modelo tende a classificar mais exemplos como infarto, reduzindo falsos negativos ao custo de aumentar os falsos positivos.

Por fim, a AUC tem seu pico na época 4, a qual possui um valor próximo à época de valor mínimo de *loss*. Após seu pico, há um decaimento dela para valores inferiores ao do início do treinamento.

Figura 4.4 – Evolução da métrica *loss* para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-50.



Fonte: Autoral.

4.2.3 – ResNet-101

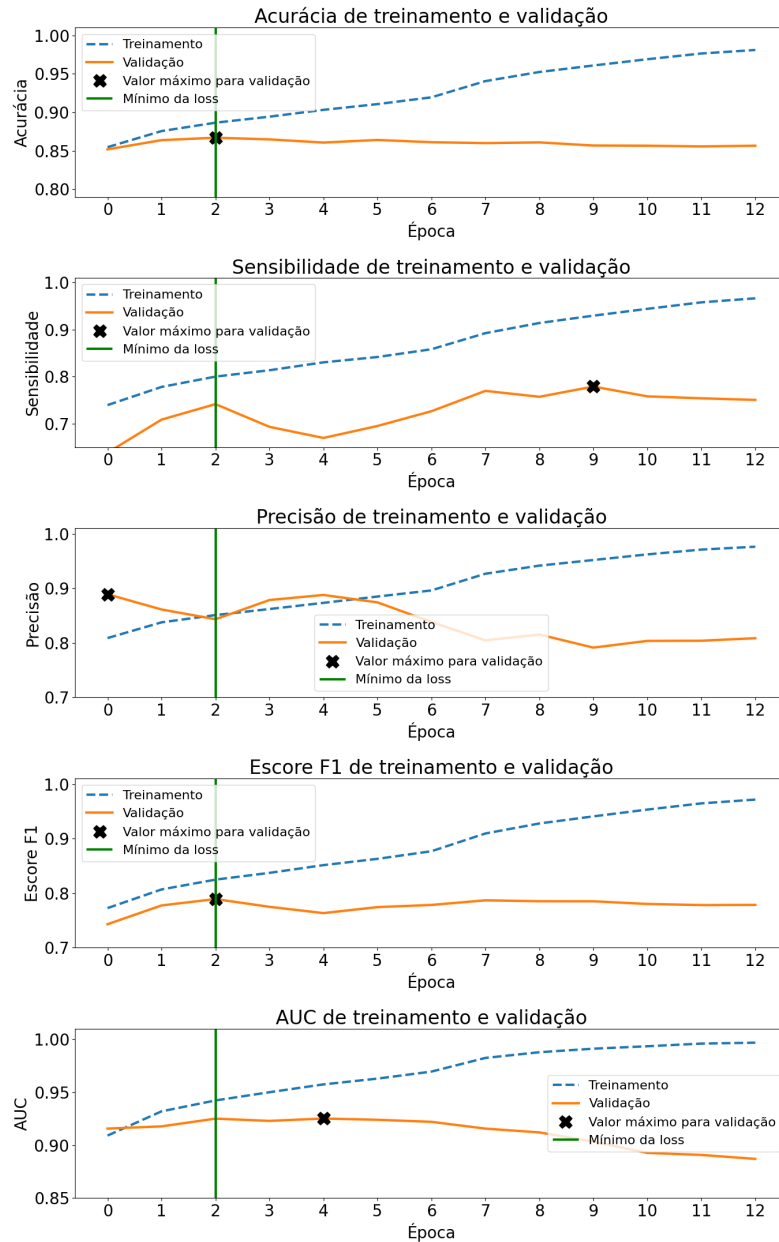
Para a arquitetura ResNet-101, cujos resultados são mostrados na Tabela 4.5, a rodada número 5 levou ao menor valor entre as rodadas para a *loss*. Considerando a evolução da *loss* de validação durante o treinamento, apresentada na Figura 4.6, foi observado algo similar ao que ocorreu para a ResNet-50. O mínimo dessa métrica esteve na época 2 e após algumas épocas houve um grande crescimento dela, indicando uma maior perda da capacidade de generalização, tendo isso ocorrido a partir da época 5.

Tabela 4.5 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura ResNet-101.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3381	0,8686	0,7294	0,8579	0,7884	0,9294
2	0,3375	0,8667	0,7306	0,8514	0,7864	0,9237
3	0,3366	0,8655	0,7732	0,8165	0,7943	0,9263
4	0,3374	0,8651	0,7519	0,8303	0,7892	0,9250
5	0,3294	0,8662	0,7572	0,8296	0,7918	0,9261
Média	0,3358	0,8664	0,7485	0,8371	0,7900	0,9261
Desvio Padrão	0,0036	0,0014	0,0186	0,0171	0,0031	0,0021

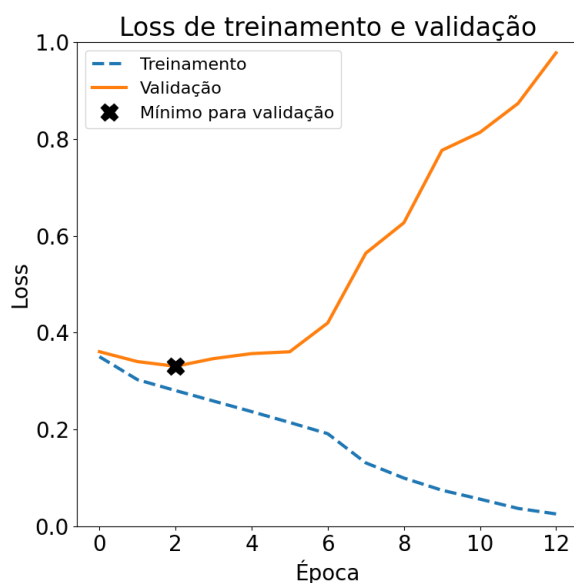
Considerando as demais métricas, apresentadas na Figura 4.7, pode ser observado que a acurácia atinge seu pico apenas na época 5, apesar de ter pouca variação durante seu crescimento.

Figura 4.5 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-50.



Fonte: Autoral.

Figura 4.6 – Evolução da métrica *loss* para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-101.



Fonte: Autoral.

Após, seu pico, no entanto, ocorre um decaimento para valores inferiores ao da época 0. Da mesma forma que ocorreu para a ResNet-50, há uma tendência de aumento da classificação de sinais em infarto com o aumento de épocas, o que pode ser visto no aumento da sensibilidade e redução da precisão, que tiveram seus picos nas épocas 9 e 4, respectivamente.

O score F1 tem seu pico na época 2, havendo então uma queda e seguida de uma estabilização em um valor intermediário entre o seu valor de pico e o inicial. Já a AUC tem seu pico na época 5, sendo ele seguido de uma redução sem estabilização.

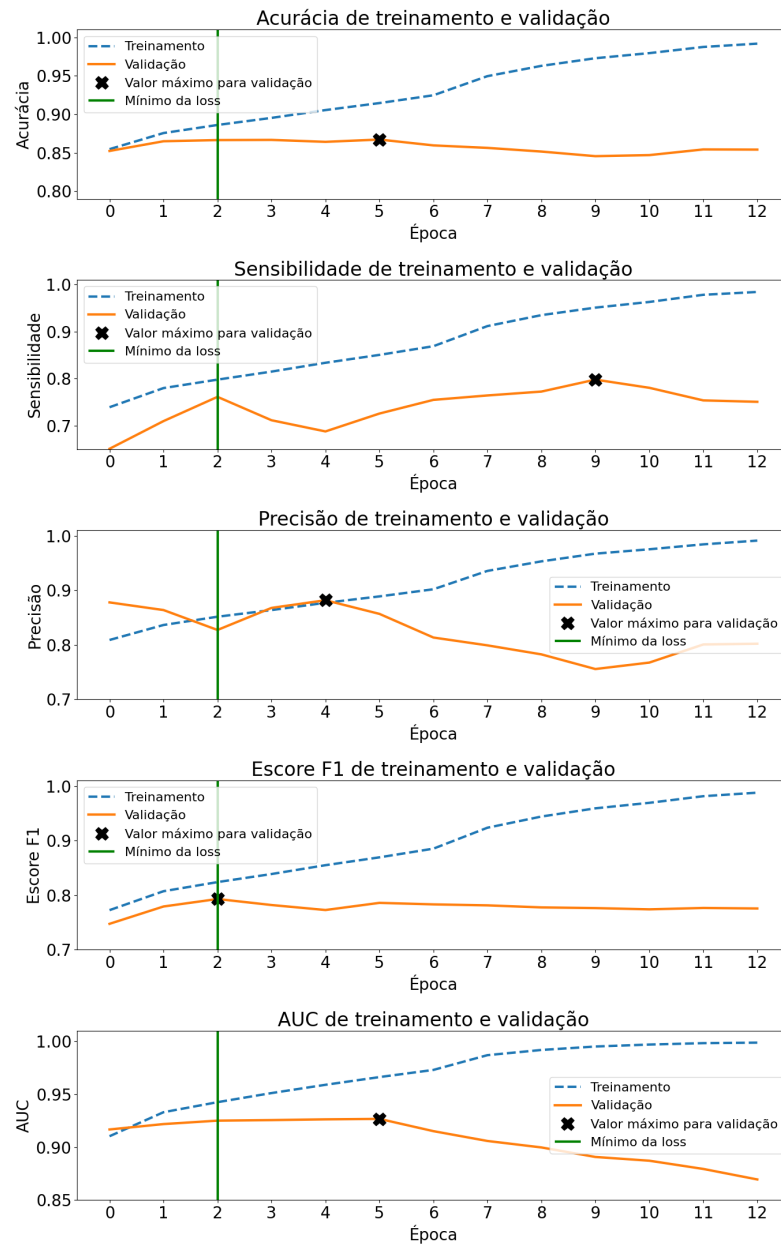
4.2.4 – DenseNet201

Analogamente, a Tabela 4.6 contém os valores das métricas para o conjunto de teste dos modelos baseados na arquitetura DenseNet201. Neste caso, a segunda rodada levou a um menor valor para a métrica *loss*.

Para essa mesma rodada, a curva de evolução da *loss* de validação possuiu um comportamento similar às demais: houve um leve decréscimo até a segunda época, havendo então um crescimento que se intensifica após a época 6, conforme apresentado na Figura 4.8.

As demais métricas dessa rodada, excluindo a sensibilidade e precisão, mostradas na Figura 4.9, seguem o comportamento da *loss*, tendo seu ponto máximo para o conjunto de validação também na segunda época e tendo em seguida um leve decaimento. Já a sensibilidade cresce gradativamente, tendo seu máximo na época 9, enquanto a precisão, com máximo na

Figura 4.7 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com EF em cada época na arquitetura ResNet-101.

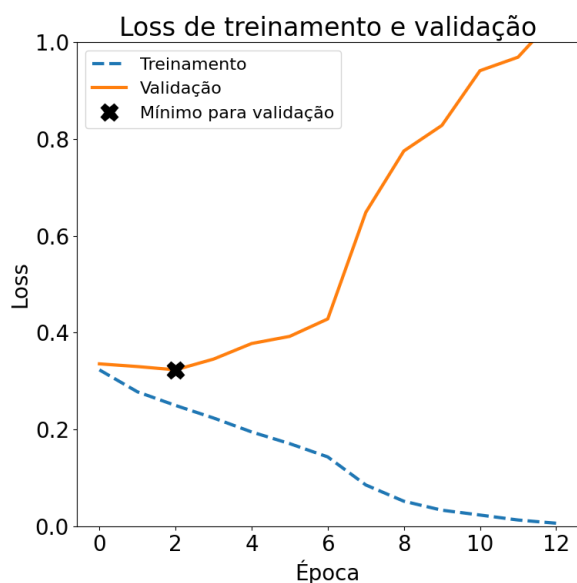


Fonte: Autoral.

Tabela 4.6 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com EF e a arquitetura DenseNet201.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3296	0,8642	0,7214	0,8517	0,7811	0,9220
2	0,3219	0,8667	0,7423	0,8421	0,7890	0,9308
3	0,3284	0,8692	0,7333	0,8565	0,7901	0,9281
4	0,3261	0,8663	0,7397	0,8430	0,7880	0,9271
5	0,3266	0,8661	0,7260	0,8534	0,7845	0,9267
Média	0,3265	0,8665	0,7325	0,8493	0,7866	0,9269
Desvio Padrão	0,0029	0,0018	0,0089	0,0064	0,0037	0,0032

Figura 4.8 – Evolução da métrica *loss* para os conjuntos de treino e validação com EF em cada época na arquitetura DenseNet201.



Fonte: Autoral.

época 0, decresce. O comportamento dessas duas métricas indica que o modelo tende a classificar os exemplos como positivos à medida que é treinado por mais épocas.

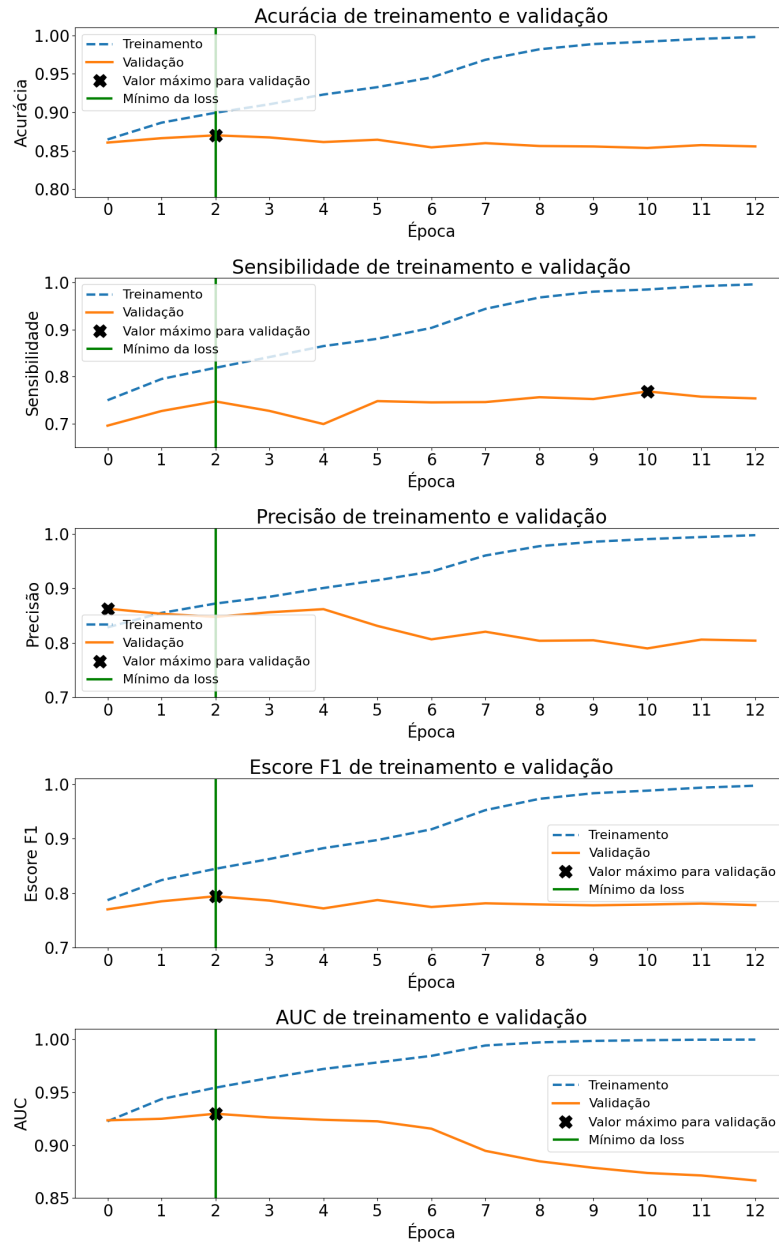
4.2.5 – Comparação entre arquiteturas

Com base nos valores médios das métricas para cada arquitetura avaliada, apresentadas nas tabelas 4.3, 4.4, 4.5 e 4.6; levando em conta todas as métricas é observada uma diferença entre elas para as arquiteturas como sendo inferior a 3 pontos percentuais (p.p.), onde a maior diferença se dá para a precisão seguida da sensibilidade, sendo ela menor que 1 p.p. para as demais.

A arquitetura DenseNet201 obteve os melhores resultados ao considerar as métricas *loss*, acurácia, AUC e precisão. Comparando os valores para as três primeiras métricas com os segundos melhores valores, os quais foram obtidos para a MobileNetV2, ResNet-101 e ResNet-101, respectivamente, observa-se que a diferença absoluta entre eles não ultrapassou 0,0021. Considerando a sensibilidade, precisão e escore F1, seus melhores valores em média foram obtidos, respectivamente, para as arquiteturas MobileNetV2, DenseNet201 e ResNet-101, havendo uma variação entre os modelos um pouco maior em comparação com as demais métricas, todavia não ultrapassando 0,0254, sendo ela para a precisão.

Observando os gráficos de evolução da métrica *loss*, nas Figuras 4.2, 4.4, 4.6 e 4.8, é perceptível que seu valor para o conjunto de validação atinge o ponto mínimo para todos os

Figura 4.9 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com EF em cada época na arquitetura DenseNet201.



Fonte: Autoral.

modelos em um número pequeno de épocas, sendo ele no máximo 8. Uma mesma tendência é observada para as demais métricas, apresentadas nas Figuras 4.3, 4.5, 4.7 e 4.9, onde seu ponto máximo é atingido em até 10 épocas.

É observado que a arquitetura em que se tem mais épocas até o ponto de mínimo da *loss* e máximo de outras métricas é a MobileNetV2. Nesse ponto, duas hipóteses são construídas: a primeira é que esse modelo, em comparação com os demais, é mais simples devido a sua menor profundidade e número de parâmetros (MathWorks, 2022). A segunda hipótese está relacionada ao conjunto de hiperparâmetros escolhidos, apresentados na Tabela 3.6, onde o *dropout* e a regularização L2, responsáveis por atenuar a complexidade da arquitetura durante o treinamento, obtidos foram maiores para esse modelo em comparação com os demais.

4.3 – Avaliação do raio de vizinhança ideal

Os resultados para a avaliação do raio de vizinhança ideal, sendo eles dados pelos valores de média de 5 rodadas utilizando a arquitetura MobileNetV2 para cada valor de raio são apresentados na Tabela 4.7, onde a escolha dessa arquitetura se deu em razão dos menores tempos de treinamento para essa arquitetura, conforme citado na Subsubseção 3.2.3.

Tabela 4.7 – Valores médios para as 5 rodadas das métricas de acordo com cada valor de percentual utilizado como raio de vizinhança.

Raio de vizinhança (%)	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3653	0,8491	0,7001	0,8247	0,7571	0,9038
2	0,3371	0,8653	0,7313	0,8469	0,7847	0,9175
3	0,3261	0,8715	0,7442	0,8549	0,7956	0,9214
4	0,3178	0,8713	0,7598	0,8418	0,7986	0,9234
5	0,3195	0,8700	0,7596	0,8383	0,7970	0,9232
6	0,3197	0,8718	0,7605	0,8434	0,7993	0,9249
7	0,3172	0,8709	0,7756	0,8298	0,8014	0,9273
8	0,3195	0,8735	0,7663	0,8430	0,8026	0,9271
9	0,3216	0,8688	0,7534	0,8397	0,7941	0,9268
10	0,3243	0,8647	0,7734	0,8157	0,7934	0,9264

Conforme descrito em Subsubseção 3.2.3, em razão do raio de vizinhança de 7% ter levado ao menor valor de *loss*, ele foi utilizado para a geração dos gráficos de recorrência utilizados nos treinamentos, cujos resultados serão apresentados na seção a seguir.

4.4 – Treinamento com gráficos de recorrência

Nas subseções a seguir estão apresentados os resultados para as arquiteturas de CNN treinadas com imagens de gráficos de recorrência. Para cada arquitetura, estão apresentadas as métricas obtidas para cada uma das 5 rodadas de treinamento e avaliação com conjunto de validação, além da média dessas métricas e seu desvio padrão. Ademais, são apresentadas as curvas de evolução da métrica *loss* e das demais métricas para o modelo com menor valor de *loss* para cada arquitetura.

4.4.1 – MobileNetV2

O resultado para as 5 rodadas treinamento com gráficos de recorrência considerando a arquitetura MobileNetV2 é apresentado na Tabela 4.8, na qual a rodada 2 levou ao menor valor para a métrica *loss*.

Tabela 4.8 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura MobileNetV2.

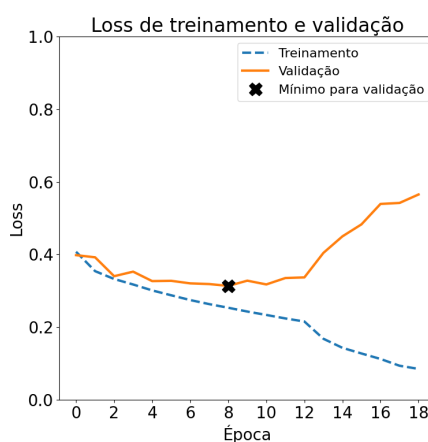
Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3328	0,8704	0,7423	0,8527	0,7937	0,9237
2	0,3143	0,8728	0,7514	0,8523	0,7987	0,9271
3	0,3179	0,8758	0,7468	0,8648	0,8015	0,9268
4	0,3195	0,8744	0,7535	0,8553	0,8012	0,9244
5	0,3245	0,8723	0,7319	0,8672	0,7938	0,9224
Média	0,3218	0,8731	0,7452	0,8585	0,7978	0,9249
Desvio Padrão	0,0072	0,0021	0,0086	0,0070	0,0038	0,0020

A Figura 4.10 mostra que para essa rodada, a métrica *loss* de validação têm uma queda mais acentuada nas duas primeiras épocas, tendo leve decréscimos até atingir seu valor mínimo na época 8, passando a crescer a partir da época 12, evidenciando a perda de capacidade de generalização.

Nela, a acurácia cresceu em torno de 0,03 da época inicial até atingir seu pico na época 7 havendo depois um leve decréscimo, mas mantendo-se acima dos valores iniciais, como pode ser observado na Figura 4.11.

Apesar da sensibilidade e precisão terem seu máximo nas épocas 9 e 4 para o conjunto de validação, respectivamente, o escore F1 têm seu máximo apenas na época 10, assim como a AUC.

Figura 4.10 – Evolução da métrica *loss* para os conjuntos de treino e validação com RP em cada época na arquitetura MobileNetV2.



Fonte: Autoral.

4.4.2 – ResNet-50

A Tabela 4.9 expõe as métricas encontradas para as 5 rodadas feitas com a arquitetura ResNet-50, onde a rodada 3 levou ao menor valor de *loss*, sendo ele 0,3054. Na Figura 4.12 é mostrado que ocorre para a ResNet-50 algo similar ao que ocorreu para a MobileNetV2, onde há uma queda maior na métrica *loss* de validação até seu ponto mínimo, que ocorre na época 7.

Tabela 4.9 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura ResNet-50.

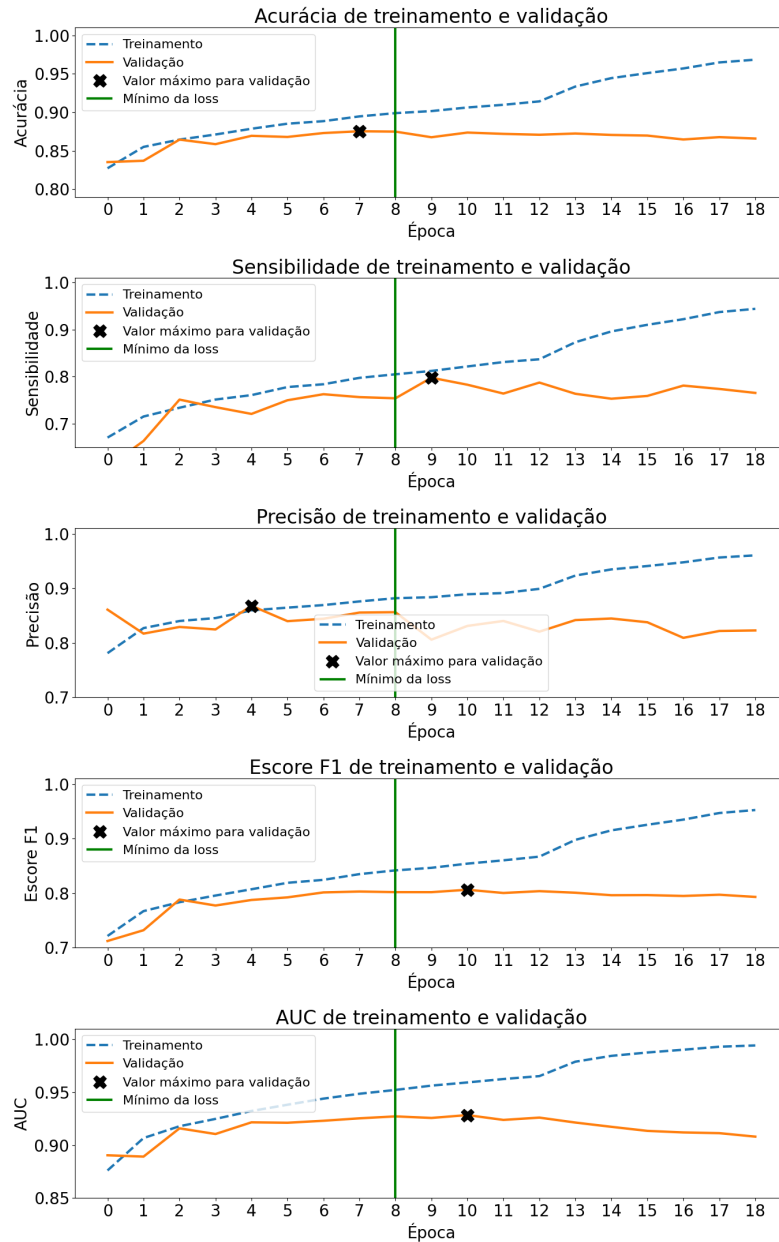
Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3144	0,8709	0,7906	0,8188	0,8045	0,9287
2	0,3139	0,8756	0,7753	0,8417	0,8071	0,9280
3	0,3054	0,8796	0,7526	0,8713	0,8076	0,9317
4	0,3328	0,8684	0,7730	0,8242	0,7978	0,9204
5	0,3111	0,8748	0,7592	0,8518	0,8029	0,9292
Média	0,3155	0,8738	0,7701	0,8416	0,8040	0,9276
Desvio Padrão	0,0103	0,0043	0,0149	0,0212	0,0040	0,0043

Após isso, há um crescimento dessa métrica até a época 12 para valores similares ao valor da primeira época, havendo então um crescimento maior a partir da época 11, evidenciando uma perda maior da capacidade de generalização do modelo.

As demais métricas, mostradas na Figura 4.13, mostram que seu crescimento é bem discreto, havendo um crescimento de 0,02 para a acurácia de validação, tendo seu pico na época 7, havendo um decréscimo discreto nas épocas seguintes.

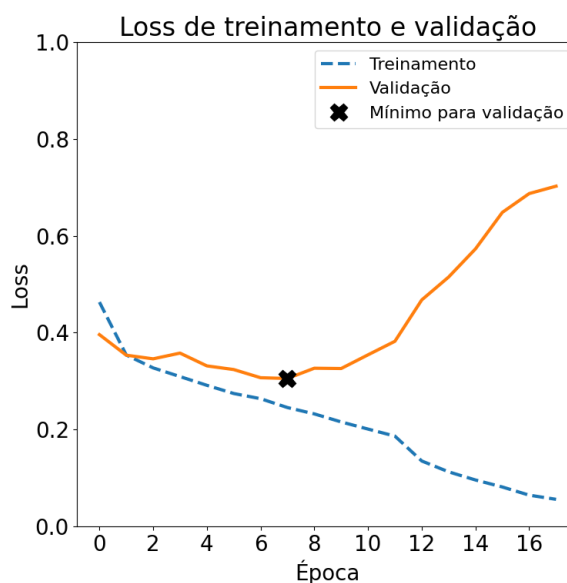
A sensibilidade e precisão têm seus picos, respectivamente, nas épocas 13 e 3, onde o compromisso entre elas é maximizado na época 10, sendo esse o máximo para a métrica escore

Figura 4.11 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com RP em cada época na arquitetura MobileNetV2.



Fonte: Autoral.

Figura 4.12 – Evolução da métrica *loss* para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-50.



Fonte: Autoral.

F1, a qual também tem um aumento em torno de 0,03, havendo após seu máximo uma leve diminuição, mas mantendo-se acima dos valores na época 0.

Por fim, a AUC para o conjunto de validação tem seu máximo na época 7, havendo então um decréscimo para valores inferiores ao valor inicial.

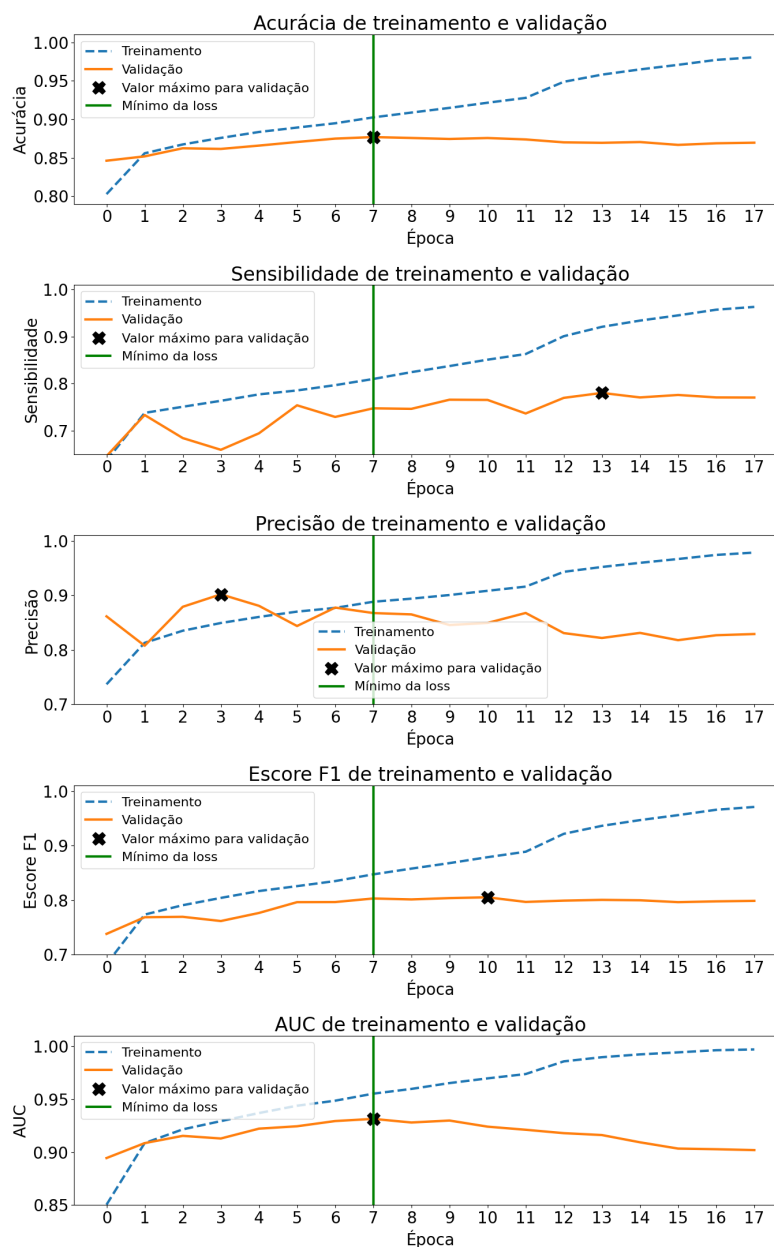
4.4.3 – ResNet-101

Para a arquitetura ResNet-101, o valor mínimo da métrica *loss* para a rodada 2, tendo ela um valor de 0,3095, conforme apresentado na Tabela 4.10. Para o modelo obtido ao fim dessa rodada, esse ponto mínimo de *loss* considerando o conjunto de validação ocorreu na época 6, mantendo então uma certa estável até a época 11, havendo então uma perda da generalização, como mostrado na Figura 4.14.

Tabela 4.10 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura ResNet-101.

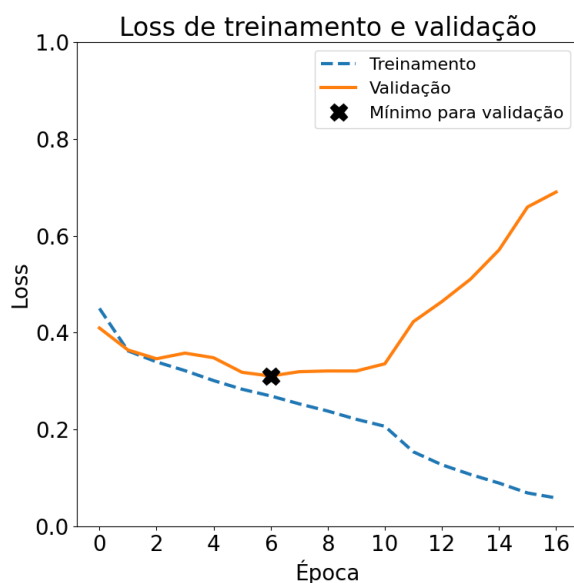
Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3253	0,8737	0,7395	0,8648	0,7973	0,9234
2	0,3095	0,8755	0,7737	0,8427	0,8067	0,9313
3	0,3117	0,8775	0,7627	0,8568	0,8070	0,9283
4	0,3194	0,8725	0,7404	0,8604	0,7959	0,9239
5	0,3215	0,8688	0,7576	0,8362	0,7950	0,9216
Média	0,3175	0,8736	0,7548	0,8522	0,8004	0,9257
Desvio Padrão	0,0067	0,0033	0,0147	0,0122	0,0060	0,0040

Figura 4.13 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-50.



Fonte: Autoral.

Figura 4.14 – Evolução da métrica *loss* para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-101.



Fonte: Autoral.

Observando as demais métricas, cuja evolução é apresentada na Figura 4.15, percebe que tanto a acurácia, quanto o score F1 e a AUC tiveram seu máximo para o conjunto de validação na época 6, na qual houve ocorre o mínimo da *loss*. Além disso, nesse caso, ainda houve um leve crescimento da precisão de validação, sendo seu pico na época 4 e havendo um decréscimo ao fim do treinamento. Já a sensibilidade teve uma tendência de crescimento com pico na época 6.

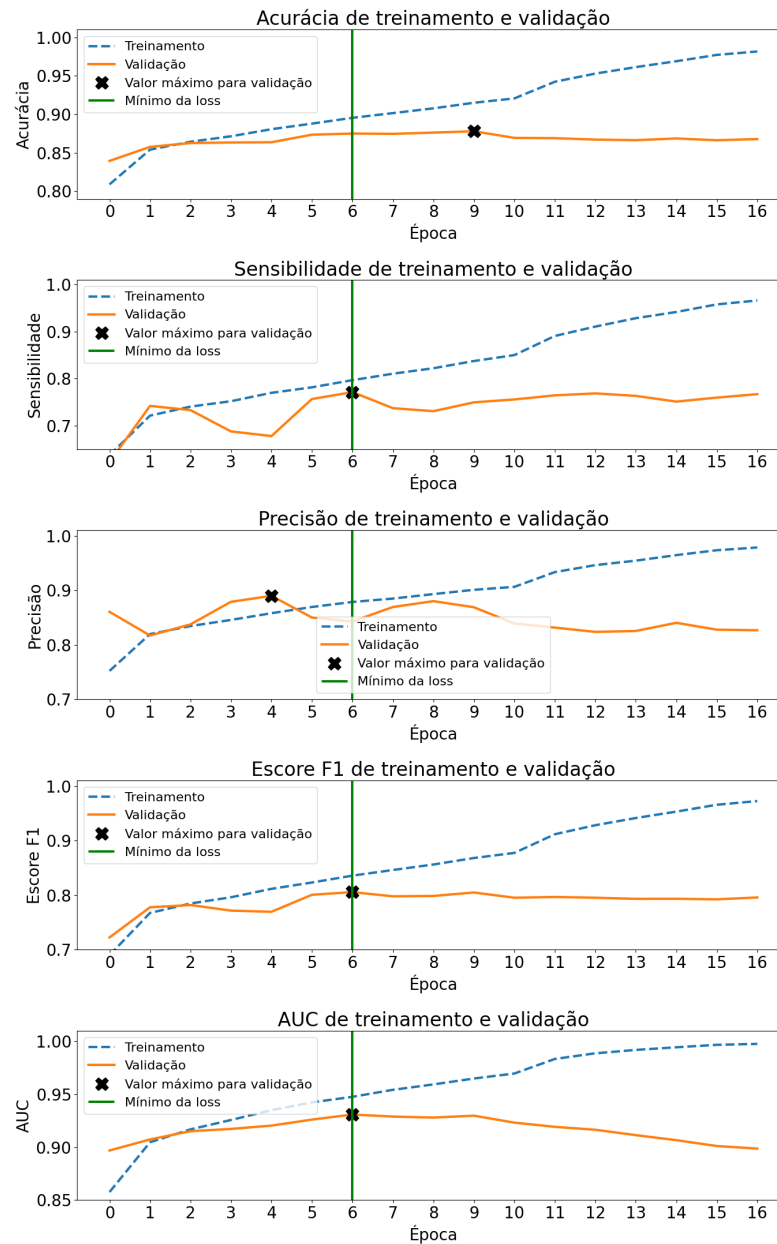
4.4.4 – DenseNet201

Por fim, as métricas obtidas nas 5 rodadas com a arquitetura DenseNet201 são apresentadas na Tabela 4.11. Nela, pode ser observado que a rodada 1 levou ao valor de *loss* de 0,3022 para o conjunto de validação, sendo ele o menor valor para essa arquitetura e considerando todas as arquiteturas e tipos de sinais testados.

A evolução da *loss* para esse modelo, apresentada na Figura 4.16, mostra que seu mínimo para o conjunto de validação é atingido rapidamente, na segunda época de treinamento, o que pode indicar uma alta complexidade do modelo e uma tendência ao sobreajuste, algo que ocorre a partir da época 6.

Considerando a Figura 4.17, a qual mostra a evolução das demais métricas, é observado que acurácia tem seu ponto máximo para o conjunto de validação na época 2, a qual ocorre o

Figura 4.15 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com RP em cada época na arquitetura ResNet-101.

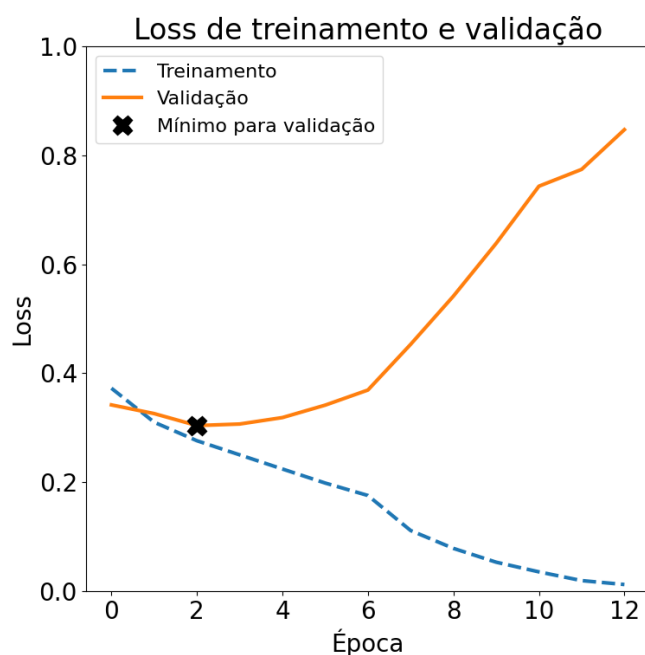


Fonte: Autoral.

Tabela 4.11 – Métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC) para o conjunto de validação nos cinco modelos treinados com RP e a arquitetura DenseNet201.

Rodada	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
1	0,3022	0,8819	0,7604	0,8715	0,8121	0,9335
2	0,3181	0,8719	0,7432	0,8563	0,7957	0,9253
3	0,3114	0,8796	0,7335	0,8884	0,8036	0,9323
4	0,3204	0,8756	0,7384	0,8717	0,7995	0,9255
5	0,3149	0,8783	0,7349	0,8829	0,8022	0,9247
Média	0,3149	0,8762	0,7447	0,8684	0,8015	0,9278
Desvio Padrão	0,0045	0,0030	0,0166	0,0189	0,0042	0,0036

Figura 4.16 – Evolução da métrica *loss* para os conjuntos de treino e validação com RP em cada época na arquitetura DenseNet201.



Fonte: Autoral.

mínimo da *loss*, assim como escore F1, havendo então um leve decaimento nessas métricas para as épocas seguintes.

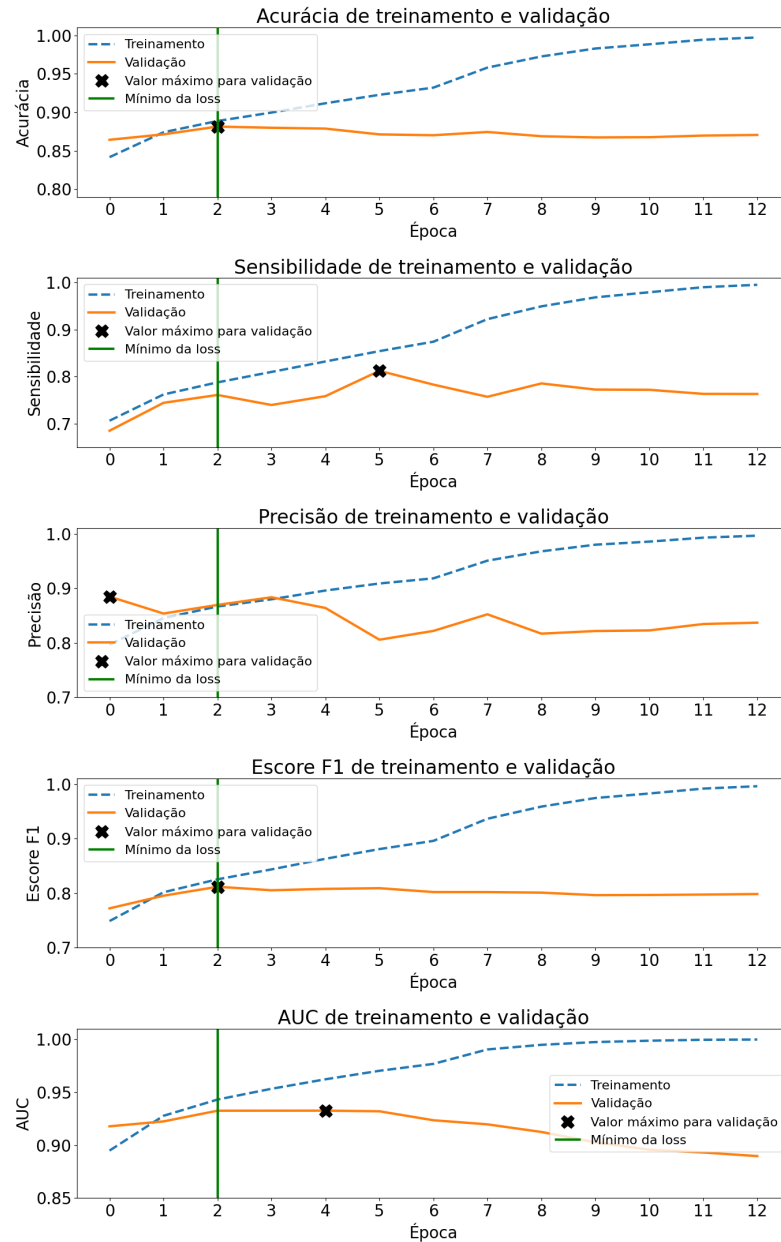
A sensibilidade possui uma tendência de crescimento, tendo seu pico na época 5 e então um leve decaimento, o que associado com o decréscimo da precisão, que teve seu pico na época 0, mostra que com o treinamento, há uma tendência em classificar mais exemplos como sendo da classe positiva.

Por fim, a AUC tem um maior crescimento até a época 2, tendo a partir daí um leve crescimento e pico na época 4. Após ela, há um decaimento até níveis inferiores ao do início do treinamento.

4.4.5 – Comparação entre arquiteturas

Ao comparar os valores das métricas para as diferentes arquiteturas, apresentados nas Tabelas 4.8, 4.9, 4.10 e 4.11, de forma análoga ao que foi feito para os modelos treinados com espaço de fase bidimensional, observa-se que a arquitetura DenseNet201 obteve em média os melhores valores para as métricas *loss*, acurácia, AUC e precisão, da mesma forma que ocorreu com o uso de imagens de espaço de fase. Para as três primeiras, as diferenças dos valores entre as arquiteturas não ultrapassaram 0,0069, enquanto para a precisão houve uma diferença de

Figura 4.17 – Evolução das métricas acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC para os conjuntos de treino e validação com RP em cada época na arquitetura DenseNet201.



Fonte: Autoral.

0,0268. Para a sensibilidade e escore F1, seus melhores valores médios foram encontrados para a arquitetura ResNet-50, havendo uma diferença máxima entre as médias de todos os modelos de 0,0254.

Considerando a evolução da métrica *loss*, observa-se um fenômeno similar ao que ocorreu para o modelo treinado com espaço de fase bidimensional: o valor mínimo dessa métrica é alcançado após um pequeno número de épocas, sendo esse valor 8, 7, 6 e 2 épocas, nessa ordem, para a MobileNetV2, ResNet-50, ResNet-101 e DenseNet201. Para as demais métricas, foi observada uma tendência similar para a maximização das métricas escore F1 e AUC, não havendo um padrão de diferenças definido para as demais métricas. Assim como para os modelos treinados com espaço de fase, pode-se presumir que esse comportamento de diferença entre as épocas de minimização da *loss* e maximização das demais métricas se deve às diferenças nos hiperparâmetros de regularização L2 e *dropout* ou a diferenças na complexidade das arquiteturas.

4.5 – Avaliação com conjunto de teste

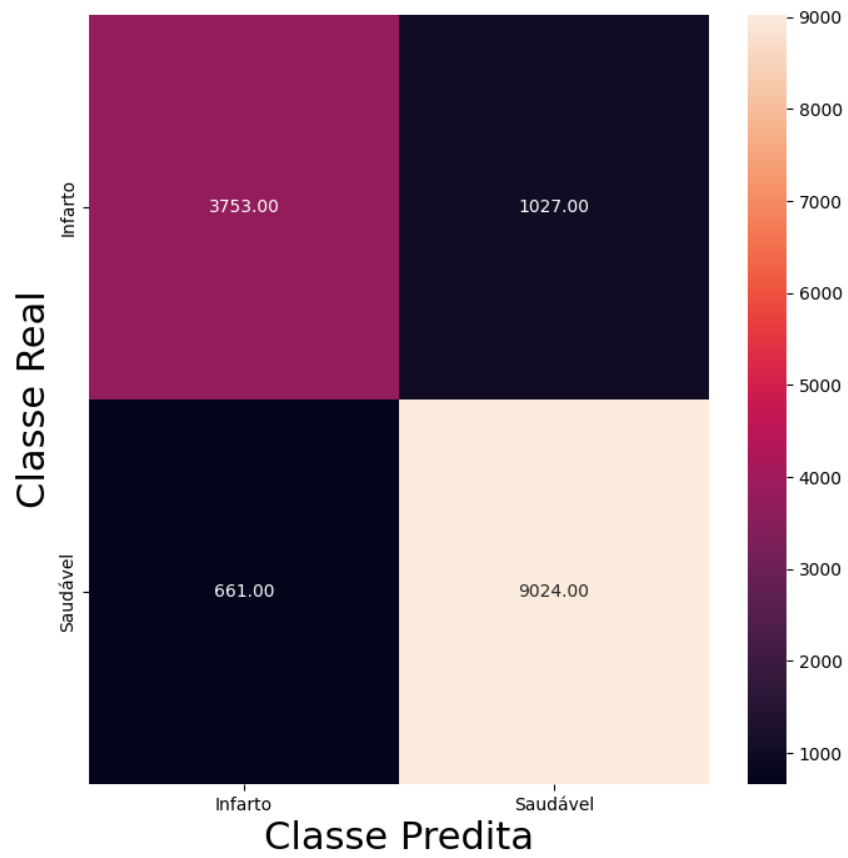
Com base nos resultados apresentados nas Seções 4.2 e 4.4, a arquitetura Densenet201 treinada com imagens de gráfico de recorrência, apresentado na subseção 4.4.4, foi a que levou ao menor valor da métrica *loss*, em média, para o conjunto de validação. Dos 5 modelos dessa arquitetura com esses sinais, o modelo 1 foi o que possuiu o menor valor dessa métrica, sendo ele então avaliado também para o conjunto de teste de imagens de gráficos de recorrência a fim de estimar o comportamento desse modelo para sinais desconhecidos.

Considerando o conjunto de teste, a matriz de confusão obtida pode ser visualizada na Figura 4.18. Ademais, foram obtidas as métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva ROC (AUC), cujos valores estão os valores apresentados na Tabela 4.12. Nela, a fim de comparação com outros trabalhos, também está apresentada especificidade obtida a partir da matriz de confusão.

Tabela 4.12 – Métricas considerando o conjunto de teste para o modelo 1 da arquitetura DenseNet201 cujo treinamento foi feito com imagens de gráficos de recorrência.

Loss	Acurácia	Sensibilidade	Especificidade	Precisão	Escore F1	AUC
0,2869	0,8833	0,7851	0,9317	0,8502	0,8164	0,9397

Figura 4.18 – Matriz de confusão obtida para os sinais do conjunto de teste.



Fonte: Autoral.

4.6 – Discussão

A etapa de avaliação do uso da técnica de *transfer learning*, apresentada nas tabelas 4.1 e 4.2 da subseção 4.1, indica que o uso de *transfer learning* é contraprodutivo para as imagens de espaço de fase bidimensional deste trabalho. Isso se dá em razão da maior média para a métrica *loss* e das menores médias para as demais métricas para os modelos treinados com a inicialização dos pesos com valores aleatórios, sendo essa suposição estendida aos gráficos de recorrência.

Uma hipótese para isso é a diferença entre essas imagens e as imagens utilizadas para o ajuste dos pesos dos modelos pré-treinados provenientes da base ImageNet (Lab, 2020), a qual tem como um dos focos pesquisas em aplicações de detecção e diferenciação de objetos (Deng *et al.*, 2009). Dessa forma, mesmo as primeiras camadas de extração de características dos modelos pré-treinados teriam que ser treinadas para seus pesos se adaptarem às imagens do presente trabalho, sendo a inicialização com pesos aleatórios mais produtiva nesses casos. Um ponto negativo desse tipo de inicialização é a necessidade de atualização dos pesos de todas as conexões do modelo, de forma que um custo maior de memória é necessário.

Considerando a análise do raio de vizinhança ideal disponível na Tabela 4.7 da Subsubseção 3.2.3, é observável que a diferença das métricas para os valores de raio de vizinhança entre 3% e 10% é, percentualmente, inferior a 1 p.p. para a métrica *loss*, acurácia, escore F1 e AUC, havendo uma variação maior para as métricas de sensibilidade e precisão.

Para essas duas métricas, se observa uma tendência de aumento da sensibilidade e redução da precisão à medida que o raio de vizinhança se aproxima do valor obtido como ideal. Isso pode ser um indicativo de que raios de vizinhança mais próximos do ideal estão relacionados com modelos que tendem a classificar exemplos como infarto, reduzindo falsos negativos e aumentando falsos positivos.

Ao examinar os resultados obtidos para o conjunto de validação, apresentados nas seções 4.2 e 4.4, é indicado que é possível identificar o infarto do miocárdio com o uso de imagens de espaço de fase e gráficos de recorrência aplicadas às arquiteturas de redes neurais convolucionais MobileNetV2, ResNet-50, ResNet-101 e DenseNet201; para a qual os gráficos de recorrência associados com a DenseNet201 levaram a um melhor desempenho.

Com base na base de dados utilizada e comparando o uso do conjunto de imagens de espaço de fase e de gráfico de recorrência de validação para as respectivas arquiteturas, observa-se que para todas elas o valor médio da métrica *loss* foi menor ao utilizar gráficos de

recorrência, conforme pode ser observado para cada par de sinais para as arquiteturas do trabalho nas Tabelas 4.3 e 4.8, 4.4 e 4.9, 4.5 e 4.10, 4.6 e 4.11.

Ao considerar as demais métricas nessas tabelas, percebe-se que com exceção da AUC para a arquitetura ResNet-101, as demais métricas tiveram melhores valores de média para os gráficos de recorrência. Já para os valores de desvio padrão dos 5 modelos observa-se uma tendência desses valores serem menores ao realizar o treinamento com imagens de espaço de fase, havendo exceções para a precisão na MobileNetV2 e para a sensibilidade e precisão para a ResNet-101.

A DenseNet201 foi, para os dois tipos de sinal, a arquitetura com melhor desempenho. Ao avaliar a diferença das médias das métricas para essa arquitetura pode-se observar que a diferença entre seus valores é inferior a, no máximo, 2,5 p.p. ao comparar percentualmente a diferença desses valores, conforme apresentado na Tabela 4.13.

Tabela 4.13 – Métricas e suas diferenças para cada tipo de sinal utilizado.

	Loss	Acurácia	Sensibilidade	Precisão	Escore F1	AUC
Espaço de fase bidimensional	0,3265	0,8665	0,7325	0,8493	0,7866	0,9269
Gráfico de recorrência	0,3134	0,8774	0,7421	0,8741	0,8026	0,9283
Diferença	-0,0131	0,0109	0,0096	0,0248	0,016	0,0014

Uma das causas dessa melhoria pode ser a necessidade, para alguns sinais, de representar a informação da dinâmica não linear do sistema em mais de duas dimensões, algo apresentado na Tabela 3.4, na qual para a maioria dos sinais a dimensão de imersão ideal foi 3. Com isso, ao utilizar uma dimensão de imersão inferior à ideal, como feito para o espaço de fase bidimensional, algumas informações relevantes para a classificação do sinal podem estar sendo perdidas.

A fim de avaliar o custo da melhoria nas métricas com o uso de imagens de gráfico de recorrência no lugar do uso do espaço de fases, se obteve o tempo de execução para a geração dos espaços de fase bidimensionais e gráficos de recorrência dos ciclos utilizados no presente trabalho, quantificados na Tabela 3.5, conforme apresentado na Figura 3.1, sem considerar a etapa de armazenamento da imagem.

Com essa quantificação foi obtido que o tempo para geração do total de imagens de espaço de fase bidimensional foi de 3202,8 segundos, enquanto para gráficos de recorrência foi de 3829,3 segundos, sendo em média, respectivamente, 33,36 ms e 39,88 ms para cada ciclo. Com isso, apesar da diferença entre o tempo de processamento dos dois tipos de sinais para toda

a base ser pouco mais de 10 minutos, ao considerar um cenário clínico, onde se tem um único sinal com uma quantidade de ciclos limitada, pode-se dizer o custo de tempo entre eles é similar.

Dessa forma, tendo em vista os pequenos ganhos obtidos no desempenho dos modelos ao utilizar imagens de gráficos de recorrência, esse tipo de sinal pode ser considerado como a opção mais adequada para a identificação de infarto do miocárdio em sinais cardíacos. Todavia, é importante levar em conta a diferença de custo computacional e em desempenho dos modelos com base no tipo de sinal utilizado em um cenário em que um processamento de uma maior quantidade de ciclos seja necessário, como para a criação de uma nova base similar à do presente para treinamento com esses dados.

Um comportamento a ser destacado em todas as curvas de *loss* apresentadas nas Figuras 4.2, 4.4, 4.6, 4.8, 4.10, 4.12, 4.14 e 4.16 é a pequena quantidade de épocas até que o ponto mínimo para essa métrica considerando o conjunto de validação, onde conforme as hipóteses propostas nas subseções 4.2.5 e 4.4.5, é proposto que esse comportamento está atrelado à complexidade das arquiteturas.

Visando avaliar essa hipótese, realizaram-se 5 rodadas de treinamento e validação de modelos com uma arquitetura "VGG8" utilizando gráficos de recorrência, sendo a arquitetura baseada na VGG16 (MathWorks, 2022), mas com menos camadas convolucionais. Os hiperparâmetros obtidos para o treinamento após sua otimização estão apresentados na Tabela 4.14.

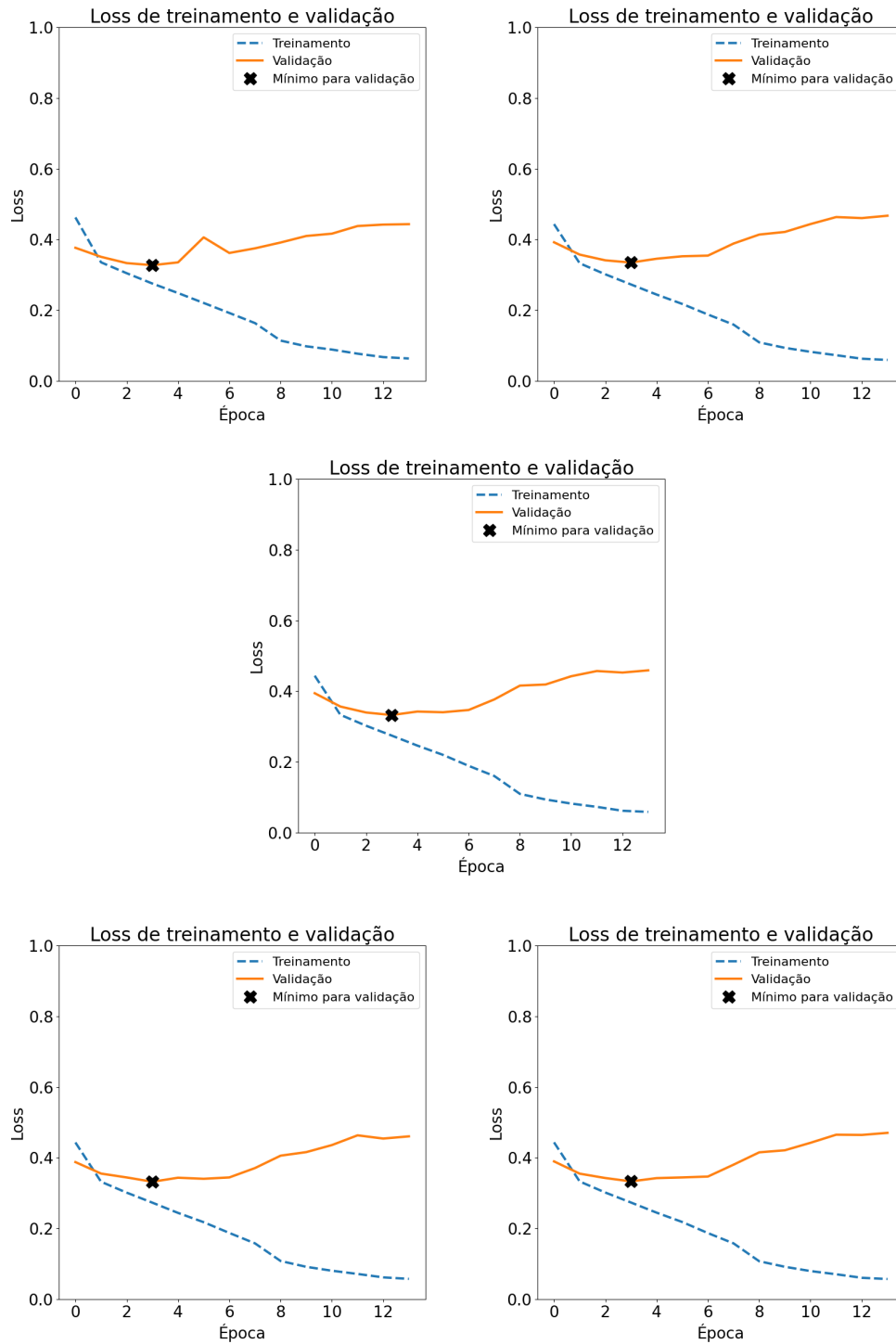
Tabela 4.14 – Valores de *Dropout*, regularização L2 e taxa de aprendizagem determinados na otimização de hiperparâmetros para a arquitetura VGG8 utilizando gráficos de recorrência.

Dropout	Regularização L2	Taxa de aprendizagem
0,2	$1 \cdot 10^{-6}$	$1,081 \cdot 10^{-4}$

Essa investigação, entretanto, mostrou que esse fenômeno também ocorre para essa arquitetura, conforme pode ser observado nas 5 curvas de *loss* obtidas e apresentadas na Figura 4.19, o que sugere que esse padrão não é uma exceção para a arquitetura. Outra hipótese está associada aos conjuntos de hiperparâmetros obtidos pelas otimizações para cada arquitetura e tipo de sinal, os quais podem não ter sido ideais em razão da quantidade de combinações testadas ter sido insuficiente para a obtenção de um conjunto adequado deles. Essa hipótese, no entanto, não é avaliada nesse trabalho.

Ao comparar a metodologia e resultados considerando a avaliação com o conjunto de teste do presente trabalho com os trabalhos citados na Seção 1.3 e posteriormente enumerados em

Figura 4.19 – Curvas para a métrica *loss* obtidas para a arquitetura VGG8 utilizando gráficos de recorrência considerando o conjunto de validação.



Fonte: Autoral.

Tabela 1.1, excluindo o trabalho de Hagiwara & Faust (2017), o qual fez uma análise qualitativa; nos dois primeiros trabalhos, os quais visam identificar sinais com arritmia e que utilizaram bases de dados com um número pequeno de sinais, não superando 140, são apresentados métodos com baixo custo computacional em que suas métricas estiveram em um mesmo nível ou superior às métricas de teste do presente trabalho.

Todavia, tanto em Amann *et al.* (2006) quanto em Roopaei *et al.* (2010), os parâmetros utilizados no método, como a quantidade de blocos da malha e limiares utilizados para a classificação são muito dependentes do conjunto de dados utilizado para sua determinação. Além disso, é possível que os sinais de espaço de fase de arritmia cardíaca possuam uma diferença maior em comparação às classes "Normal" e "Infarto", o que facilita sua separação e, por consequência, aumenta os valores das métricas de classificação.

Os métodos aplicados por Amann *et al.* (2006) e Roopaei *et al.* (2010) também são utilizados no trabalho de Costa *et al.* (2018) para a obtenção dos parâmetros de saída desses métodos, os utilizando como entrada para uma rede neural MLP que visa identificar o infarto do miocárdio, além de suas variações com relação à parede afetada. Costa *et al.* (2018), em contraste com o presente trabalho, possui métodos com um custo computacional menor, tanto pelo fato de se utilizarem as derivações eletrocardiográficas obtidas pelo método de Frank, de forma que sua reconstrução não é necessária, pela utilização de tempos de atraso fixos para todos os sinais para a reconstrução de seu espaço de fase bidimensional, quanto pela utilização de redes neurais mais rasas em comparação com as CNNs utilizadas no presente trabalho.

Todavia, ainda são necessárias algumas etapas adicionais para se gerar o modelo, como determinar o tamanho de bloco a ser utilizado para cada classificação de infarto, além do cálculo dos parâmetros apresentados por Roopaei *et al.* (2010) e a seleção da combinação adequada deles. Ao mesmo tempo, no presente trabalho é necessária a determinação do tempo de atraso e da dimensão de imersão ideais.

Ao levar em consideração as métricas para a identificação do infarto do miocárdio do presente trabalho com as obtidas por Costa *et al.* (2018) para o grupo sem diferenciação por parede afetada, observa-se que as métricas obtidas no segundo são superiores em comparação com o primeiro. Algumas hipóteses podem ser levantadas para essa diferença entre esses resultados: a primeira se deve à pequena quantidade de sinais utilizados para o conjunto de teste, onde apenas 40 sinais foram utilizados para fazer a estimativa da atuação do modelo, onde é possível que em razão da separação, o conjunto de teste contenha poucos sinais das subclasses

de infarto minoritárias, de forma que seria favorecida a classificação correta dos sinais desse conjunto.

A segunda hipótese está envolvida com a possibilidade da existência de um viés no modelo treinado em razão de não ser mencionado nenhum conjunto de validação, sendo possível que o conjunto de teste tenha sido utilizado como critério para a seleção da combinação dos parâmetros de espaço de fase a serem utilizados ou para a otimização de hiperparâmetros do modelo (Stuart & Peter, 2010).

Métricas superiores em comparação o presente trabalho também foram obtidas por Sridhar *et al.* (2021), o qual também utilizou sinais provenientes da base PTB, empregando, nesse caso, a derivação II e fazendo a segmentação desses sinais e, com isso, obtendo 20622 sinais de 2s, dos quais os 14 parâmetros não lineares mais significativos, de 45 obtidos, foram utilizados para sua classificação por um SVM. Entre as limitações desse trabalho estão o tamanho do conjunto de dados utilizado, a possibilidade da ocorrência de um vazamento de dados, já que não foi descrito nenhum método sobre a separação adequada dos conjuntos de treino e teste a fim de garantir que segmentos de 2s provenientes de um mesmo sinal não estejam nos dois conjuntos.

Ao comparar a metodologia empregada em Mathunjwa *et al.* (2022) com a do presente trabalho, algumas diferenças podem ser observadas quanto ao pré-processamento realizado nos sinais de ECG, considerando as etapas para geração dos gráficos de recorrência, onde em Mathunjwa *et al.* (2022) é somente feita a segmentação dos sinais de ECG, sendo então gerados seus gráficos de recorrência sem a limiarização, enquanto no presente trabalho há como etapa adicional a geração do VCG a partir do ECG, sendo então feita a segmentação dos sinais, a geração de 3 gráficos de recorrência para cada segmento e sua combinação em uma única imagem. Destaca-se também o emprego de dois modelos de CNN em série por Mathunjwa *et al.* (2022), sendo o primeiro responsável por detectar classes mais distintas de sinais, como fibrilação ventricular e ruídos, enquanto o segundo é responsável por classificar diferentes arritmias de sinais normais.

Ademais, enquanto Mathunjwa *et al.* (2022) fez uso de um conjunto de técnicas para aumento de dados (*data augmentation*) por meio de translações nas imagens ou espelhamento horizontal, além da alteração das intensidades dos canais RGB nas imagens de treinamento; no presente trabalho nenhuma técnica desse conjunto foi empregada, o que poderia ter impacto positivo nas métricas de classificação e na evolução da métrica *loss* de validação. Houve também,

em Mathunjwa *et al.* (2022), uma etapa para determinação do número de camadas para as duas CNNs de seu modelo.

Ao comparar os resultados apresentados na Tabela 4.12 com os resultados obtidos por Mathunjwa *et al.* (2022), percebe-se que as métricas em comum aos dois trabalhos são inferiores neste trabalho. Conforme já citado na discussão dos trabalhos de Amann *et al.* (2006) e Roopaei *et al.* (2010), uma possibilidade é que as classes de arritmia cardíaca envolvidas possuem maior distinção em relação às classes "Normal" e "Infarto", o que facilita sua separação e, por consequência, aumenta os valores das métricas de classificação.

Outra hipótese está relacionada com os 128 exames utilizados por Mathunjwa *et al.* (2022), já que não é mencionado que sinais, onde cada sinal corresponde a um segmento de 2 segundos de um exame, provenientes de um mesmo exame estão apenas em um dos conjuntos de treino, validação ou teste. Em razão de esses segmentos terem semelhanças para um mesmo sujeito há a possibilidade de estar havendo um vazamento de dados (*data leakage*) e com isso, a generalização da rede não estar sendo corretamente avaliada, havendo uma superestimação dessas métricas, conforme já discutido na Seção 1.3.

CONSIDERAÇÕES FINAIS

Neste trabalho é comparada a utilização de imagens de espaço bidimensional e de gráficos de recorrência, obtidos a partir das três derivações vetocardiográficas reconstruídas pelo método de Kors com base em um eletrocardiograma de 12 derivações (Kors *et al.*, 1990). Essas imagens são utilizadas como entrada para redes neurais convolucionais, para as quais se consideraram as arquiteturas MobileNetV2, ResNet-50, ResNet-101 e DenseNet201, sendo essas quatro arquiteturas também comparadas.

Considerando as imagens de espaço de fase bidimensional e a arquitetura MobileNetV2, foi avaliado o uso da técnica de *transfer learning*, onde foi observado que a inicialização com pesos aleatórios é mais produtiva em comparação com o uso dos pesos provenientes do treinamento com a base ImageNet, sendo, a partir daí, essa inicialização mais produtiva utilizada na criação dos demais modelos.

Para o treinamento dos modelos utilizando gráficos de recorrência foi necessário determinar dois parâmetros, sendo eles a dimensão de imersão adequada para o conjunto de sinais a ser utilizado e o raio de vizinhança. O primeiro foi determinado para cada sinal do qual seria gerado um gráfico de recorrência, enquanto o segundo foi selecionado com base em testes com um intervalo de valores, onde um subconjunto de sinais para cada um deles foi utilizado para o treinamento e teste com o conjunto de validação considerando a arquitetura MobileNetV2, sendo 7% considerado o melhor valor por levar ao menor valor da métrica *loss* para o teste.

Determinados esses parâmetros, foi realizado o treinamento dos modelos que utilizam as imagens de espaço de fase bidimensional e dos que utilizam gráficos de recorrência, onde para os dois casos, foi mostrado que a arquitetura DenseNet201 possui um melhor desempenho. Ao comparar o uso das imagens de espaço de fase com as de gráficos de recorrência, um maior desempenho é observado os modelos treinados com gráficos de recorrência. Também foram analisadas as diferenças entre as métricas considerando os dois tipos de sinais, sendo

elas inferiores a 0,002 para todas as métricas ao considerar suas médias para a arquitetura DenseNet201.

Um ponto comum aos modelos com menor valor de *loss* para todas as arquiteturas e sinais é o pequeno número de épocas necessário para que se atinja o valor mínimo dessa métrica para o conjunto de validação, onde esse fenômeno pode estar associado com a seleção de um conjunto de hiperparâmetros inadequado ou com a complexidade da arquitetura utilizada.

Devido à metodologia de se utilizar o modelo obtido para o ponto mínimo da métrica *loss* de validação, as métricas obtidas a partir da análise do melhor modelo com o conjunto de teste estiveram próximas das obtidas para o conjunto de validação, o que demonstra que o modelo obtido foi capaz de generalização. Essas métricas tiveram os valores de 0,8833, 0,7851, 0,9317, 0,8502, 0,8164 e 0,9397 para a acurácia, sensibilidade, especificidade, precisão, score F1 e AUC, respectivamente. Apesar de seus valores não terem sido superiores quando comparados com os demais trabalhos discutidos, o presente trabalho se destaca pela maior quantidade e diversidade de sinais utilizados para seu desenvolvimento, além de precauções para evitar qualquer vazamento de dados e superestimação dessas métricas.

Ademais, ao comparar o custo computacional necessário para a obtenção das imagens de espaço de fase bidimensional com o que se tem para gráficos de recorrência, foi observado que a diferença, em média, para os métodos é de 6,52 ms, o que representou, para o conjunto de dados utilizado no presente trabalho, uma diferença inferior a 11 minutos. Tendo esses resultados em mente, é possível avaliar qual processamento deve ser utilizado segundo a quantidade de ciclos a ser processada, considerando o custo computacional e a diferença, em média, nas métricas obtidas para a arquitetura DenseNet201 para os dois tipos de sinal.

Com base no que está apresentado nesta seção e no Capítulo 4, pode-se dizer que o objetivos do presente trabalho foram atingidos: houve a determinação da arquitetura mais adequada, dentre àquelas cuja avaliação foi proposta, e do tipo de sinal mais adequado, além das diferenças nas métricas obtidas para os dois tipos de sinal e seus custos computacionais.

5.1 – Contribuições da pesquisa

A maior contribuição desta pesquisa é a determinação dos gráficos de recorrência como tipo de sinal mais adequado para a identificação do infarto do miocárdio em comparação

com o espaço de fase. Além disso, também são avaliadas as diferenças em relação ao custo computacional para a obtenção de cada um deles e diferenças nos valores das métricas obtidas.

Adicionalmente, o trabalho também contribui com a observação de que para a maioria dos sinais do trabalho, o valor de dimensão ideal foi 3, algo não considerado em trabalhos anteriores, os quais fizeram uso apenas do espaço de fase bidimensional. Essa observação somada com as diferenças observadas entre as métricas para espaço de fase podem servir como ponto de partida para outras avaliações do impacto do uso de uma dimensão de imersão não ideal.

Outras contribuições deste trabalho são a de que o uso da técnica de *transfer learning* com os pesos da base ImageNet não é produtivo para os sinais do presente trabalho e a determinação da arquitetura DenseNet201 como a arquitetura que leva a um melhor desempenho na classificação ao considerar as arquiteturas avaliadas. Por fim, o trabalho levanta a hipótese, entretanto, que essas arquiteturas tenham alta complexidade para a classificação dos sinais deste trabalho, sendo mostrado que outras arquiteturas mais simples devem ser avaliadas.

5.2 – Sugestões para trabalhos futuros

Sugestões para trabalhos futuros são propostas com base em alguns pontos não avaliados no presente trabalho, mas presentes em trabalhos relacionados, como o de Mathunjwa *et al.* (2022). O primeiro deles é a avaliação do uso de técnicas de aumento de dados, onde deve ser avaliado se o processo de aumento de dados em imagens de espaço de fase e gráficos de recorrência é algo viável ou se a informação presente nessas imagens, relevante para o diagnóstico das patologias, pode ser comprometida. Já o segundo está relacionado com o tamanho da arquitetura, no qual diferentes modelos com valores menores de profundidade podem ser testados de maneira progressiva a fim de determinar a profundidade ideal. Com o uso de uma diversidade maior de dados e arquiteturas menos complexas, há a possibilidade de reduzir o comportamento observado para as curvas de evolução da métrica *loss* para o conjunto de validação observada nesse trabalho. Outra proposta para trabalho futuros que visa reduzir esse comportamento é aumentar a quantidade de testes de conjuntos de hiperparâmetros.

Além disso, com base na metodologia empregada por Costa *et al.* (2018), um estudo onde sejam avaliados os desempenho modelos conforme a parede cardíaca acometida pelo

infarto poderia esclarecer mais sobre as vantagens e desvantagens desse método como forma de identificação de infartos.

Um trabalho proposto com base nos valores obtidos para a Tabela 3.4 consiste em investigar a característica dos sinais que tiveram dimensão de imersão ideal igual a 4 a fim de determinar a causa dessas discrepâncias e, caso sejam considerados dados legítimos, se alguma informação diagnóstica pode ser extraída deles, de maneira similar a casos reportados por Osborne & Overbay (2019).

Por fim, é sugerido também um estudo para análise de classificações propostas por esse método com avaliações de cardiologistas especializados em diagnóstico de infarto, a fim de analisar quais critérios nos sinais de eletrocardiograma podem ter impactado na predição para cada sinal dada pelo modelo.

5.3 – Trabalhos Publicados

- ▣ de Sousa, R. D.; Barbosa, J. R.; de Oliveira, A. B. S.; Regis, C. D. M. *Analysis of the optimal time delay for reconstruction of phase space of vcg signals to identify anterior wall infarction*. In: **Simpósio Brasileiro de Telecomunicações e Processamento de Sinais**, 2021.
- ▣ Silva, E. T. A.; Júnior, D. M. D. S.; de Sousa, R. D.; Regis, C. D. M. Classificação do infarto de miocárdio a partir da extração de parâmetros de espaços de fase com o uso dos Descritores de Haralick. In: **Simpósio Brasileiro de Telecomunicações e Processamento de Sinais**, 2022, Santa Rita do Sapucaí - MG.

REFERÊNCIAS BIBLIOGRÁFICAS

- Acharya, U Rajendra, Fujita, Hamido, Sudarshan, Vidya K, Oh, Shu Lih, Adam, Muhammad, Koh, Joel EW, Tan, Jen Hong, Ghista, Dhanjoo N, Martis, Roshan Joy, Chua, Chua K, *et al.* 2016. Automated detection and localization of myocardial infarction using electrocardiogram: a comparative study of different leads. *Knowledge-Based Systems*, **99**, 146–156.
- Acharya, U Rajendra, Fujita, Hamido, Oh, Shu Lih, Hagiwara, Yuki, Tan, Jen Hong, & Adam, Muhammad. 2017. Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Information Sciences*, **415**, 190–198.
- Aggarwal, Charu C, *et al.* 2018. Neural networks and deep learning. *Springer*, **10**, 978–3.
- Albawi, Saad, Mohammed, Tareq Abed, & Al-Zawi, Saad. 2017. Understanding of a convolutional neural network. *Pages 1–6 of: 2017 International Conference on Engineering and Technology (ICET)*. Ieee.
- Amann, Anton, Tratnig, Robert, & Unterkofler, Karl. 2006. Detecting ventricular fibrillation by time-delay methods. *IEEE Transactions on Biomedical Engineering*, **54**(1), 174–177.
- Andrade, JP de, Piegas, Leopoldo Soares, Timerman, Ari, Feitosa, Gilson, Rossi Neto, JM, Nicolau, José Carlos, & Mattos, Luiz Alberto. 2009. IV Diretriz da Sociedade Brasileira de Cardiologia sobre tratamento do infarto agudo do miocárdio com supradesnível do segmento ST. *Arq Bras Cardiol*, **93**(6 Suppl 2), e179–264.
- Blaus, Bruce. 2014. Medical gallery of Blausen medical 2014. *Wiki J. Med*, **1**(10).
- Bousseljot, R, Kreiseler, D, & Schnabel, A. 1995. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. *Biomedizinische Technik/Biomedical Engineering*, **40**(s1), 317–318.

-
- Brasil, OPAS/OMS. 2017. *OPAS/OMS Brasil - Doenças Cardiovasculares*. Disponível em: http://www.paho.org/bra/index.php?option=com_content&view=article&id=5253:doencas-cardiovasculares&Itemid=839. Acesso em: Abril de 2022.
- Bueno, Danilo Potengy. 2011. O ECG no prognóstico do IAM. *Revista da Sociedade de Cardiologia*.
- Carreiras, Carlos, Alves, Ana Priscila, Lourenço, André, Canento, Filipe, Silva, Hugo, Fred, Ana, et al. 2015. *BioSPPy: Biosignal Processing in Python*. [Online; accessed <today>].
- Chen, Yen-Lin, Chiang, Yuan, Chiu, Pei-Hsin, Huang, I-Chen, Xiao, Yu-Bai, Chang, Shu-Wei, & Huang, Chang-Wei. 2021. High-Dimensional Phase Space Reconstruction with a Convolutional Neural Network for Structural Health Monitoring. *Sensors*, **21**(10), 3514.
- Chollet, François, et al. 2015. *Keras*. <https://keras.io>.
- Chollet, Francois. 2021. *Deep learning with Python*. Simon and Schuster.
- Cortez, Daniel, Sharma, Nandita, Devers, Christopher, Devers, Erin, & Schlegel, Todd T. 2014. Visual transform applications for estimating the spatial QRS–T angle from the conventional 12-lead ECG: Kors is still most Frank. *Journal of electrocardiology*, **47**(1), 12–19.
- Cortez, DL, & Schlegel, TT. 2010. When Deriving the Spatial QRS-T Angle from the 12-lead Electrocardiogram, which Transform is more Frank: Regression or Inverse Dower? *Journal of Electrocardiology*, **4**(43), 302–309.
- Costa, Cecília. 2017. *Identificação do Infarto do Miocárdio por meio da Análise do Cardiograma Vetorial e da Reconstrução do Espaço de Fases*. M.Phil. thesis, Instituto Federal da Paraíba.
- Costa, Cecília M, Silva, Ittalo S, de Sousa, Rafael D, Hortegal, Renato A, & Regis, Carlos Danilo M. 2018. The association between reconstructed phase space and Artificial Neural Networks for vectorcardiographic recognition of myocardial infarction. *Journal of electrocardiology*, **51**(3), 443–449.
- Daoud, Maisa, & Mayo, Michael. 2019. A survey of neural network-based cancer prediction models from microarray data. *Artificial intelligence in medicine*, **97**, 204–214.
- de Cardiologia, Sociedade Brasileira. 2003. *Diretriz de interpretação de eletrocardiograma de repouso*,. Tech. rept. Arquivos brasileiros de Cardiologia.

-
- Dehnavi, Ali Reza Mehri, Farahabadi, Iman, Rabbani, Hossain, Farahabadi, Amin, Mahjoob, Mohamad Parsa, & Dehnavi, Nasser Rajabi. 2011. Detection and classification of cardiac ischemia using vectorcardiogram signal via neural network. *Journal of research in medical sciences: the official journal of Isfahan University of Medical Sciences*, **16**(2), 136.
- Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, & Fei-Fei, Li. 2009. Imagenet: A large-scale hierarchical image database. *Pages 248–255 of: 2009 IEEE conference on computer vision and pattern recognition*. Ieee.
- Dokur, Zümray, & Ölmez, Tamer. 2001. ECG beat classification by a novel hybrid neural network. *Computer methods and programs in biomedicine*, **66**(2-3), 167–181.
- Dower, Gordon E, Machado, H Bastos, & Osborne, JA. 1980. On deriving the electrocardiogram from vectorcardiographic leads. *Clinical cardiology*, **3**(2), 87–95.
- Eckmann, Jean-Pierre, Kamphorst, S Oliffson, Ruelle, David, *et al.* 1995. Recurrence plots of dynamical systems. *World Scientific Series on Nonlinear Science Series A*, **16**, 441–446.
- Edenbrandt, Lars, & Pahlm, Olle. 1988. Vectorcardiogram synthesized from a 12-lead ECG: superiority of the inverse Dower matrix. *Journal of electrocardiology*, **21**(4), 361–367.
- Faouzi, Johann, & Janati, Hicham. 2020. pyts: A Python Package for Time Series Classification. *Journal of Machine Learning Research*, **21**(46), 1–6.
- Frank, Ernest. 1956. An accurate, clinically practical system for spatial vectorcardiography. *circulation*, **13**(5), 737–749.
- Fraser, Andrew M, & Swinney, Harry L. 1986. Independent coordinates for strange attractors from mutual information. *Physical review A*, **33**(2), 1134.
- Gardner, Matt W, & Dorling, SR. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, **32**(14-15), 2627–2636.
- Gaziano, Thomas, Reddy, K Srinath, Paccaud, Fred, Horton, Sue, & Chaturvedi, Vivek. 2006. Cardiovascular disease. *In: Disease Control Priorities in Developing Countries. 2nd edition*. The International Bank for Reconstruction and Development/The World Bank.

-
- Ginefra, Paulo, Barbosa, Eduardo Correa, Ribeiro, Ricardo, Bomfim, Alfredo, Boghossian, Silvia Helena Cardoso, Barbosa, Paulo Roberto Benchimol, & Veloso, Henrique. 2006. Infarto do miocárdio da parede inferior associado ao hemibloqueio esquerdo anterior: um dilema na clínica cardiológica. *Revista da SOCERJ*.
- Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaron. 2016. *Deep learning*. MIT press.
- Govindan, RB, Narayanan, K, & Gopinathan, MS. 1998. On the evidence of deterministic chaos in ECG: Surrogate and predictability analysis. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **8**(2), 495–502.
- Guyton, Arthur C, & John, E. 2011. Hall textbook of medical physiology. *Jackson, Mississippi: Saunders Elsevier*.
- HacksCooking. 2013. *e-Health Sensor Platform V2. 0 for Arduino and Raspberry Pi*. Disponível em: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical>. Acesso em: Março de 2018.
- Hagiwara, Yuki, & Faust, Oliver. 2017. Nonlinear analysis of coronary artery disease, myocardial infarction, and normal ECG signals. *Journal of Mechanics in Medicine and Biology*, **17**(07), 1740006.
- Hagiwara, Yuki, Fujita, Hamido, Oh, Shu Lih, Tan, Jen Hong, San Tan, Ru, Ciaccio, Edward J, & Acharya, U Rajendra. 2018. Computer-aided diagnosis of atrial fibrillation based on ECG signals: A review. *Information Sciences*, **467**, 99–114.
- Hammad, Mohamed, Alkinani, Monagi H, Gupta, BB, El-Latif, Abd, & Ahmed, A. 2021. Myocardial infarction detection based on deep neural network on imbalanced data. *Multimedia Systems*, 1–13.
- Hampton, John. 2014. *150 ECG Casos Clínicos*. Elsevier Brasil.
- Haykin, Simon O. 2008. *Neural Networks and Learning Machines*. 3rd edn. Ontario: Pentrice Hall.
- Haykin, Simon S, et al. 2009. *Neural networks and learning machines/Simon Haykin*.

-
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. 2016. Deep residual learning for image recognition. *Pages 770–778 of: Proceedings of the IEEE conference on computer vision and pattern recognition.*
- Huang, Gao, Liu, Zhuang, Van Der Maaten, Laurens, & Weinberger, Kilian Q. 2017. Densely connected convolutional networks. *Pages 4700–4708 of: Proceedings of the IEEE conference on computer vision and pattern recognition.*
- Ilakiyaselvan, N, Khan, A Nayeemulla, & Shahina, A. 2020. Deep learning approach to detect seizure using reconstructed phase space images. *Journal of Biomedical Research*, **34**(3), 240.
- Isin, Ali, & Ozdalili, Selen. 2017. Cardiac arrhythmia detection using deep learning. *Procedia computer science*, **120**, 268–275.
- Jiang, Fei, Jiang, Yong, Zhi, Hui, Dong, Yi, Li, Hao, Ma, Sufeng, Wang, Yilong, Dong, Qiang, Shen, Haipeng, & Wang, Yongjun. 2017. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, **2**(4), 230–243.
- JM, Balajee, *et al.* 2018. Data wrangling and data leakage in machine learning for healthcare. *JETIR- International Journal of Emerging Technologies and Innovative Research.*
- Kennel, Matthew B, Brown, Reggie, & Abarbanel, Henry DI. 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, **45**(6), 3403.
- Kermany, Daniel S, Goldbaum, Michael, Cai, Wenjia, Valentim, Carolina CS, Liang, Huiying, Baxter, Sally L, McKeown, Alex, Yang, Ge, Wu, Xiaokang, Yan, Fangbing, *et al.* 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, **172**(5), 1122–1131.
- Khan, Sibghatullah I, & Pachori, Ram Bilas. 2021. Automated detection of posterior myocardial infarction from vectorcardiogram signals using Fourier–Bessel series expansion based empirical wavelet transform. *IEEE Sensors Letters*, **5**(5), 1–4.
- Kors, JA, Van Herpen, G, Sittig, AC, & Van Bommel, JH. 1990. Reconstruction of the Frank vectorcardiogram from standard electrocardiographic leads: diagnostic comparison of different methods. *European Heart Journal*, **11**(12), 1083–1092.

-
- Lab, Stanford Vision. 2020. *ImageNet*. Available in: <https://image-net.org/about.php>. Acesso em: Julho de 2022.
- Lima, Flávia Emília Leite de, Menezes, Tarciana Nobre de, Tavares, Miriam Paulichenco, Szarfarc, Sophia Cornbluth, & Fisberg, Regina Mara. 2000. Fatty acids and cardiovascular diseases: a review. *Revista de Nutrição*, **13**(2), 73–80.
- Ljung, Lennart, & Glad, Torkel. 1994. *Modeling of dynamic systems*. Philadelphia, PA: Prentice Hall.
- Macfarlane, Peter W, & Edenbrandt, L. 1991. 12-lead vectorcardiography in ischemic heart disease. *Journal of electrocardiology*, **24**, 188–193.
- MacKay, David JC, & Mac Kay, David JC. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.
- Mathunjwa, Bhekumuzi M, Lin, Yin-Tsong, Lin, Chien-Hung, Abbod, Maysam F, & Shieh, Jiann-Shing. 2021. ECG arrhythmia classification by using a recurrence plot and convolutional neural network. *Biomedical Signal Processing and Control*, **64**, 102262.
- Mathunjwa, Bhekumuzi M, Lin, Yin-Tsong, Lin, Chien-Hung, Abbod, Maysam F, Sadrawi, Muammar, & Shieh, Jiann-Shing. 2022. ECG Recurrence Plot-Based Arrhythmia Classification Using Two-Dimensional Deep Residual CNN Features. *Sensors*, **22**(4), 1660.
- MathWorks, The. 2022. *Pretrained Deep Neural Networks*. Available in: <https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html>. Acesso em: Julho de 2022.
- Moffa, Paulo Jorge. 2001. *Eletrocardiograma normal e patológico*. 1 edn. Roca Brasil.
- Myers, Audun D, Yesilli, Melih, Tymochko, Sarah, Khasawneh, Firas, & Munch, Elizabeth. 2020. Teaspoon: A comprehensive python package for topological signal processing. In: *TDA {&} Beyond*.
- Narin, Ali, Kaya, Ceren, & Pamuk, Ziynet. 2021. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, **24**(3), 1207–1220.

-
- Nielsen, Finn Årup. 2018. Linking imagenet wordnet synsets with wikidata. *Pages 1809–1814 of: Companion Proceedings of the The Web Conference 2018*.
- Organization, World Health. 2021. *Cardiovascular diseases (CVDs)s*. Available in: [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)). Acesso em: Abril de 2022.
- Osborne, Jason W, & Overbay, Amy. 2019. The power of outliers (and why researchers should always check for them). *Practical Assessment, Research, and Evaluation*, **9**(1), 6.
- O’Shea, Keiron, & Nash, Ryan. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Pastore, CA, Pinho, JA, Pinho, C, Samesima, N, Pereira Filho, HG, Kruse, JCL, Paixão, A, Pérez-Riera, AR, Ribeiro, AL, Oliveira, CAR, *et al*. 2016. III Diretrizes da Sociedade Brasileira de Cardiologia sobre análise e emissão de laudos eletrocardiográficos. *Arquivos Brasileiros de Cardiologia*, **106**(4), 1–23.
- Pollack, Charles V, Amin, Alpesh, Wang, Tracy, Deitelzweig, Steven, Cohen, Marc, Slattery, David, Fanikos, John, DiLascia, Christopher, Tuder, Regan, & Kaatz, Scott. 2020. Contemporary NSTEMI management: the role of the hospitalist. *Hospital Practice*, **48**(1), 1–11.
- Ramachandran, Prajit, Zoph, Barret, & Le, Quoc V. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
- Roopaei, Mehdi, Boostani, Reza, Sarvestani, R Rohani, Taghavi, Mohammad Ali, & Azimifar, Zohreh. 2010. Chaotic based reconstructed phase space features for detecting ventricular fibrillation. *Biomedical Signal Processing and Control*, **5**(4), 318–327.
- Rubin, Jonathan, Parvaneh, Saman, Rahman, Asif, Conroy, Bryan, & Babaeizadeh, Saeed. 2017. Densely connected convolutional networks and signal quality analysis to detect atrial fibrillation using short single-lead ECG recordings. *Pages 1–4 of: 2017 Computing in cardiology (cinc)*. IEEE.
- Samala, Ravi K, Chan, Heang-Ping, Hadjiiski, Lubomir, & Helvie, Mark A. 2021. Risks of feature leakage and sample size dependencies in deep feature extraction for breast mass classification. *Medical physics*, **48**(6), 2827–2837.

-
- Sandler, Mark, Howard, Andrew, Zhu, Menglong, Zhmoginov, Andrey, & Chen, Liang-Chieh. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. *Pages 4510–4520 of: Proceedings of the IEEE conference on computer vision and pattern recognition.*
- Santos, Laurita dos, Barroso, Joaquim J, Godoy, Moacir F de, Macau, Elbert EN, & Freitas, Ubiratan S. 2014. Recurrence quantification analysis as a tool for discrimination among different dynamics classes: The heart rate variability associated to different age groups. *Pages 125–136 of: Translational recurrences.* Springer.
- Snoek, Jasper, Larochelle, Hugo, & Adams, Ryan P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, **25**.
- Sridhar, Chaitra, Lih, Oh Shu, Jahmunah, V, Koh, Joel EW, Ciaccio, Edward J, San, Tan Ru, Arunkumar, N, Kadry, Seifedine, & Rajendra Acharya, U. 2021. Accurate detection of myocardial infarction using non linear features with ECG signals. *Journal of Ambient Intelligence and Humanized Computing*, **12**(3), 3227–3244.
- Stewart, Ian. 2000. The Lorenz attractor exists. *Nature*, **406**(6799), 948–949.
- Strodthoff, Nils, Wagner, Patrick, Schaeffter, Tobias, & Samek, Wojciech. 2020. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. *arXiv preprint arXiv:2004.13701*.
- Strogatz, Steven H. 1994. *Nonlinear dynamics and chaos*. Philadelphia, PA: Westview Press.
- Stuart, Russell, & Peter, Norvig. 2010. *Artificial Intelligence A Modern Approach Third Edition*.
- Sukienik, Bernardo. 2015. *Atlas de eletrocardiografia*. Elsevier Brasil.
- Takens, Floris. 1981. Detecting strange attractors in turbulence. *Pages 366–381 of: Dynamical systems and turbulence, Warwick 1980*. Springer.
- Takens, Floris. 1983. Invariants related to dimension and entropy. *Atas do*, **13**, 353–359.
- Tampu, Iulian Emil, Eklund, Anders, & Haj-Hosseini, Neda. 2022. Inflation of test accuracy due to data leakage in deep learning-based classification of OCT images. *Scientific Data*, **9**(1), 580.
- TensorFlow. 2022a. *Classification on imbalanced data*. Available in: https://www.tensorflow.org/tutorials/structured_data/imbalanced_data. Acesso em: Julho de 2022.

- TensorFlow. 2022b. *Transfer learning and fine-tuning*. Available in: https://www.tensorflow.org/tutorials/images/transfer_learning. Acesso em: Julho de 2022.
- TensorFlow. 2023. *Introduction to the Keras Tune*. Available in: https://www.tensorflow.org/tutorials/keras/keras_tuner. Acesso em: Agosto de 2023.
- Thygesen, Kristian, Alpert, Joseph S, & White, Harvey D. 2007. Universal definition of myocardial infarction. *circulation*, **116**(22), 2634–2653.
- Vafaie, Mehrshad. 2016. State-of-the-art diagnosis of myocardial infarction. *Diagnosis*, **3**(4), 137–142.
- Vargas, Ana Caroline Gomes, Paes, Aline, & Vasconcelos, Cristina Nader. 2016. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: *Proceedings of the xxix conference on graphics, patterns and images*, vol. 1. sn.
- Vieira, Vinícius Jefferson Dias. 2014. *Avaliação de distúrbios da voz por meio de análise de quantificação de recorrência*. M.Phil. thesis, Instituto Federal da Paraíba.
- Wagner, Patrick, Strodthoff, Nils, Bousseljot, Ralf-Dieter, Kreiseler, Dieter, Lunze, Fatima I, Samek, Wojciech, & Schaeffter, Tobias. 2020. PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, **7**(1), 1–15.
- White, Harvey D, & Chew, Derek P. 2008. Acute myocardial infarction. *The Lancet*, **372**(9638), 570–584.
- Yagis, Ekin, Atnafu, Selamawet Workalemahu, García Seco de Herrera, Alba, Marzi, Chiara, Sceda, Riccardo, Giannelli, Marco, Tessa, Carlo, Citi, Luca, & Diciotti, Stefano. 2021. Effect of data leakage in brain MRI classification using 2D convolutional neural networks. *Scientific reports*, **11**(1), 22544.
- Yang, Hui, Bukkapatnam, Satish TS, & Komanduri, Ranga. 2012. Spatiotemporal representation of cardiac vectorcardiogram (VCG) signals. *Biomedical engineering online*, **11**(1), 16.
- Yang, Hui, Chen, Yun, Kan, Chen, & Liu, Gang. 2013. Spatiotemporal differentiation of myocardial infarctions. *IEEE Transactions on Automation Science and Engineering*, **10**(4), 10.

- Yang, Qigui, Chen, Guangrong, & Zhou, Tianshou. 2006. A unified Lorenz-type system and its canonical form. *International Journal of Bifurcation and Chaos*, **16**(10), 2855–2871.
- Yasue, Hirofumi, Nakagawa, Hitoshi, Itoh, Teruhiko, Harada, Eisaku, & Mizuno, Yuji. 2008. Coronary artery spasm—clinical features, diagnosis, pathogenesis, and treatment. *Journal of cardiology*, **51**(1), 2–17.
- Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, & Lipson, Hod. 2014. How transferable are features in deep neural networks? *Advances in neural information processing systems*, **27**.
- Zbilut, Joseph P, & Webber Jr, Charles L. 1992. Embeddings and delays as derived from quantification of recurrence plots. *Physics letters A*, **171**(3-4), 199–203.
- Zhang, Ning, Shen, Shui-Long, Zhou, Annan, & Xu, Ye-Shuang. 2019. Investigation on performance of neural networks using quadratic relative error cost function. *IEEE Access*, **7**, 106642–106652.
- Zimmerman, Michael W, Povinelli, Richard J, Johnson, Michael T, & Ropella, Kristina M. 2003. A reconstructed phase space approach for distinguishing ischemic from non-ischemic ST changes using Holter ECG data. *Pages 243–246 of: Computers in Cardiology, 2003*. IEEE.
-

APÊNDICES

QUANTIDADE DE AMOSTRAS ATRASADAS POR DERIVAÇÃO VETOCARDIOGRÁFICA

Cada ciclo cardíaco foi tratado como um sinal individual, sendo a quantidade de amostras a serem atrasadas para cada um deles determinada com base no primeiro mínimo local da informação mútua. Seja τ a quantidade de amostras atrasadas para a determinação da reconstrução do espaço de fase e do gráfico de recorrência correspondente àquele ciclo, são apresentados nas tabelas A.1, A.2, A.3; para cada τ , a quantidade de ciclos que utilizou esse valor.

Essa informação é apresentada de maneira sumarizada também na Figura 3.3, no Capítulo 3.

A Tabela A.1 apresenta os valores para a derivação vetocardiográfica X.

Tabela A.1 – Quantidade de ciclos para cada valor de quantidade de amostras atrasadas para a derivação X.

Quantidade de amostras atrasadas	Quantidade de ciclos
1	1
4	4
5	75
6	403
7	1000
8	2172
9	3940
10	6502
11	8010
12	7982
13	7469
14	7502
15	7858
16	8264
17	8235
18	7567
19	5993
20	4446
21	3434
22	2226
23	1375
24	819
25	408
26	204
27	79
28	36
29	9
30	2
32	3

Da mesma maneira, para a derivação Y, os valores são apresentados na Tabela A.2.

Tabela A.2 – Quantidade de ciclos para cada valor de quantidade de amostras atrasadas para a derivação Y.

Quantidade de amostras atrasadas	Quantidade de ciclos
2	1
4	25
5	463
6	2085
7	3886
8	5429
9	6763
10	7446
11	7586
12	7300
13	7472
14	7908
15	8262
16	8272
17	7124
18	5618
19	3972
20	2585
21	1598
22	1018
23	592
24	308
25	185
26	70
27	31
28	14
29	3
30	1
32	1

Por fim, as quantidades de ciclos para cada quantidade de amostras atrasadas para a derivação vetocardiográfica Z são apresentadas na Tabela A.3.

Tabela A.3 – Quantidade de ciclos para cada valor de quantidade de amostras atrasadas para a derivação Z.

Quantidade de amostras atrasadas	Quantidade de ciclos
2	1
4	3
5	59
6	418
7	1180
8	2240
9	3488
10	4631
11	5735
12	6942
13	8607
14	10537
15	12375
16	12530
17	10786
18	7634
19	4484
20	2286
21	1120
22	519
23	225
24	129
25	49
26	25
27	9
28	3
29	1
30	1
31	1

EVOLUÇÃO DAS MÉTRICAS PARA OS DEMAIS MODELOS

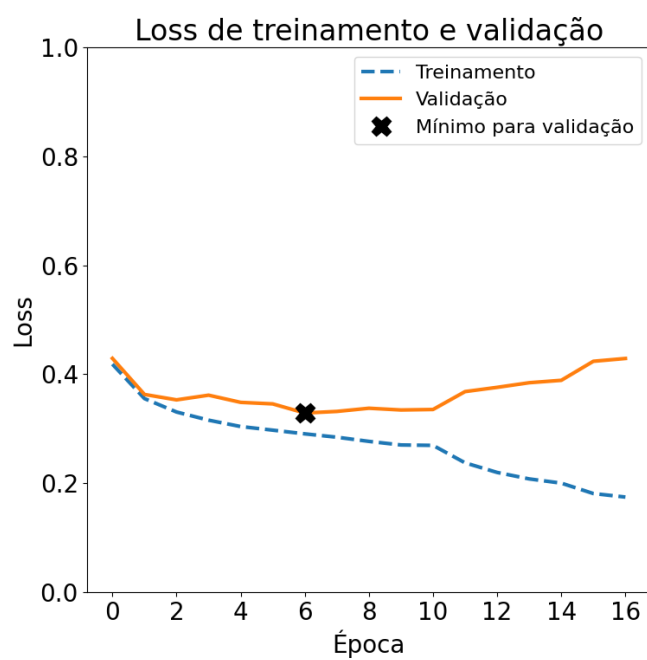
Neste apêndice se encontram as demais imagens para a evolução das métricas *loss*, acurácia, sensibilidade, precisão, escore F1 e área abaixo da curva (AUC) para os demais modelos obtidos nas rodadas em que não tiveram o valor mínimo de *loss* para cada arquitetura e que não foram apresentados no Capítulo 4, tanto para os modelos treinados com imagens de espaço de fase quanto para aqueles treinados com gráficos de recorrência.

B.1 – Treinamento com imagens de espaço de fase

B.1.1 – MobileNetV2

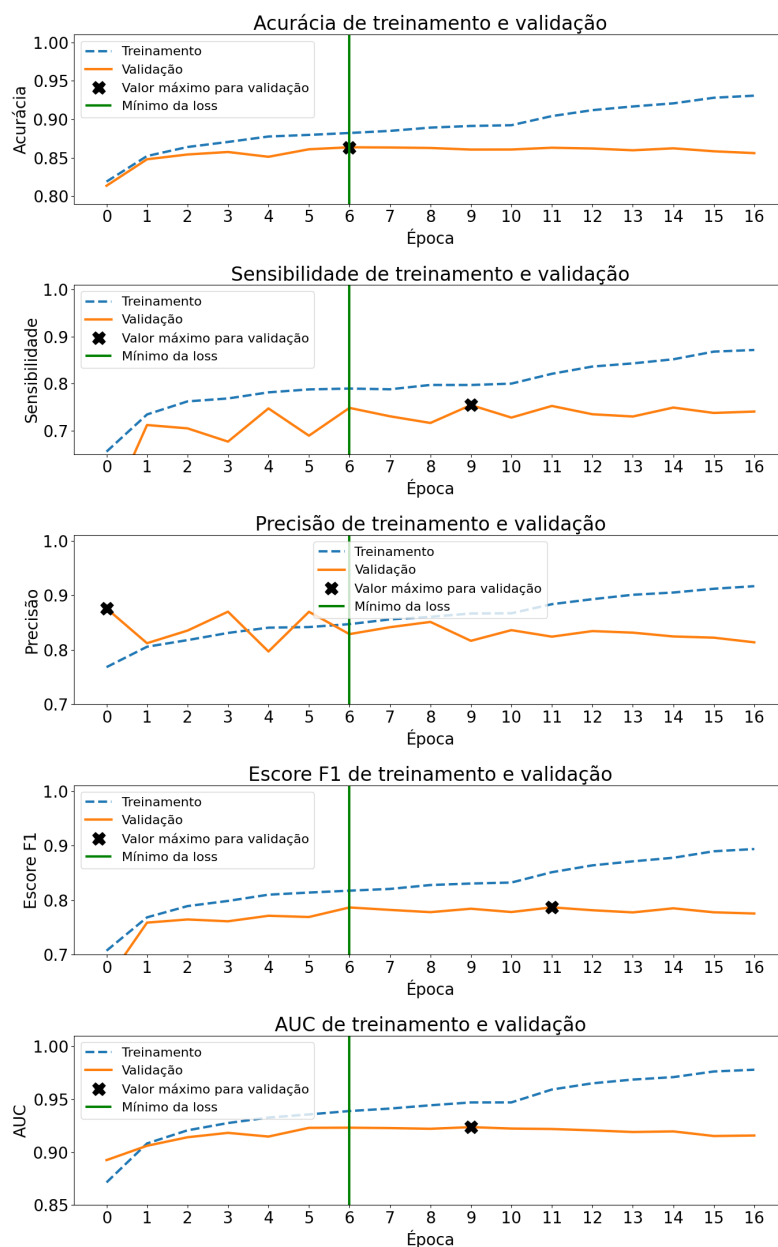
Para a rodada 2:

Figura B.1 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura MobileNetV2.



Fonte: Autoral.

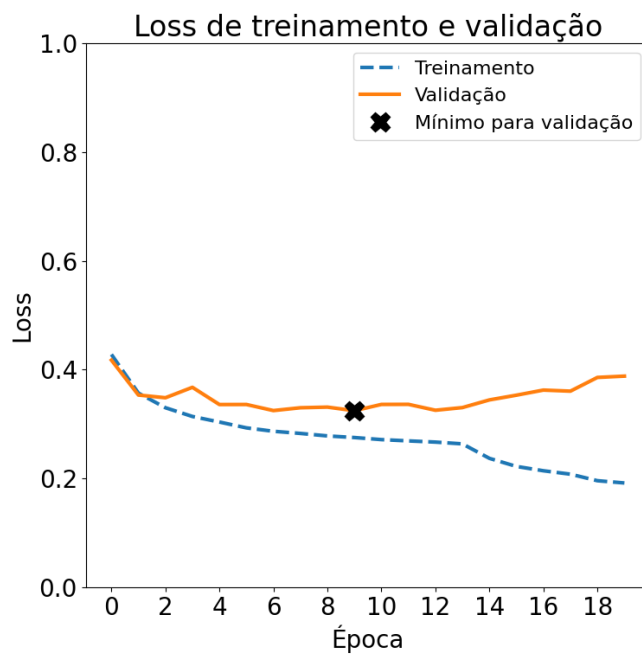
Figura B.2 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura MobileNetV2.



Fonte: Autoral.

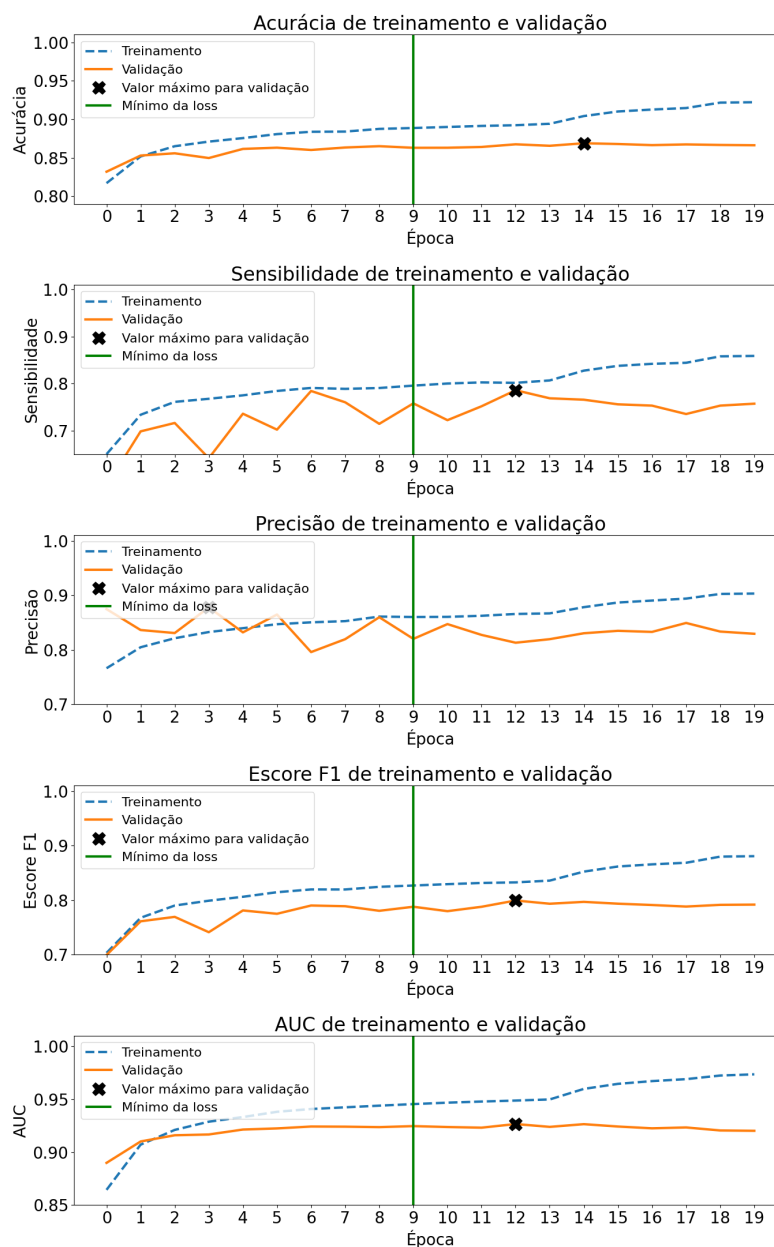
Considerando a rodada 3:

Figura B.3 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura MobileNetV2.



Fonte: Autoral.

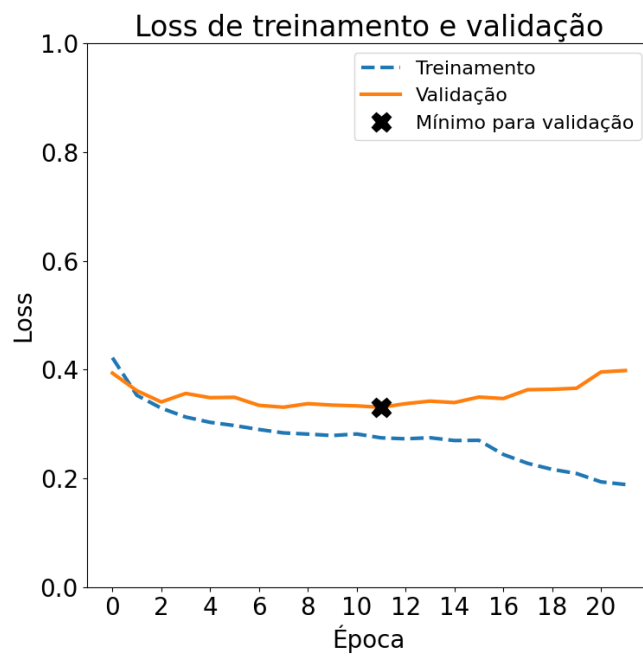
Figura B.4 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura MobileNetV2.



Fonte: Autoral.

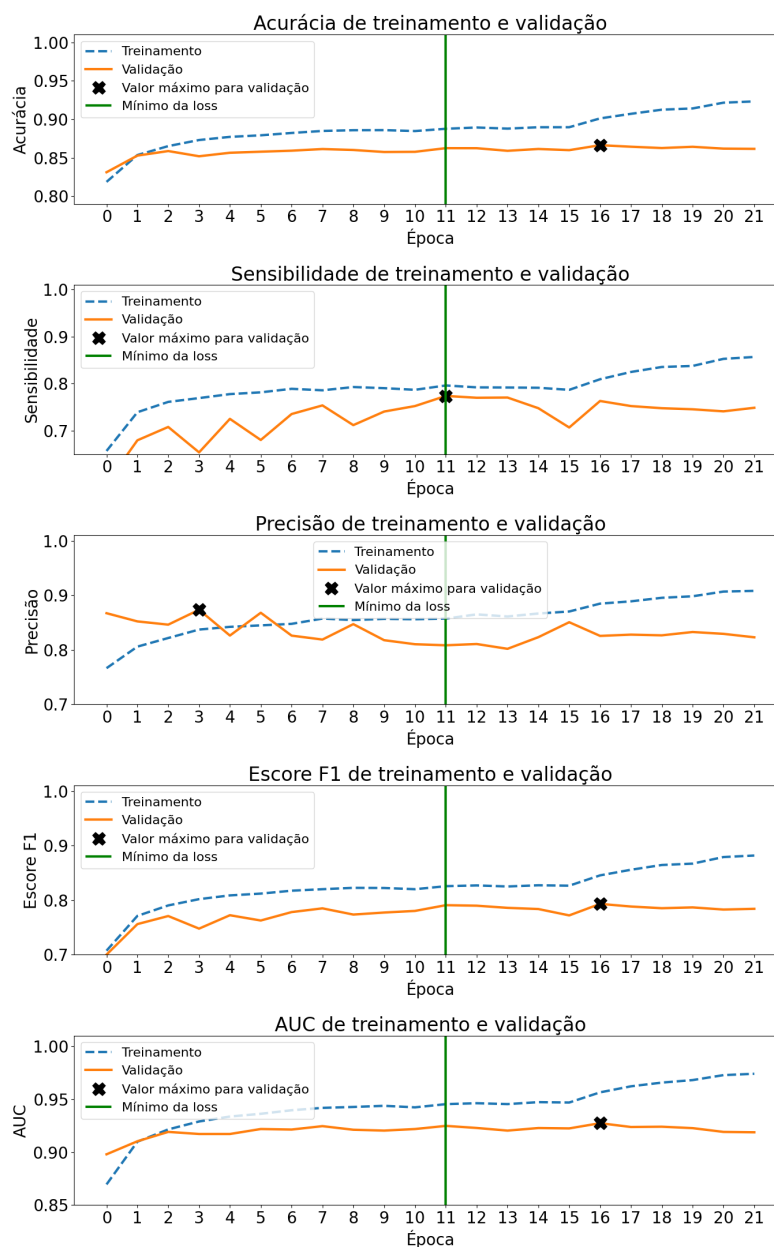
A rodada 4:

Figura B.5 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura MobileNetV2.



Fonte: Autoral.

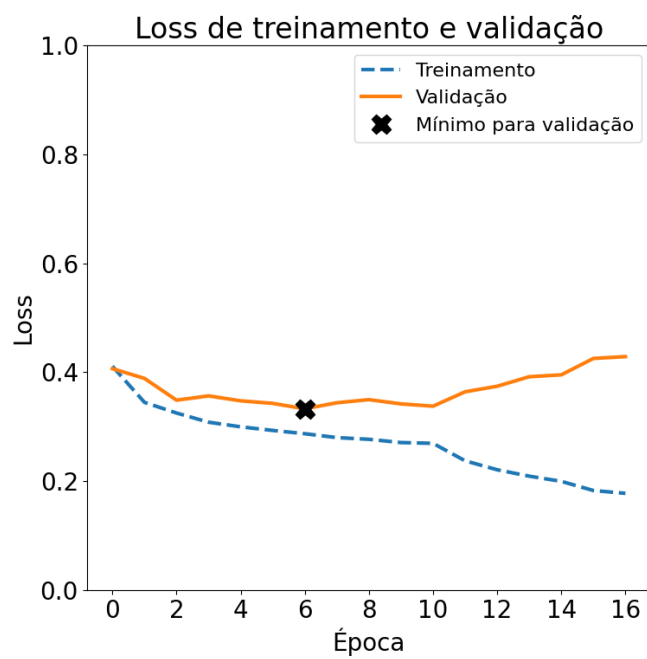
Figura B.6 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura MobileNetV2.



Fonte: Autoral.

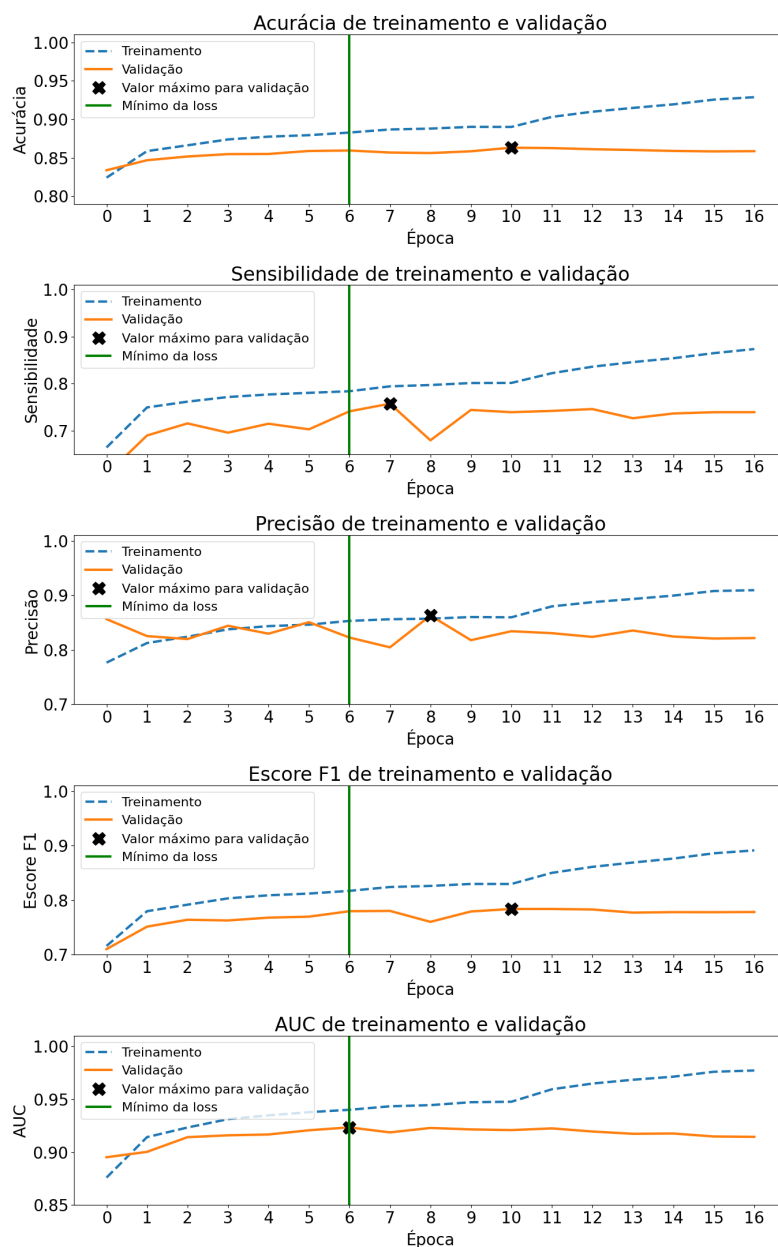
Por fim, a rodada 5:

Figura B.7 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura MobileNetV2.



Fonte: Autoral.

Figura B.8 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura MobileNetV2.

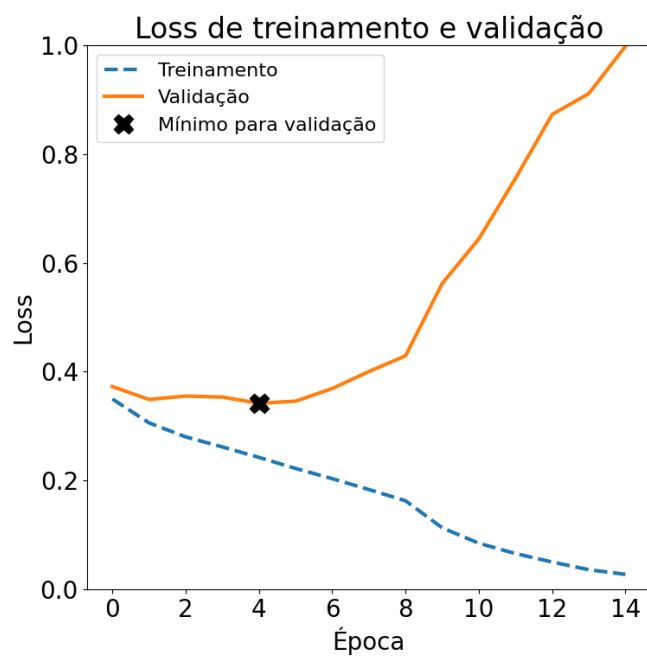


Fonte: Autoral.

B.1.2 – ResNet-50

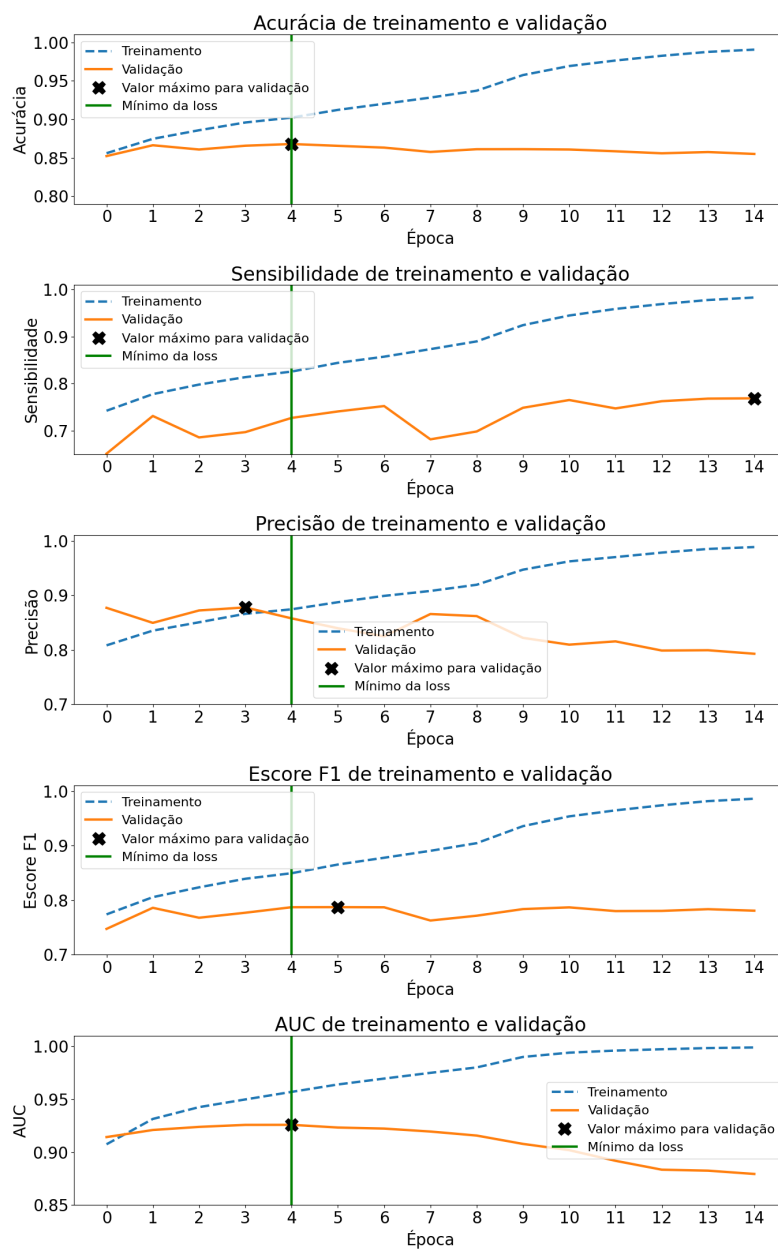
A rodada 1:

Figura B.9 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-50.



Fonte: Autoral.

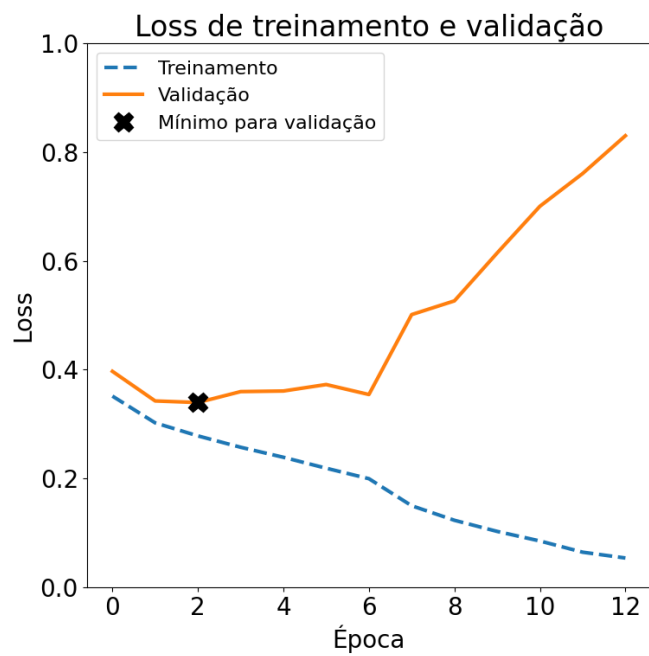
Figura B.10 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-50.



Fonte: Autoral.

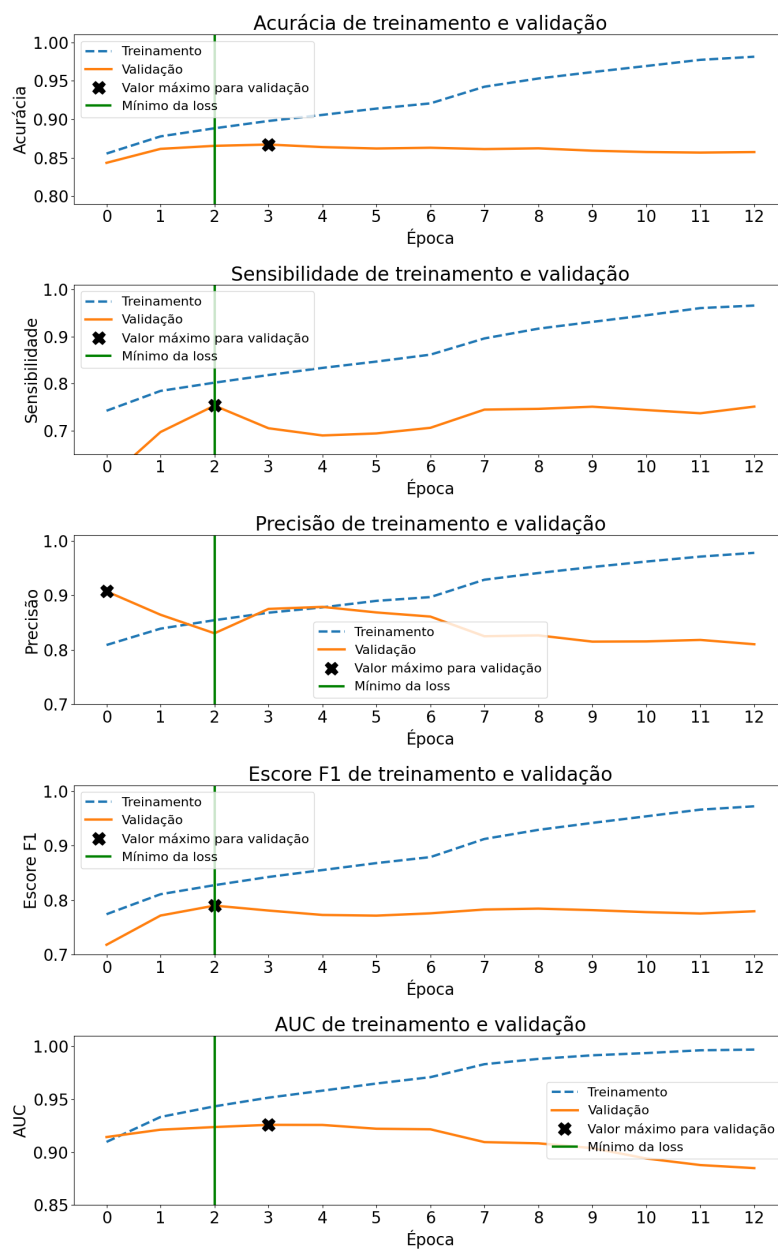
Para a rodada 2:

Figura B.11 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-50.



Fonte: Autoral.

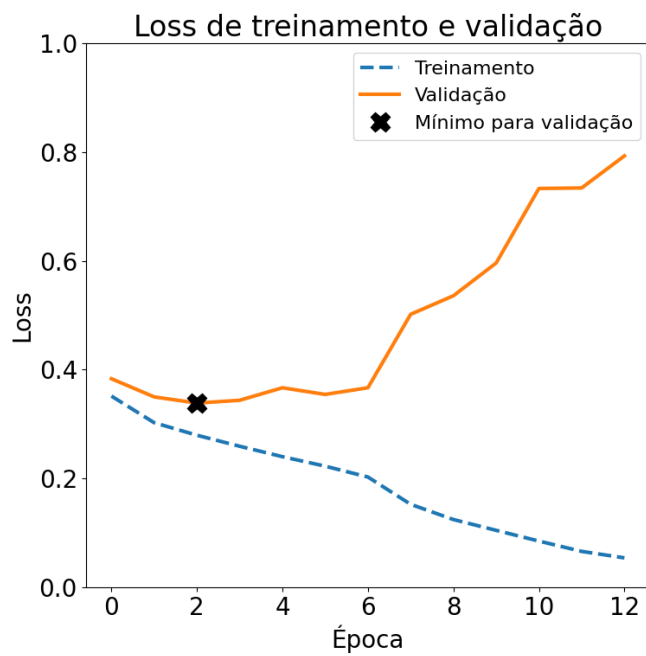
Figura B.12 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-50.



Fonte: Autoral.

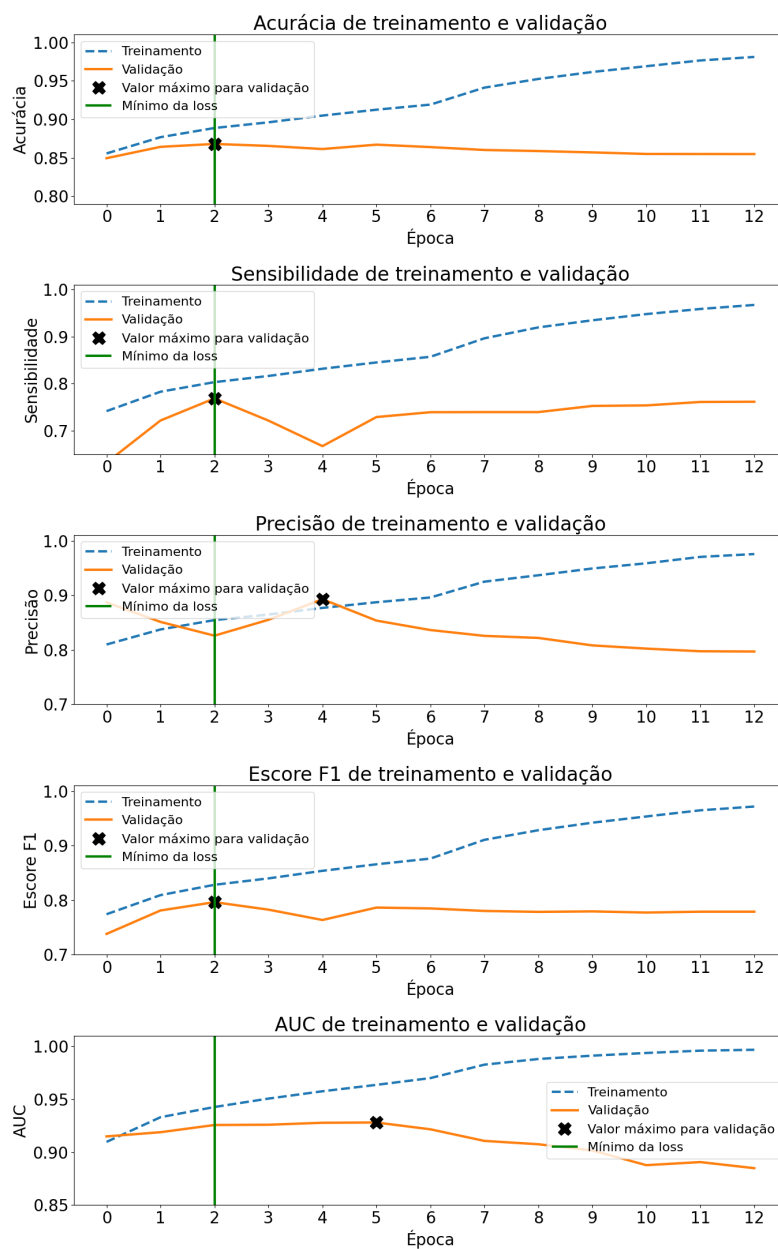
Considerando a rodada 3:

Figura B.13 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-50.



Fonte: Autoral.

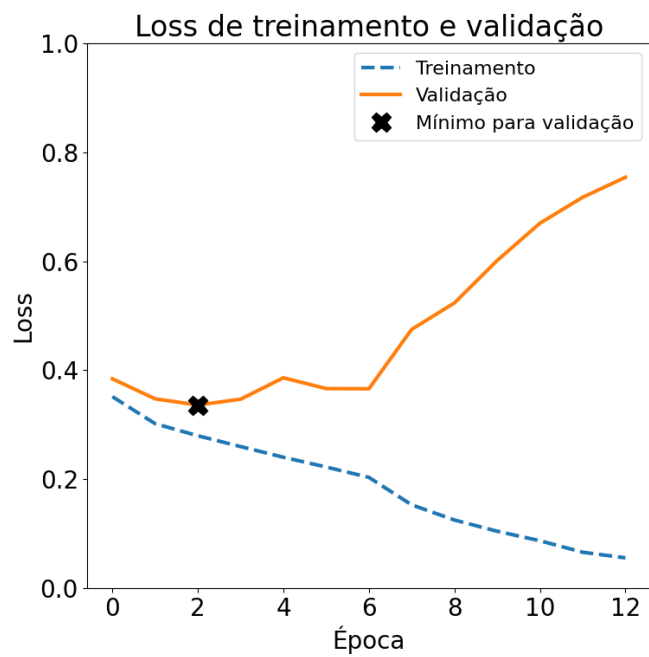
Figura B.14 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-50.



Fonte: Autoral.

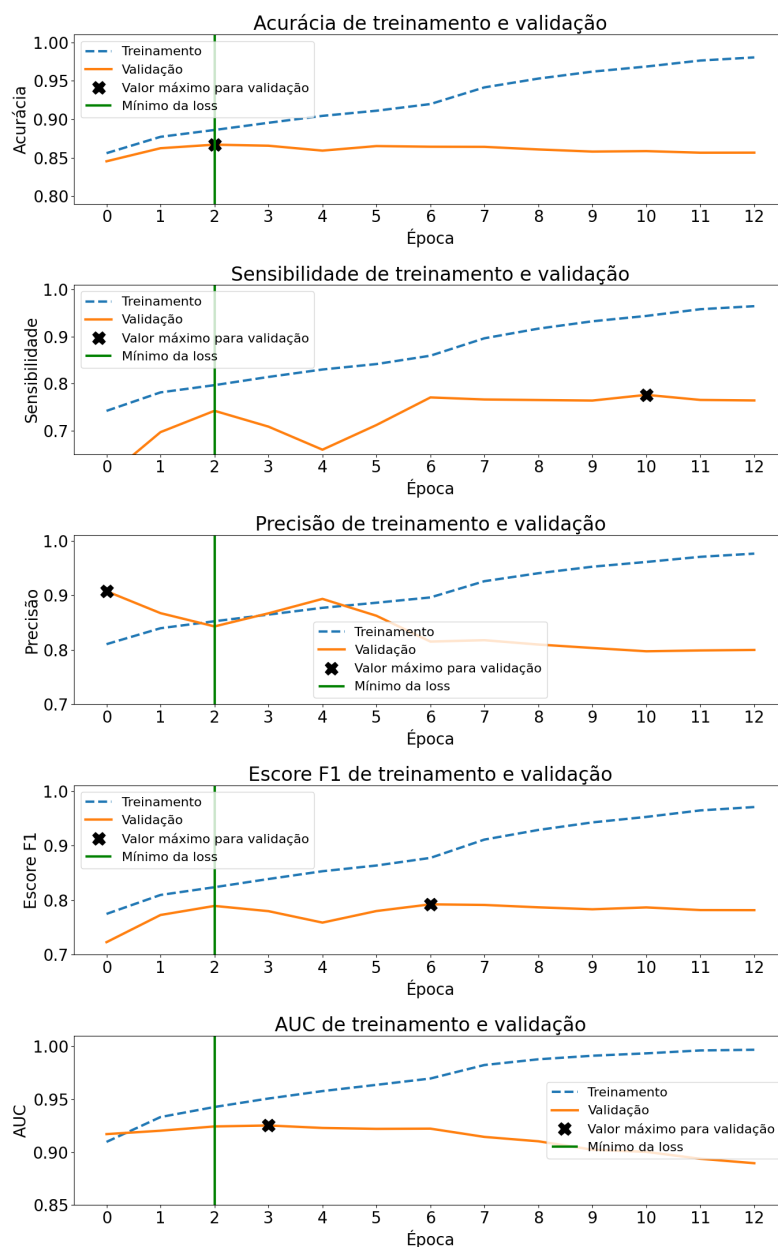
Por fim, a rodada 5:

Figura B.15 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura ResNet-50.



Fonte: Autoral.

Figura B.16 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura ResNet-50.

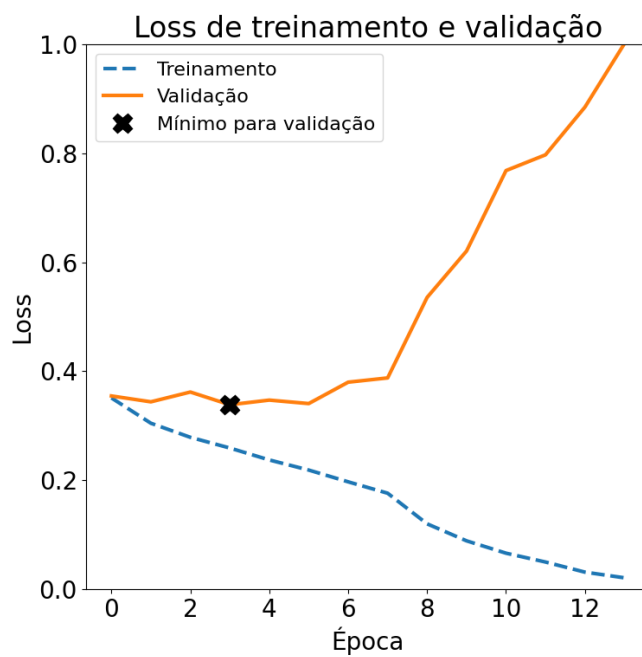


Fonte: Autoral.

B.1.3 – ResNet-101

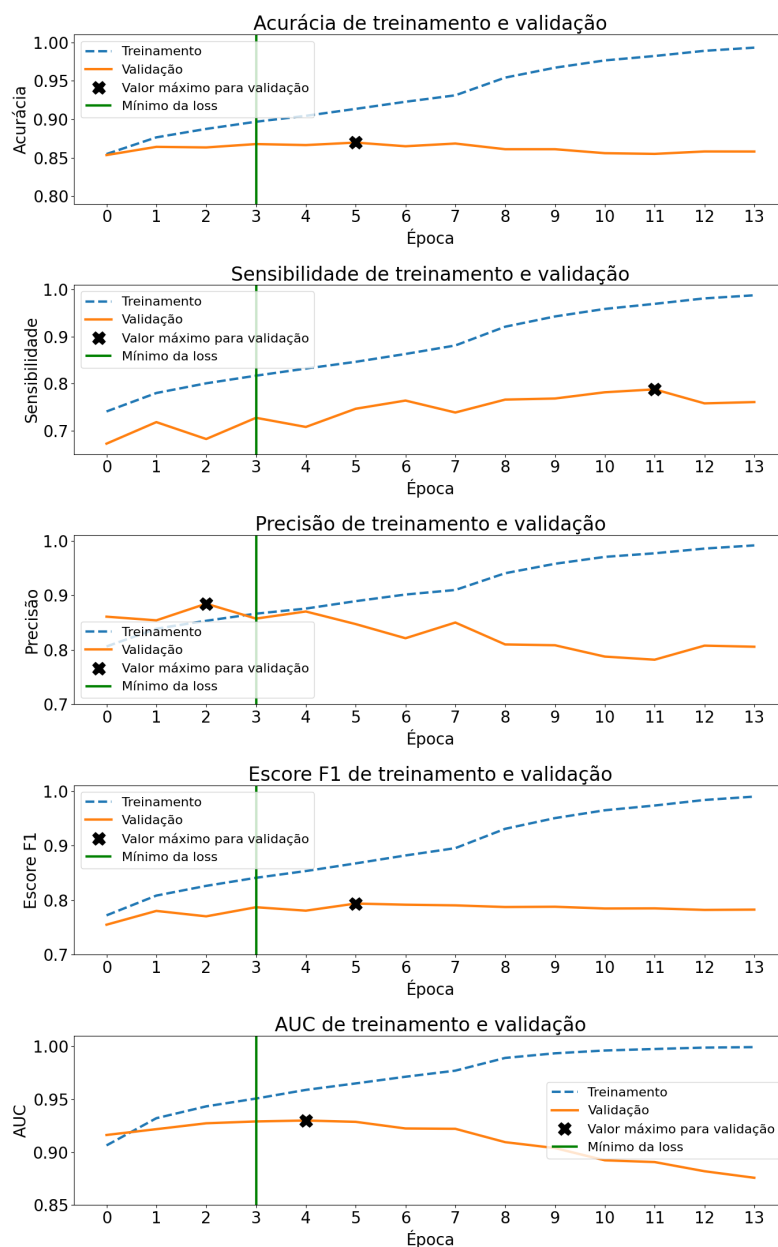
Se obteve para a rodada 1:

Figura B.17 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-101.



Fonte: Autoral.

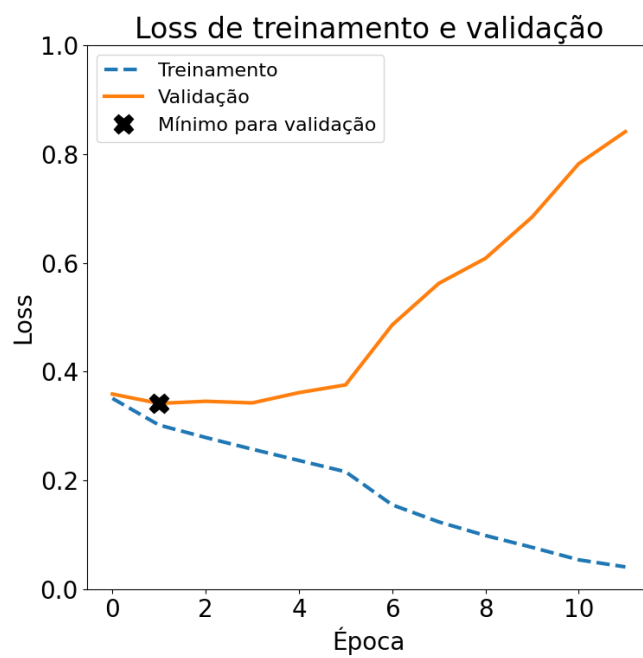
Figura B.18 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura ResNet-101.



Fonte: Autoral.

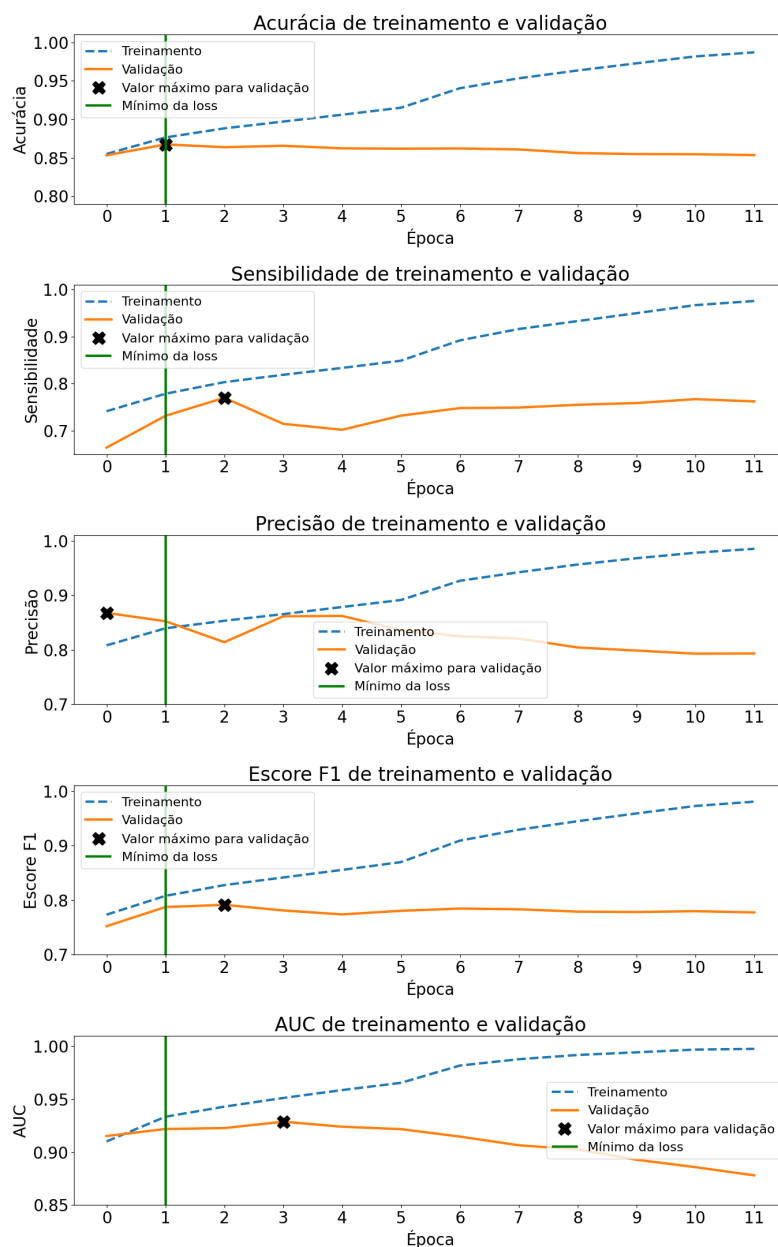
Considerando a rodada 2:

Figura B.19 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-101.



Fonte: Autoral.

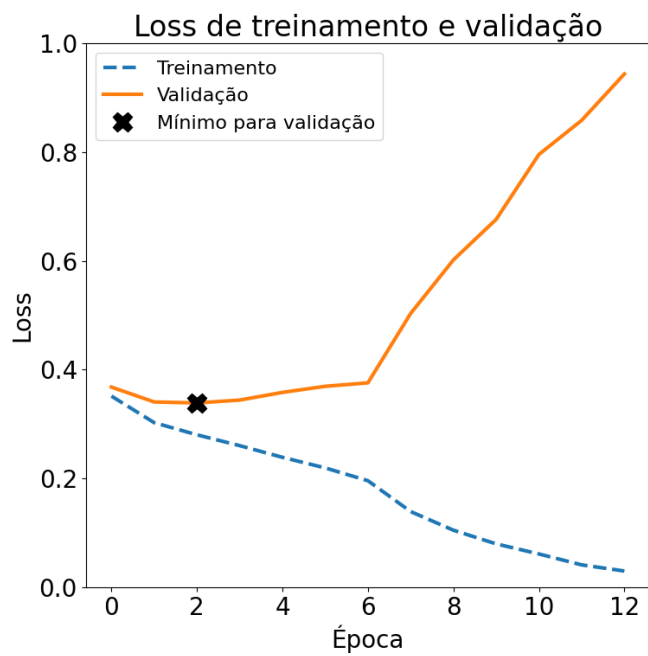
Figura B.20 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 2 com a arquitetura ResNet-101.



Fonte: Autoral.

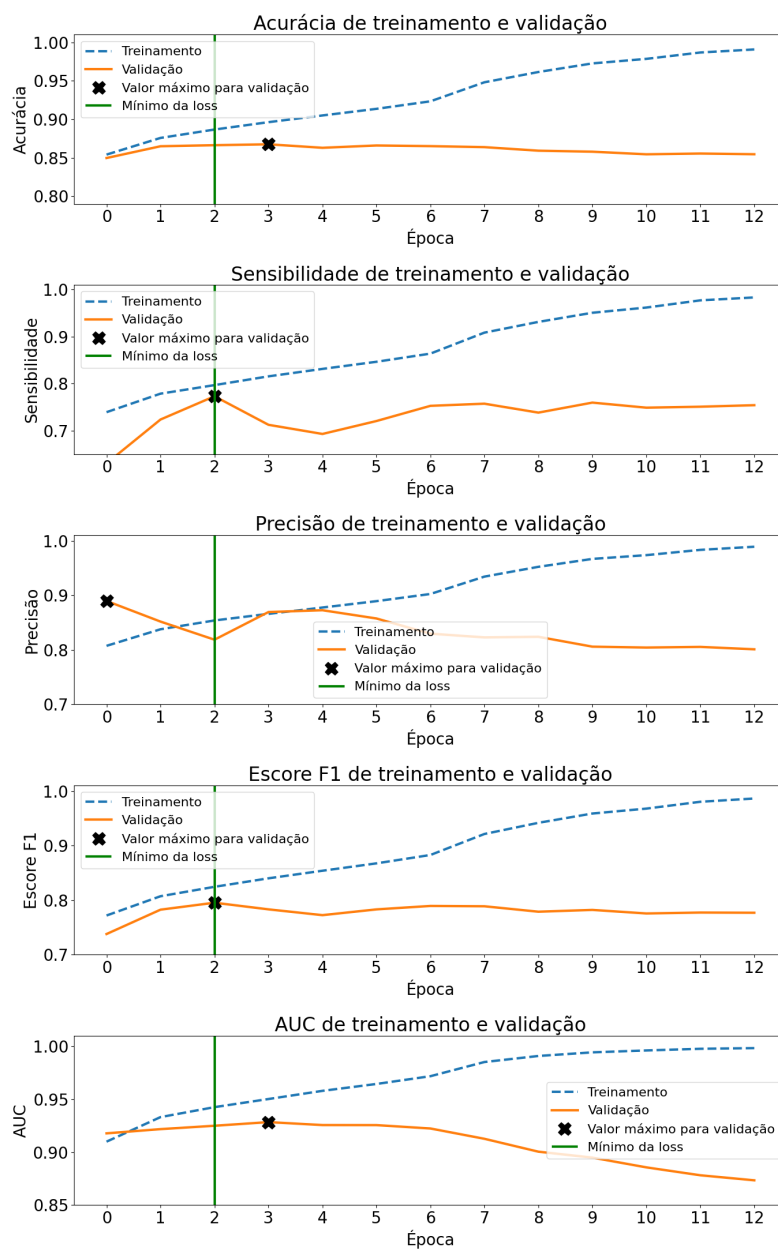
Para a rodada 3:

Figura B.21 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-101.



Fonte: Autoral.

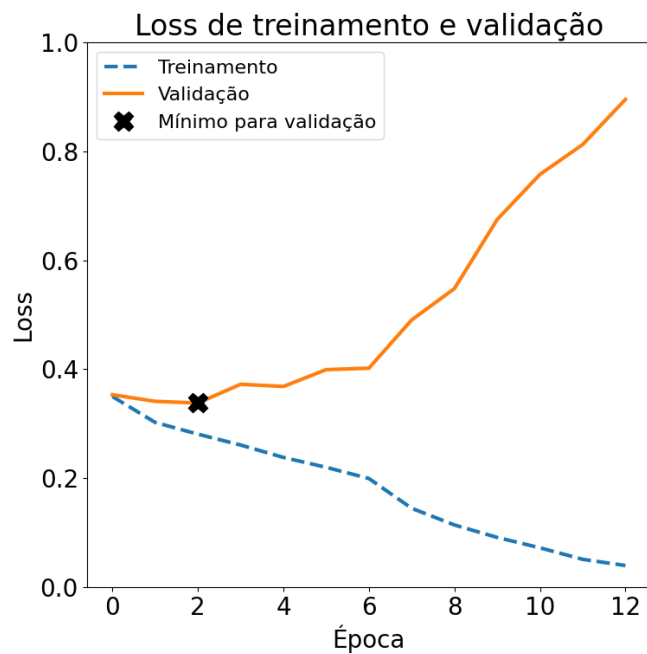
Figura B.22 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura ResNet-101.



Fonte: Autoral.

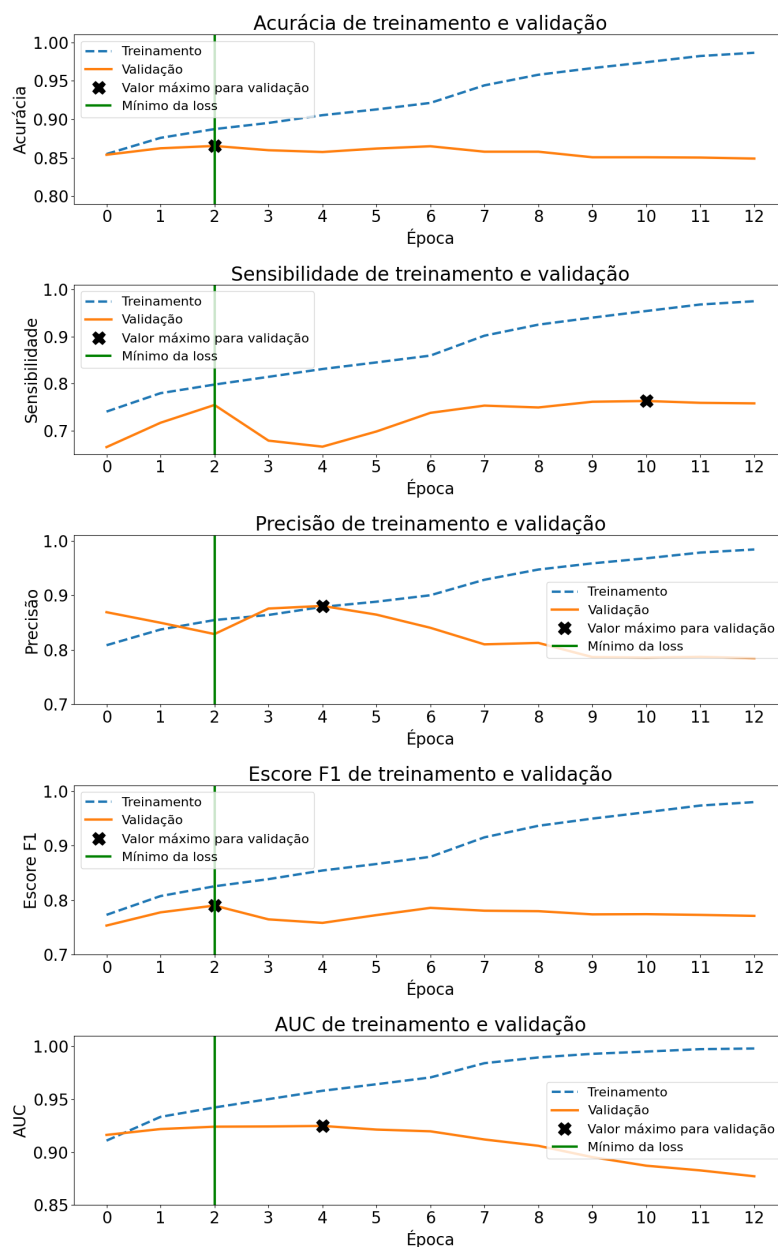
Por fim, para a rodada 4:

Figura B.23 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura ResNet-101.



Fonte: Autoral.

Figura B.24 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura ResNet-101.

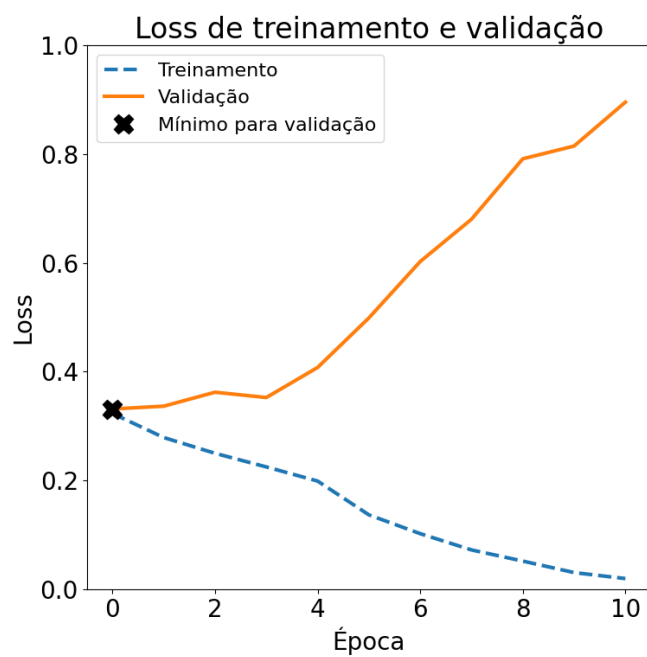


Fonte: Autoral.

B.1.4 – DenseNet201

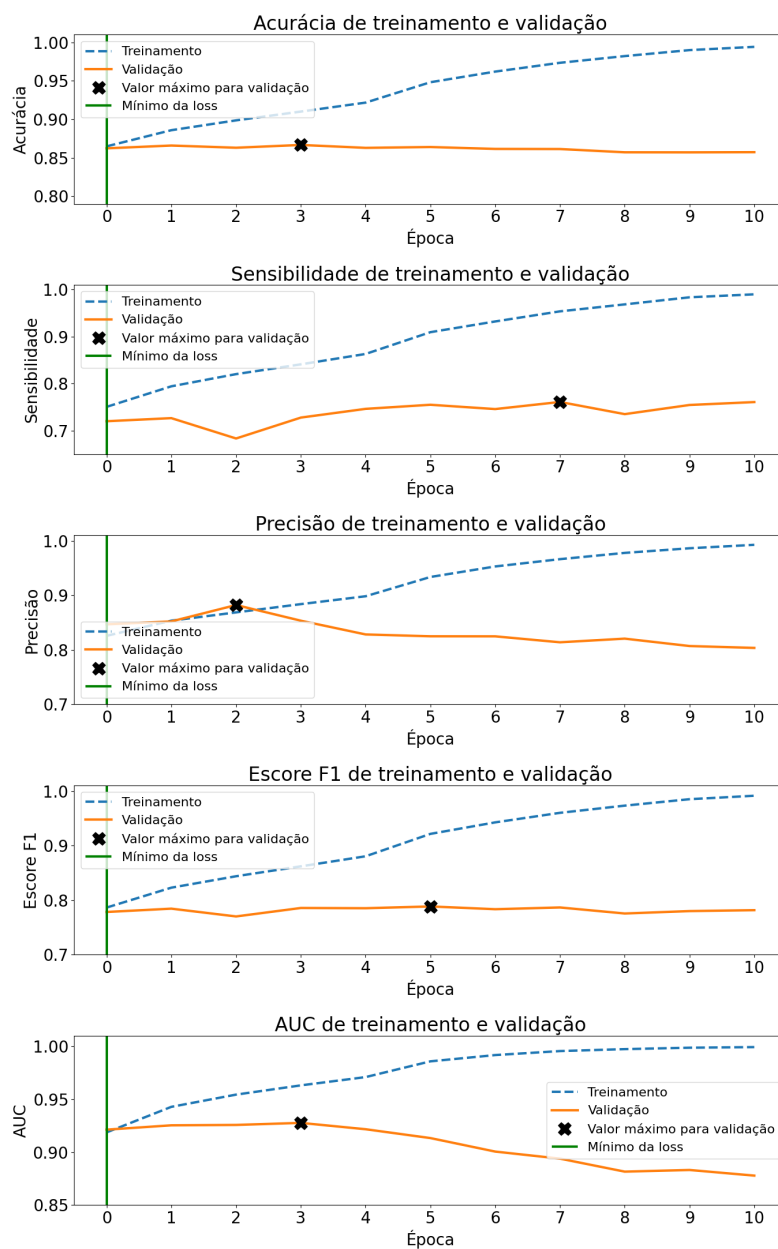
Para a rodada 1:

Figura B.25 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura DenseNet201.



Fonte: Autoral.

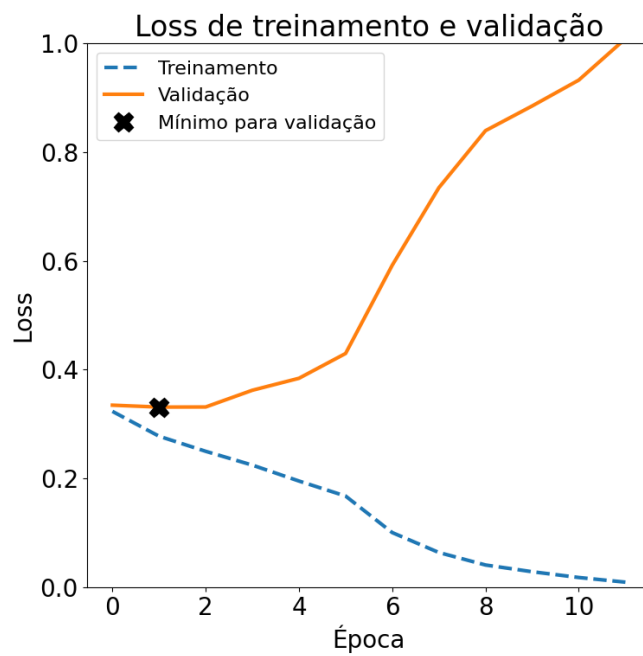
Figura B.26 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 1 com a arquitetura DenseNet201.



Fonte: Autoral.

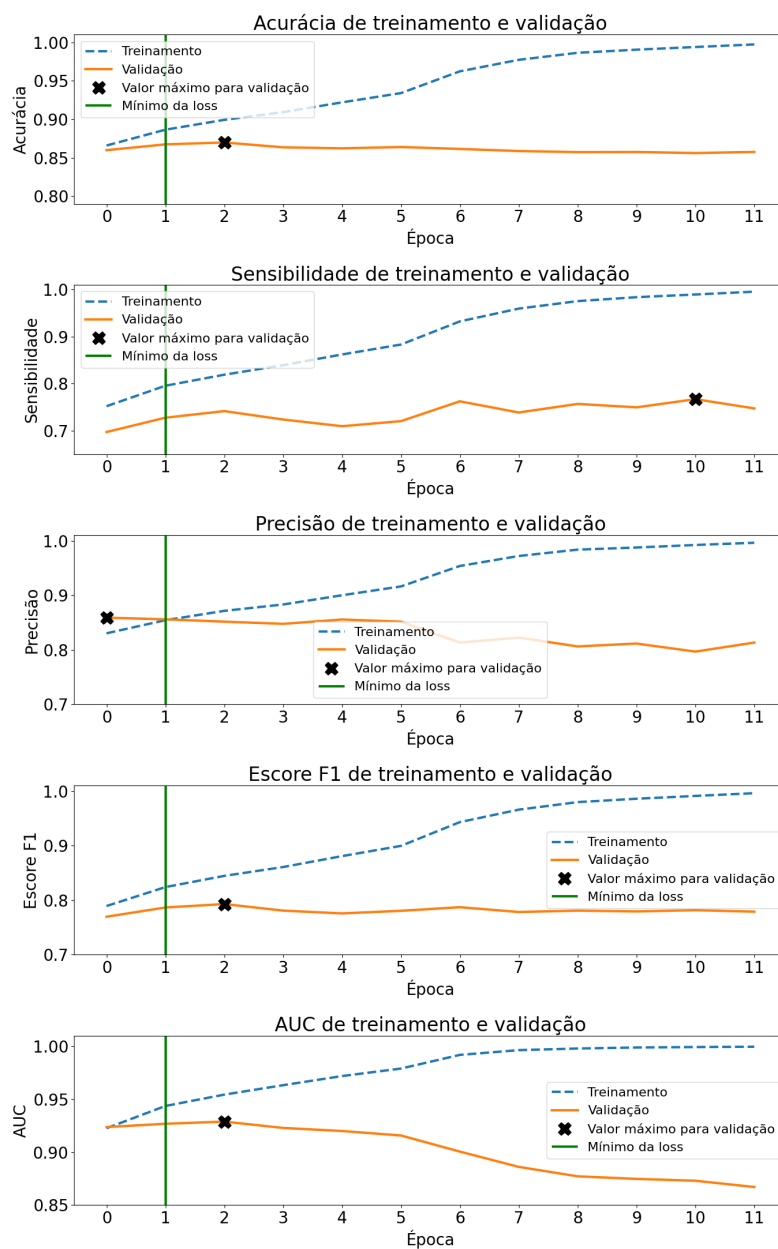
Considerando a rodada 3:

Figura B.27 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura DenseNet201.



Fonte: Autoral.

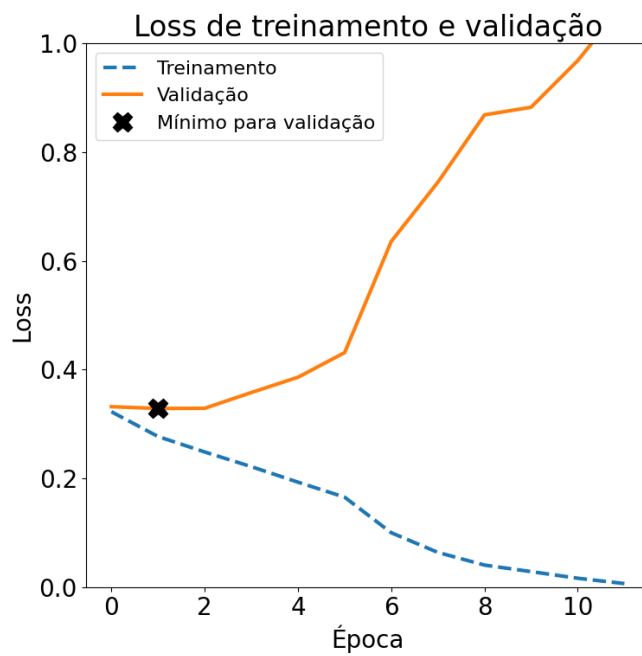
Figura B.28 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 3 com a arquitetura DenseNet201.



Fonte: Autoral.

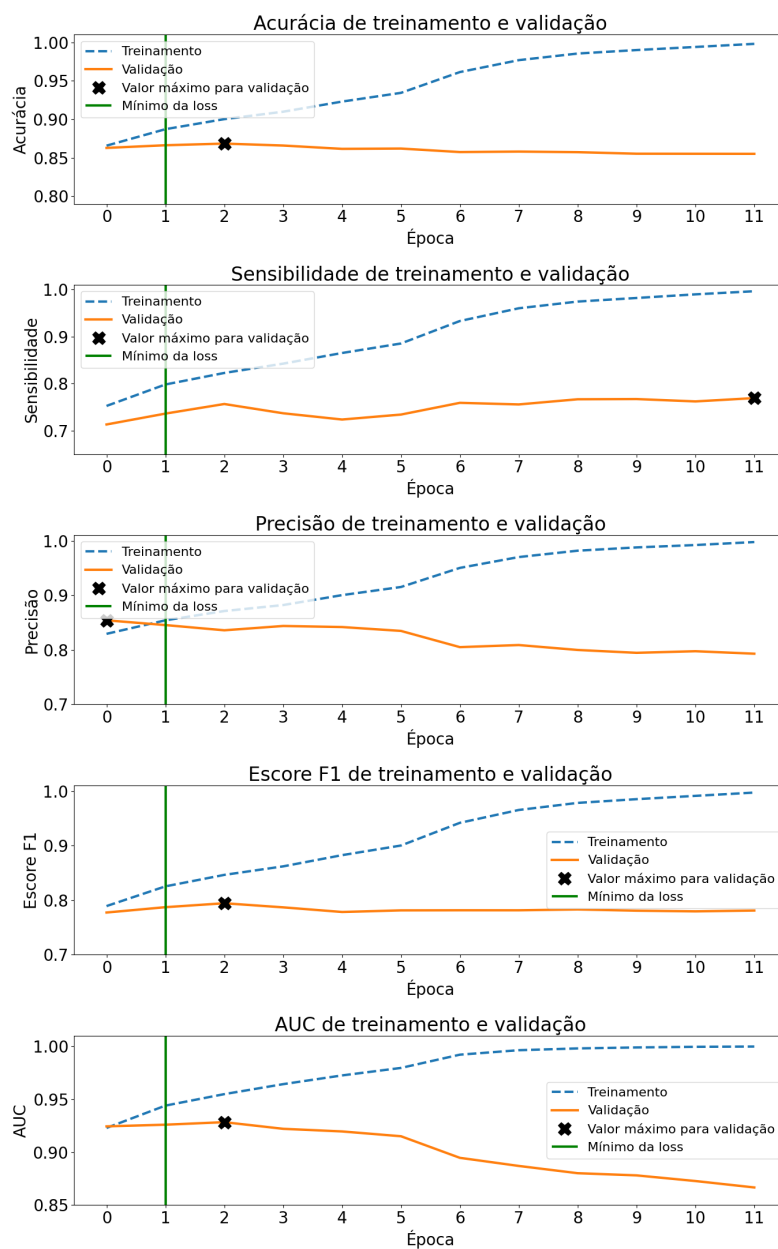
Para a rodada 4:

Figura B.29 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura DenseNet201.



Fonte: Autoral.

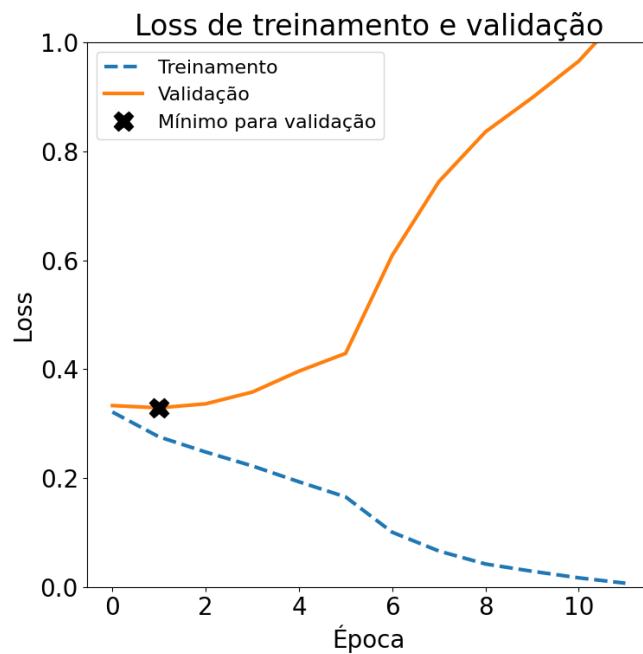
Figura B.30 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 4 com a arquitetura DenseNet201.



Fonte: Autoral.

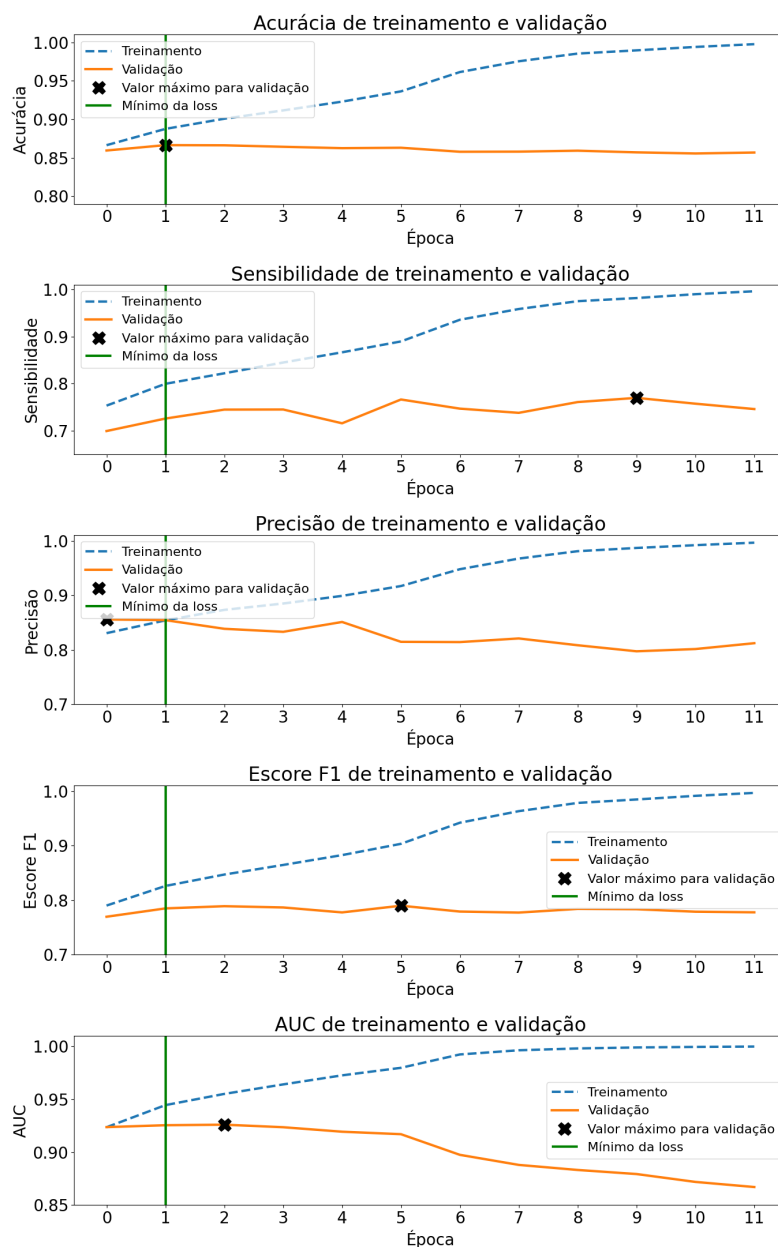
Por fim, para a rodada 5:

Figura B.31 – Evolução por época da métrica *loss* para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura DenseNet201.



Fonte: Autoral.

Figura B.32 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação com imagens de EF para a rodada 5 com a arquitetura DenseNet201.



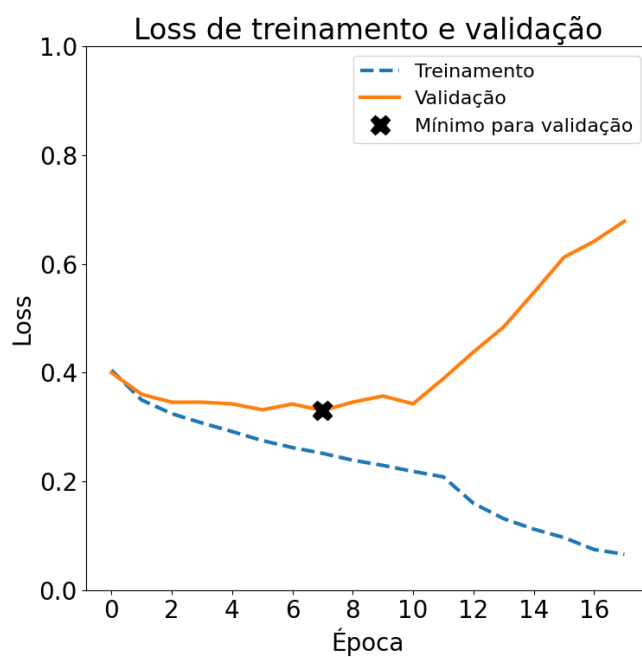
Fonte: Autoral.

B.2 – Treinamento com gráficos de recorrência

B.2.1 – MobileNetV2

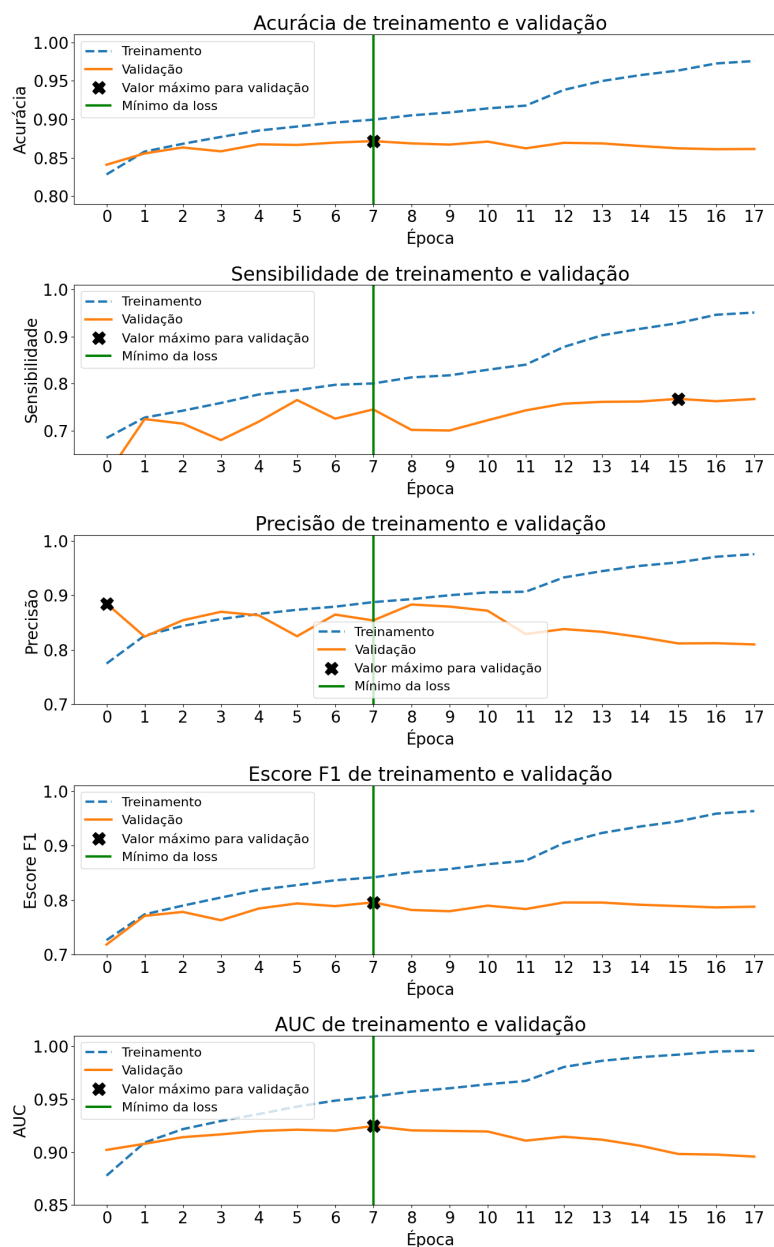
Para a rodada 1:

Figura B.33 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura MobileNetV2.



Fonte: Autoral.

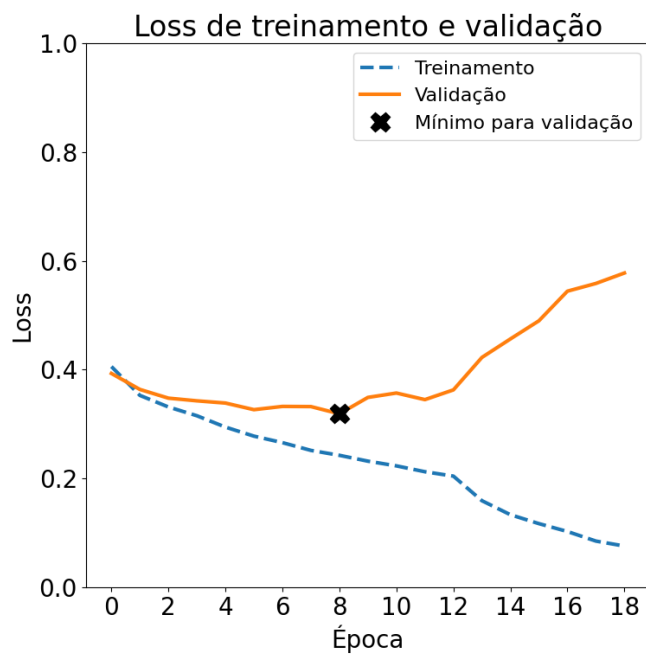
Figura B.34 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura MobileNetV2.



Fonte: Autoral.

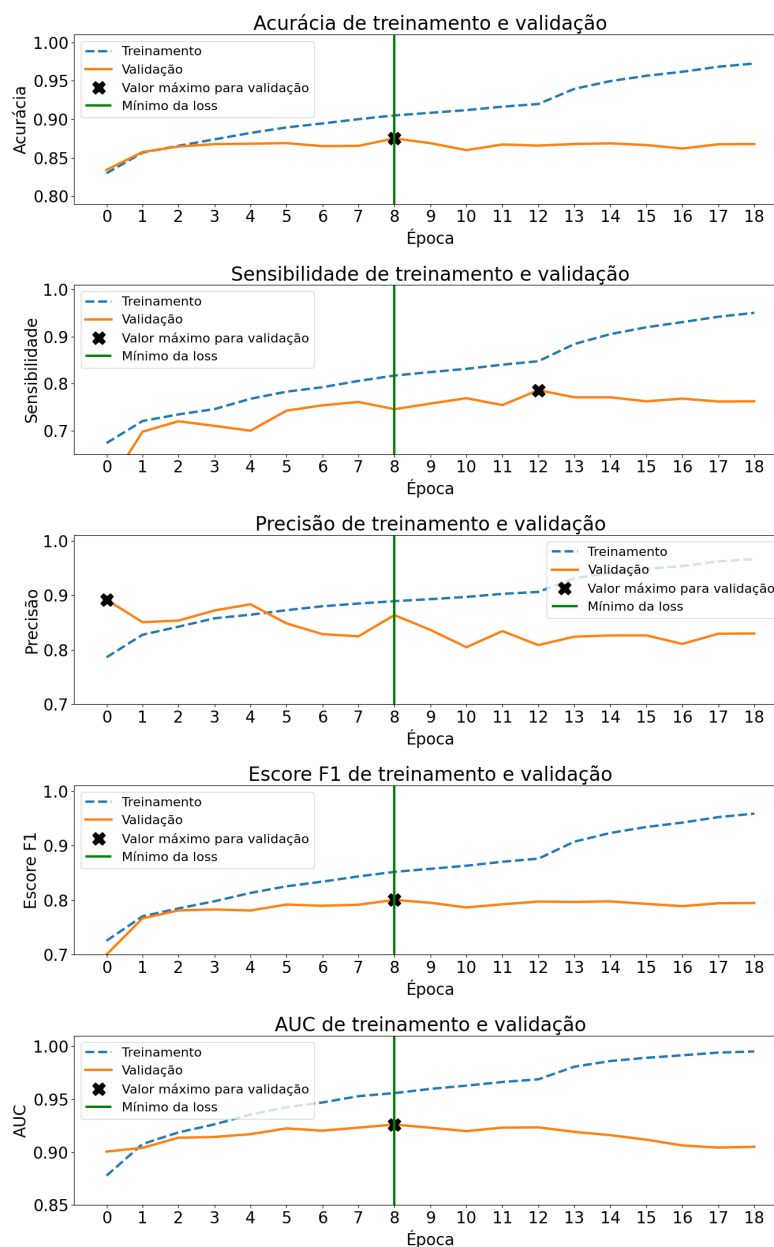
Considerando a rodada 3:

Figura B.35 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura MobileNetV2.



Fonte: Autoral.

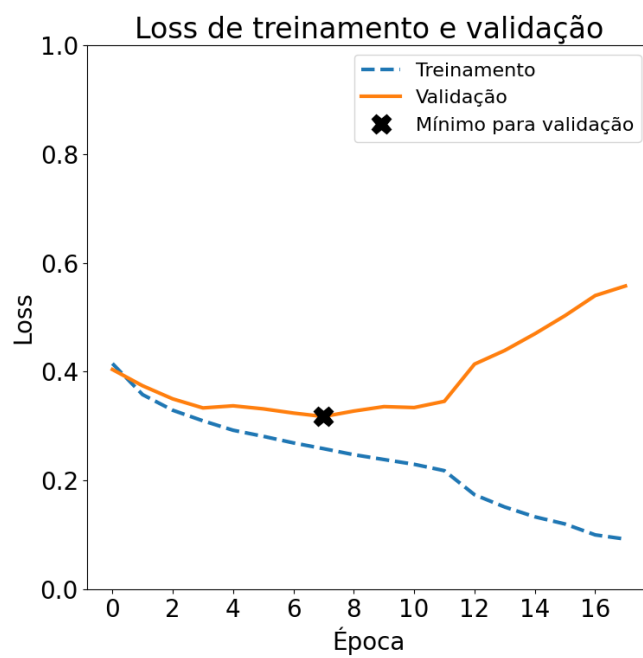
Figura B.36 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura MobileNetV2.



Fonte: Autoral.

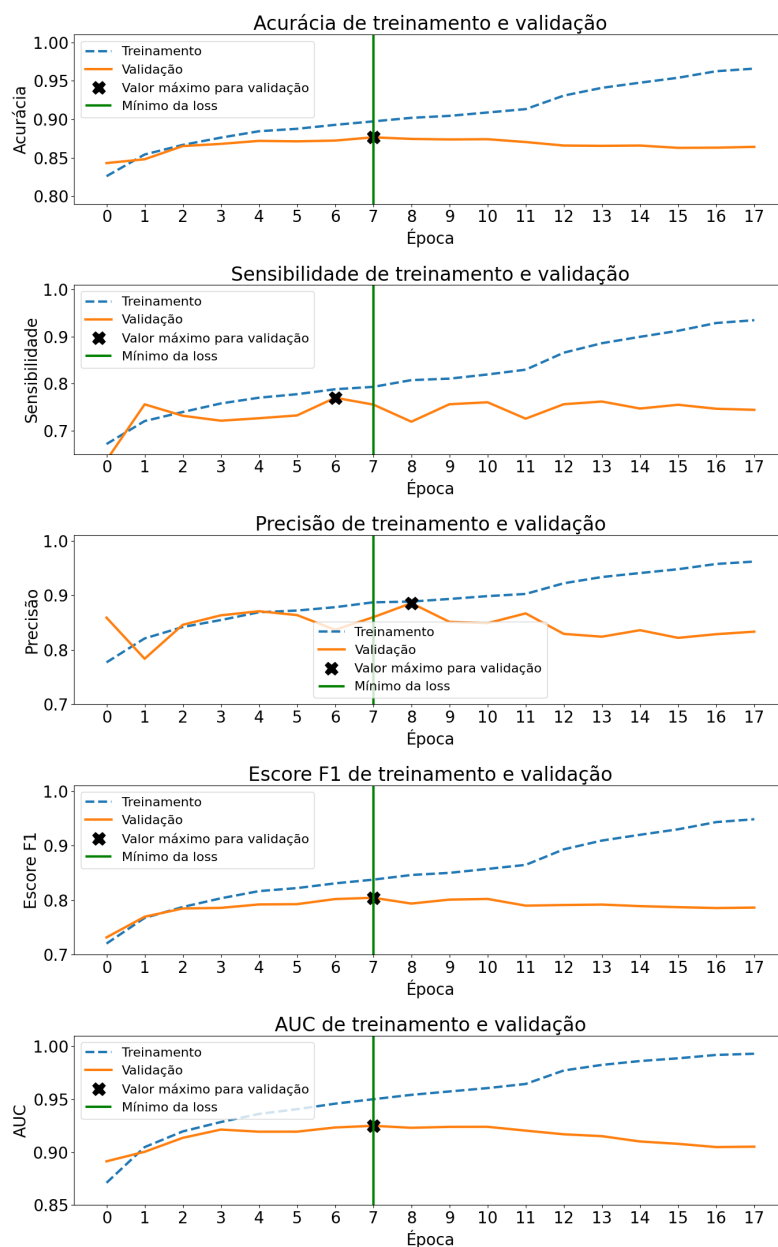
A rodada 4:

Figura B.37 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura MobileNetV2.



Fonte: Autoral.

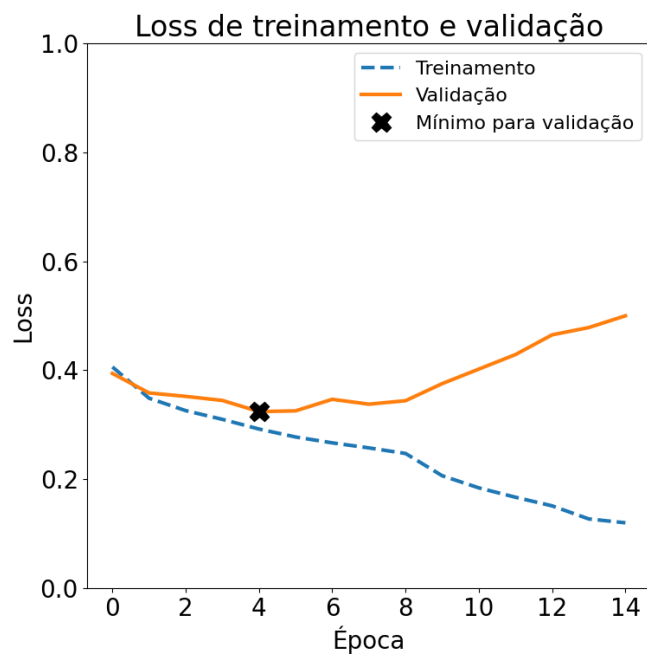
Figura B.38 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura MobileNetV2.



Fonte: Autoral.

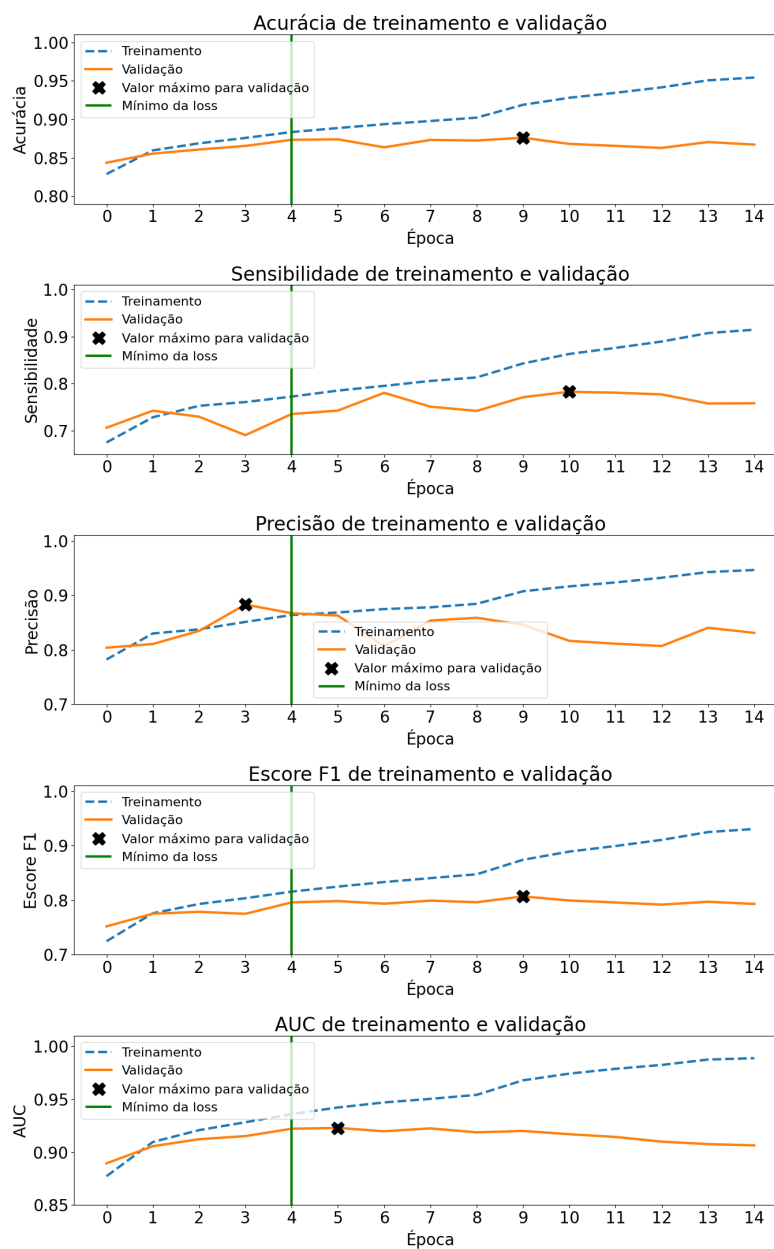
Por fim, a rodada 5:

Figura B.39 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura MobileNetV2.



Fonte: Autoral.

Figura B.40 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura MobileNetV2.

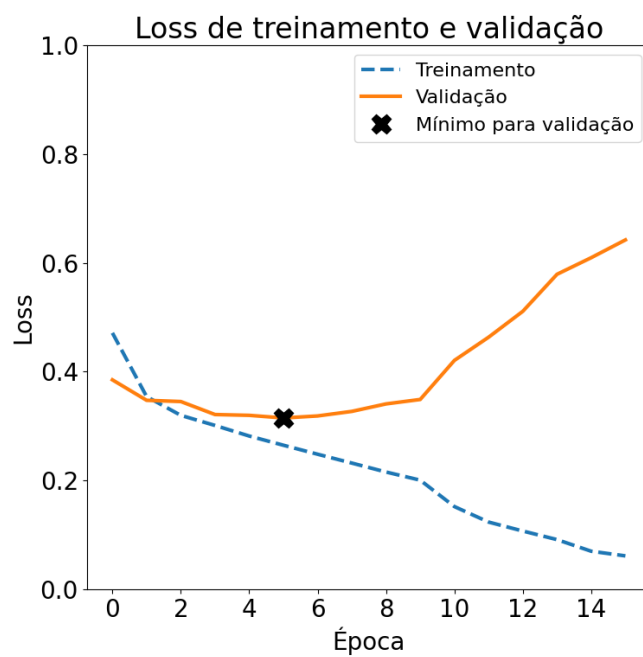


Fonte: Autoral.

B.2.2 – ResNet-50

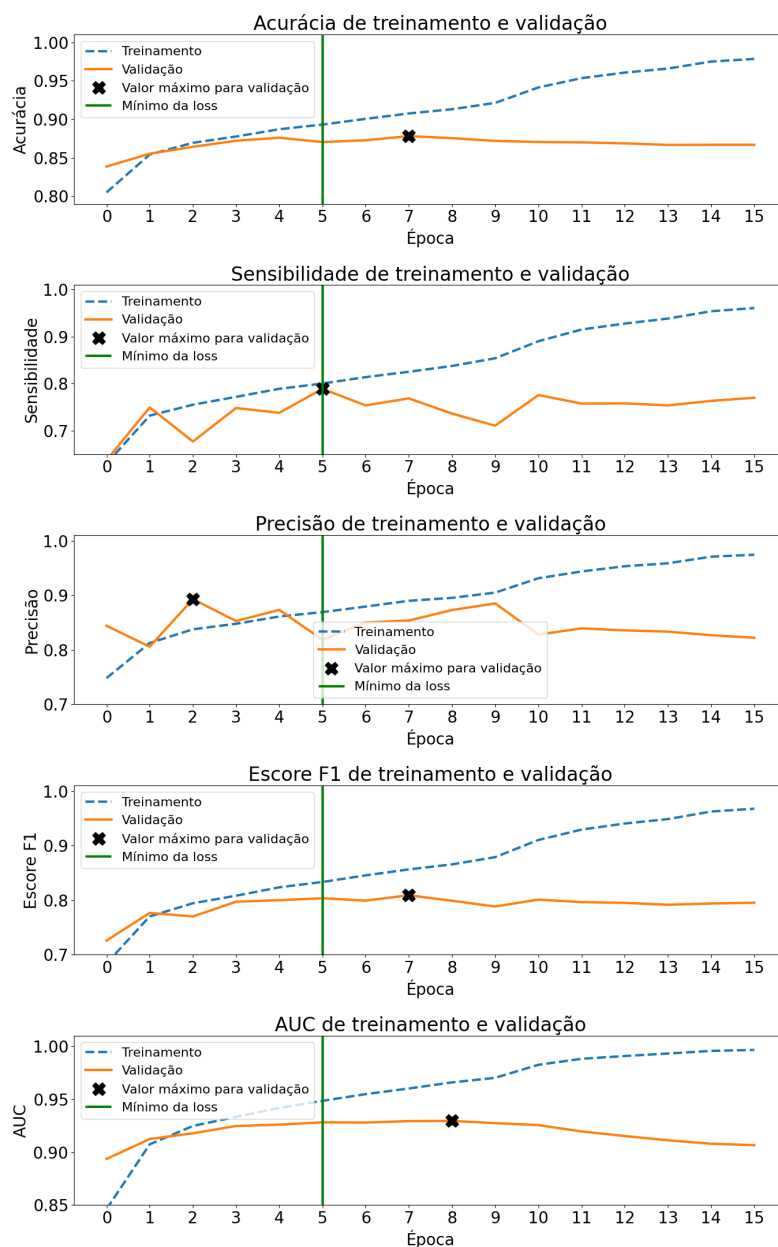
A rodada 1:

Figura B.41 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-50.



Fonte: Autoral.

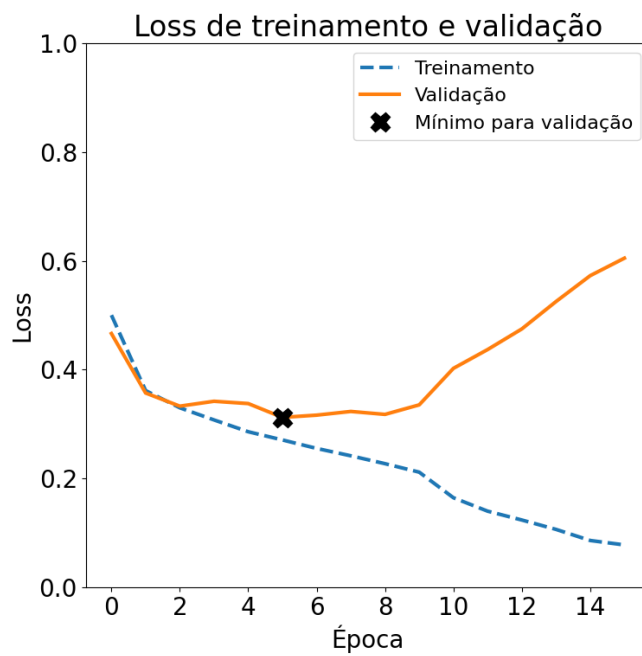
Figura B.42 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-50.



Fonte: Autoral.

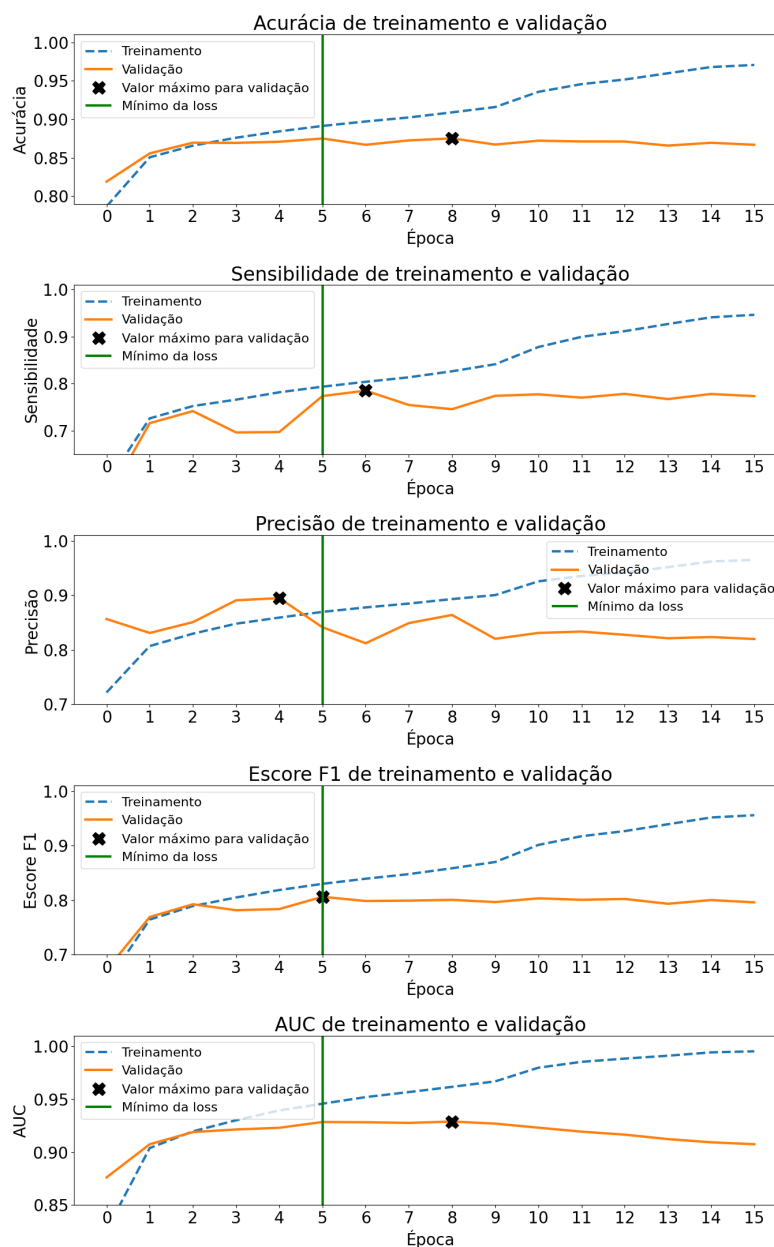
Para a rodada 2:

Figura B.43 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura ResNet-50.



Fonte: Autoral.

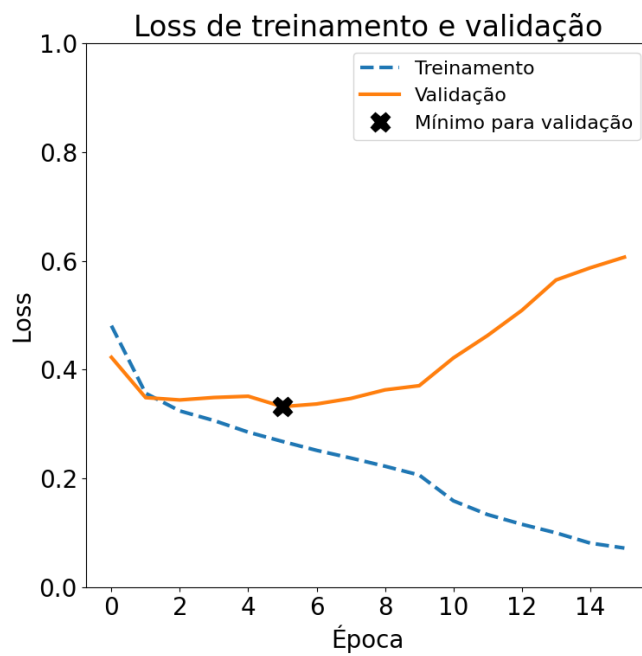
Figura B.44 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura ResNet-50.



Fonte: Autoral.

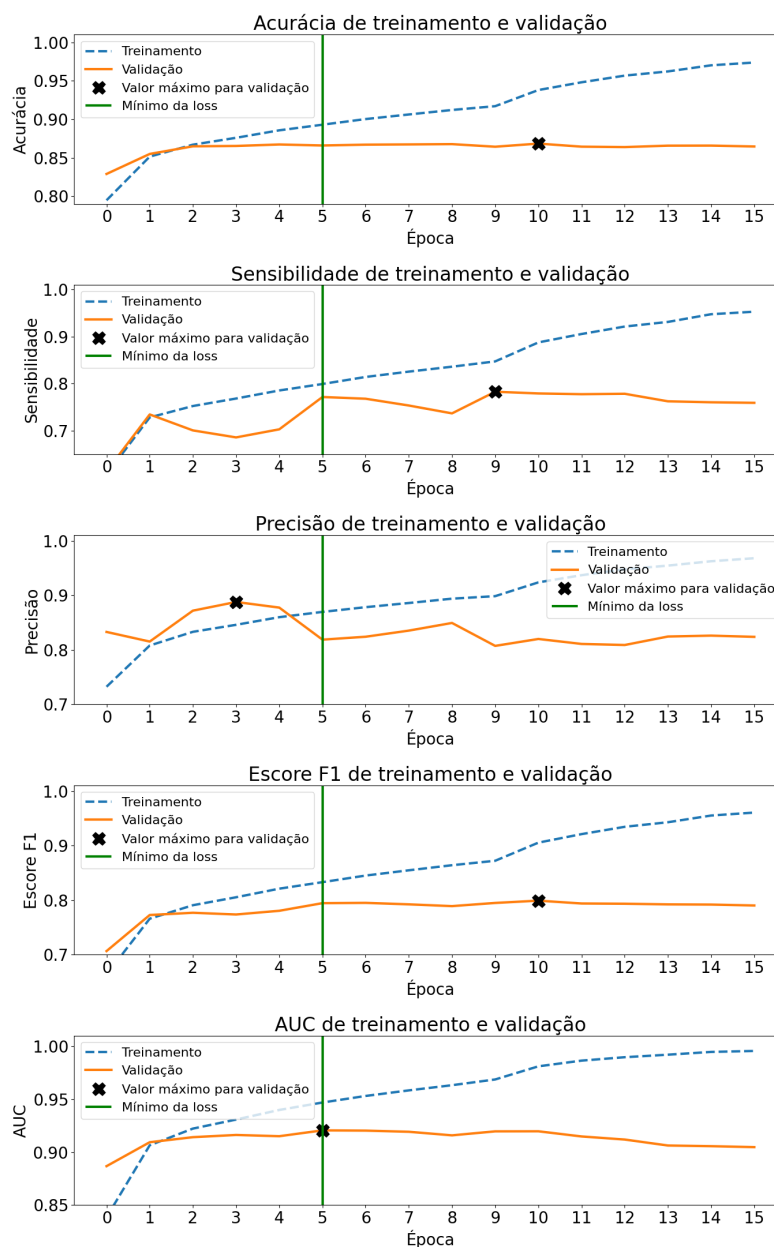
Considerando a rodada 4:

Figura B.45 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-50.



Fonte: Autoral.

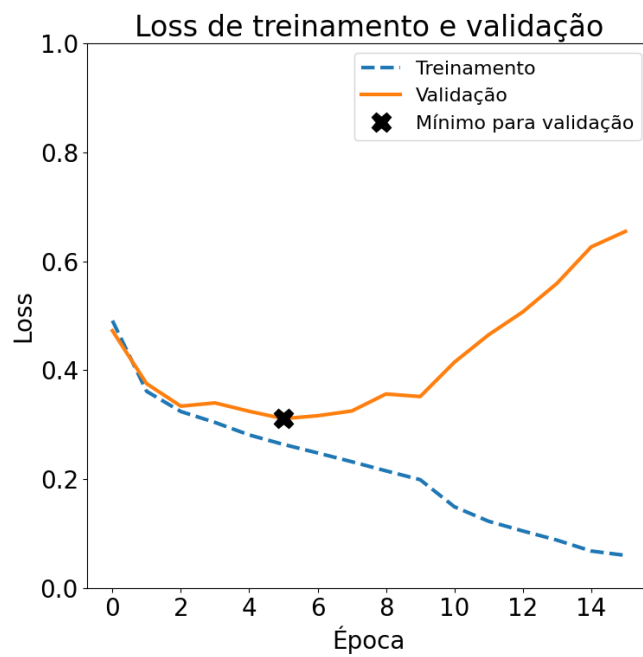
Figura B.46 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-50.



Fonte: Autoral.

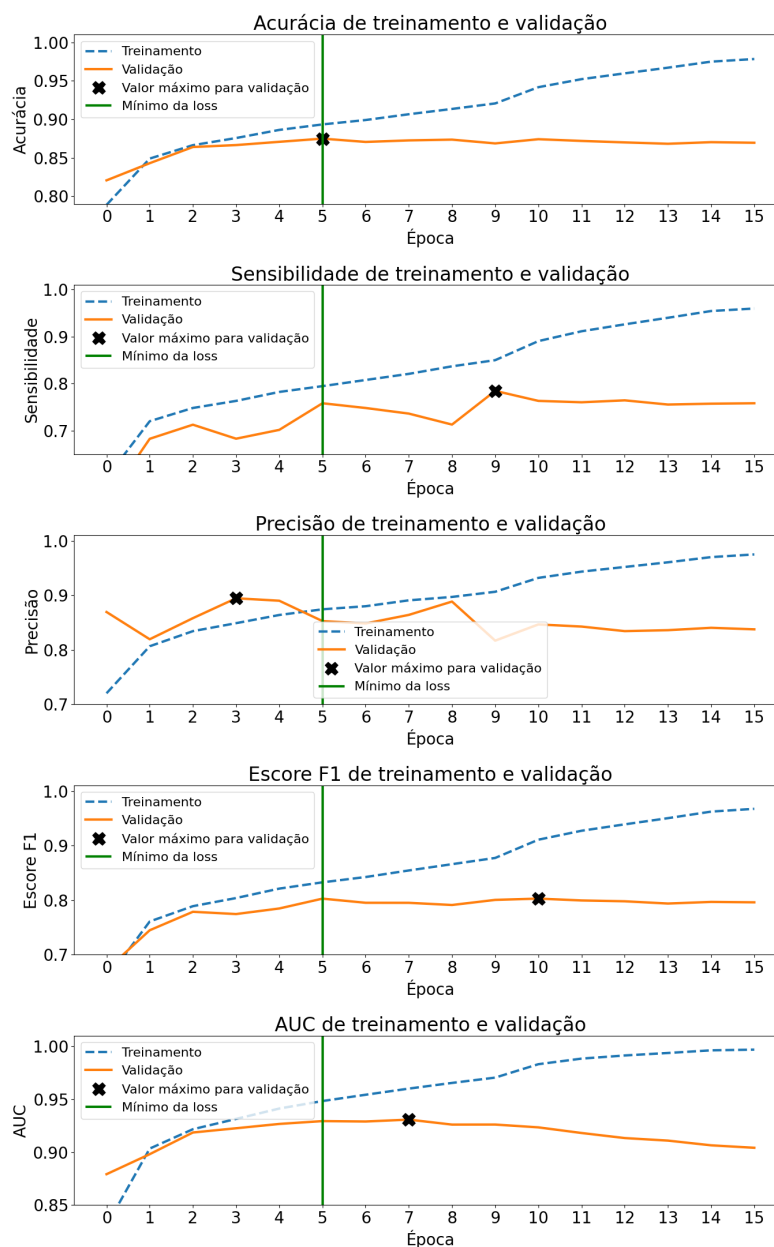
Por fim, a rodada 5:

Figura B.47 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-50.



Fonte: Autoral.

Figura B.48 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-50.

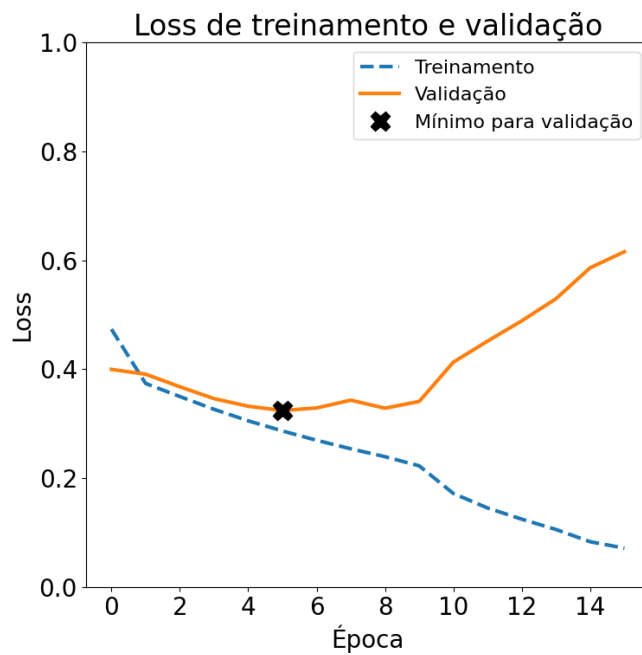


Fonte: Autoral.

B.2.3 – ResNet-101

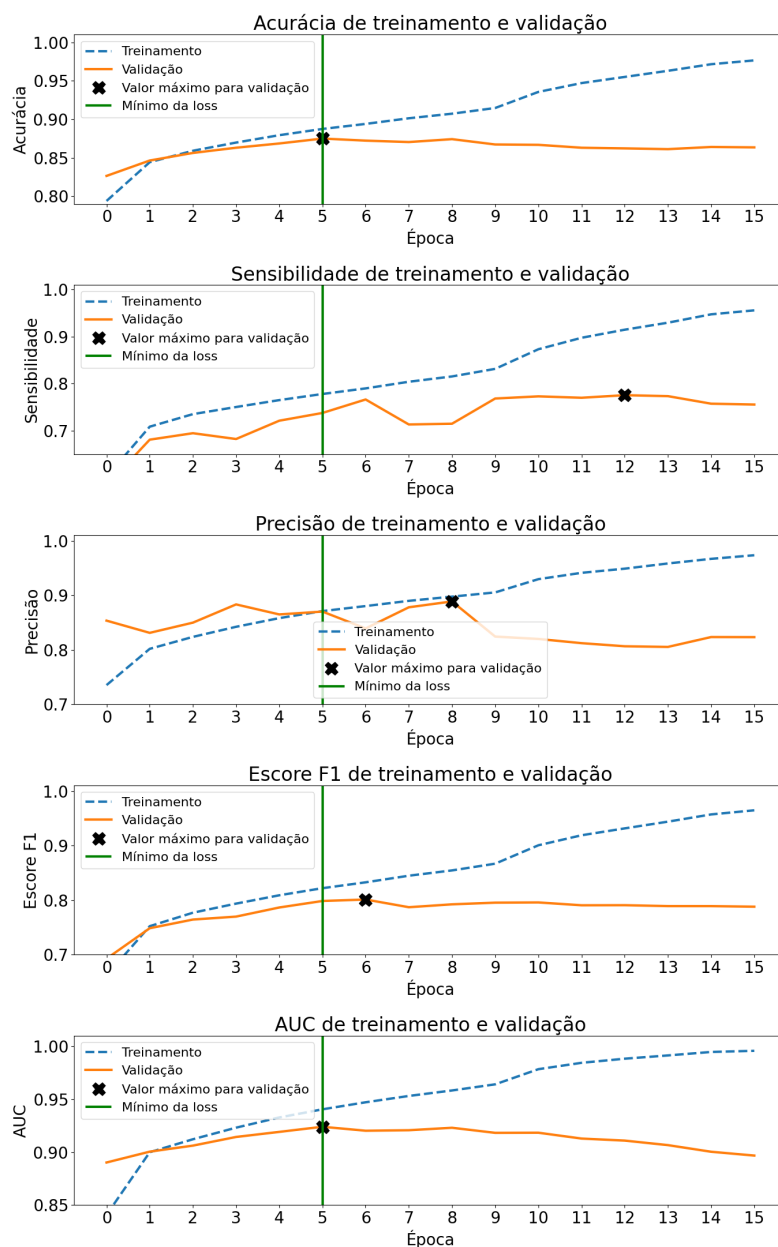
Se obteve para a rodada 1:

Figura B.49 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-101.



Fonte: Autoral.

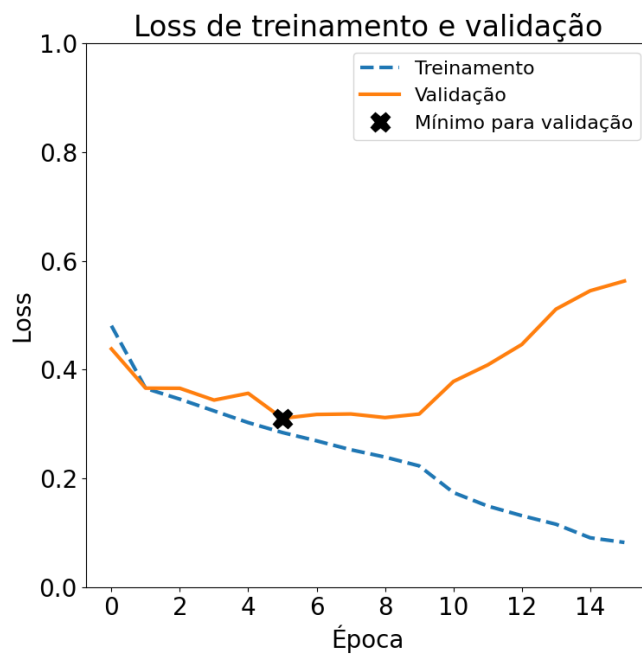
Figura B.50 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 1 com a arquitetura ResNet-101.



Fonte: Autoral.

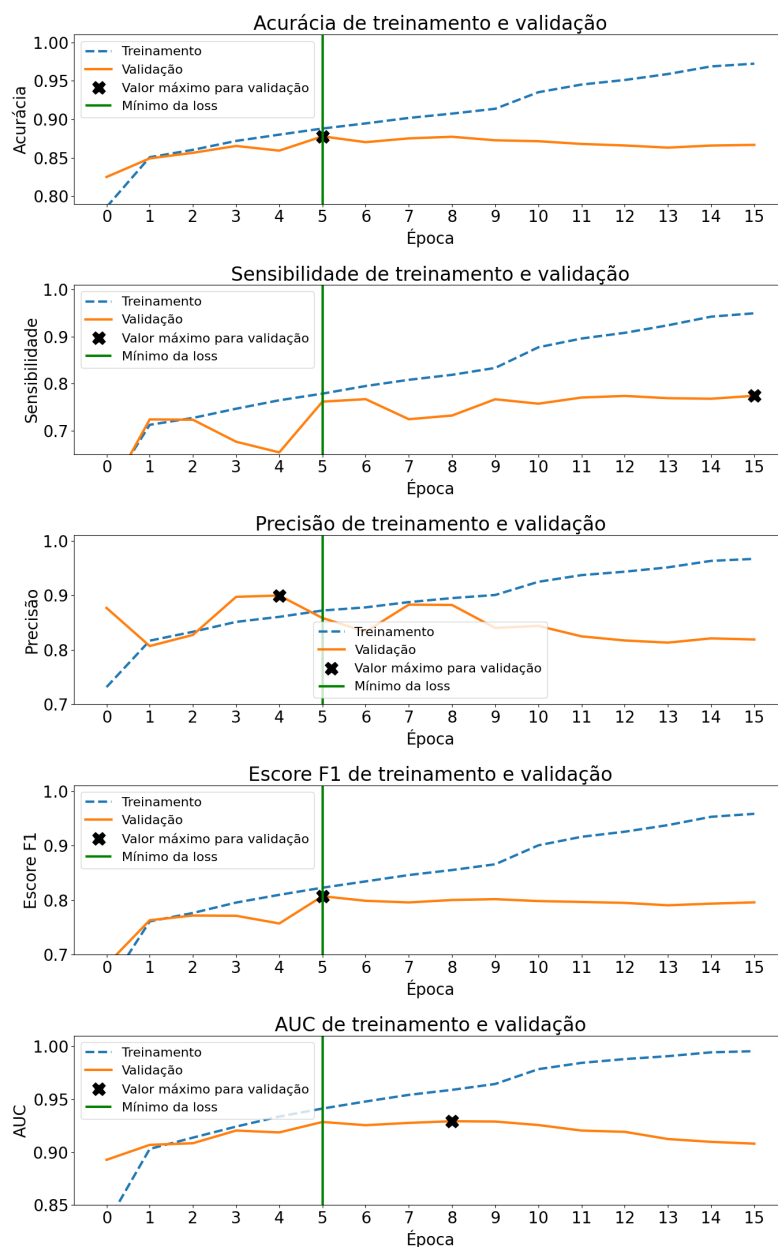
Para a rodada 3:

Figura B.51 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura ResNet-101.



Fonte: Autoral.

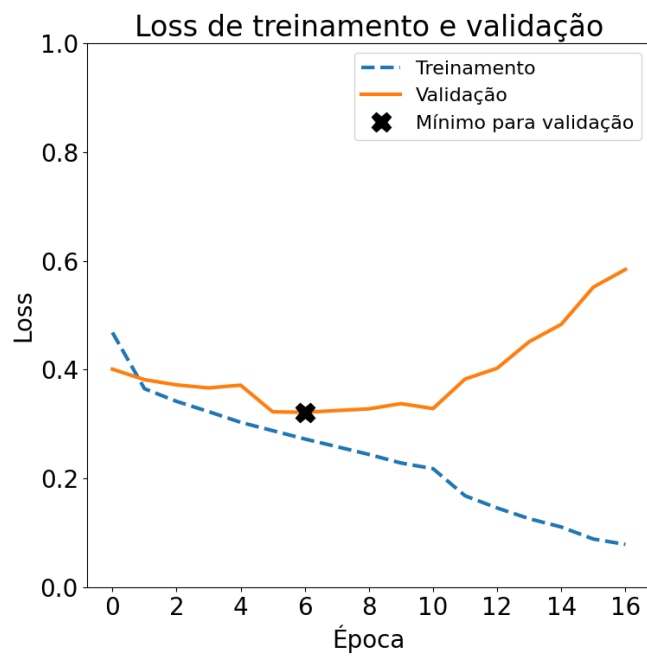
Figura B.52 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura ResNet-101.



Fonte: Autoral.

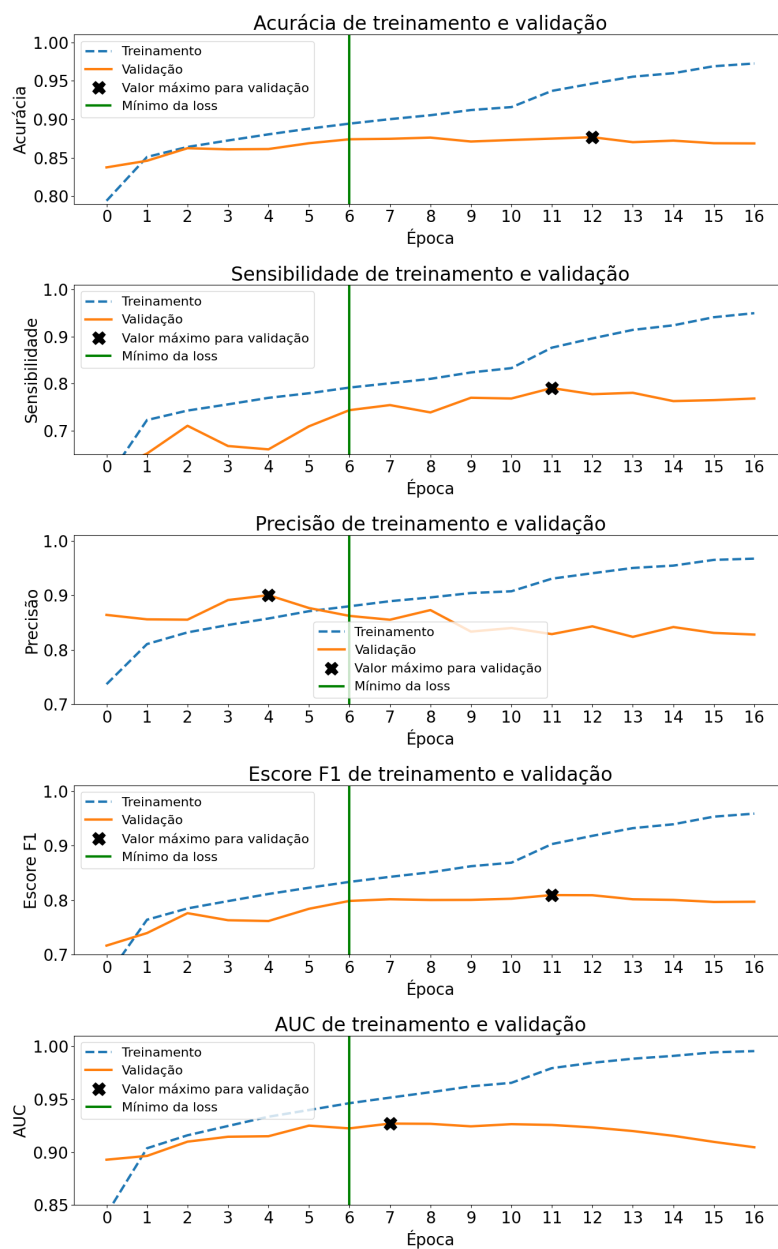
Considerando a rodada 4:

Figura B.53 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-101.



Fonte: Autoral.

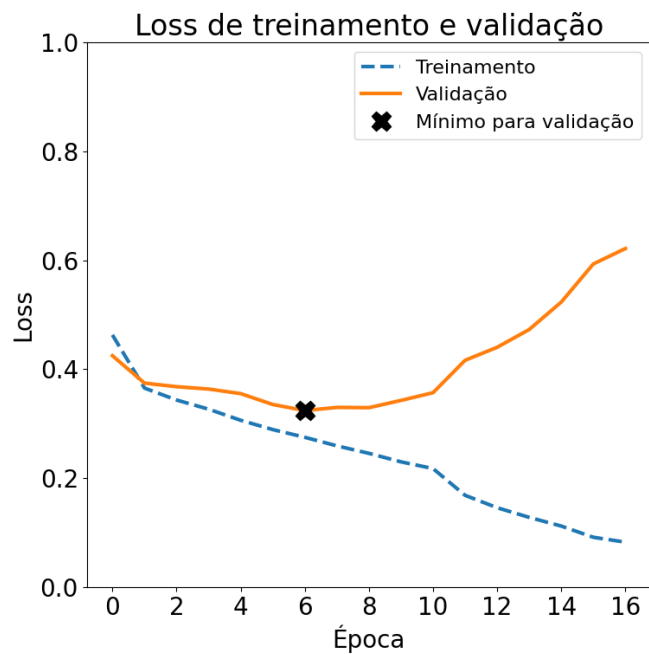
Figura B.54 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura ResNet-101.



Fonte: Autoral.

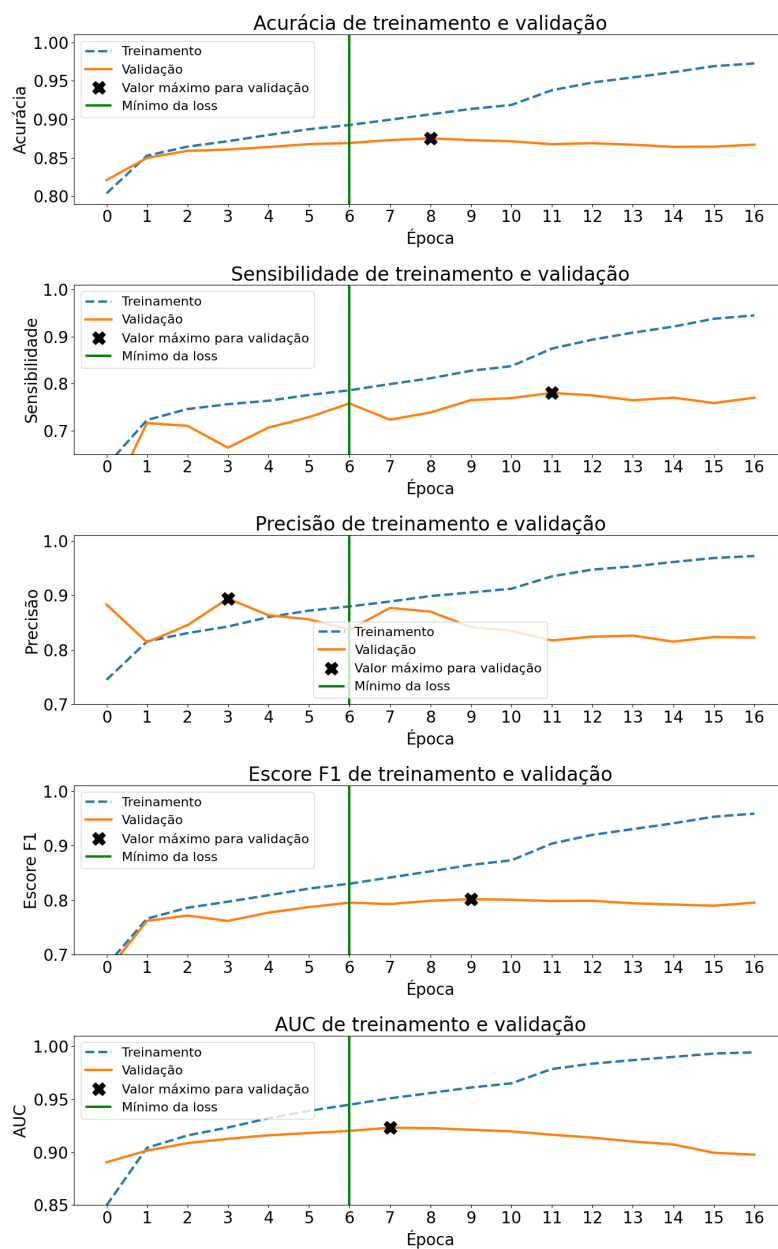
Por fim, para a rodada 5:

Figura B.55 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-101.



Fonte: Autoral.

Figura B.56 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura ResNet-101.

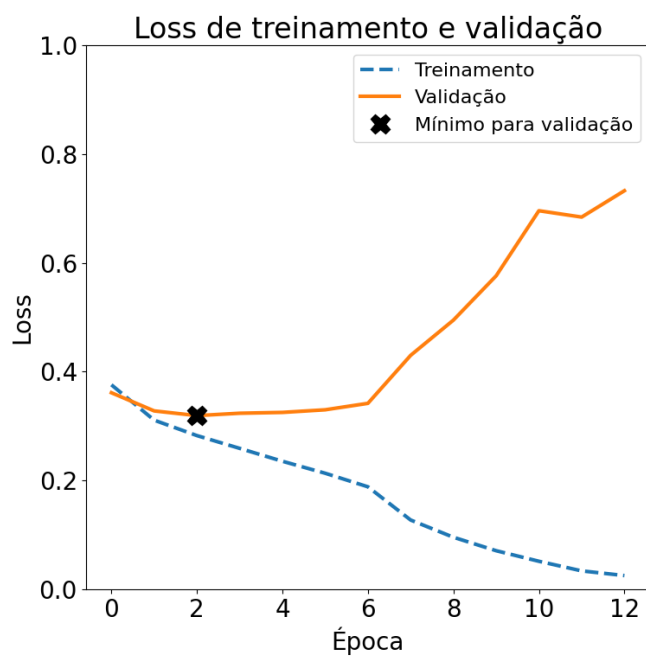


Fonte: Autoral.

B.2.4 – DenseNet201

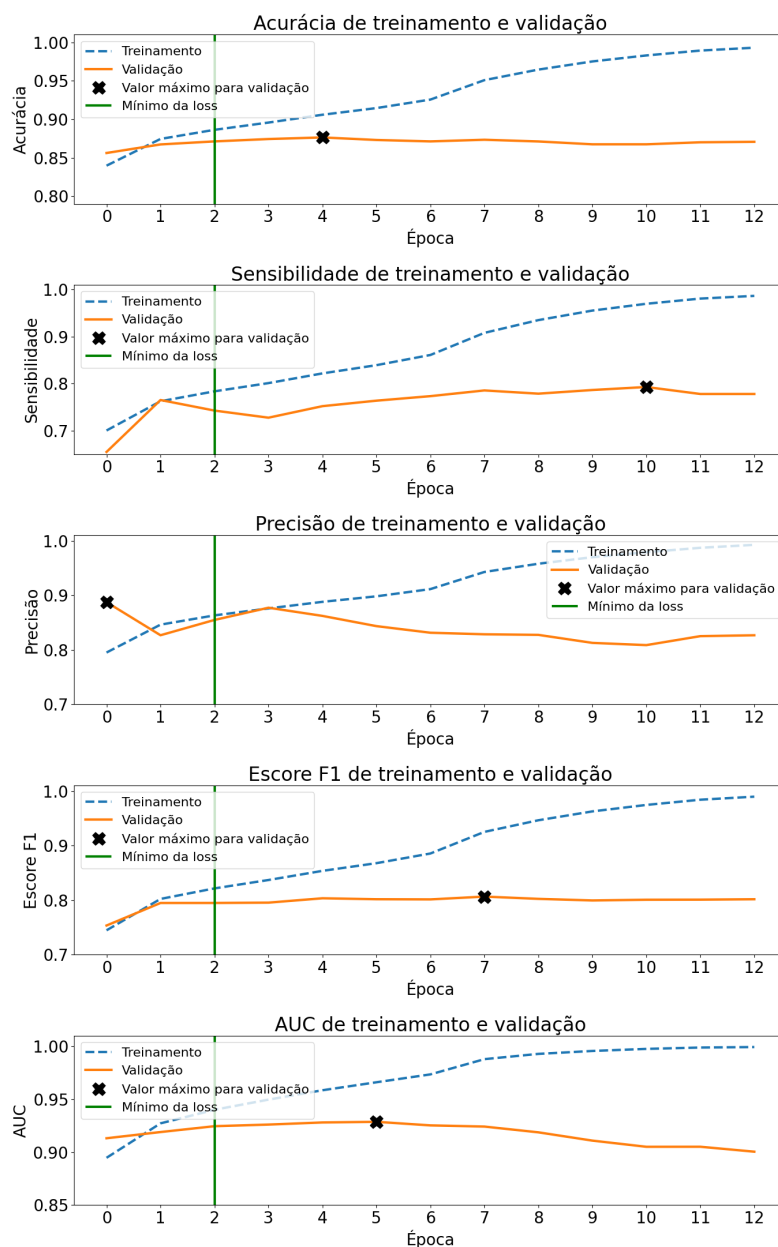
Para a rodada 2:

Figura B.57 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura DenseNet201.



Fonte: Autoral.

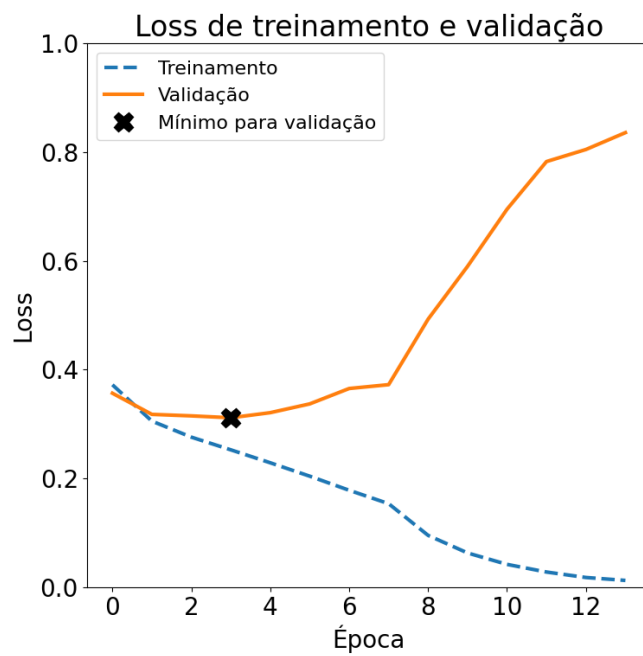
Figura B.58 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 2 com a arquitetura DenseNet201.



Fonte: Autoral.

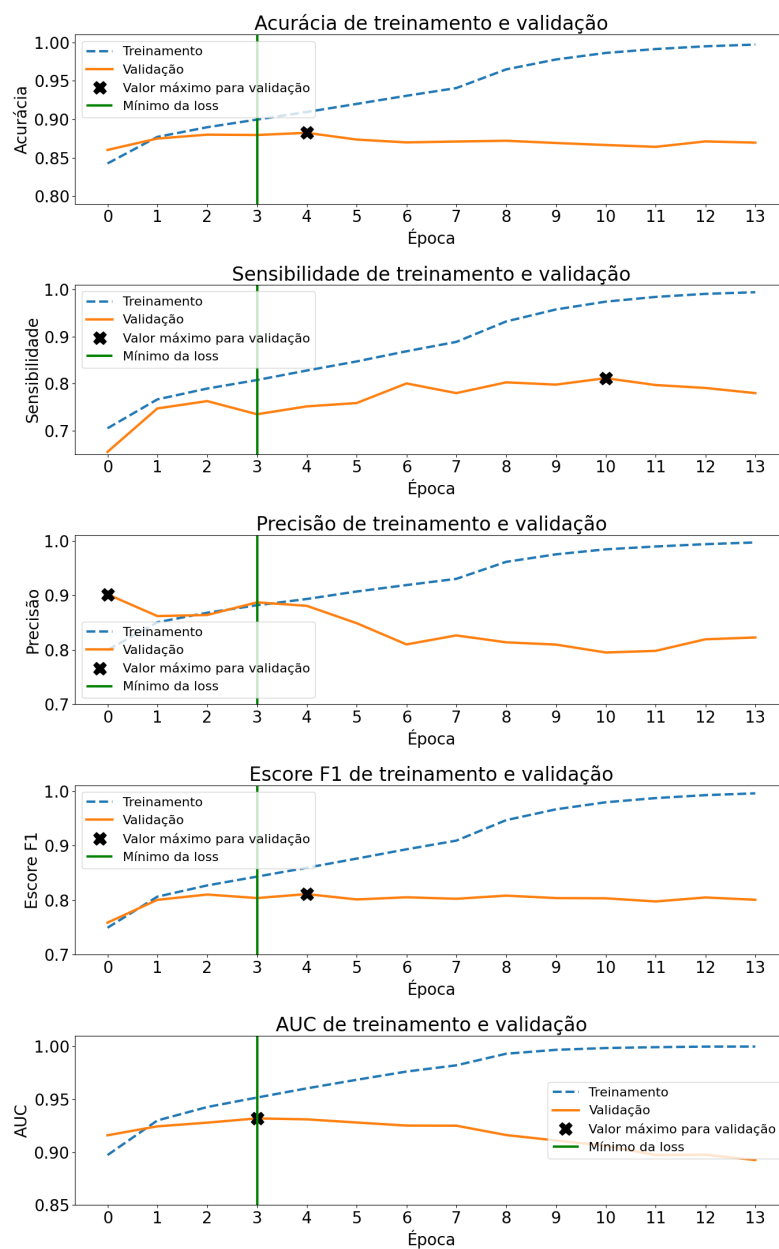
Considerando a rodada 3:

Figura B.59 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura DenseNet201.



Fonte: Autoral.

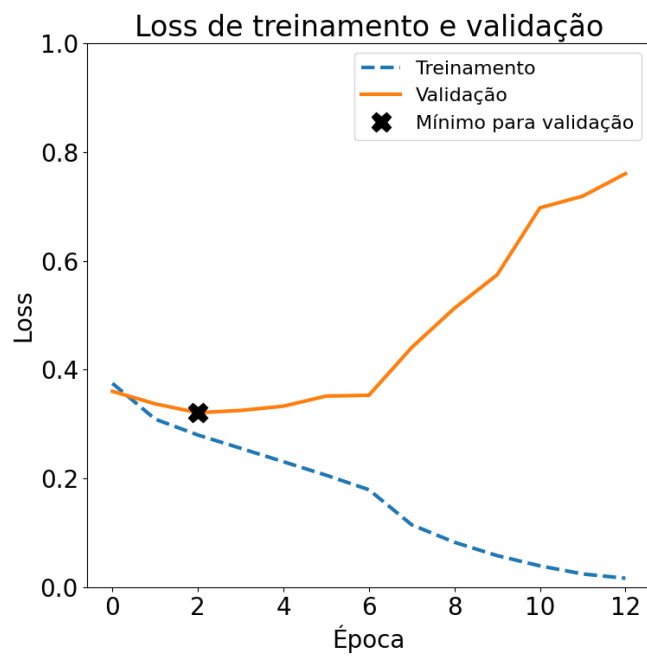
Figura B.60 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 3 com a arquitetura DenseNet201.



Fonte: Autoral.

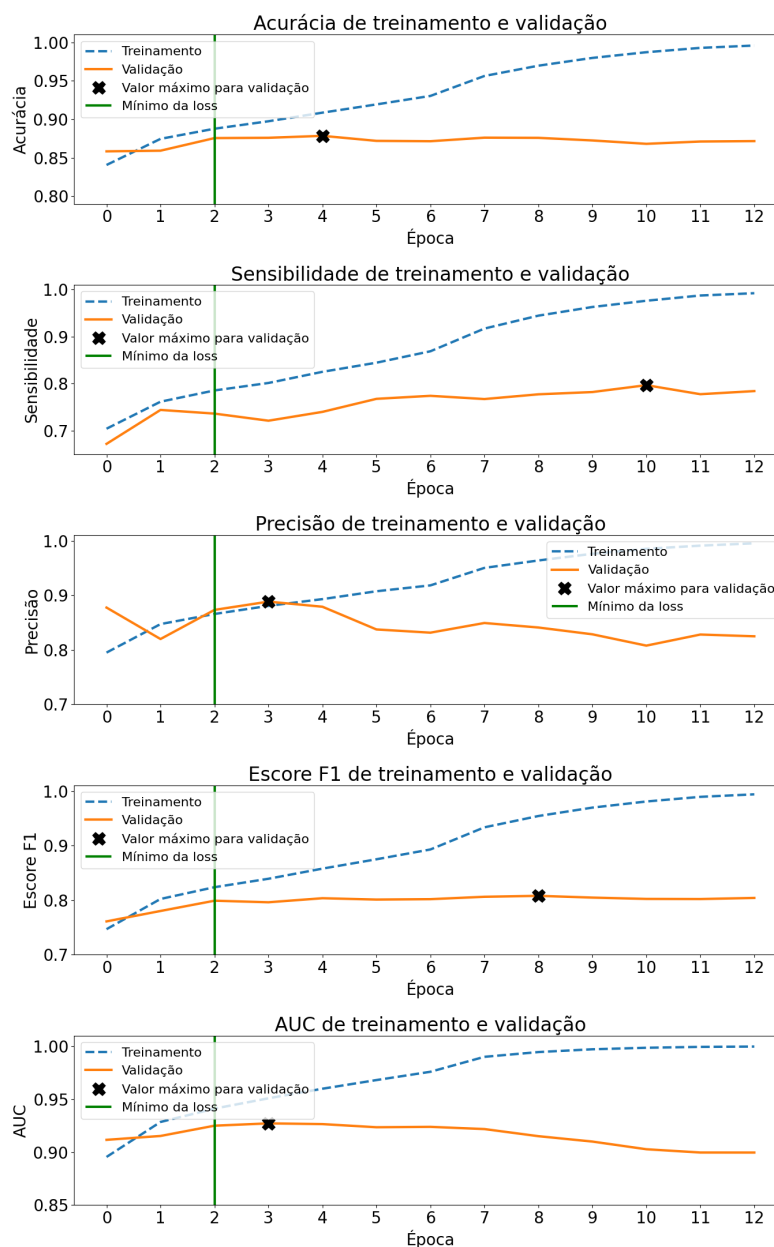
Para a rodada 4:

Figura B.61 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura DenseNet201.



Fonte: Autoral.

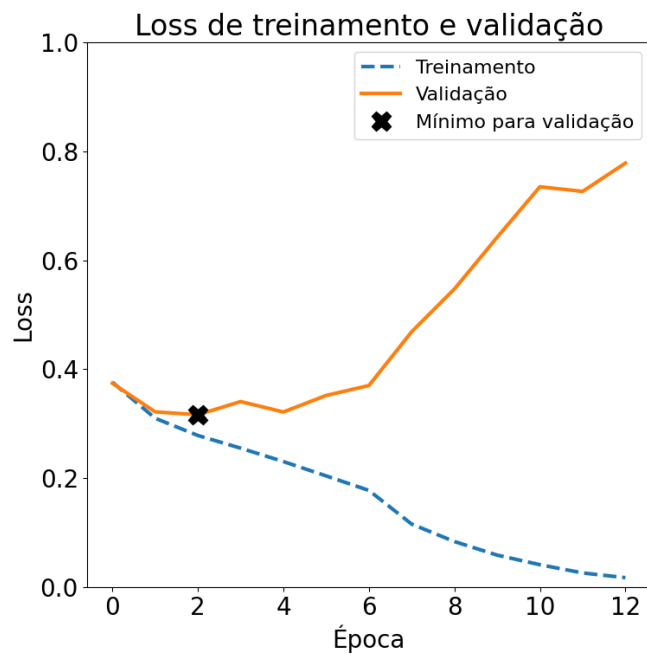
Figura B.62 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 4 com a arquitetura DenseNet201.



Fonte: Autoral.

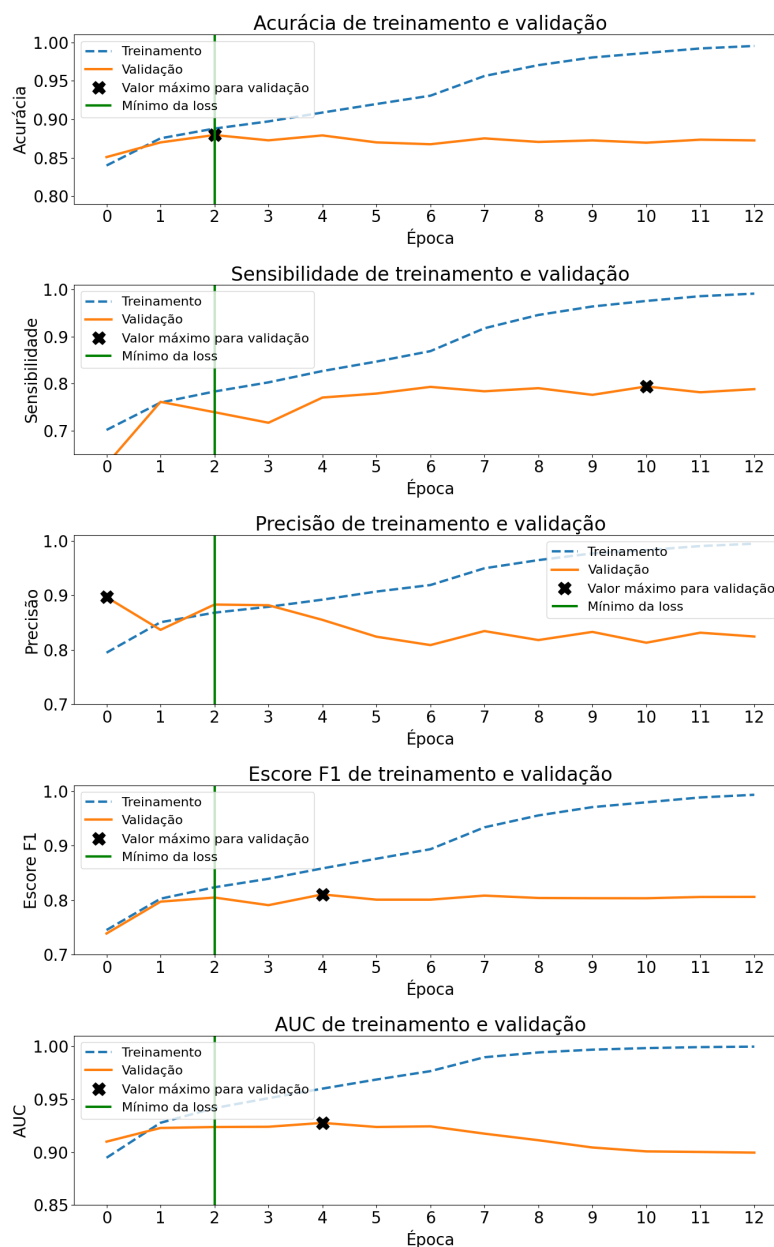
Por fim, para a rodada 5:

Figura B.63 – Evolução por época da métrica *loss* para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura DenseNet201.



Fonte: Autoral.

Figura B.64 – Evolução por época das métricas acurácia, sensibilidade, precisão, escore F1 e AUC para os conjuntos de treino e validação utilizando RP para a rodada 5 com a arquitetura DenseNet201.



Fonte: Autoral.