UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO

THALYSSA BAIOCCO RODRIGUES

ESTIMATIVA DA POSE 3D DE INSTRUMENTOS CIRÚRGICOS EM FLUOROSCOPIA ATRAVÉS DE REDES NEURAIS PROFUNDAS

VITÓRIA
2022

THALYSSA BAIOCCO RODRIGUES

ESTIMATIVA DA POSE 3D DE INSTRUMENTOS CIRÚRGICOS EM FLUOROSCOPIA ATRAVÉS DE REDES NEURAIS PROFUNDAS

Parte manuscrita do Projeto de Graduação da aluna Thalyssa Baiocco Rodrigues, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Orientadores: Dr. Pascal Cathier e Dr. Thierry Lefevre
Coorientadora: Profa. Dra. Raquel Frizera Vassallo

VITÓRIA
2022

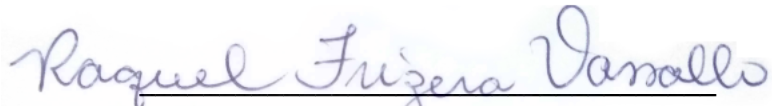
THALYSSA BAIOCCO RODRIGUES

ESTIMATIVA DA POSE 3D DE INSTRUMENTOS CIRÚRGICOS EM FLUOROSCOPIA ATRAVÉS DE REDES NEURAIS PROFUNDAS

Parte manuscrita do Projeto de Graduação da aluna Thalyssa Baiocco Rodrigues, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

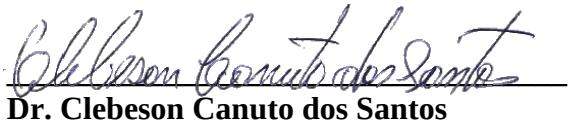
Aprovada em 22 de março de 2022.

COMISSÃO EXAMINADORA:



Profa. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo
Coorientadora

Profa. Dra. Mariana Rampinelli Fernandes
IFES - Vitória
Examinadora



Dr. Clebeson Canuto dos Santos
ISVision - Soluções Inteligentes
Examinador

RESUMO

O intuito deste projeto é obter um sistema capaz de estimar a pose 3D (posição e rotação) de um instrumento médico a partir de uma imagem de raios-X, para auxiliar médicos em intervenções cirúrgicas pouco invasivas. Este tipo de intervenção tem mudado positivamente o cenário das cirurgias cardíacas, apresentando menor risco ao paciente e exigindo um tempo de recuperação mais curto quando comparada a práticas cirúrgicas tradicionais. A abordagem utilizada é baseada na estimação da transformação 3D entre duas projeções 2D do instrumento médico: a imagem de raios-X observada, e uma projeção de referência para a qual a pose 3D que a gerou é conhecida. Duas redes neurais profundas foram utilizadas para tal finalidade: ProjectNet, uma rede para gerar uma projeção 2D artificial do objeto de interesse a partir de uma pose 3D de entrada, e DeltaPoseNet, uma rede para estimar a transformação 3D entre a projeção artificial e a imagem observada do objeto. Além da translação e da rotação, o objeto de interesse apresenta um parâmetro de configuração que implica transformações não-rígidas à projeção do objeto. Com a ProjectNet, foi possível gerar projeções do objeto de maneira contínua para todos os parâmetros de sua pose e de sua configuração, ainda que o conjunto de treinamento da rede fosse limitado a uma representação discreta de determinadas configurações. Com a DeltaPoseNet, foi possível codificar uma medida de similaridade baseada na estimativa de diferença de pose entre duas imagens de maneira consistente: ou seja, a diferença de pose estimada sendo de fato menor quanto mais próximas as imagens são uma da outra.

Palavras-chave: Redes neurais profundas. Processamento de imagens médicas. Cirurgia cardíaca. Intervenções cirúrgicas pouco invasivas. Intervenção/Reparação Transcateter da Válvula Mitral.

ABSTRACT

The purpose of this project is to obtain a system capable of estimating the 3D pose of a medical instrument from an X-ray image, to assist doctors in minimally invasive surgical interventions. This type of intervention has positively changed the cardiac surgery scenario, presenting less risk to the patient, and requiring a shorter recovery time when compared to traditional surgical practices. The approach used is based on estimating the 3D transformation between two 2D projections of the medical instrument: the observed X-ray image, and a reference projection for which the 3D pose that generated it is known. Two deep neural networks were used for this purpose: ProjectNet, a network for generating an artificial 2D projection of the object of interest from a 3D pose as input, and DeltaPoseNet, a network for estimating the 3D transformation between the artificial projection and the observed image of the object. In addition to translation and rotation, the object of interest has a configuration parameter that causes non-rigid transformations to the object projection. With ProjectNet, it was possible to generate projections of the object in a continuous manner for all its pose and configuration parameters, even though the network's training set was limited to a discrete representation of certain configurations. With DeltaPoseNet, it was possible to encode a similarity measure based on the estimated pose difference between two images in a consistent manner: that is, the estimated pose difference being smaller the closer the images are to each other.

Keywords: Deep neural networks. Medical image processing. Cardiac surgery. Minimally invasive surgical interventions. Transcatheter Mitral Valve repair (TMVr).

LISTA DE FIGURAS

Figura 1 – MitraClip, instrumento médico utilizado para cirurgias cardíacas que foi objeto de interesse deste projeto 13
Figura 2 – Diferentes partes de um coração humano 14
Figura 3 – Ilustração do princípio da abordagem utilizada, a ser detalhada no Capítulo 2 15
Figura 4 – Ilustração do método iterativo de estimação de pose de DeepIM 18
Figura 5 – Ilustração do método proposto: a) geração de uma projeção do objeto a partir de uma pose de entrada; b) estimativa da transformação de pose (diferença de pose) entre a projeção gerada e uma imagem observada 23
Figura 6 – Ilustração da parametrização usada para a pose 3D 24
Figura 7 – Ilustração dos efeitos de mudanças aplicadas aos diferentes parâmetros de pose 3D 24
Figura 8 – Exemplos de projeções a partir dos sete modelos 3D existentes para o MitraClip (correspondendo a sete ângulos diferentes entre os braços do MitraClip) 26
Figura 9 – Exemplos de projeções mantidas para formar o conjunto de dados (1-3) e de projeção não mantida (4) 26
Figura 10 - Distribuição dos valores dos parâmetros nos dados disponíveis. Parâmetros de translação dados em mm (x, y, z), parâmetros de rotação dados em graus (α, τ, ρ) 27
Figura 11 – Esquemático da arquitetura da DeltaPoseNet 28
Figura 12 – Esquemático da arquitetura da ProjectNet 34
Figura 13 – Ilustração do sistema que liga ProjectNet e DeltaPoseNet como uma função a ser otimizada 36
Figura 14 – a) Curvas da função de perda de treino e teste para DeltaPoseNet$_{V1}$ (treinada com conjunto de dados contendo todos os modelos do MitraClip) – pares gerados **sem** restrições de distância máxima; b) erros para cada parâmetro separadamente. Podemos observar que a perda global de teste é alta e começa a divergir, o que se deve a um mau desempenho na previsão de α e τ. 40
Figura 15 – a Curvas da função de perda de treino e teste para DeltaPoseNet$_{V1}$ (treinada com conjunto de dados contendo todos os modelos do MitraClip) – pares gerados com uma distância máxima de 0,8rad (~46°) em α e τ ($\Delta\alpha \leq 0{,}8\text{rad}$ e $\Delta\tau \leq 0{,}8\text{rad}$); b) erro correspondente para cada parâmetro separadamente. 41
Figura 16 – Comparação por parâmetro entre os treinamentos da DeltaPoseNet$_{V1}$ com e sem restrições de distância máxima em α e τ 41

Figura 17 – Comparação entre versões da DeltaPoseNet$_{V1}$ treinadas com e sem blocos residuais 42
Figura 18 – a) Curvas da função de perda de treino e teste para DeltaPoseNet$_{V2}$ (treinada com modelos A, C, E, G do MitraClip) – pares gerados com uma distância máxima de 0,8rad (~46°) em α e τ; b) erro correspondente para cada parâmetro separadamente 43
Figura 19 – Comparação entre DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$ no conjunto de teste. Na parte superior esquerda, podemos ver a evolução da função $L\Delta\mathrm{Pose}$ sobre o conjunto de teste ao longo das épocas de treinamento da rede. Os outros gráficos mostram a evolução do erro para os diferentes parâmetros de pose 44
Figura 20 – Diferenças angulares entre os modelos de objetos, mostrando que, entre os modelos B, D e F, o modelo F é o que apresenta a maior distância de ângulo de braço comparado aos seus vizinhos na representação discreta (21,91° e 49,34°). 47
Figura 21 – Evolução da função de perda ao longo das épocas durante o treinamento da ProjectNet 47
Figura 22 – Influência da alteração dos diferentes parâmetros na projeção de saída da ProjectNet. Os valores dos parâmetros foram tomados uniformemente espaçados no intervalo dos parâmetros dos dados de treinamento, mostrados na Figura 10..... 48
Figura 23 – Evolução do parâmetro ρ de $-\pi$ (projeção à extrema esquerda) a $+\pi$ (projeção à extrema direita), com y fixo a $+10$ 49
Figura 24 – Evolução do parâmetro ρ de $-\pi$ (projeção à extrema esquerda) a $+\pi$ (projeção à extrema direita), com y fixo a -10 49
Figura 25 – Histograma conjunto para os dados de treinamento mostrando os parâmetros y e ρ 50
Figura 26 – Ilustração de situações em que a projeção de saída foi a esperada (verde) e onde surgiram problemas (vermelho). Pode-se ver a relação com a distribuição dos dados de treinamento: a rede teve dificuldades em regiões que não estavam representadas pelos dados de treinamento 50
Figura 27 – Saídas da ProjectNet para diferentes valores de ângulo θarm entre os braços do MitraClip (valores não representados pelo conjunto de dados), mostrando a capacidade da rede de interpolar novas configurações do clipe..... 51
Figura 28 – Funções de perda sobre o conjunto de treinamento e de teste durante o treinamento da DeltaPoseNet$_{V3}$ (nos gráficos, Exp. 1, com pares de todos os modelos de

objetos) e DeltaPoseNet$_{V4}$ (nos gráficos, Exp. 2, com pares de modelos de objetos A, C, E, G). Os pares de imagens consistem em uma DRR e uma projeção da ProjectNet.....53
Figura 29 – Erro entre os valores previstos e a *ground-truth* sobre os dados de teste, para os diferentes parâmetros de pose, durante o treinamento da DeltaPoseNet$_{V3}$ (nos gráficos, Exp. 1, com pares de todos os modelos de objetos) e DeltaPoseNet$_{V4}$ (nos gráficos, Exp. 2, com pares de modelos de objetos A, C, E, G). Os pares de imagens consistem em uma DRR e uma projeção da ProjectNet.....53
Figura 30 – Transição de pose através da atualização iterativa dos parâmetros de pose inicial usando a diferença de pose prevista pela DeltaPoseNet.....56
Figura 31 – Avaliação da medida de similaridade $Spose$ sobre alguns exemplos. Inicia-se a partir de uma postura centrada e evolui-se para a mesma postura que a postura alvo, calculando o valor de $Spose$ a cada nova projeção gerada. Os gráficos à direita mostram a evolução da medida $Spose$.....57
Figura 32 – Curvas da função de perda de treino e teste para DeltaPoseNet treinada considerando a diferença relativa de pose entre as imagens de entrada (azul) e considerando a diferença de pose absoluta (laranja) (conjunto de dados contendo todos os modelos do MitraClip).....64
Figura 33 – Erros durante o treinamento para os diferentes parâmetros de pose sobre o conjunto de **treinamento**, comparando as versões absolutas e relativas da DeltaPoseNet....65
Figura 34 - Erros durante o treinamento para os diferentes parâmetros de pose sobre o conjunto de **teste**, comparando as versões absolutas e relativas da DeltaPoseNet.....65

LISTA DE TABELAS

Tabela 1 – Resultados da DeltaPoseNet$_{V1}$ por modelo do MitraClip (pares de imagem compostos por duas DRRs) 45
Tabela 2 – Resultados da DeltaPoseNet$_{V2}$ por modelo do MitraClip (pares de imagem compostos por duas DRRs) 45
Tabela 3 – Comparação do desempenho das diferentes versões da DeltaPoseNet sobre os conjuntos de teste correspondentes 52
Tabela 4 – Resultados da DeltaPoseNet$_{V3}$ por modelo do MitraClip (pares de imagem compostos por uma DRR e uma projeção da ProjectNet) 54
Tabela 5 – Resultados da DeltaPoseNet$_{V4}$ por modelo do MitraClip (pares de imagem compostos por uma DRR e uma projeção da ProjectNet) 54

LISTA DE ABREVIATURAS E SIGLAS

BB	Caixa delimitadora (do inglês, *Bounding Box*)
CNN	Rede Neural Convolucional (do inglês, *Convolutional Neural Network*)
DL	Aprendizagem profunda (do inglês, *Deep Learning*)
DRR	Radiografia Reconstruída Digitalmente (do inglês, *Digitally Reconstructed Radiograph*)
FC	Camadas totalmente conectadas (do inglês, *Fully Connected layers*)
PnP	Referência ao algoritmo Perspective-n-Point.
TMVr	Intervenção/Reparação Transcateter da Válvula Mitral (do inglês, *Transcatheter Mitral Valve repair*)
RGB	Referência aos canais vermelho, verde e azul de uma imagem colorida (do inglês, *Red, Green, Blue)*
RGB-D	Referência ao conjunto imagem RGB e imagem de profundidade (do inglês, *Red, Green, Blue, Depth)*

LISTA DE SÍMBOLOS

x, y, z	Parâmetros de translação (mm)
α, τ, ρ	Parâmetros de rotação (°)
δ	Faz referência a um parâmetro qualquer (entre $x, y, z, \alpha, \tau, \rho$)
$\boldsymbol{\Delta pose}$	Vetor de diferença de pose (incluindo parâmetros de translação e rotação)
θ_{arm}	Ângulo entre os "braços" do MitraClip

SUMÁRIO

1 INTRODUÇÃO 13
1.1 Visão geral e motivação 13
1.2 Trabalhos relacionados 16
1.3 Estrutura do texto 21

2 METODOLOGIA 22
2.1 Formulação do problema 23
2.2 Materiais e recursos 25
2.3 DeltaPoseNet 27
2.3.1 Treinamento da DeltaPoseNet 29
2.3.1.1 Função de perda 29
2.3.1.2 Conjunto de dados e treinamento 30
2.4 ProjectNet 33
2.4.1 Treinamento da ProjectNet 34
2.4.1.1 Função de perda 34
2.4.1.2 Conjunto de dados e treinamento 35
2.5 Project-DeltaPose-Net: conectando DeltaPoseNet e ProjectNet 35
2.5.1 Treinamento da Project-DeltaPose-Net 36
2.5.2 Refinamento de pose 37

3 RESULTADOS 39
3.1 DeltaPoseNet$_{V1}$ – Treinamento usando todos os modelos disponíveis do MitraClip 39
3.2 DeltaPoseNet$_{V2}$ – Treinamento usando modelos A, C, E, G do MitraClip 42
3.3 Comparação entre DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$ no conjunto de teste 44
3.4 ProjectNet 47
3.5 Project-DeltaPose-Net 51
3.5.1 Treinamento da DeltaPoseNet$_{V3}$ e da DeltaPoseNet$_{V4}$ 51
3.5.2 Refinamento de pose 55

4 CONCLUSÕES 58

REFERÊNCIAS BIBLIOGRÁFICAS 60

APÊNDICE A – EXPERIMENTOS COM DELTAPOSENET UTILIZANDO DIFERENÇA DE POSE RELATIVA.....64

1 INTRODUÇÃO

1.1 Visão geral e motivação

Um tópico de grande interesse no campo médico são os procedimentos cardíacos minimamente invasivos, pois consistem em intervenções mais seguras e que proporcionam uma recuperação mais rápida para o paciente, quando comparadas com intervenções clássicas “a peito aberto”. Tais procedimentos são realizados inserindo instrumentos cirúrgicos no corpo do paciente através de uma pequena incisão e, em seguida, direcionando-os para o local desejado através do sistema sanguíneo. O percurso dos instrumentos é monitorado pelo médico através de um sistema de imagens em tempo real baseado em raios-X chamado fluoroscopia.

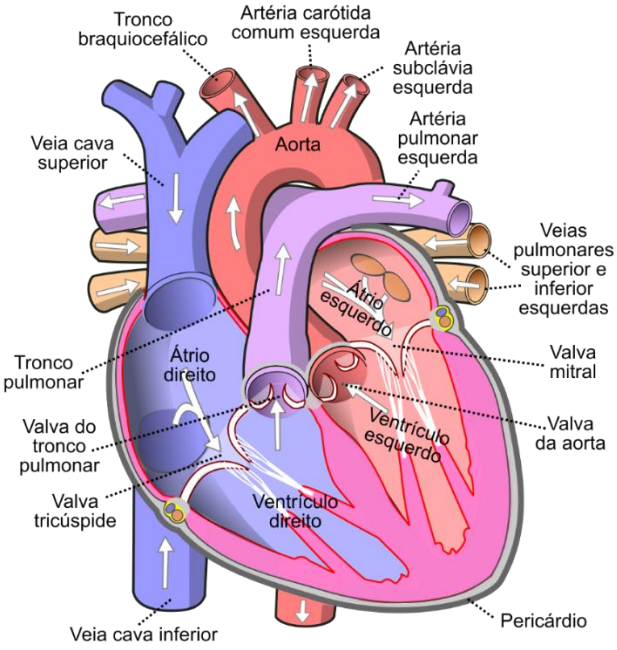
Uma das soluções da Philips, empresa com a qual este projeto foi desenvolvido, envolve o *tracking* por fluoroscopia do “MitraClip” (Figura 1), objeto utilizado em um procedimento chamado Intervenção/Reparação Transcateter da Válvula Mitral (em inglês, *Transcatheter Mitral Valve repair* - TMVr). Este procedimento visa reverter uma condição conhecida como regurgitação da válvula mitral. Em um coração saudável, válvulas delicadas, como a válvula mitral, asseguram um fluxo de sangue unidirecional através das câmaras do coração. A regurgitação da válvula mitral é caracterizada por uma anormalidade que impede que a válvula se feche perfeitamente, causando fluxo de sangue na direção oposta: em vez de fluir apenas para a aorta durante a fase de expulsão do sangue, uma parte do sangue flui de volta do ventrículo esquerdo para o átrio esquerdo (a Figura 2 ilustra as diferentes partes de um coração humano).

Figura 1 – MitraClip, instrumento médico utilizado para cirurgias cardíacas que foi objeto de interesse deste projeto

Fonte: Fava, 2019.

O procedimento TMVr é muito menos invasivo do que uma cirurgia a peito aberto, que pode não ser uma opção para alguns pacientes (ABBOTT GROUP OF COMPANIES, 2020). É importante destacar que as estruturas tipo clipe do MitraClip podem ser abertas e fechadas. Assim, ele é transportado através do sistema sanguíneo com o clipe fechado, é aberto para passar através da válvula mitral e depois é fechado novamente, de forma a manter os tecidos da válvula mitral juntos.

Figura 2 – Diferentes partes de um coração humano



Fonte: SanarFlix, 2019.

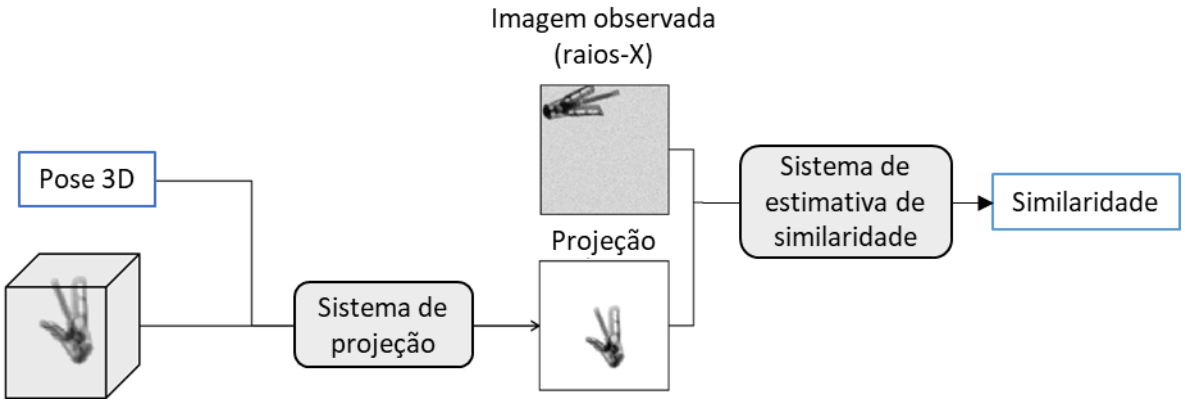
Durante cirurgias cardíacas, é interessante para o médico utilizar diferentes modalidades de imagem. Imagens de raios-X permitem uma boa visualização do MitraClip, mas tecidos moles, como os do coração, não são bem visíveis. Por outro lado, tais tecidos são bem visíveis em imagens de ultrassom. Assim, é interessante a associação destas duas modalidades de imagem.

Neste contexto, é de interesse da Philips fornecer uma solução que apresente imagens provenientes de ambas as modalidades para o médico durante a cirurgia. Esta solução pode ser resumida da seguinte forma: conhecendo-se a posição da sonda de ultrassom com relação a

fonte de raios-X, se fosse possível encontrar a pose 3D[1] do MitraClip com relação a fonte de raios-X, seria possível encontrar a pose com relação a sonda de ultrassom. Assim, poder estimar a pose 3D do MitraClip a partir de uma imagem de raios-X permitiria uma melhor integração das duas modalidades de imagem.

Tendo essa aplicação como motivação, o objetivo deste projeto é desenvolver um sistema capaz de aprender uma medida de similaridade entre duas imagens (projeções 2D) de um objeto. Como esquematizado na Figura 3, a ideia é que, tendo um conhecimento do modelo 3D do objeto desejado, este sistema possa ser usado para estimar a pose 3D (translação e rotação) deste objeto em uma imagem de raios-X (imagem observada), encontrando a pose de entrada que maximiza esta medida de similaridade. Isto pode ser visto como um problema de registro de imagens, onde a similaridade é tanto maior quanto mais próxima a pose usada para gerar a projeção estiver da pose 3D real do objeto na imagem observada. A metodologia utilizada será detalhada no Capítulo 2. Aqui o objetivo é apenas introduzir uma noção por trás da abordagem escolhida.

Figura 3 – Ilustração do princípio da abordagem utilizada, a ser detalhada no Capítulo 2



Fonte: Produção da própria autora.

[1] Define-se “pose 3D” como o conjunto de parâmetros que caracteriza a posição e a rotação de um objeto.

1.2 Trabalhos relacionados

A localização de objetos no espaço 3D a partir de imagens é um problema fundamental no campo da visão computacional e tem diversas aplicações, tais como reconhecimento de objetos, realidade virtual, manipulação de robôs, realidade aumentada, interpretação de cenas e navegação autônoma. Este problema é comumente designado como estimativa de pose 6D, visto que consiste a estimar a translação 3D e a rotação 3D do objeto, formulação que engloba 6 diferentes parâmetros de pose (SAHIN; GARCIA-HERNAND; SOCK; KIM, 2020).

As abordagens atuais tratam do problema da estimativa de pose 3D direta ou indiretamente. No grupo das abordagens indiretas, uma prática comum é construir correspondências 2D-3D e resolvê-las posteriormente usando um algoritmo Perspective-n-Point (PnP) (LEPETIT; MORENO-NOGUER; FUA, 2009). Os métodos tradicionais recuperam as correspondências 2D-3D extraindo características locais de uma imagem e combinando-as com características em um modelo 3D do objeto (COLLET; MARTINEZ; SRINIVASA, 2011) (ROTHGANGER; LAZEBNIK; SCHMID; PONCE, 2006). Tais métodos, no entanto, mostram uma tendência de baixo desempenho quando lidam com objetos não texturizados, já que a extração de características da imagem depende da informação da textura do objeto. Para superar esta limitação, trabalhos mais recentes propuseram diferentes abordagens para recuperar as correspondências 2D-3D. Alguns métodos exploraram a estimativa de coordenadas 3D de pixels ou *keypoints* pertencentes ao objeto de interesse na imagem de entrada (BRACHMANN; KRULL; MICHEL; GUMHOLD; SHOTTON; ROTHER, 2014) (KRULL; BRACHMANN; MICHEL; YANG; GUMHOLD; ROTHER, 2015) (BRACHMANN; MICHEL; KRULL; YANG; GUMHOLD; ROTHER, 2016). Seguindo esta abordagem baseada em coordenadas, métodos como BB8 (RAD; LEPETIT, 2017) consistem em prever as projeções 2D dos cantos da caixa delimitadora (do inglês, *bounding box* - BB) englobando o modelo 3D do objeto.

Em HybridPose (SONG; SONG; HUANG, 2020), são geradas representações intermediárias a partir da imagem de entrada, numa tentativa de extrair diferentes informações geométricas: *keypoints*, vetores de borda (do inglês, *edge vectors*) e correspondências de simetria. As três representações são estimadas por três redes neurais baseadas na ResNet (HE; ZHANG; REN; SUN, 2016) e são então utilizadas para obter uma estimativa inicial de pose através da solução de um sistema linear. Os autores alegam que o uso da representação híbrida proposta torna seu

modelo mais robusto a situações de oclusão e truncamento. De acordo com suas observações, os *keypoints* contribuem para a precisão da estimativa da translação, enquanto os vetores de borda e as correspondências de simetria ajudam a estabilizar a estimativa da rotação.

Em i3PosNet (KÜGLER; SEHRING; STEFANOV; STENIN; KRISTIN; KLENZNER; SCHIPPER; MUKHOPADHYAY, 2020), o problema da estimativa de pose das imagens de raios-X é abordado indiretamente, com uma rede neural convolucional (do inglês, *Convolutional Neural Network* – CNN) que estima posições de múltiplos *keypoints* pertencentes ao objeto de interesse na imagem de entrada. Os parâmetros de pose do objeto são então obtidos a partir desses *keypoints* através de considerações geométricas. i3PosNet é baseada em *patches* (isto é, subdivisões/porções da imagem original) e, assim como os outros métodos discutidos, usa uma estratégia iterativa de refinamento de pose.

No grupo de métodos diretos para a estimativa da translação e da rotação 3D, o problema ou é convertido em um problema de regressão (XIANG; SCHMIDT; NARAYANAN; FOX, 2018) (SONG; SONG; HUANG, 2020) ou um problema de classificação (HINTERSTOISSER; LEPETIT; ILIC; HOLZER; BRADSKI; KONOLIGE; NAVAB, 2012), caso em que o espaço de pose é discretizado.

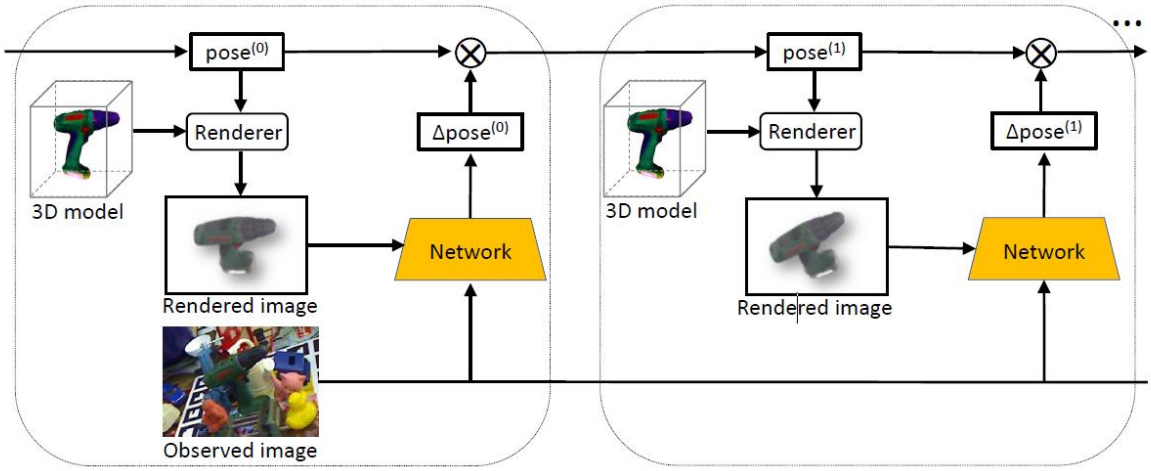
A rede PoseCNN (XIANG; SCHMIDT; NARAYANAN; FOX, 2018) regride a translação 3D e a rotação 3D separadamente: ela consiste de um *backbone* para extrair características, que são então passadas por três ramificações diferentes para obter a segmentação semântica da imagem, e as estimativas de translação e rotação. Os autores também introduziram uma nova função de perda (*ShapeMatch-Loss*) para resolver o problema de ruídos e inconsistências na função de perda devidos a objetos simétricos (por exemplo, elevado valor da função de perda quando a estimativa da rede é, na verdade, correta se for levada em consideração a simetria do objeto), o que influencia negativamente no treinamento da rede.

Li, Wang e Ji (2019) propõem unificar a estratégia indireta baseada em coordenadas e a estratégia direta baseada em regressão para estimar a pose do objeto. Mais precisamente, a translação é estimada diretamente, enquanto a rotação, indiretamente. A imagem é passada através de uma rede (*backbone*) para gerar características (*features*) intermediárias que são então passadas através de duas outras redes: uma estima a translação; a outra estima um

mapa/carta de coordenadas (isto é, coordenadas 3D para cada pixel pertencente ao objeto), que é então usado para solucionar a rotação através de PnP, com a ajuda do algoritmo RANSAC (FISCHLER, M. A.; BOLLES, 1981) para filtrar os *outliers*.

A forma como a maioria desses métodos é desenvolvida os torna especializados em um conjunto limitado de categorias de objetos, não sendo, assim, capazes de generalizar para outras categorias. Li, Wang, Ji, Xiang e Fox (2020) desenvolveram uma estratégia que permite que sua rede seja mais independente das formas e características dos objetos. Sua abordagem, ilustrada na Figura 4 e chamada Deep Iterative Matching, ou DeepIM, consiste em comparar imagens renderizadas do objeto de interesse com a imagem de entrada. Eles treinam uma rede para prever a transformação de pose relativa entre uma imagem renderizada (obtida a partir de uma estimativa inicial de pose) e a imagem observada. Esta informação é então utilizada para refinar iterativamente a estimativa de pose inicial e assim obter uma imagem renderizada que corresponda melhor à imagem observada. Diferentemente da maioria dos métodos, que utilizam parâmetros de pose em relação às coordenadas centradas na câmera, DeepIM propõe uma representação diferente da pose: a translação é tomada em relação às coordenadas centradas na câmera, mas a rotação é tomada em relação às coordenadas centradas no objeto. A combinação desta representação e a previsão de uma diferença de pose (em vez da pose do objeto diretamente) é o que os autores afirmam ter permitido sua abordagem de generalizar para categorias de objetos não vistas.

Figura 4 – Ilustração do método iterativo de estimação de pose de DeepIM



Fonte: Li, Wang, Ji, Xiang e Fox (2020).

Uma abordagem similar é utilizada por Miao, Wang e Liao (2016), que trataram o problema como um registro 2D/3D, o que equivale a estimar os parâmetros de transformação que proporcionariam o melhor registro entre uma Radiografia Reconstruída Digitalmente (DRR) e uma imagem de raio-X. Os regressores CNN são, assim, empregados para estimar diretamente os parâmetros de transformação, explorando as informações embutidas nas aparências das DRRs e das imagens de raios-X.

O refinamento de uma estimativa inicial de pose é uma estratégia comum adotada por diferentes métodos para aumentar a precisão, o que é feito ou usando características (*features*) de imagem determinadas à mão (isto é, escolhidas diretamente por um ser humano) (TJADEN; SCHWANECKE; SCHÖMER, 2017), ou através de funções de perda a serem otimizadas (RAD; LEPETIT, 2017). Uma abordagem poderosa comumente usada para o refinamento da pose é o algoritmo de Iterative Closest Point (ICP) (BESL; MCKAY, 1992) (KEHL; MANHARDT; TOMBARI; ILIC; NAVAB, 2017) (SUNDERMEYER; MARTON; DURNER; BRUCKER; TRIEBEL, 2018) (XIANG; SCHMIDT; NARAYANAN; FOX, 2018). Já o método PoseCNN (XIANG; SCHMIDT; NARAYANAN; FOX, 2018) utiliza uma versão altamente personalizada do ICP. Entretanto, as implementações de ICP de melhor desempenho muitas vezes não são suficientemente eficientes para aplicações em tempo real.

Algumas outras técnicas de refinamento de pose são baseadas na otimização iterativa de uma métrica em função dos parâmetros de transformação, isto é, um valor escalar que representa a qualidade do registro. Essa ideia se aproxima mais do objetivo deste projeto, uma vez que o interesse é obter uma medida ideal de similaridade. Essa é a base da abordagem de refinamento de pose em DeepIM (LI; WANG; JI; XIANG; FOX, 2020). O refinamento iterativo com redes neurais também foi adotado por Rad e Lepetit (2017) e em DenseFusion (WANG; XU; ZHU; MARTÍN-MARTÍN; LU; FEI-FEI; SAVARESE, 2019), mas suas redes não foram projetadas para diminuir diretamente a diferença de pose.

Em DenseFusion, uma rede principal funciona como um estimador de pose inicial, enquanto outra rede, como um estimador de pose residual (isto é, diferença de pose). Usando um sensor RGB-D, eles dispõem de imagens RGB (*Red, Green, Blue*) e de imagens de profundidade (*Depth*). Em um primeiro momento, f*eatures* são extraídas da imagem RGB para formar uma representação (*embedding*) de cor. A imagem de profundidade é usada para gerar um *point*

cloud (usando os parâmetros intrínsecos conhecidos da câmera), a partir do qual é, então, criado um *embedding* de *features* geométricas. Ambos *embeddings* são densos, isto é, com uma codificação para cada pixel, e são fundidos para formar uma representação global para cada um deles. Tal representação é usada para estimar uma pose para cada pixel. Dentre todas as poses, a pose inicial do objeto é tomada como sendo a que maximiza uma determinada medida de confiança. O *point cloud* é, então, atualizado com esta pose inicial estimada. Um novo *embedding* de geometria é, então, computado e passado através da rede de estimativa de pose residual juntamente com o *embedding* de cor (usada como referência para calcular o residual). A pose residual estimada é então usada em cada iteração para atualizar o *point cloud* e calcular uma nova pose residual.

Em HybridPose (SONG; SONG; HUANG, 2020), o módulo de refinamento de pose emprega o método Gauss-Newton para otimização numérica para otimizar uma função com base na diferença entre a pose prevista e a pose *ground-truth* (isto é, o valor real da saída, que se espera que o modelo preveja/estime).

Mais perto do domínio deste projeto, alguns trabalhos também exploraram aplicações para imagens de raios-X (MIAO; WANG; LIAO, 2016) (KÜGLER; SEHRING; STEFANOV; STENIN; KRISTIN; KLENZNER; SCHIPPER; MUKHOPADHYAY, 2020). Em i3PosNet (KÜGLER; SEHRING; STEFANOV; STENIN; KRISTIN; KLENZNER; SCHIPPER; MUKHOPADHYAY, 2020), o problema de estimativa de pose das imagens de raios-X é abordado indiretamente, com uma CNN que estima posições de múltiplos *keypoints* pertencentes ao objeto de interesse na imagem de entrada. Os parâmetros de pose do objeto são então derivados desses *keypoints* através de considerações geométricas. A i3PosNet usa estratégia baseada no processamento de *patches* (isto é, subdivisões da imagem) e também usa uma estratégia iterativa de refinamento de pose.

Em (MIAO; WANG; LIAO, 2016), os autores lidaram com o problema de registro 2D/3D, o que equivale a estimar os parâmetros da transformação que melhor registrariam uma Radiografia Reconstruída Digitalmente (DRR) para uma imagem de raios-X. Similarmente à abordagem utilizada em DeepIM (LI; WANG; JI; XIANG; FOX, 2020), foram empregados regressores CNN para estimar diretamente os parâmetros de transformação, explorando as informações embutidas nas aparências das DRRs e das imagens de raios-X.

1.3 Estrutura do texto

Este documento está dividido da seguinte forma: o Capítulo 2 discute os métodos utilizados para o desenvolvimento do projeto, detalha a formulação do problema teórico, introduz a notação utilizada e apresenta os materiais e recursos disponibilizados; o Capítulo 3 apresenta e discute os resultados obtidos; e o Capítulo 4 conclui o documento com uma discussão sobre os resultados e as possíveis direções para trabalhos futuros.

2 METODOLOGIA

Tendo como motivação o contexto da fluoroscopia e o interesse em conhecer a pose 3D do MitraClip a partir de uma imagem de raios-X, o objetivo deste projeto foi o desenvolvimento de um sistema capaz de estimar a diferença de pose (ou seja, a transformação da pose) entre duas imagens dadas, de forma semelhante ao que foi feito em (MIAO; WANG; LIAO, 2016) E (LI; WANG; JI; XIANG; FOX, 2020). Idealmente, ao estimar a diferença de pose, este sistema codificaria uma medida de similaridade entre as duas imagens de entrada: uma imagem observada, e uma imagem gerada artificialmente, para a qual são conhecidos os parâmetros de pose (ou seja, uma projeção do objeto de interesse a partir de uma determinada pose conhecida).

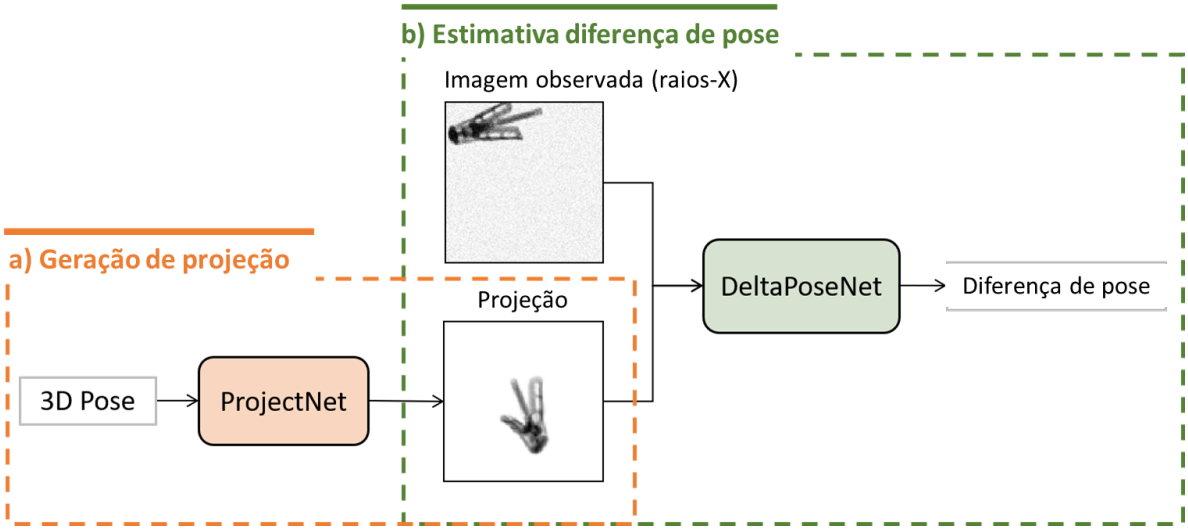
Esta medida de similaridade é útil para uma variedade de aplicações, desde avaliar a qualidade de um registro e fornecer uma base para melhorá-lo, até a extensão para a estimativa de pose 3D. A aplicação para estimativa de pose 3D é baseada na ideia de que, dada uma imagem observada A e uma projeção B (obtida a partir de uma determinada pose), é possível obter a diferença de pose entre as imagens e assim estimar a pose na imagem observada A.

Com esta abordagem em mente, surgem duas questões: 1) como obter projeções a partir de uma determinada pose e de um modelo 3D do objeto; 2) como estimar a diferença de pose entre duas imagens dadas. Estas foram, portanto, as principais questões de interesse durante o desenvolvimento deste projeto e as abordagens utilizadas serão discutidas nas subseções seguintes.

Na Seção 2.1, é apresentada a formulação do problema a ser resolvido. Na Seção 2.2, são apresentados os materiais e recursos disponíveis para o projeto, incluindo detalhes sobre os dados utilizados. Nas Seções 2.3 e 2.4, as redes DeltaPoseNet e a ProjectNet são apresentadas, respectivamente. Ambas são modelos de CNNs profundas (redes neurais convolucionais profundas), a primeira com o objetivo de prever a diferença de pose entre duas imagens do objeto; a segunda, desenvolvida para gerar projeções a partir de uma determinada pose de entrada.

Como ilustrado na Figura 5, os dois blocos podem ser colocados juntos a fim de prever a diferença de pose (e, por conseguinte, a semelhança) entre uma imagem observada (raios-X) e uma projeção gerada. Na Seção 2.5, esta combinação de ProjectNet e DeltaPoseNet é discutida.

Figura 5 – Ilustração do método proposto: a) geração de uma projeção do objeto a partir de uma pose de entrada; b) estimativa da transformação de pose (diferença de pose) entre a projeção gerada e uma imagem observada



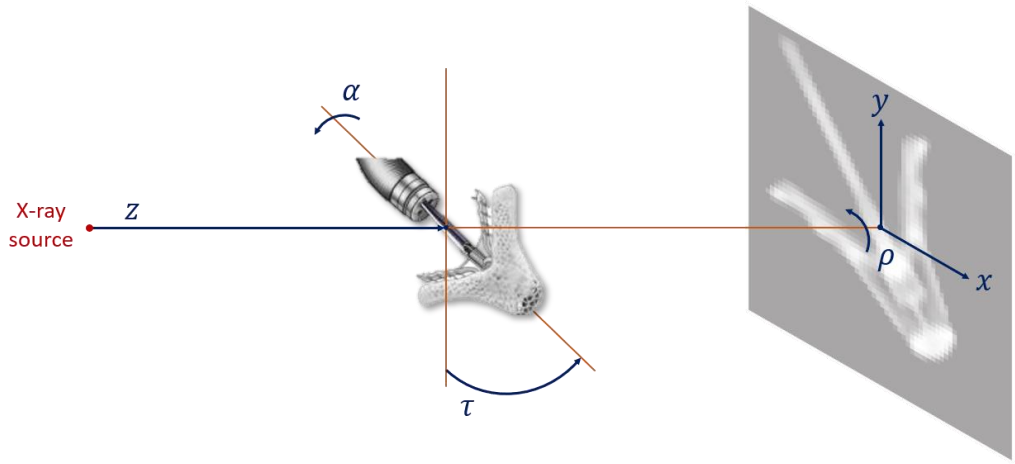
Fonte: Produção da própria autora.

2.1 Formulação do problema

No contexto deste projeto, o interesse é obter a estimativa da diferença de pose 3D entre duas imagens, ou seja, os parâmetros de transformação 3D que permitem passar da pose do objeto em uma imagem para a pose do objeto na outra imagem. Para parametrizar a pose 3D, ou uma transformação 3D, utilizamos seis componentes: x, y, z representando a translação, e α, τ, ρ representando a rotação. Os parâmetros de rotação serão referenciados neste trabalho como axis (α), tilt (τ) e plane (ρ). Diferentemente da abordagem geral utilizada na literatura para a estimativa da pose da câmera, utilizamos uma parametrização desacoplada entre translação e rotação. Definimos os parâmetros de translação com relação à fonte de raios-X, enquanto o centro de rotação é o centro do objeto. Esta representação é semelhante à usada em i3PosNet (KÜGLER; SEHRING; STEFANOV; STENIN; KRISTIN; KLENZNER; SCHIPPER; MUKHOPADHYAY, 2020) e garante que uma mudança de rotação não provoque uma mudança de translação na projeção.

Como ilustrado na Figura 6, x e y representam a translação 2D no plano da imagem, enquanto z representa a distância do objeto em relação à fonte de raios-X. Para os parâmetros de rotação, o ângulo axis, α, representa a rotação do objeto em torno de seu eixo de rotação principal; o ângulo tilt, τ, quantifica a inclinação do instrumento em relação ao plano da imagem; e o ângulo plane, ρ, representa o ângulo entre o eixo de rotação principal do objeto projetado no plano da imagem e o eixo horizontal da imagem.

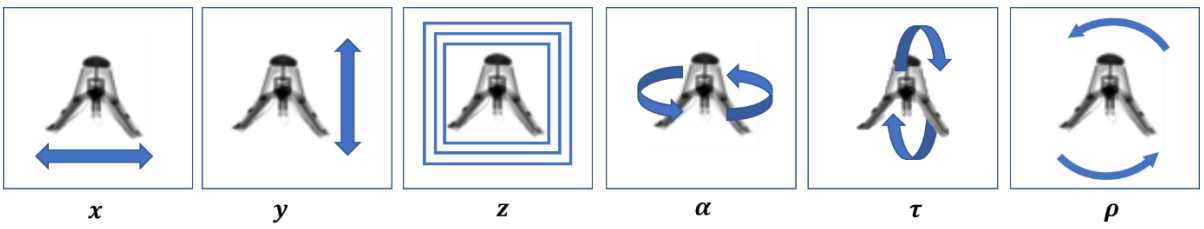
Figura 6 – Ilustração da parametrização usada para a pose 3D



Fonte: Produção da própria autora.

Os 6 parâmetros podem ser separados em parâmetros de transformação dentro e fora do plano (KAISER; JOHN; HEIMANN; BROST; NEUMU; ROSE, 2014). Os parâmetros de transformação dentro do plano são x, y e ρ, e seus efeitos são aproximadamente transformações rígidas 2D (MIAO; WANG; LIAO, 2016), como ilustrado na Figura 7. Os parâmetros para fora do plano são α e τ, que causam mudanças de forma, e z, que afeta a escala do objeto na imagem (Figura 7).

Figura 7 – Ilustração dos efeitos de mudanças aplicadas aos diferentes parâmetros de pose 3D



Fonte: Produção da própria autora.

Outro elemento de interesse ao parametrizar a projeção do MitraClip é o ângulo entre seus "braços" (θ_{arm}), que pode variar. Diferentemente dos seis parâmetros discutidos anteriormente, a alteração do ângulo entre os braços causa uma transformação não-rígida.

Em termos de notação, ao longo deste documento será usado δ para fazer uma referência geral a qualquer um dos parâmetros (por exemplo, ao falar de uma situação que se aplica a todos eles).

2.2 Materiais e recursos

Para a realização deste projeto, foram utilizados materiais e recursos fornecidos pelo laboratório Medisys da Philips Research Paris. O Medisys é um laboratório especializado em processamento de imagens médicas (redução de ruído, detecção de objetos, estimativa de pose 3D etc.), bem como em modelagem digital (modelagem de redes arteriais e de diversas patologias), com múltiplas aplicações. Reúne cerca de trinta engenheiros de pesquisa, trabalhando em temas desafiadores desta área de pesquisa (inteligência artificial, extração de contornos, registro de imagem etc.), em colaboração com grupos acadêmicos (INRIA, CEREMADE, EPFL, Institut Mines-ParisTech etc.) e vários instituições clínicas na França e no exterior.

Para os experimentos, a Philips disponibilizou uma GPU de 12 GB (GeForce GTX 1080 Ti da NVIDIA). Os códigos foram escritos em Python (3.7.6) usando o *framework* TensorFlow (2.1.0).

Os dados iniciais disponíveis para este projeto foram compostos por sete modelos 3D do MitraClip: cada um deles com um ângulo diferente entre seus braços. Isso permitiu obter uma representação discreta de diferentes configurações possíveis do instrumento. A partir destes sete modelos, foram obtidas projeções 2D do MitraClip para diferentes poses, utilizando a técnica de *ray-tracing*, uma abordagem comum para gerar Radiografias Reconstruídas Digitalmente (DRRs) (SHEROUSE; NOVINS; CHANEY, 1990). Detalhes sobre esta técnica estão fora do escopo deste projeto, uma vez que as DRRs haviam sido previamente geradas pelos orientadores deste trabalho, na Philips, para formar o conjunto de dados base a ser utilizado. O conjunto de dados base foi assim composto por estas projeções, juntamente com seus respectivos parâmetros de pose (seguindo a formulação explicada na Seção 2.1). Exemplos de

projeções para os sete modelos diferentes e o ângulo correspondente entre braços podem ser vistos na Figura 8.

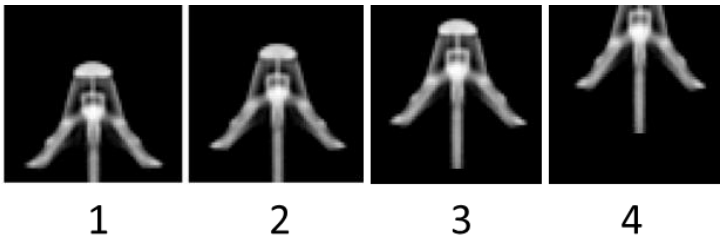
As projeções tinham um tamanho de 62×62 pixels, com apenas um canal (em escala de cinza). No contexto das imagens de raios-X, as informações estão principalmente relacionadas à intensidade dos pixels, e à forma e à escala do objeto.

Figura 8 – Exemplos de projeções a partir dos sete modelos 3D existentes para o MitraClip (correspondendo a sete ângulos diferentes entre os braços do MitraClip)

Fonte: Produção da própria autora.

O ideal seria que as DRRs fossem geradas a partir de parâmetros seguindo uma distribuição uniforme. Entretanto, elas foram geradas de forma a garantir que a ponta do MitraClip estivesse presente na projeção, de forma a garantir que uma parte considerável do objeto estaria presente na imagem. Por isso, foi feita uma amostragem aleatória dos parâmetros de pose, mas só foram consideradas para o conjunto de dados as projeções que continham a ponta do clipe. Isto é, se uma projeção gerada a partir de uma pose $[x, y, z, \alpha, \tau, \rho]$ não contivesse a ponta do MitraClip, tal projeção era descartada. Como exemplo, as projeções 1-3 da Figura 9 seriam mantidas, enquanto a projeção 4 seria descartada. Tal seleção das projeções afetou a distribuição de cada parâmetro no conjunto de dados, o que pode ser verificado na Figura 10 (distribuições não uniformes).

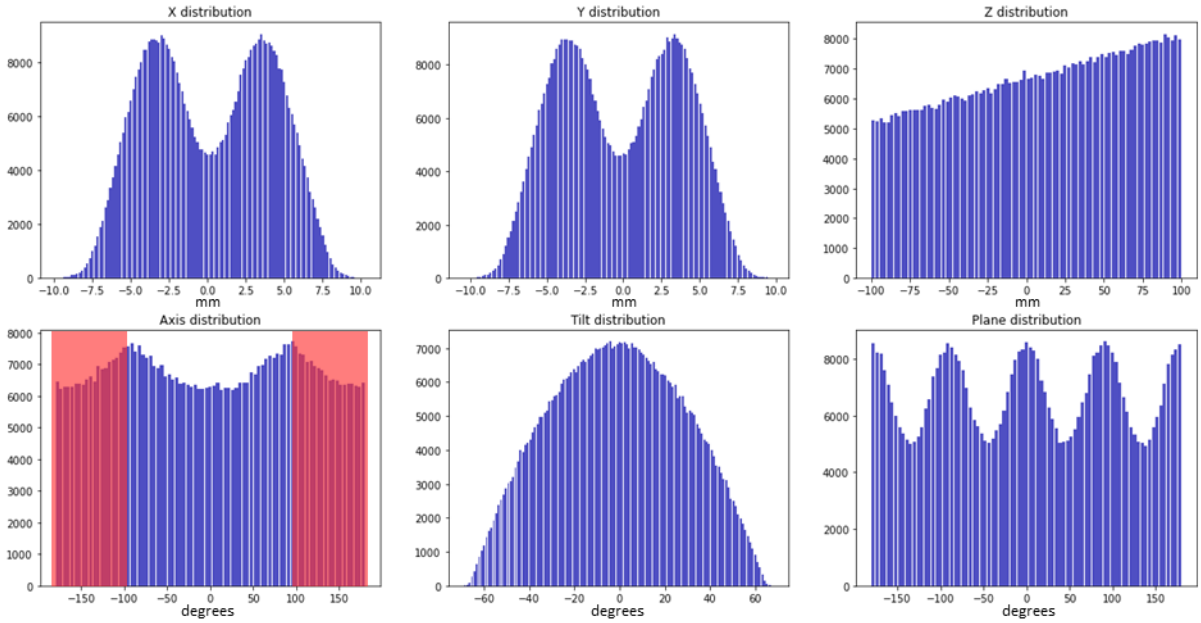
Figura 9 – Exemplos de projeções mantidas para formar o conjunto de dados (1-3) e de projeção não mantida (4)



Fonte: Produção da própria autora.

É importante destacar que o MitraClip tem uma simetria espelhada em torno de seu eixo de rotação principal, o que faz com que projeções geradas a partir dos mesmos parâmetros x, y, z, τ, ρ mas com uma diferença entre elas de 180° em α ($\Delta\alpha = \pi$) pareçam iguais. Para contornar esta ambiguidade, forçamos o parâmetro α para o intervalo $\left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$: valores fora deste intervalo (regiões em vermelho na Figura 10) foram, assim, redefinidos para seu valor correspondente dentro do intervalo (por exemplo, $\frac{3\pi}{4}$ torna-se $\frac{3\pi}{4} - \pi = -\frac{\pi}{4}$).

Figura 10 - Distribuição dos valores dos parâmetros nos dados disponíveis. Parâmetros de translação dados em mm (x, y, z), parâmetros de rotação dados em graus (α, τ, ρ)



Fonte: Produção da própria autora.

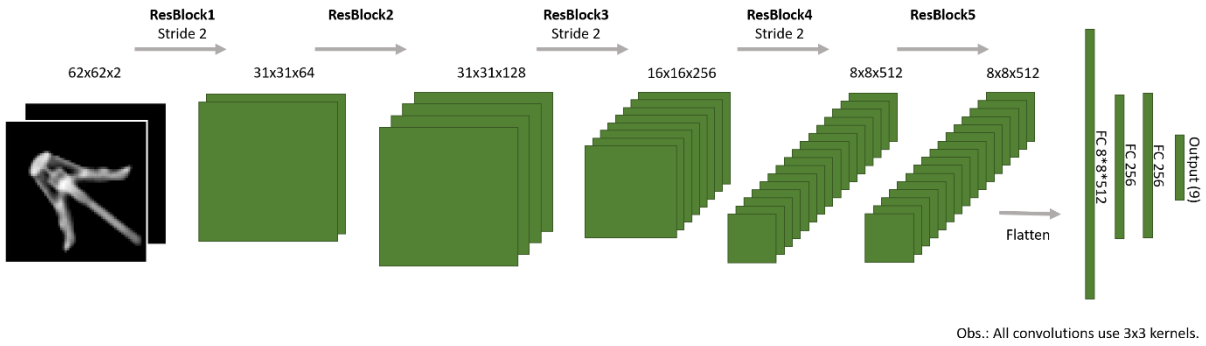
2.3 DeltaPoseNet

A principal questão de interesse deste projeto é a estimativa da diferença entre as poses 3D que originaram duas projeções de um mesmo objeto. Para este fim, foi projetada a DeltaPoseNet, uma CNN que recebe duas imagens como entrada e que produz um vetor de transformação que codifica a diferença de pose entre as duas imagens dadas. Um esquema desta abordagem é ilustrado na Figura 5b. As imagens de entrada seriam a imagem observada e uma projeção artificial do objeto, obtida a partir de uma determinada pose de entrada. Tendo tanto a pose 3D que gerou a projeção quanto o vetor de transformação 3D com relação à imagem observada, é possível deduzir a pose 3D que deu origem à imagem observada.

Em DeepIM (LI; WANG; JI; XIANG; FOX, 2020), os autores abordaram um problema semelhante usando um modelo de rede baseado na FlowNet (DOSOVITSKIY; FISCHER; ILG; HAUSSER; HAZIRBAS; GOLKOV; VAN DER SMAGT; CREMERS; BROX, 2015). Em alinhamento com o que foi sugerido em (WANG; CLARK; WEN; TRIGONI, 2017), os autores defendem os potenciais desta rede, originalmente projetada para a previsão de fluxo óptico, para a estimativa dos parâmetros de transformação entre duas imagens. Levando em consideração os bons resultados reivindicados pelos autores, foi decidido explorar uma arquitetura baseada na parte codificadora/contrativa da FlowNet. No caso deste trabalho, uma vez que se deseja estimar por regressão a diferença de pose, não há necessidade da parte decodificadora/expansiva (com convoluções transpostas) da FlowNet.

Um esquema da rede utilizada pode ser visto na Figura 11. Uma diferença principal com relação ao modelo da FlowNet é que as convoluções diretas foram substituídas por blocos residuais (HE; ZHANG; REN; SUN, 2016). Estes blocos são caracterizados pela soma da entrada do bloco (inalterada) com os *feature maps* de saída (resultantes das convoluções do bloco), o que conserva informação de características de nível inferior (*low level*) através da rede. A ideia por trás dos blocos residuais é que a função mais simples que uma camada deve codificar é a identidade (ou seja, a saída sendo igual à entrada). A DeltaPoseNet tem, assim, uma arquitetura semelhante a um codificador e consiste em uma sequência de blocos residuais, seguida por uma sequência de camadas totalmente conectadas. Na Seção 3.1, será mostrada uma comparação entre o uso de convoluções diretas e de blocos residuais.

Figura 11 – Esquemático da arquitetura da DeltaPoseNet



Fonte: Produção da própria autora.

A entrada da DeltaPoseNet é um par de imagens, assim composta por dois canais: um sendo a imagem observada, o outro, a projeção artificial. Sua saída $\boldsymbol{\Delta pose}$ é um vetor de tamanho 9 com a seguinte forma:

$$\boldsymbol{\Delta pose} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \text{sen}(\Delta\alpha) \\ \cos(\Delta\alpha) \\ \text{sen}(\Delta\tau) \\ \cos(\Delta\tau) \\ \text{sen}(\Delta\rho) \\ \cos(\Delta\rho) \end{bmatrix} \tag{1}$$

onde $\Delta\delta$ é a diferença estimada para um dos parâmetros da pose (lembrando que δ denota quaisquer dos seis parâmetros de interesse). Na Seção 2.3.1.1 será discutida a justificativa para o uso de seno e cosseno para o tratamento dos parâmetros angulares.

2.3.1 Treinamento da DeltaPoseNet

2.3.1.1 Função de perda

A função de perda utilizada para treinamento da rede foi o erro quadrático médio entre a *ground-truth* $\boldsymbol{\Delta pose_{GT}}$ e a previsão $\boldsymbol{\Delta pose_{pred}}$, como segue:

$$L_{\Delta\text{Pose}} = \frac{1}{N \cdot M} \sum_{k=1}^{N} \sum_{i=1}^{M} \left(\boldsymbol{\Delta pose_{k_{i_{GT}}}} - \boldsymbol{\Delta pose_{k_{i_{pred}}}} \right)^2 \tag{2}$$

onde $M = 9$ é o número de elementos no vetor $\boldsymbol{\Delta pose}$ e N é o número de amostras de pares de imagens. A *ground-truth* utilizada e a normalização aplicada aos parâmetros são discutidas na Seção 2.3.1.2.

Computar diretamente as distâncias angulares poderia causar valores de perda inconsistentes, devido ao *loop* em torno de 2π (por exemplo, se a rede estima que as projeções têm uma diferença angular de $2\pi\ rad$ e a *ground-truth* correspondente é $0\ rad$, a perda computada será alta, quando na verdade deveria ser baixa, dado que as duas situações são visualmente

equivalentes). Para evitar esta inconsistência, foram usados o seno e o cosseno das diferenças angulares, como mostrado na equação (1), para a $\boldsymbol{\Delta pose}$.

Inicialmente pretendia-se estimar a diferença de pose relativa (isto é, com sinal positivo ou negativo), mas o treinamento da rede não atingiu as expectativas. Portanto, optou-se por regredir a diferença absoluta como forma de simplificar o problema. Os resultados das experiências com a distância relativa estão detalhados no Apêndice A (página 64), enquanto o restante deste relatório abrange apenas os resultados obtidos com a formulação absoluta.

2.3.1.2 Conjunto de dados e treinamento

A rede foi treinada com imagens sintéticas: DRRs obtidas através da técnica de *ray-tracing* a partir dos sete modelos de objetos disponíveis (como explicado na Seção 2.2). O conjunto de dados utilizado para a DeltaPoseNet consistiu em pares de imagens gerados anteriormente ao treinamento, juntamente com a correspondente *ground-truth*: a diferença absoluta de pose entre as poses das imagens, isto é:

$$\boldsymbol{\Delta pose_{GT}} = \begin{bmatrix} |\Delta x| \\ |\Delta y| \\ |\Delta z| \\ \text{sen}(|\Delta\alpha|) \\ \cos(|\Delta\alpha|) \\ \text{sen}(|\Delta\tau|) \\ \cos(|\Delta\tau|) \\ \text{sen}(|\Delta\rho|) \\ \cos(|\Delta\rho|) \end{bmatrix} \tag{3}$$

Para cada modelo de objeto, foram gerados 100.000 pares de imagens, dos quais 80.000 foram separados para treinamento e 20.000 para teste. Ao total, para todos os modelos, foram então gerados 700.000 pares de imagens (560.000 de treinamento e 140.000 de teste). Dois conjuntos de dados foram considerados:

- Conjunto de dados 1: pares de imagens de todos os sete modelos de objetos (conforme Figura 8). Formado pelos 700.000 pares de imagens: 560.000 no conjunto de treinamento e 140.000 no conjunto de teste.
- Conjunto de dados 2: pares de imagens dos modelos de objetos A, C, E, G (conforme Figura 8), onde $\theta_{arm_A} < \theta_{arm_C} < \theta_{arm_E} < \theta_{arm_G}$, tendo o modelo A o menor valor

para o ângulo entre braços (braços fechados) e o modelo G o maior (braços com o máximo de abertura). Formado, então, pelos 400.000 pares de imagens correspondentes aos modelos A, C, E, G: 320.000 no conjunto treinamento e 80.000 no conjunto teste.

Como apresentado na Seção 2.2, as imagens têm dimensão 62×62 pixels. O conjunto de dados 1 contém o conjunto 2 mais pares de imagens para os outros modelos de objetos (B, D, F). Para cada par de imagens, ambas as imagens pertenciam ao mesmo modelo, ou seja, não havia diferença de ângulo entre os braços. O interesse em se considerar o conjunto de dados 2 é poder verificar o desempenho da rede sobre modelos de objeto não vistos durante o treino (no caso, modelos B, D e F), verificando se é possível aprender a interpolar resultados para tais modelos de maneira satisfatória ou próxima aos resultados obtidos com o treinamento usando o conjunto de dados 1.

Os valores correspondentes à translação, Δx, Δy e Δz, foram normalizados na *ground-truth*, conforme as seguintes relações:

$$\begin{aligned} \Delta x_{norm} &= \frac{\Delta x}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \\ \Delta y_{norm} &= \frac{\Delta y}{\max(\boldsymbol{y}) - \min(\boldsymbol{y})} \\ \Delta z_{norm} &= \frac{\Delta z}{\max(\boldsymbol{z}) - \min(\boldsymbol{z})} \end{aligned} \tag{4}$$

onde $\max(\boldsymbol{\delta})$ e $\min(\boldsymbol{\delta})$ correspondem respectivamente aos valores máximo e mínimo de δ entre todos os dados de treinamento.

Foi considerada a diferença absoluta de pose, portanto os valores normalizados estavam na faixa $[0,1]$. Não houve necessidade de normalizar os valores dos ângulos, uma vez que $\boldsymbol{\Delta pose}$

contém os valores dos senos e cossenos dos ângulos, conforme equação (1), que são valores já normalizados.

Uma situação especial teve que ser considerada ao calcular a diferença de pose: a simetria espelhada em relação ao ângulo α. Como discutido na Seção 2.2, o parâmetro α foi forçado ao intervalo $\left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$ para eliminar a ambiguidade devida à simetria do objeto.

Neste caso, a diferença de ângulo $\Delta\alpha$ é cíclica em $\frac{\pi}{2}$. Isto é, fixando um objeto em $\alpha_1 = -\frac{\pi}{2}$ e variando α_2 de um outro objeto entre $\left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$, a diferença $|\Delta\alpha|$ observada visualmente entre as projeções aumenta à medida que α_2 se aproxima de 0, e, em seguida, começa a diminuir à medida que α_2 se aproxima de $\frac{\pi}{2}$.

Ou seja, os objetos em $\alpha_1 = -\frac{\pi}{2}$ e em $\alpha_2 = +\frac{\pi}{2}$ estão, na verdade, em uma posição equivalente (a uma distância zero), mas o simples cálculo $|\alpha_1 - \alpha_2|$ resultaria em uma distância de π. Para tratar essa incoerência, adotamos a seguinte redefinição de $|\Delta\alpha|$ na *ground-truth* (equação (3)):

$$|\Delta\alpha|_{new} = |\Delta\alpha|_{original}\,, \qquad se\ |\Delta\alpha|_{original} \leq \frac{\pi}{2}$$
$$|\Delta\alpha|_{new} = \pi - |\Delta\alpha|_{original}\,, \quad se\ |\Delta\alpha|_{original} > \frac{\pi}{2}$$

de modo a garantir que $|\Delta\alpha|_{\alpha_1=-\frac{\pi}{2},\ \alpha_2=-\pi/2} = |\Delta\alpha|_{\alpha_1=-\frac{\pi}{2},\ \alpha_2=\pi/2} = 0$. Se esta situação não fosse considerada, a rede estaria lidando com valores de perda inconsistentes durante o treinamento, o que foi, de fato, observado durante a execução de alguns experimentos.

Por motivos que serão discutidos na Seção 3.1, optou-se por limitar a diferença em α e τ entre as imagens nos pares passados para a DeltaPoseNet: máximo de 0,8 rad (~46°). Isto é, os pares de imagens dos conjuntos de dados foram gerados garantindo $|\Delta\alpha| \leq 0{,}8$ rad e $|\Delta\tau| \leq 0{,}8$ rad.

Duas versões diferentes da rede foram treinadas, cada uma usando um dos dois conjuntos de dados descritos nesta seção. A DeltaPoseNet treinada com o conjunto de dados 1 foi nomeada DeltaPoseNet$_{V1}$ e a treinada com o conjunto de dados 2, DeltaPoseNet$_{V2}$. Em ambos os

treinamentos, foi usado um tamanho de *batch* de 500 pares de imagens. Assim, para o treinamento com o conjunto de dados 1 (contendo 560.000 pares de imagens de treino), cada época tinha 1.120 passos e levava cerca de 21 min para rodar. Para o caso com o conjunto de dados 2 (com 320.000 pares de imagens de treino), cada época tinha 640 passos, o que levava cerca de 12 min cada.

O aumento de dados (do inglês, *data augmentation*) foi realizado durante o treinamento: correção gama para mudar a luminância das imagens; adição de ruído gaussiano; ajuste de log contraste; *flip* horizontal; *flip* vertical; rotação de 90° (com as respectivas alterações da *ground-truth* para levar em conta o impacto das três últimas transformações sobre a diferença de pose).

Os pesos da DeltaPoseNet foram inicializados usando o inicializador padrão do TensorFlow: inicializador uniforme Xavier.

2.4 ProjectNet

A DeltaPoseNet foi desenvolvida com a finalidade de estimar a diferença de pose entre duas imagens e assim fornecer uma medida de similaridade entre elas. O intuito é minimizar esta diferença de pose (ou seja, maximizar a similaridade), o que significa encontrar a pose 3D que gera a projeção mais próxima da imagem observada. Neste contexto, foi explorada a geração de projeções artificiais usando uma rede neural que foi nomeada ProjectNet. O intuito era poder comparar a qualidade das projeções e o tempo de execução com relação à técnica de *ray-tracing.*

A entrada da ProjectNet é composta pelos parâmetros de pose 3D e sua arquitetura tipo decodificador consiste em uma sequência de camadas totalmente conectadas seguidas por uma sequência de convoluções transpostas. Este tipo de arquitetura tem sido amplamente utilizado na literatura, especialmente para tarefas de segmentação, onde camadas convolucionais são usadas para extrair características na parte codificadora e, em seguida, camadas de convoluções transpostas são usadas para restaurar o tamanho original da imagem na parte decodificadora, possibilitando assim obter um valor de predição para cada pixel da imagem. No caso da ProjectNet, apenas a parte decodificadora interessa, já que a entrada é uma determinada pose

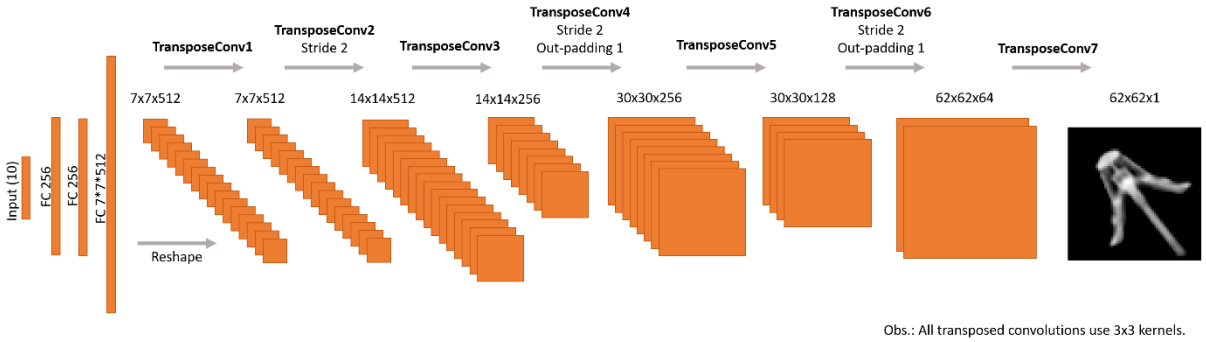
para o objeto, que pode ser visto como um código a partir do qual se quer extrair a imagem correspondente. A entrada da ProjectNet é então um vetor da forma:

$$\boldsymbol{pose_{in}} = (\theta_{arm}, x, y, z, \sin(\alpha), \cos(\alpha), \sin(\tau), \cos(\tau), \sin(\rho), \cos(\rho))^T \quad (5)$$

que codifica a pose do objeto do qual se quer obter uma projeção. Como esquematizado na Figura 12, este vetor de entrada é passado por 3 camadas totalmente conectadas (do inglês, *Fully Connected* – FC) e a saída destas camadas FC é redimensionada para formar mapas de características (*feature maps*) 2D que são então passados como entrada para a parte convolucional da rede. Todas as convoluções transpostas têm *kernels* 3x3.

As convoluções transpostas têm sido amplamente utilizadas como uma técnica de *up-sampling* porque seus parâmetros podem ser aprendidos, diferentemente da maioria das estratégias clássicas de *up-sampling*, tais como vizinho mais próximo (*nearest-neighbors*), interpolação bi-linear, *max-unpooling*, etc (Zeiler; Taylor; Fergus, 2011).

Figura 12 – Esquemático da arquitetura da ProjectNet



Fonte: Produção da própria autora.

2.4.1 Treinamento da ProjectNet

2.4.1.1 Função de perda

A ProjectNet produz um único canal como saída: a projeção do objeto correspondente à pose dada como entrada. Como função de perda a ser minimizada, utilizou-se o erro quadrático médio entre a projeção de saída e a *ground-truth*, seguindo a equação (6):

$$L_{Proj} = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{1}{H \cdot W}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}(x_{k_{GT_{i,j}}} - x_{k_{Pred_{i,j}}})^2\right) \quad (6)$$

onde N é a quantidade de amostras de dados; H e W são a altura e largura das imagens, respectivamente; $x_{k_{GT_{i,j}}}$ é o pixel (i, j) da imagem *ground-truth* k e $x_{k_{Pred_{i,j}}}$ é o pixel (i, j) da imagem estimada k.

2.4.1.2 Conjunto de dados e treinamento

O conjunto de dados é composto por DRRs (*ground-truths*) e os parâmetros de pose a partir dos quais eles foram gerados (entradas). Os valores dos pixels nas imagens da *ground-truth* foram normalizados o intervalo [0,1].

Todos os modelos disponíveis para o MitraClip foram utilizados nos experimentos com a ProjectNet. Todas as DRRs fornecidas foram divididas em conjuntos de treinamento e de teste. O conjunto de treinamento foi composto por 560.000 imagens únicas, enquanto 140.000 imagens fizeram o conjunto de teste (representando 80% e 20% de todos os dados, respectivamente).

Cada *batch* foi composto por 500 imagens, portanto cada época teve um total de 1.120 passos e demorou cerca de 23 min utilizando a GPU disponível. Para o treinamento da ProjectNet, seus pesos foram inicializados usando o inicializador padrão do TensorFlow: inicializador uniforme Xavier.

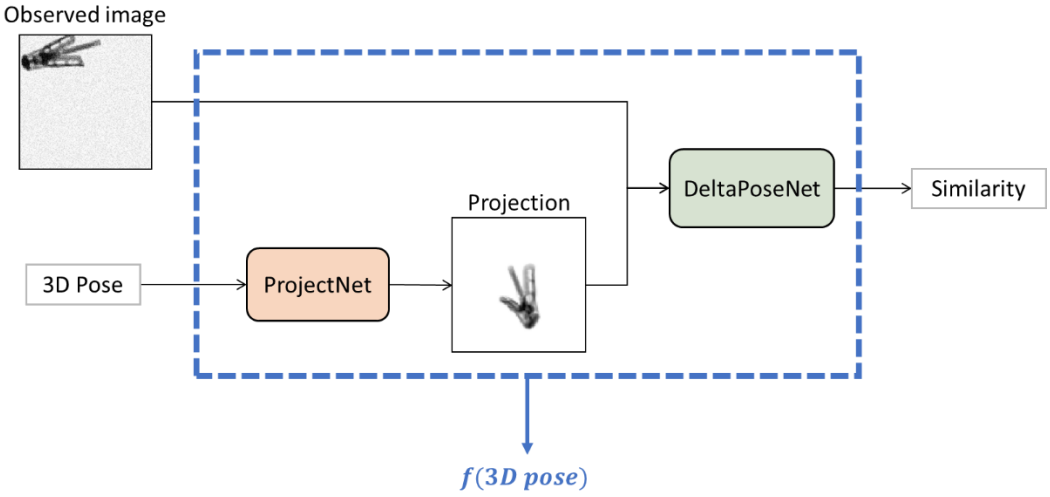
2.5 Project-DeltaPose-Net: conectando DeltaPoseNet e ProjectNet

Tendo DeltaPoseNet e ProjectNet, pode-se explorar a combinação de ambas, seguindo o esquema da Figura 5, onde a imagem observada seria uma DRR e a projeção seria uma saída dada pela ProjectNet. O objetivo é ver até que ponto a combinação das duas redes produz resultados satisfatórios. A combinação das duas redes pode ser vista como uma função dos parâmetros de pose de entrada:

$$f(x, y, z, \alpha, \tau, \rho) = \boldsymbol{\Delta pose} \quad (7)$$

O sistema que une as duas redes propostas será referenciado como Project-DeltaPose-Net e está ilustrado na Figura 13.

Figura 13 – Ilustração do sistema que liga ProjectNet e DeltaPoseNet como uma função a ser otimizada



Fonte: Produção da própria autora.

O interesse de combinar as duas redes é explorar o caráter diferenciável das projeções da ProjectNet para a otimização da diferença de pose estimada. Ou seja, otimizar a função codificada pelo sistema Project-DeltaPose-Net:

$$\boldsymbol{\Delta pose_{Optim}} = \operatorname{argmin} f(x, y, z, \alpha, \tau, \rho) \tag{8}$$

2.5.1 Treinamento da Project-DeltaPose-Net

Os resultados obtidos com a combinação das duas redes treinadas como descrito nas Seções 2.3.1 (isto é, DeltaPoseNet previamente treinada com um par de DRRs) e 2.4.1 não foram satisfatórios para o caso em que uma das DRRs foi substituída por uma projeção da ProjectNet. Por mais que as projeções da ProjectNet se assemelhem às DRRs, certamente apresentam diferentes artefatos que precisam ser aprendidos pela DeltaPoseNet. Assim, optou-se por retreinar a DeltaPoseNet.

Na Seção 2.3.1, a entrada de dois canais da DeltaPoseNet é composta por duas DRRs. Nesta seção, foram considerados pares de imagens compostos por 1) uma DRR do conjunto de dados original e 2) uma projeção da ProjectNet. Duas novas versões da DeltaPoseNet foram treinadas,

com dois conjuntos de dados diferentes (similarmente ao que foi feito para a DeltaPoseNet original, como descrito na Seção 2.3.1.2):

- Conjunto de dados 3: pares $(DRR, projeção\ da\ ProjectNet)$ para todos os sete modelos de objetos (conforme Figura 8). Formado por 200.000 pares de imagens: 160.000 no conjunto de treinamento e 40.000 no conjunto de teste.
- Conjunto de dados 4: pares $(DRR, projeção\ da\ ProjectNet)$ para os modelos de objetos A, C, E, G (conforme Figura 8), onde $\theta_{arm_A} < \theta_{arm_C} < \theta_{arm_E} < \theta_{arm_G}$, tendo o modelo A o menor valor para o ângulo entre braços (braços fechados) e o modelo G o maior (braços com o máximo de abertura). Formado por 200.000 pares de imagens correspondentes aos modelos A, C, E, G: 160.000 no conjunto de treinamento e 40.000 no conjunto de teste.

A DeltaPoseNet treinada com o conjunto de dados 3 foi chamada de DeltaPoseNet$_{V3}$ e a treinada com o conjunto de dados 4, de DeltaPoseNet$_{V4}$.

Os dados foram gerados conforme as mesmas especificações descritas na Seção 2.3.1.2. Para garantir as restrições $\Delta\alpha \leq 0{,}8$ rad e $\Delta\tau \leq 0{,}8$ rad, para cada par de imagem uma DRR era tomada como referência e os parâmetros de pose passados para ProjectNet eram amostrados uniformemente dentro da distribuição dos dados (Figura 10), mas impondo as restrições sobre α e τ de forma a garantir uma distância máxima de 0,8 rad com relação à DRR de referência.

Para ambos os experimentos, foi utilizado um tamanho de *batch* de 500 pares de imagens, cada época tendo assim 400 passos e demorando cerca de 8min para rodar.

Data augmentation foi aplicada como na Seção 2.3.1.2. Os pesos da DeltaPoseNet foram novamente inicializados usando o inicializador padrão do TensorFlow: inicializador uniforme Xavier.

2.5.2 Refinamento de pose

Como discutido na seção Trabalhos relacionados, o refinamento da pose é uma estratégia muito comum adotada por diferentes métodos que lidam com o problema da estimativa da pose, a fim de melhorar uma estimativa inicial. No caso deste trabalho, uma abordagem seria considerar

todo o esquema do ProjectNet e DeltaPoseNet como uma função a ser minimizada (equação (8)) (conforme esquema ilustrado na Figura 13), ou seja, encontrar os parâmetros de pose de entrada na ProjectNet que minimizem a diferença de pose prevista pela DeltaPoseNet (correspondendo assim à projeção mais próxima possível da imagem observada).

Decidiu-se analisar como a projeção evoluiria se a pose de entrada da ProjectNet fosse iterativamente atualizada utilizando a diferença de pose estimada pela DeltaPoseNet. Assim, utilizou-se uma imagem DRR do conjunto de dados original como imagem de referência/observada, e gerou-se uma primeira projeção usando a ProjectNet, a partir de uma pose de entrada centrada. Então, as duas imagens foram passadas através da DeltaPoseNet$_{V3}$ (treinada com o conjunto de dados 3, composto de pares ($DRR, projeção\ da\ ProjectNet$) para todos os sete modelos do MitraClip – conforme Seção 2.5.1) para obter a estimativa da diferença de pose $\boldsymbol{\Delta pose_{pred}}$. Esta estimativa foi usada para atualizar a pose inicial de entrada. Esta atualização foi feita seguindo uma abordagem do tipo *grid-search*, para a qual considerou-se três opções para cada parâmetro:

1. Manter o valor atual ($\delta_{new} = \delta_{old}$);
2. Incrementar o valor atual com a diferença estimada ($\delta_{new} = \delta_{old} + \Delta\delta_{pred}$);
3. Decrementar o valor atual com a diferença estimada ($\delta_{new} = \delta_{old} - \Delta\delta_{pred}$).

Foram analisadas todas as combinações possíveis de parâmetros considerando estas três opções, gerando uma projeção para cada combinação e obtendo a diferença de pose correspondente com a DeltaPoseNet$_{V3}$. A partir da diferença de pose, avaliou-se a seguinte medida de similaridade:

$$S_{pose} = \Delta x_{pred} + \Delta \mathrm{y}_{\mathrm{pred}} + \Delta \mathrm{z}_{\mathrm{pred}} + \Delta \alpha_{\mathrm{pred}} + \Delta \tau_{pred} + \Delta \rho_{pred} \quad (9)$$

Tendo três possibilidades para cada parâmetro e um total de seis parâmetros (o ângulo entre braços sendo forçado a ser o mesmo que na imagem de referência), foram avaliadas assim $3^6 = 729$ combinações de parâmetros em cada iteração. A combinação resultando na menor S_{pose} era selecionada e passava-se para a próxima iteração.

3 RESULTADOS

3.1 DeltaPoseNet$_{V1}$– Treinamento usando todos os modelos disponíveis do MitraClip

Nesta seção são analisados os resultados obtidos com a DeltaPoseNet treinada utilizando todos os modelos do MitraClip disponíveis (conjunto de dados 1, como detalhado na Seção 2.3.1.2), a qual foi chamada de DeltaPoseNet$_{V1}$. Como pode ser visto na Figura 14, a rede apresentou um desempenho fraco em relação à estimativa dos parâmetros $\Delta\alpha$ e $\Delta\tau$, o que impactou a função de perda global (sobre todos os parâmetros) sobre o conjunto de teste. Estes resultados são um reflexo da dificuldade real que a tarefa representa: alterações em α e τ causam modificações de forma do objeto na projeção, o que é um padrão difícil de aprender para a rede, especialmente com a simetria espelhada do MitraClip. Para superar este problema, decidiu-se limitar a diferença máxima em α e τ entre as imagens nos pares passados para a DeltaPoseNet: 0,8 rad (~46°) (isto é, os 700.000 pares de imagens foram gerados novamente, garantindo que $\Delta\alpha \leq 0{,}8$ rad e $\Delta\tau \leq 0{,}8$ rad).

Os resultados para este novo conjunto de dados são mostrados na Figura 15, onde é possível ver que a curva de perda de teste converge bem. O valor médio de $L_{\Delta\mathrm{Pose}}$ (equação (2)) no conjunto de teste foi de 0,03 (±0,05). Os erros médios foram:

- Em Δx, 0,21 mm (±0,17 mm);
- Em Δy, 0,23 mm (±0,18 mm);
- Em Δz, 9,12 mm (±7,53 mm);
- Em Δα, 4,43° (±4,81°);
- Em Δτ, 4,59° (±4,89°);
- Em Δρ, 1,47° (±1,24°).

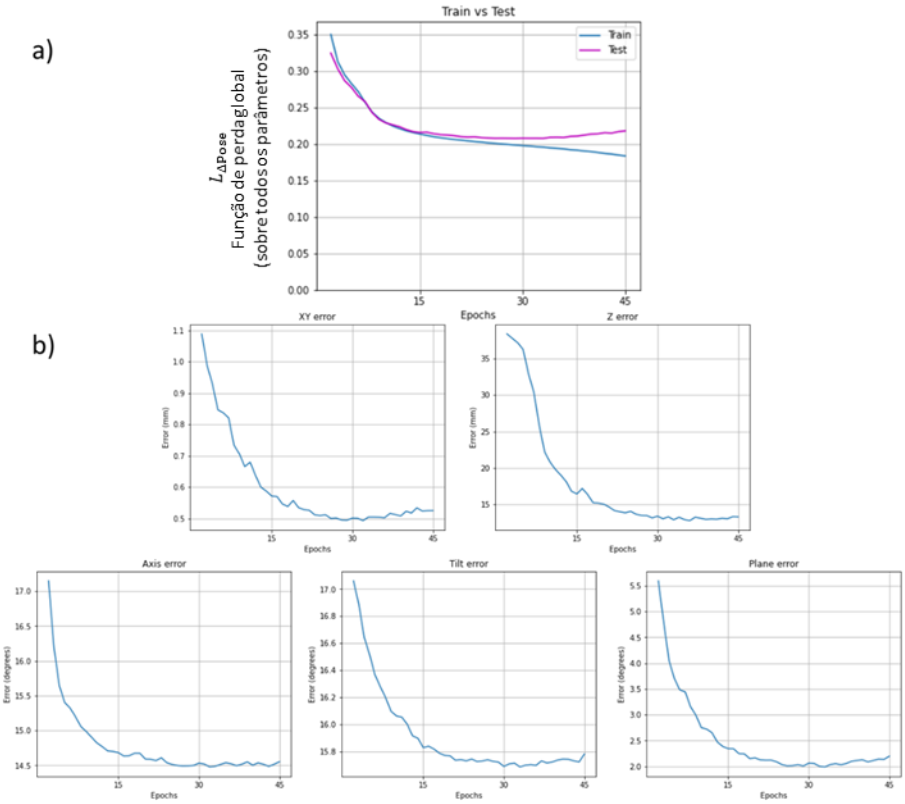
Os resultados obtidos por modelo do MitraClip são mostrados na Tabela 1 e serão discutidos na Seção 3.3.

Uma comparação por parâmetro entre os casos com e sem restrições de distância máxima em α e τ é mostrada na Figura 16, onde é possível observar uma melhora significativa nas previsões

de α e τ. As experiências no restante deste documento foram todas feitas considerando estas restrições de diferença máxima.

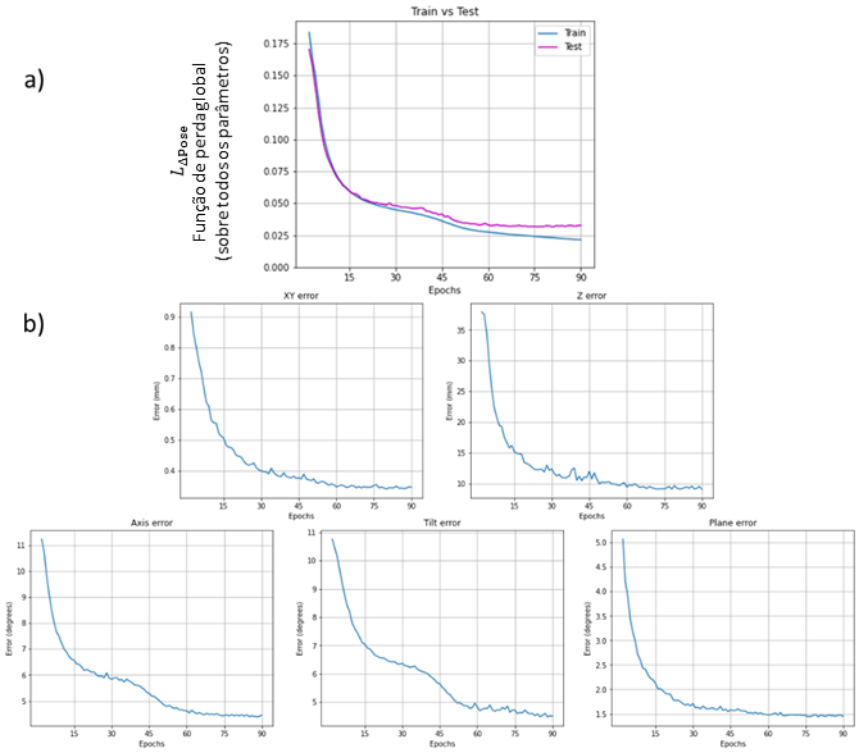
Até a Figura 16, os resultados foram todos obtidos com uma arquitetura de rede baseada em blocos residuais. Para analisar a sua influência, foi treinada uma versão sem blocos residuais, usando as mesmas condições de treino: mesmos dados e mesmos hiperparâmetros. A Figura 17 mostra uma comparação entre a rede treinada com os blocos residuais e a rede treinada com blocos convolucionais comuns. Tal comparação permite confirmar que o uso destes blocos foi de fato melhor para o desempenho geral da rede. Não houve muitas melhorias para os parâmetros Δx e Δy, que são considerados os mais fáceis de serem estimados, mas todos os outros parâmetros se beneficiaram com o uso de blocos residuais.

Figura 14 – a) Curvas da função de perda de treino e teste para DeltaPoseNet$_{V1}$ (treinada com conjunto de dados contendo todos os modelos do MitraClip) – pares gerados **sem** restrições de distância máxima; b) erros para cada parâmetro separadamente. Podemos observar que a perda global de teste é alta e começa a divergir, o que se deve a um mau desempenho na previsão de α e τ.



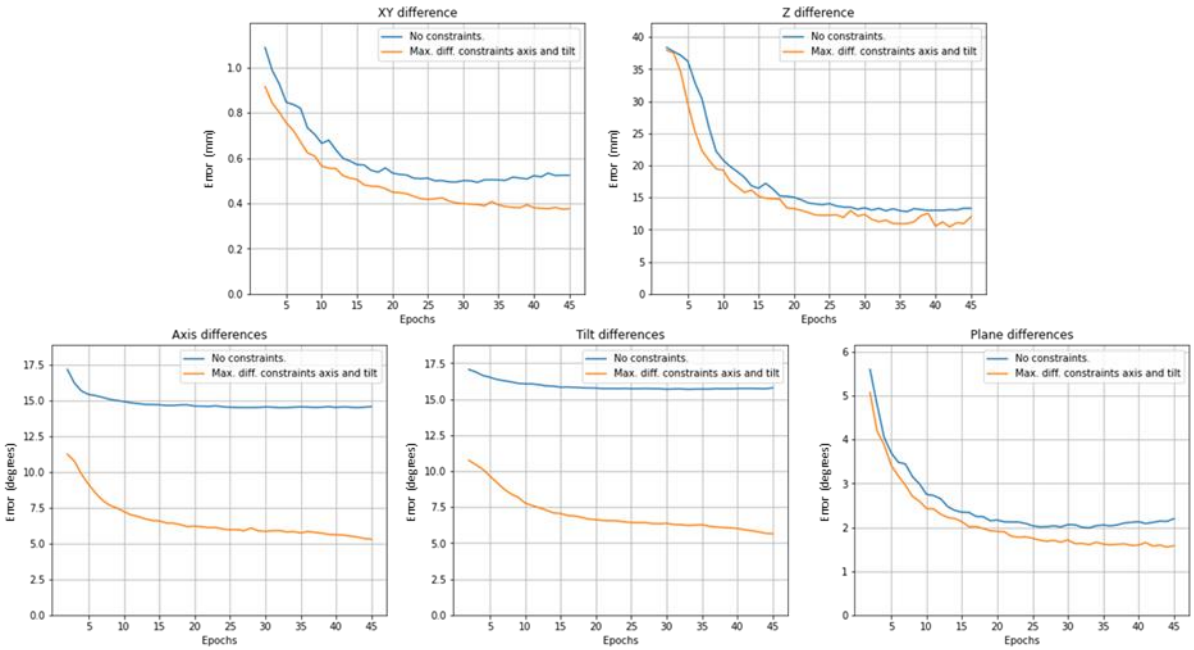
Fonte: Produção da própria autora.

Figura 15 – a Curvas da função de perda de treino e teste para DeltaPoseNet$_{V1}$ (treinada com conjunto de dados contendo todos os modelos do MitraClip) – pares gerados com uma distância máxima de 0,8rad (~46°) em α e τ ($\Delta\alpha \leq 0{,}8\text{rad}$ e $\Delta\tau \leq 0{,}8\text{rad}$); b) erro correspondente para cada parâmetro separadamente.



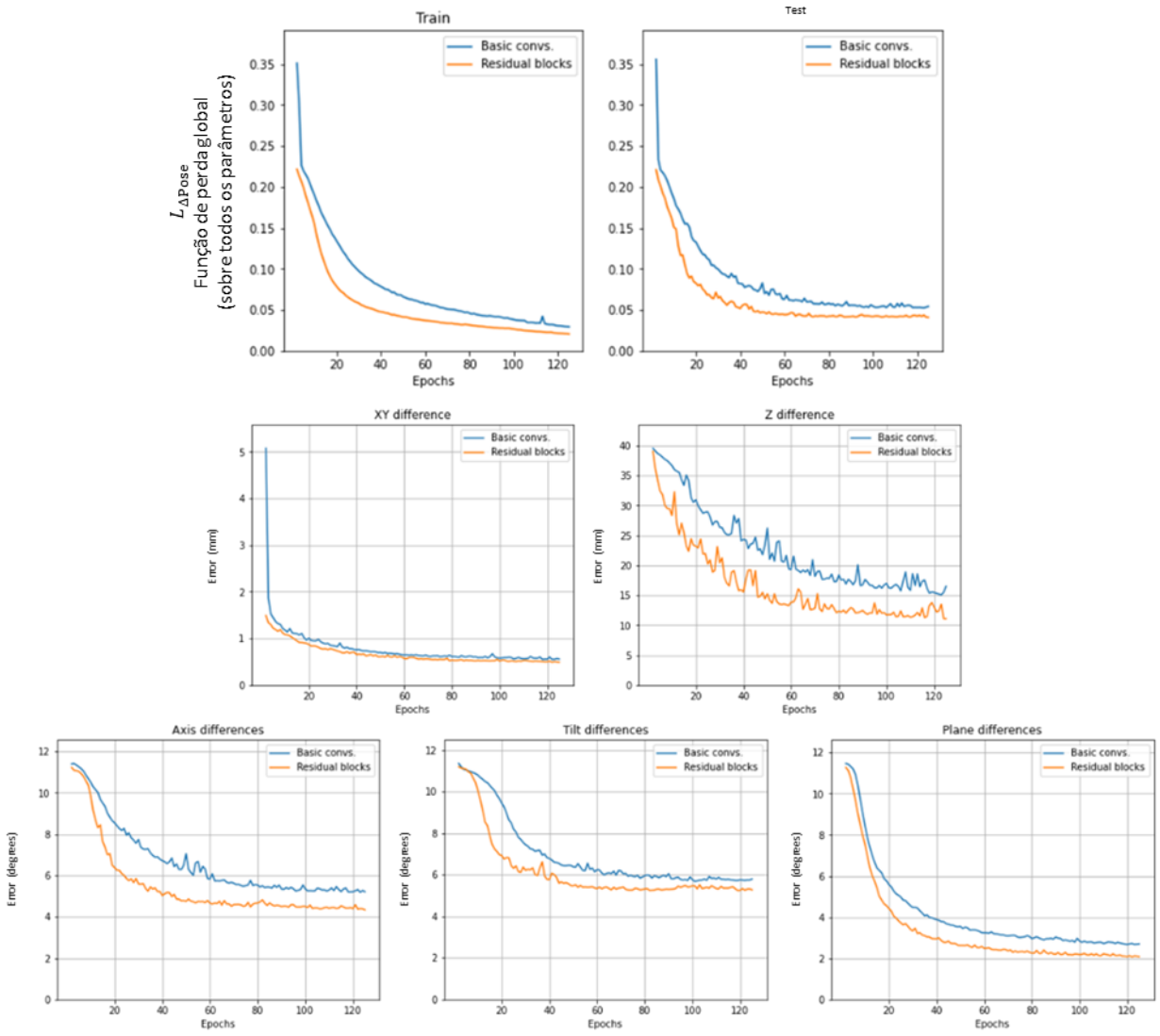
Fonte: Produção da própria autora.

Figura 16 – Comparação por parâmetro entre os treinamentos da DeltaPoseNet$_{V1}$ com e sem restrições de distância máxima em α e τ



Fonte: Produção da própria autora.

Figura 17 – Comparação entre versões da DeltaPoseNet$_{V1}$ treinadas com e sem blocos residuais



Fonte: Produção da própria autora.

3.2 DeltaPoseNet$_{V2}$ – Treinamento usando modelos A, C, E, G do MitraClip

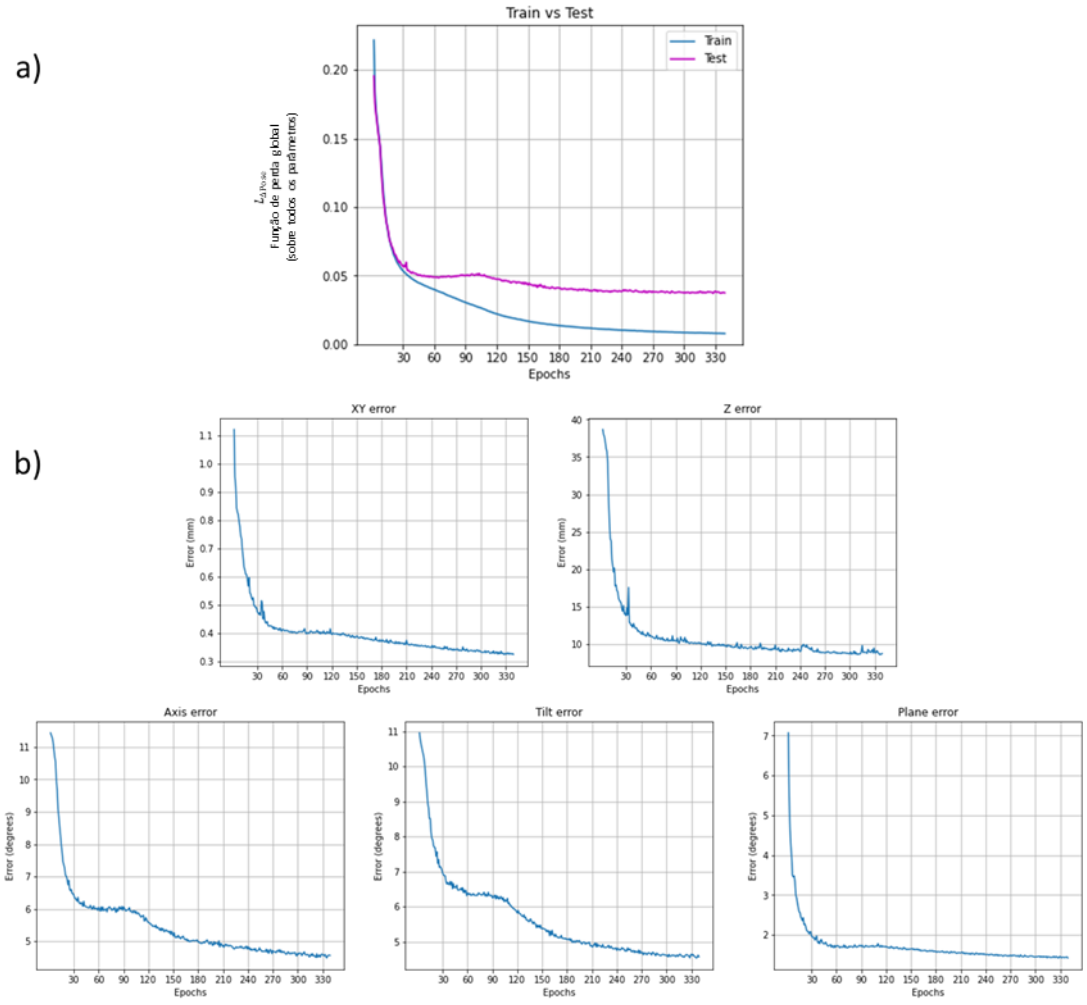
Nesta seção são analisados os resultados obtidos com a DeltaPoseNet treinada utilizando os modelos de objeto A, C, E e G (conjunto de dados 2 como detalhado na Seção 2.3.1.2), que foi chamada de DeltaPoseNet$_{V2}$. A Figura 18 mostra as curvas de perda de treinamento e de teste para este experimento. O valor médio de $L_{\Delta\text{Pose}}$ no conjunto de teste foi de 0,05 (±0,08). Os erros médios foram:

- Em Δx, 0,23 mm (±0,19 mm);
- Em Δy, 0,2339 mm (±0,20 mm);

- Em Δz, 10,89 mm (±10,17 mm);
- Em Δα, 4,88° (±5,90°);
- Em Δτ, 5,13° (±6,15°);
- Em Δρ, 1,59° (±1,62°).

Os resultados foram em geral ligeiramente inferiores aos apresentados pela rede treinada com todos os modelos, mas não significativamente. Na Seção 3.3, serão discutidas as possíveis razões para estes valores de erro mais altos, comparando os desempenhos por modelo do objeto. O objetivo desta experiência foi analisar até que ponto a DeltaPoseNet era capaz de prever a diferença de pose para modelos que não foram vistos durante o treinamento. Isto é discutido mais detalhadamente na Seção 3.3.

Figura 18 – a) Curvas da função de perda de treino e teste para DeltaPoseNet$_{V2}$ (treinada com modelos A, C, E, G do MitraClip) – pares gerados com uma distância máxima de 0,8rad (~46°) em α e τ; b) erro correspondente para cada parâmetro separadamente



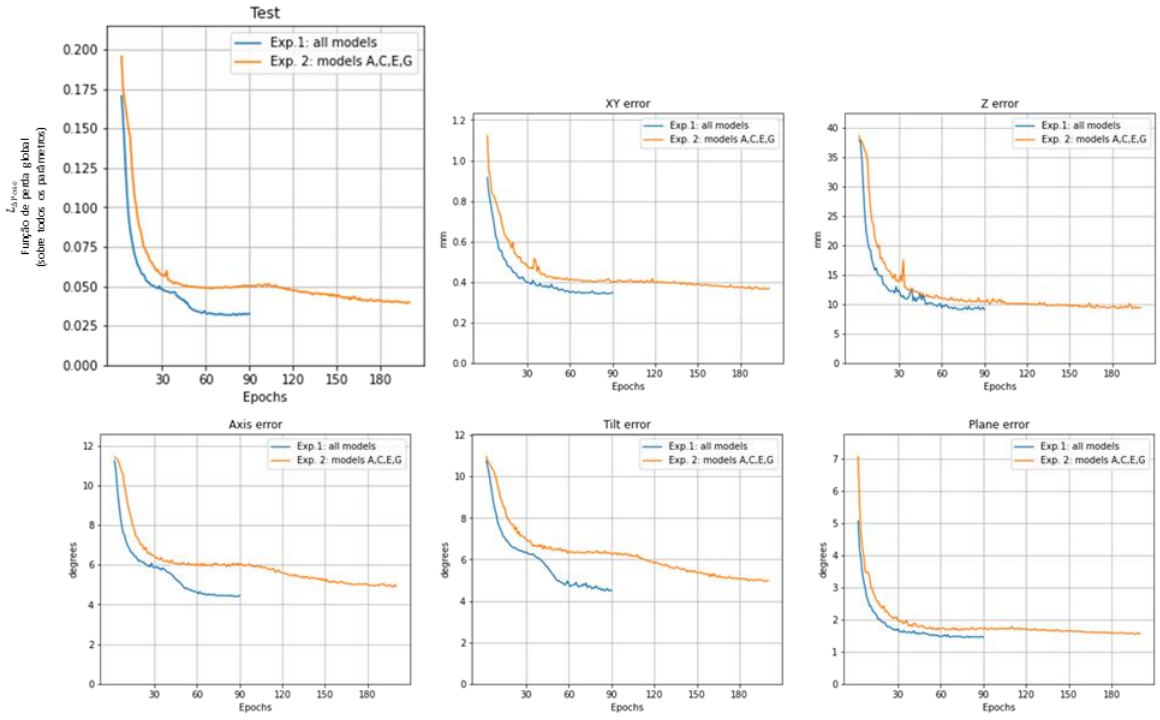
Fonte: Produção da própria autora.

3.3 Comparação entre DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$ no conjunto de teste

Nesta seção, são analisadas e comparadas as duas versões previamente treinadas da DeltaPoseNet: DeltaPoseNet$_{V1}$ (treinada com todos os modelos do MitraClip – Seção 3.1) e DeltaPoseNet$_{V2}$ (treinada com os modelos A, C, E e G – Seção 3.2).

Na Figura 19, podemos ver a comparação da evolução da função de perda e do erro médio no conjunto de testes durante o treinamento das redes DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$. Podemos observar que, embora a DeltaPoseNet$_{V1}$ tenha convergido mais rapidamente, ambas as versões chegam a um nível equiparável de valores finais.

Figura 19 – Comparação entre DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$ no conjunto de teste. Na parte superior esquerda, podemos ver a evolução da função $L_{\Delta\text{Pose}}$ sobre o conjunto de teste ao longo das épocas de treinamento da rede. Os outros gráficos mostram a evolução do erro para os diferentes parâmetros de pose



Fonte: Produção da própria autora.

Também foi feita a comparação por modelo do MitraClip, obtendo os resultados das duas versões da DeltaPoseNet sobre a parcela do conjunto de teste correspondente a somente um dos modelos por vez (como apresentado na Seção 2.3.1.2, foram gerados 100.000 pares de imagens por modelo, dos quais 20.000 foram reservados para teste). A Tabela 1 e a Tabela 2 mostram

os valores médios de erro e perda por modelo do MitraClip, para DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$, respectivamente.

Tabela 1 – Resultados da DeltaPoseNet$_{V1}$ por modelo do MitraClip (pares de imagem compostos por duas DRRs)

DeltaPoseNet$_{V1}$ (treinada com todos os modelos do MitraClip)							
Erro ↓	Modelo A	Modelo B	Modelo C	Modelo D	Modelo E	Modelo F	Modelo G
Δx (mm)	0.2089 ±0.1667	0.2021 ±0.1584	0.1970 ±0.1580	0.2057 ±0.1664	0.2155 ±0.1739	0.2275 ±0.1820	0.2195 ±0.1797
Δy (mm)	0.2295 ±0.1818	0.2175 ±0.1744	0.2117 ±0.1717	0.2160 ±0.1727	0.2336 ±0.1878	0.2411 ±0.1947	0.2428 ±0.1965
Δz (mm)	9.4725 ±7.7713	8.6926 ±7.0863	8.7025 ±7.2282	8.9653 ±7.5123	9.5126 ±7.7952	9.5703 ±7.8255	8.7793 ±7.2404
$\Delta \alpha$ (°)	4.6230 ±4.9030	4.1883 ±4.7408	4.1207 ±4.6307	4.3243 ±4.7316	4.5649 ±4.9316	4.5759 ±4.8315	4.6688 ±4.8515
$\Delta \tau$ (°)	4.8128 ±4.8709	4.5064 ±4.9027	4.4317 ±4.8972	4.4896 ±4.9416	4.5129 ±4.8012	4.6320 ±4.9415	4.8801 ±4.9891
$\Delta \rho$ (°)	1.3988 ±1.1468	1.3492 ±1.1269	1.3732 ±1.1838	1.3775 ±1.1752	1.6034 ±1.3484	1.6357 ±1.3726	1.5344 ±1.2769
$L_{\Delta\text{Pose}}$	**0.0330 ±0.0518**	**0.0299 ±0.0513**	**0.0293 ±0.0493**	**0.0307 ±0.0507**	**0.0323 ±0.0511**	**0.0329 ±0.0523**	**0.0333 ±0.0506**

Fonte: Produção da própria autora.

Tabela 2 – Resultados da DeltaPoseNet$_{V2}$ por modelo do MitraClip (pares de imagem compostos por duas DRRs)

DeltaPoseNet$_{V2}$ (treinada com modelos A, C, E, G do MitraClip)							
Erro ↓	Modelo A	Modelo B	Modelo C	Modelo D	Modelo E	Modelo F	Modelo G
Δx (mm)	0.1959 ±0.1590	0.2165 ±0.1715	0.1984 ±0.1593	0.2246 ±0.1782	0.2192 ±0.1777	0.3507 ±0.2688	0.2095 ±0.1737
Δy (mm)	0.1981 ±0.1619	0.2149 ±0.1686	0.1994 ±0.1617	0.2262 ±0.1770	0.2209 ±0.1770	0.3584 ±0.2768	0.2088 ±0.1732
Δz (mm)	8.3622 ±7.0106	10.6143 ±8.5762	8.8878 ±7.6569	10.8560 ±8.9044	9.4161 ±7.7918	19.6768 ±15.6622	8.4540 ±7.0197
$\Delta \alpha$ (°)	4.4796 ±5.8624	4.6675 ±6.0989	4.3238 ±5.5729	4.6511 ±5.8227	4.8091 ±6.0016	6.6542 ±6.1872	4.6989 ±5.6713
$\Delta \tau$ (°)	4.6282 ±5.8196	4.9043 ±5.7859	4.4030 ±5.7669	5.5112 ±6.6431	4.5177 ±5.6751	7.2313 ±7.0360	4.7644 ±5.8625
$\Delta \rho$ (°)	1.2908 ±1.0286	1.3711 ±1.1546	1.3841 ±1.1962	1.3930 ±1.2167	1.6195 ±1.4020	2.5065 ±2.8870	1.4534 ±1.2183
$L_{\Delta\text{Pose}}$	**0.0371 ±0.0770**	**0.0411 ±0.0784**	**0.0356 ±0.0726**	**0.0455 ±0.0844**	**0.0392 ±0.0758**	**0.0769 ±0.0834**	**0.0380 ±0.0713**

Fonte: Produção da própria autora.

Colocando os valores médios de $L_{\Delta\text{Pose}}$ para DeltaPoseNet$_{V1}$ da Tabela 1 em ordem, obtém-se:

$$L_{\Delta pose_C} < L_{\Delta pose_B} < L_{\Delta pose_D} < L_{\Delta pose_E} < L_{\Delta pose_F} < L_{\Delta pose_A} < L_{\Delta pose_G}$$

Portanto, junto com o modelo F, os modelos A, E e G foram os modelos para os quais a DeltaPoseNet$_{V1}$ apresentou os maiores valores de perda. Esta é provavelmente a razão pela qual o desempenho da DeltaPoseNet$_{V2}$ (treinado somente com os modelos A, C, E, G) foi ligeiramente inferior do que o da DeltaPoseNet$_{V1}$ (como exposto anteriormente na Seção 3.2): primeiro, a DeltaPoseNet$_{V2}$ teve que lidar com modelos que estavam mais distantes entre si (em termos de configuração do ângulo do braço), o que aumenta a dificuldade da tarefa; segundo, nos experimentos com DeltaPoseNet$_{V1}$ (Seção 3.1), a perda global média e os erros médios sofrem influência dos erros obtidos sobre os modelos B e D, o que reduz o valor dos erros e perda médios.

Ao analisar os resultados para DeltaPoseNet$_{V2}$ (Tabela 2), tem-se a seguinte ordem de valores de perda:

$$L_{\Delta pose_C} < L_{\Delta pose_A} < L_{\Delta pose_G} < L_{\Delta pose_E} < L_{\Delta pose_B} < L_{\Delta pose_D} < L_{\Delta pose_F}.$$

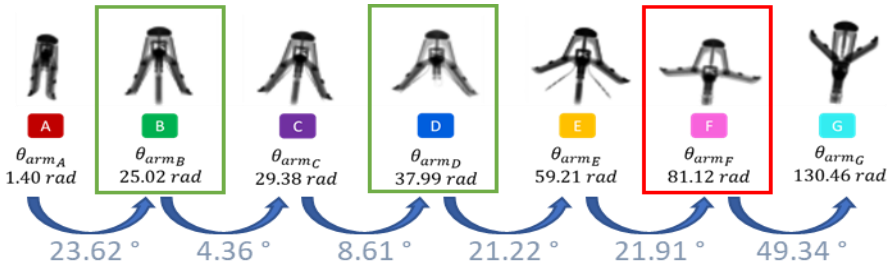
Como esperava-se, os valores de perda foram menores para os modelos vistos durante o treinamento (A, C, E, G). É interessante notar, porém, que a rede se mostra capaz de dar uma estimativa satisfatória da diferença de pose mesmo para os modelos do MitraClip que não foram vistos durante o treinamento (B, D, F) e que são interpolações daqueles vistos durante o treinamento (em termos do ângulo entre os braços do MitraClip). Este resultado mostra que a rede pode ser treinada com uma representação discreta do objeto e ainda ser capaz de atuar em um contexto real, no qual novas configurações contínuas serão encontradas.

As maiores discrepâncias observadas entre os modelos vistos durante o treinamento (A, C, E, G) e os demais (B, D, F) ocorrem para os parâmetros Δz, $\Delta\tau$ e $\Delta\alpha$, o que é explicado pelo grau de complexidade mais elevado para a identificação da influência destes parâmetros sobre a projeção do objeto.

Ainda considerando DeltaPoseNet$_{V2}$, a maior perda foi registrada para o modelo F. Isto pode ser explicado pelo fato de que este modelo é o que está mais distante de seus "vizinhos", ou seja, a diferença entre seu ângulo de braço (θ_{arm_F}) e os de seus vizinhos ($\theta_{arm_E}, \theta_{arm_G}$) é maior (como ilustrado em Figura 20): $\Delta\theta_{arm_{EF}} = 21{,}91°$ and $\Delta\theta_{arm_{FG}} = 49{,}34°$, enquanto para os modelos B e D, tem-se: $\Delta\theta_{arm_{AB}} = 23{,}62°$, $\Delta\theta_{arm_{BC}} = 4{,}36°$, $\Delta\theta_{arm_{CD}} = 8{,}61°$ e

$\Delta\theta_{arm_{DE}} = 21{,}22°$ (ou seja, estes modelos estão consideravelmente mais próximos de seus "vizinhos"). Além disso, entre os modelos vistos durante o treinamento (A, C, E, G), foram observados erros maiores para os modelos E e G, o que também justifica um desempenho pior para o modelo F, já que o modelo F é uma interpolação dos modelos E e G.

Figura 20 – Diferenças angulares entre os modelos de objetos, mostrando que, entre os modelos B, D e F, o modelo F é o que apresenta a maior distância de ângulo de braço comparado aos seus vizinhos na representação discreta (21,91° e 49,34°).

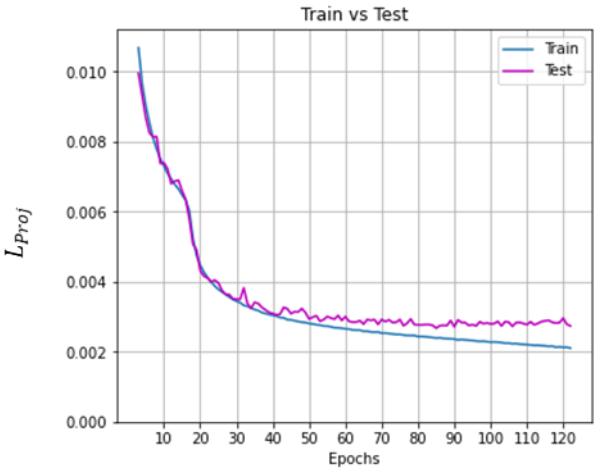


Fonte: Produção da própria autora.

3.4 ProjectNet

A função de perda L_{Proj} (equação (6)) usada para o treinamento da ProjectNet foi o erro médio quadrático entre a saída e a projeção de referência (*ground-truth*). As curvas para as perdas sobre os conjuntos de treinamento e de teste ao longo das épocas são mostradas na Figura 21. O modelo final selecionado foi o modelo obtido na última época do treino.

Figura 21 – Evolução da função de perda ao longo das épocas durante o treinamento da ProjectNet

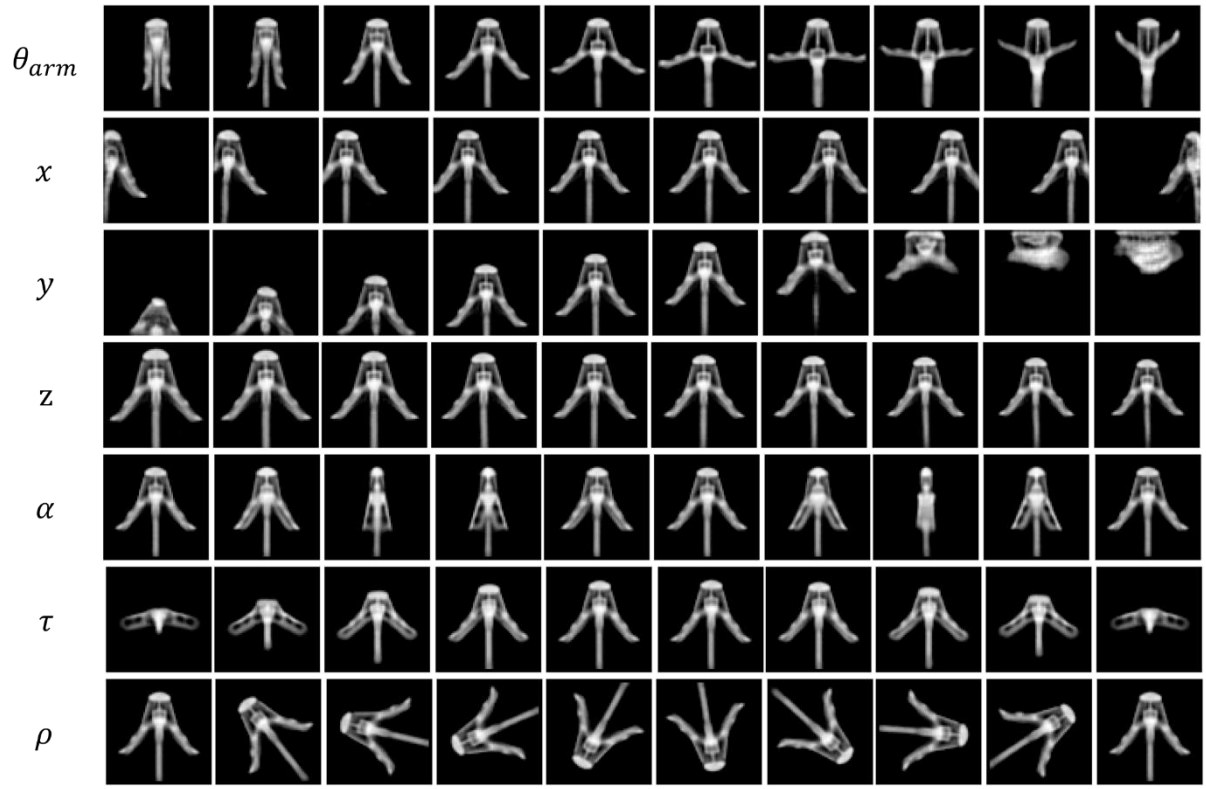


Fonte: Produção da própria autora.

O valor médio da perda L_{Proj} (equação (6)) sobre o conjunto de testes para o modelo final da ProjectNet foi de $0{,}0027$, com um desvio padrão de $7.9374 \cdot 10^{-5}$. O tempo médio para gerar uma projeção usando a GPU disponível foi de $7{,}94ms$ (medido sobre 1000 amostras). Para efeitos de comparação, o tempo médio para gerar uma DRR clássica utilizando a técnica de *ray-tracing* é de $20ms$. Assim, é mais rápido gerar uma projeção utilizando a ProjectNet. Além disso, a projeção é diferenciável, pois é gerada com uma rede neural implementada em TensorFlow, o que abre a possibilidade de explorar o gradiente envolvido em sua geração.

Para avaliar o desempenho qualitativo da ProjectNet, a Figura 22 mostra diferentes projeções obtidas. Em cada linha pode-se observar os resultados das projeções variando apenas um dos parâmetros, no intervalo dos parâmetros de dados de treinamento (como mostrado na Figura 10). É possível ver que as transições são em geral suaves, contínuas e consistentes.

Figura 22 – Influência da alteração dos diferentes parâmetros na projeção de saída da ProjectNet. Os valores dos parâmetros foram tomados uniformemente espaçados no intervalo dos parâmetros dos dados de treinamento, mostrados na Figura 10



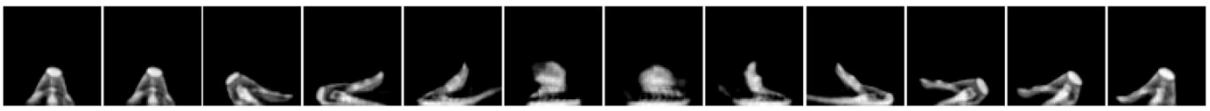
Fonte: Produção da própria autora.

Um resultado particular pode ser observado na imagem mais à direita na linha correspondente à coordenada y na Figura 22. Este comportamento é verificado para os casos em que a ponta do clipe não está presente na imagem, como pode ser verificado na Figura 23 e na Figura 24. Na Figura 23, observa-se que, fixando $y = 10$, o efeito aparece quando ρ se afasta de 0 (seja em direção a $-\pi$ ou em direção a $+\pi$), uma vez que é nessa região que a ponta do clipe deixa de aparecer na projeção. Já a Figura 24 mostra que, fixando $y = -10$, o efeito aparece quando ρ se aproxima de 0 (seja a partir de $-\pi$ ou de $+\pi$), pelo mesmo motivo.

Figura 23 – Evolução do parâmetro ρ de $-\pi$ (projeção à extrema esquerda) a $+\pi$ (projeção à extrema direita), com y fixo a $+10$

Fonte: Produção da própria autora.

Figura 24 – Evolução do parâmetro ρ de $-\pi$ (projeção à extrema esquerda) a $+\pi$ (projeção à extrema direita), com y fixo a -10

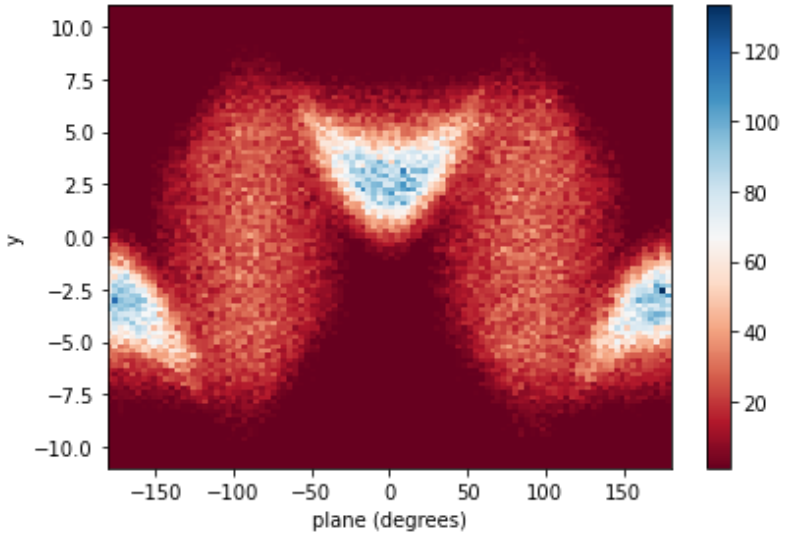


Fonte: Produção da própria autora.

Este comportamento é uma consequência da forma como o conjunto de dados foi gerado: não ter a ponta do clipe na projeção foi usado como critério de exclusão, ou seja, apenas DRRs contendo a ponta do clipe foram mantidas. Como foi exposto na Seção 2.2, isto fez com que a distribuição dos parâmetros não fosse uniforme. Como consequência, algumas combinações de parâmetros nunca são representadas no conjunto de dados. De fato, pode-se verificar na Figura 25 e na Figura 26 que situações como $\rho \cong 0$ e $y \cong -10$ não estão bem representadas nos dados de treinamento, o que explica a dificuldade mostrada pela rede em aprender estas representações.

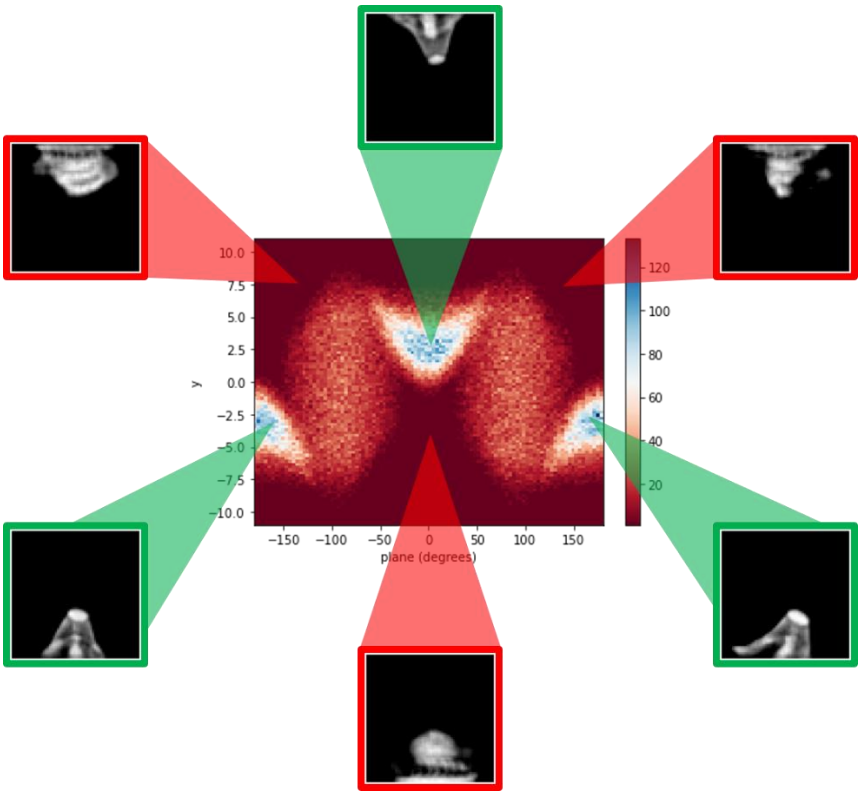
Assim, os resultados estranhos verificados para a ProjectNet são compreensíveis: eles ocorrem quando as configurações de pose estão longe dos exemplos vistos durante o treinamento, ou seja, a rede não pode extrapolar além do que está representado no conjunto de treinamento. A ProjectNet é, portanto, consistente nas projeções que gera, apresentando dificuldades apenas devido a um problema no conjunto de treinamento.

Figura 25 – Histograma conjunto para os dados de treinamento mostrando os parâmetros y e ρ



Fonte: Produção da própria autora.

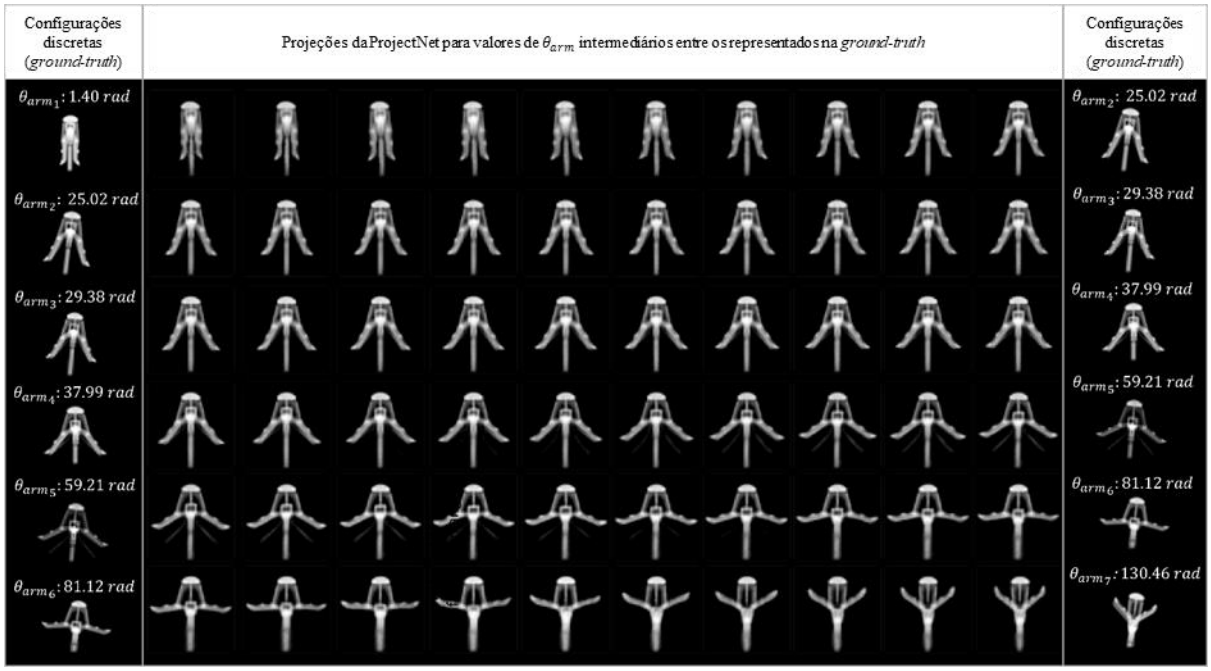
Figura 26 – Ilustração de situações em que a projeção de saída foi a esperada (verde) e onde surgiram problemas (vermelho). Pode-se ver a relação com a distribuição dos dados de treinamento: a rede teve dificuldades em regiões que não estavam representadas pelos dados de treinamento



Fonte: Produção da própria autora.

Além disso, embora seja possível gerar DRRs através da técnica de *ray-tracing*, a grande vantagem da ProjectNet é a capacidade de interpolar novas configurações contínuas do objeto a partir de uma representação discreta no conjunto de dados. Isto é, ProjectNet é capaz de gerar projeções para qualquer ângulo θ_{arm} entre os braços do MitraClip, apesar de o conjunto de dados utilizado conter somente a representação de sete configurações diferentes de θ_{arm} (conforme Figura 8). Isto pode ser verificado na Figura 27, onde são mostradas diferentes projeções obtidas para valores de ângulo θ_{arm} entre os braços do MitraClip, os quais são valores intermediários interpolados entre os valores representados pelos sete modelos disponíveis no conjunto de dados.

Figura 27 – Saídas da ProjectNet para diferentes valores de ângulo θ_{arm} entre os braços do MitraClip (valores não representados pelo conjunto de dados), mostrando a capacidade da rede de interpolar novas configurações do clipe



Fonte: Produção da própria autora.

3.5 Project-DeltaPose-Net

3.5.1 Treinamento da DeltaPoseNet$_{V3}$ e da DeltaPoseNet$_{V4}$

Como explicado na Seção 2.5.1, duas novas versões da DeltaPoseNet foram treinadas para permitir a sua combinação com a ProjectNet (isto é, usando como entrada um par de imagens consistindo de uma DRR e uma projeção dada pela ProjectNet). Tais versões são referidas como

"DeltaPoseNet$_{V3}$" (versão treinada com o conjunto de dados 3) e como "DeltaPoseNet$_{V4}$" (versão treinada com o conjunto de dados 4).

Na Figura 28 e na Figura 29, pode-se observar que as curvas de perdas e erros durante o treinamento foram bastante próximas para ambos os experimentos. Os valores das perdas e dos erros sobre seus respectivos conjuntos de testes são mostrados na Tabela 3. É possível perceber que seus valores são muito próximos, mostrando que as duas versões da rede têm desempenhos semelhantes.

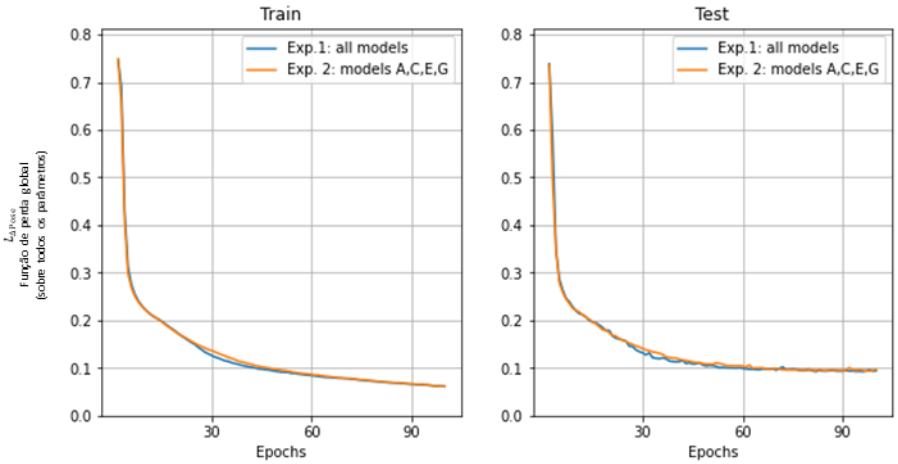
A partir desta tabela, pode-se também comparar seus desempenhos com os obtidos com os experimentos descritos na Seção 3.3, onde os pares de imagens foram compostos apenas por DRRs. Note que os desempenhos das versões treinadas com projeções da ProjectNet (DeltaPoseNet$_{V3}$ e DeltaPoseNet$_{V4}$) são inferiores aos obtidos pelas versões treinadas apenas com DRRs (DeltaPoseNet$_{V1}$ e DeltaPoseNet$_{V2}$). Isto sugere que, embora as projeções da ProjectNet sejam consistentes, a rede ainda não aprendeu perfeitamente todas as mudanças sutis causadas pelas variações dos diferentes parâmetros de pose. Em todo caso, os resultados são satisfatórios e promissores, motivando um futuro trabalho de melhoria do modelo de rede da ProjectNet.

Tabela 3 – Comparação do desempenho das diferentes versões da DeltaPoseNet sobre os conjuntos de teste correspondentes

Erro ↓	**DeltaPoseNet$_{V1}$**	**DeltaPoseNet$_{V2}$**	**DeltaPoseNet$_{V3}$**	**DeltaPoseNet$_{V4}$**
$L_{\Delta Pose}$	0.03 (±0.05)	0.05 (±0.08)	0.09 (±0.09)	0.10 (±0.10)
Δx **(mm)**	0.21mm (±0.17mm)	0.23mm (±0.19mm)	0.46mm (±0.38mm)	0.47mm (±0.38mm)
Δy **(mm)**	0.23mm (±0.18mm)	0.23mm (±0.20mm)	0.47mm (±0.39mm)	0.49mm (±0.40mm)
Δz **(mm)**	9.12mm (±7.53mm)	10.89mm (±10.17mm)	20.98mm (±16.97mm)	21.72mm (±17.77mm)
$\Delta \alpha$ **(°)**	4.43° (±4.81°)	4.88° (±5.90°)	7.60° (±6.16°)	7.73 ° (±6.27°)
$\Delta \tau$ **(°)**	4.59° (±4.89°)	5.13° (±6.15°)	8.05° (±6.29°)	8.18 ° (±6.47°)
$\Delta \rho$ **(°)**	1.47° (±1.24°)	1.59° (±1.62°)	4.23° (±4.88°)	4.12 ° (±4.87°)

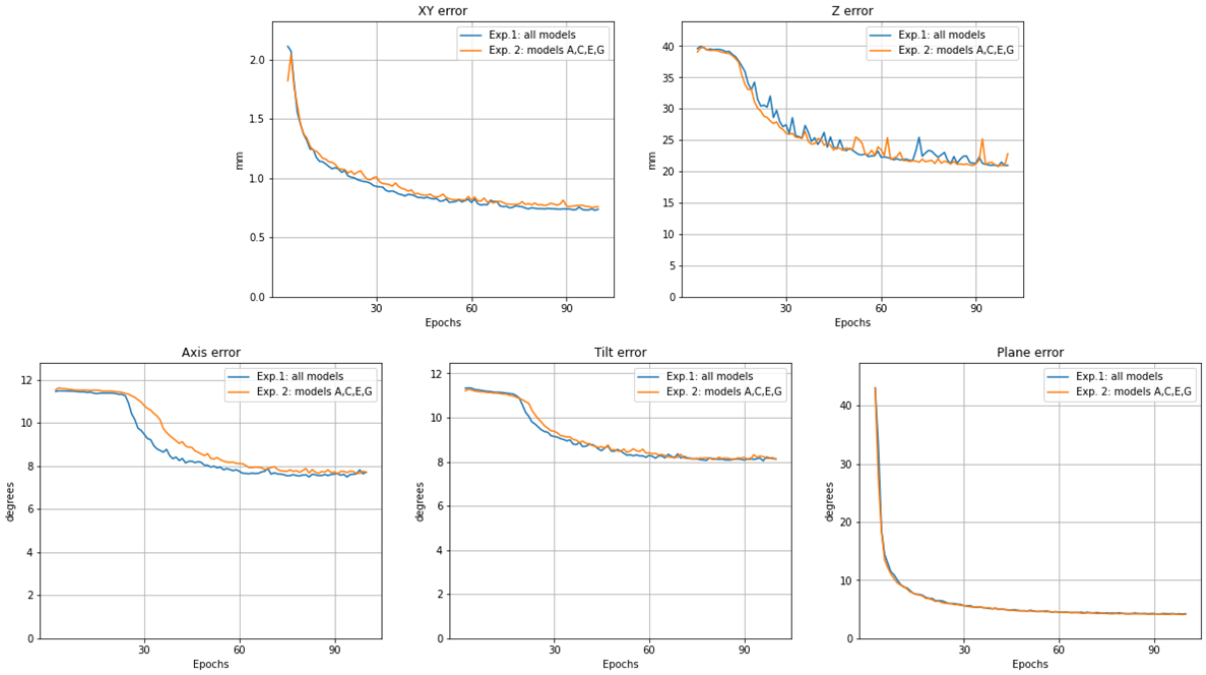
Fonte: Produção da própria autora.

Figura 28 – Funções de perda sobre o conjunto de treinamento e de teste durante o treinamento da DeltaPoseNet$_{V3}$ (nos gráficos, Exp. 1, com pares de todos os modelos de objetos) e DeltaPoseNet$_{V4}$ (nos gráficos, Exp. 2, com pares de modelos de objetos A, C, E, G). Os pares de imagens consistem em uma DRR e uma projeção da ProjectNet



Fonte: Produção da própria autora.

Figura 29 – Erro entre os valores previstos e a *ground-truth* sobre os dados de teste, para os diferentes parâmetros de pose, durante o treinamento da DeltaPoseNet$_{V3}$ (nos gráficos, Exp. 1, com pares de todos os modelos de objetos) e DeltaPoseNet$_{V4}$ (nos gráficos, Exp. 2, com pares de modelos de objetos A, C, E, G). Os pares de imagens consistem em uma DRR e uma projeção da ProjectNet



Fonte: Produção da própria autora.

A Tabela 4 e a Tabela 5 mostram a comparação por modelo de objeto da DeltaPoseNet$_{V3}$ e da DeltaPoseNet$_{V4}$, respectivamente.

Tabela 4 – Resultados da DeltaPoseNet$_{V3}$ por modelo do MitraClip (pares de imagem compostos por uma DRR e uma projeção da ProjectNet)

DeltaPoseNet$_{V3}$ (treinada com todos os modelos do MitraClip)							
Erro ↓	Modelo A	Modelo B	Modelo C	Modelo D	Modelo E	Modelo F	Modelo G
Δx (mm)	0.4620 ±0.3733	0.4328 ±0.3505	0.4239 ±0.3412	0.4401 ±0.3533	0.4624 ±0.3708	0.4869 ±0.3927	0.5018 ±0.4115
Δy (mm)	0.4800 ±0.3963	0.4359 ±0.3570	0.4383 ±0.3580	0.4453 ±0.3667	0.4716 ±0.3821	0.4927 ±0.4028	0.5208 ±0.4308
Δz (mm)	22.3973 ±17.6416	19.8833 ±15.8862	19.5030 ±15.7134	19.8263 ±16.0291	20.7836 ±16.7877	21.3463 ±17.1994	21.9064 ±17.7745
$\Delta \alpha$ (°)	8.5243 ±6.3525	7.3435 ±6.0720	7.1718 ±5.9779	7.2491 ±6.0575	7.3209 ±6.1143	7.5323 ±6.1331	8.0325 ±6.3581
$\Delta \tau$ (°)	8.8257 ±6.6201	7.9331 ±6.2539	7.7712 ±6.1506	7.7364 ±6.1926	7.6627 ±6.0580	7.7326 ±6.0759	8.2497 ±6.3149
$\Delta \rho$ (°)	4.4472 ±5.0666	3.8163 ±3.8254	3.7319 ±3.7958	3.7663 ±3.8495	4.0079 ±4.0925	4.2290 ±4.6696	4.9264 ±6.4308
$L_{\Delta Pose}$	**0.1064 ±0.0919**	**0.0851 ±0.0766**	**0.0820 ±0.0750**	**0.0835 ±0.0772**	**0.0861 ±0.0802**	**0.0905 ±0.0868**	**0.1046 ±0.1164**

Fonte: Produção da própria autora.

Tabela 5 – Resultados da DeltaPoseNet$_{V4}$ por modelo do MitraClip (pares de imagem compostos por uma DRR e uma projeção da ProjectNet)

DeltaPoseNet$_{V4}$ (treinada com modelos A, C, E, G do MitraClip)							
Erro ↓	Modelo A	Modelo B	Modelo C	Modelo D	Modelo E	Modelo F	Modelo G
Δx (mm)	0.4462 ±0.3642	0.4423 ±0.3523	0.4360 ±0.3532	0.4563 ±0.3649	0.4713 ±0.3837	0.5243 ±0.4208	0.4852 ±0.3952
Δy (mm)	0.4682 ±0.3847	0.4652 ±0.3787	0.4613 ±0.3733	0.4735 ±0.3838	0.4975 ±0.4022	0.5517 ±0.4404	0.5058 ±0.4183
Δz (mm)	21.0763 ±16.7124	21.2558 ±17.1672	20.5112 ±16.7024	20.5363 ±16.6776	20.7503 ±17.0233	26.5082 ±21.2352	20.5043 ±17.1083
$\Delta \alpha$ (°)	8.1918 ±6.3273	7.5025 ±6.1959	7.3164 ±6.0798	7.4356 ±6.1675	7.4278 ±6.2066	8.2597 ±6.5540	7.7962 ±6.2849
$\Delta \tau$ (°)	8.7034 ±6.6717	8.0874 ±6.3632	7.9253 ±6.3518	8.1659 ±6.5980	7.8326 ±6.3068	8.3551 ±6.5282	8.0483 ±6.3059
$\Delta \rho$ (°)	3.9867 ±4.4278	3.7964 ±3.8673	3.7502 ±4.2014	3.7359 ±3.7964	3.9302 ±4.0788	4.6337 ±5.1908	4.3139 ±5.3186
$L_{\Delta Pose}$	**0.0991 ±0.0861**	**0.0903 ±0.0806**	**0.0871 ±0.0886**	**0.0897 ±0.0831**	**0.0890 ±0.0835**	**0.1134 ±0.1029**	**0.0956 ±0.0992**

Fonte: Produção da própria autora.

Pode-se tirar conclusões semelhantes às da Seção 3.3: mesmo no caso em que uma das DRRs foi substituída por uma projeção da ProjectNet, observa-se que a rede treinada com apenas alguns dos modelos ainda consegue dar previsões satisfatórias para modelos que não foram

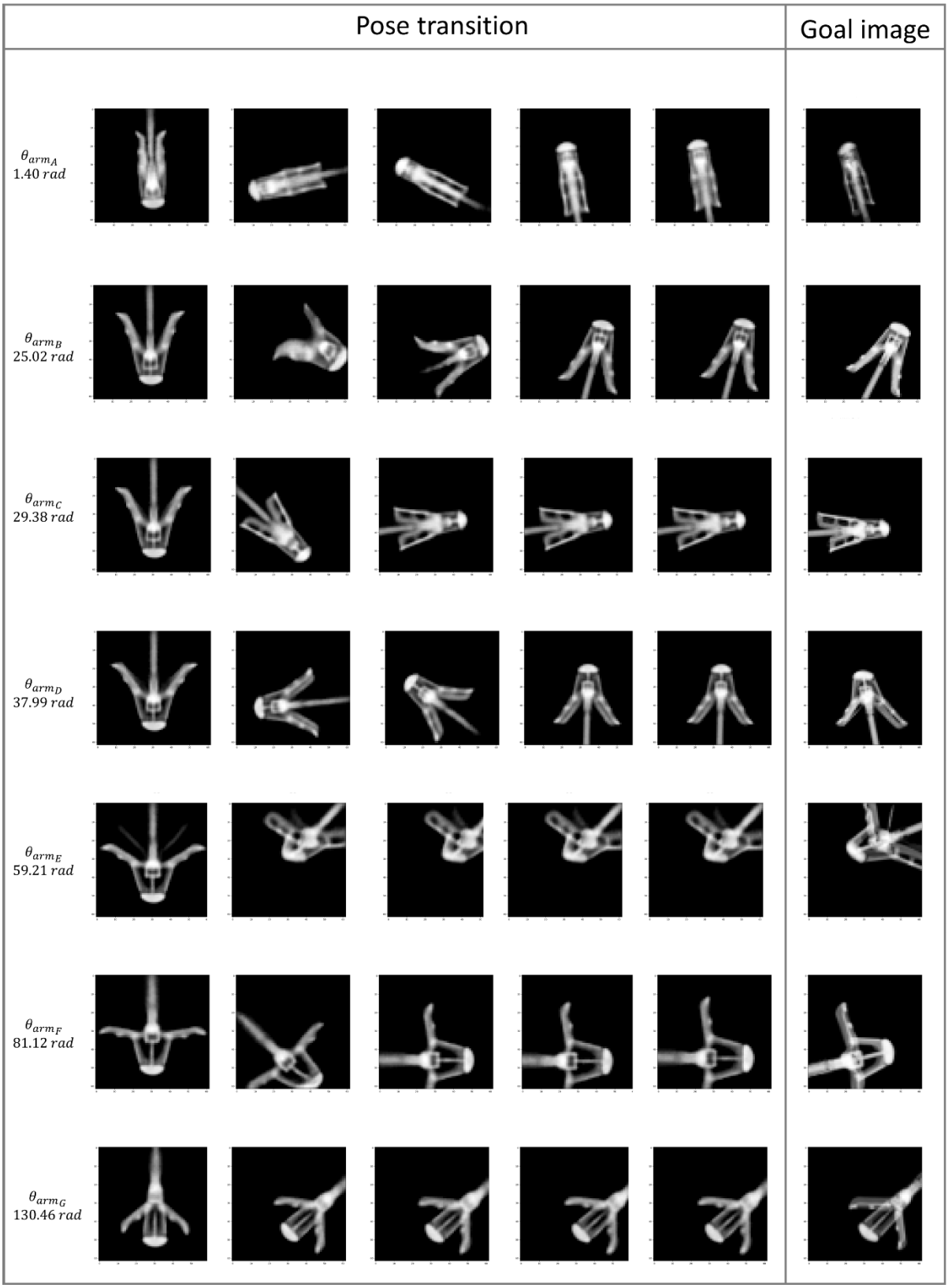
vistos durante o treinamento, mas que são uma interpolação dos que o foram. Isto fornece uma visão interessante sobre o potencial da sua utilização para comparação de imagens reais de raios-X com as projeções geradas pela ProjectNet.

3.5.2 Refinamento de pose

Alguns exemplos de resultados obtidos com o refinamento de pose como descrito na Seção 2.5.2 são mostrados na Figura 30. O processo iterativo geralmente chega à melhor projeção após algumas iterações (~3 iterações). Esta abordagem está longe de ser a ideal, mas dá uma visão sobre o potencial de explorar a minimização da diferença de pose usando os métodos desenvolvidos neste projeto (associando ProjectNet e DeltaPoseNet).

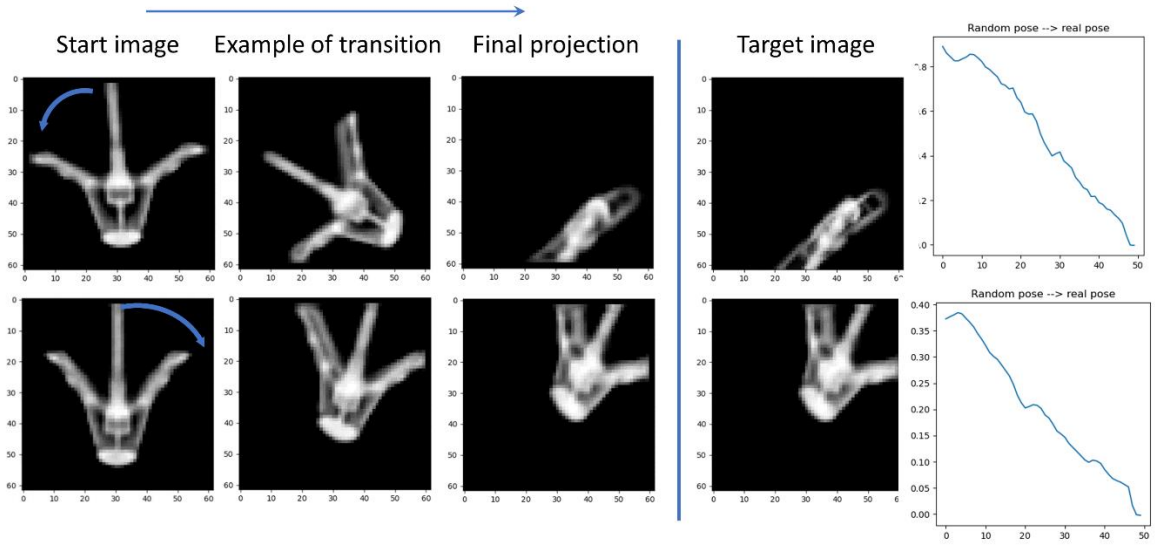
Para avaliar a consistência da medida de similaridade (isto é, verificar se ela diminui à medida que as imagens se aproximam umas das outras), foi calculado seu valor para diferentes projeções, partindo de uma postura centrada e evoluindo exatamente para a mesma postura que a imagem de referência. Alguns resultados são mostrados na Figura 31, o que permite validar S_{pose} como uma medida de similaridade consistente.

Figura 30 – Transição de pose através da atualização iterativa dos parâmetros de pose inicial usando a diferença de pose prevista pela DeltaPoseNet



Fonte: Produção da própria autora.

Figura 31 – Avaliação da medida de similaridade S_{pose} sobre alguns exemplos. Inicia-se a partir de uma postura centrada e evolui-se para a mesma postura que a postura alvo, calculando o valor de S_{pose} a cada nova projeção gerada. Os gráficos à direita mostram a evolução da medida S_{pose}



Fonte: Produção da própria autora.

4 CONCLUSÕES

Durante a realização deste projeto, foi desenvolvido um sistema capaz de codificar uma medida de similaridade consistente, baseada na estimativa de diferença de pose da DeltaPoseNet entre duas imagens. Ou seja, a diferença de pose estimada é de fato menor quanto mais próximas as imagens são uma da outra. Também foram obtidos resultados muito interessantes em relação à ProjectNet, uma rede capaz de gerar projeções diferenciáveis do objeto de interesse em um curto espaço de tempo. Além disso, a associação das duas redes mostrou resultados promissores no que diz respeito à estimativa da pose 3D do objeto de interesse.

Foram utilizados diferentes modelos 3D do MitraClip e foi demonstrado que o sistema desenvolvido é capaz de lidar com configurações de objetos que não foram vistas durante o treinamento. Tanto a ProjectNet quanto a DeltaPoseNet são capazes de interpolar para configurações de objetos nunca vistas anteriormente. A ProjectNet aprendeu a representação de uma transformação contínua não-rígida: a variação do ângulo entre os braços do MitraClip. Assim, a rede é capaz de gerar projeções contínuas, mesmo tendo sido treinada com uma representação discreta do objeto.

Uma extensão interessante do sistema desenvolvido seria ir além dos parâmetros de pose que a DeltaPoseNet atualmente prevê (equação (1)) e incluir a previsão da diferença entre os ângulos dos braços $\Delta\theta_{arm}$. Até agora, nos pares de imagens usados para treinar a DeltaPoseNet, ambas as imagens pertenciam ao mesmo modelo, ou seja, com o mesmo ângulo entre os braços do MitraClip. Seria ideal ter um sistema capaz de lidar bem com a previsão do $\Delta\theta_{arm}$, já que não se sabe de antemão qual será a configuração do ângulo do braço na imagem de raios-X observada. Acredita-se que esta seja uma tarefa consideravelmente complexa, já que nem sempre é possível dizer qual é o ângulo entre os braços do clipe a partir de apenas uma imagem. Por exemplo, uma situação particularmente difícil é quando o clipe está posicionado com $\alpha = \pm\frac{\pi}{2}\ rad$, pois os braços do clipe estão alinhados, um ocluindo o outro. Assim, seria muito provável que fosse necessário considerar uma abordagem diferente (por exemplo, considerando *frames* de imagem em uma sequência em vez de apenas uma imagem observada). Em tal situação, a ProjectNet já estaria pronta para lidar com esta nova característica da DeltaPoseNet, uma vez que o ângulo θ_{arm} já é um de seus parâmetros de entrada.

Nos experimentos realizados, foram limitadas as diferenças angulares máximas em α e τ, dado que a DeltaPoseNet apresentou dificuldades quando treinada sem tal restrição (Seção 3.1). Para superar esta questão, relacionada à simetria do MitraClip e à mudança que ela causa na projeção, outras funções de perda poderiam ser exploradas em trabalhos futuros, tais como as sugeridas em (LI; WANG; JI; XIANG; FOX, 2020), (RAD; LEPETIT, 2017) e (XIANG; SCHMIDT; NARAYANAN; FOX, 2018), embora isso implasse no uso do modelo 3D do objeto a ser computado (o que é o caso da maioria das perdas de que tentam resolver casos de simetria de objeto).

Na Seção 3.5.1 foi discutido o fato de que alguns resultados foram inferiores quando foram usadas as projeções da ProjectNet, em vez de apenas as DRRs do conjunto de dados da base. Isto sugere que a ProjectNet não foi capaz de aprender completamente as mudanças sutis associadas a cada parâmetro de pose, especialmente a mudança de escala causada por variações em z. Portanto, uma primeira extensão natural deste projeto seria explorar mudanças na ProjectNet a fim de melhorar a qualidade das projeções.

Com relação à tarefa de refinamento da pose, embora tenha sido utilizada uma simples abordagem estilo *grid-search*, os resultados forneceram uma visão sobre o potencial de explorar a minimização da diferença de pose codificada pelo sistema desenvolvido neste projeto (ou seja, combinando ProjectNet e DeltaPoseNet). Como continuação, diferentes algoritmos de otimização poderiam ser considerados, seja explorando os gradientes da rede, por exemplo, através da descida de gradientes; ou utilizando outras técnicas de otimização multivariável, tais como o método de Powell (POWELL, 1964). Independentemente da técnica de otimização, seria certamente importante concentrar algum esforço para garantir que a função codificada pelo sistema desenvolvido seja suave. Caso contrário, os métodos de otimização terão dificuldades. Assim, seria uma boa ideia considerar a regularização dos pesos da rede, por exemplo, através de técnicas como *dropout* e/ou a normalização espectral.

Este trabalho mostrou resultados promissores na estimativa da diferença de pose, que pode ser usada para aplicações em registro de imagem e estimativa de pose 3D. Como foi discutido nesta seção, este projeto abre diferentes possibilidades para pesquisas futuras.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBOTT GROUP OF COMPANIES. **MitraClip (Transcatheter Mitral Valve Repair)**. Disponível em: https://mitraclip.com/mitraclip-procedure/. Acesso em: 18 ago. 2020.

BESL, P. J.; MCKAY, N. D. A method for registration of 3D shapes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 14, n. 2, p. 239-256, fev. 1992.

BRACHMANN, E.; KRULL, A.; MICHEL, F.; GUMHOLD, S.; SHOTTON, J.; ROTHER, C. Learning 6D Object Pose Estimation Using 3D Object Coordinates. *In*: EUROPEAN CONFERENCE ON COMPUTER VISION, 13., 2014, Zurich. **Proceedings** […]. Zurich: Springer, 2014. p. 536-551.

BRACHMANN, E.; MICHEL, F.; KRULL, A.; YANG, M. Y.; GUMHOLD, S.; ROTHER, C. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016, Las Vegas. **Proceedings** […]. Las Vegas: IEEE, 2016. p. 3364-3372.

COLLET, A.; MARTINEZ, M.; SRINIVASA, S. S. The MOPED framework: Object Recognition and Pose Estimation for Manipulation. **The International Journal of Robotics Research (IJRR)**, v. 30, n. 10, p. 1284-1306, 2011.

DOSOVITSKIY, A.; FISCHER, P.; ILG, E.; HAUSSER, P.; HAZIRBAS, C.; GOLKOV, V.; VAN DER SMAGT, P.; CREMERS, D.; BROX, T. Flownet: Learning optical convolutional networks. *In:* IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2015. **Proceedings** […]. IEEE ICCV, 2015. p. 2758-2766.

FAVA, C. **MitraClip in the Real World: Mid-Term Progress**. 2019. Disponível em: https://solaci.org/en/2019/09/16/mitraclip-in-the-real-world-mid-term-progress/. Acesso em: 18 ago. 2020.

FISCHLER, M. A.; BOLLES, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. **Communications of the ACM**, v. 24, n. 6, p. 381-395, 1981.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. *In:* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. **Proceedings** […]. IEEE CVPR, 2016, p. 770-778.

HINTERSTOISSER, S.; LEPETIT, V.; ILIC, S.; HOLZER, S.; BRADSKI, G.; KONOLIGE, K.; NAVAB, N. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. *In*: Asian Conference on Computer Vision (ACCV), 2012.

KAISER, M.; JOHN, M.; HEIMANN, T.; BROST, A.; NEUMU, T.; ROSE, G. 2D/3D Registration of TEE Probe from Two Non-orthogonal C-arm Directions. *In*: Medical Image Computing and Computer-assisted Intervention: MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention, 2014. **Proceedings** […], MICCAI, 2014, v. 17, n. Pt 1, p. 283-290.

KEHL, W.; MANHARDT, F.; TOMBARI, F.; ILIC, S.; NAVAB, N. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2017, Venice. **Proceedings** […]. Venice: IEEE, 2017. p. 1530-1538.

KRULL, A.; BRACHMANN, E.; MICHEL, F.; YANG, M. Y.; GUMHOLD, S.; ROTHER, C. Learning Analysis-By-Synthesis for 6D Pose Estimation in RGB-D Images. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2015, Santiago. **Proceedings** […]. Santiago: IEEE, 2015. p. 954-962.

KÜGLER, D.; SEHRING, J.; STEFANOV, A.; STENIN, I.; KRISTIN, J.; KLENZNER, T.; SCHIPPER, J.; MUKHOPADHYAY, A. i3PosNet: instrument pose estimation from X-ray in temporal bone surgery. **International Journal of Computer Assisted Radiology and Surgery**, v. 15, p. 1137–1145, 2020.

LEPETIT, V.; MORENO-NOGUER, F.; FUA, P. EPnP: An Accurate O(n) Solution to the PnP Problem. **International Journal of Computer Vision**, v. 81, n. 2, p. 155-166, fev. 2009.

LI, Y.; WANG, G.; JI, X.; XIANG, Y.; FOX, D. DeepIM: Deep Iterative Matching for 6D Pose Estimation. **International Journal of Computer Vision**, v. 128, p. 657-678, 2020.

LI, Z.; WANG, G.; JI, X. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation. *In*: IEEE/CVF INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2019, Seoul. **Proceedings** […]. Seoul: IEEE, 2019. p. 7677-7686.

MIAO, S.; WANG, Z. J.; LIAO, R. A CNN Regression Approach for Real-time 2D/3D. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1352-1363, maio 2016.

POWELL, M. J. D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. **Computer Journal**, v. 7, n. 2, p. 155-162, 1964.

RAD, M.; LEPETIT, V. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2017, Venice. **Proceedings** […]. Venice: IEEE, 2017. p. 3848-3856.

ROTHGANGER, F.; LAZEBNIK, S.; SCHMID, C.; PONCE, J. 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints. **International Journal of Computer Vision**, v. 66, p. 231-259, 2006.

SAHIN, Caner; GARCIA-HERNAND, Guillermo; SOCK, Juil; KIM, Tae-Kyun. A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators. **Image and Vision Computing**, v. 96, n. 103898, abr. 2020

SANARFLIX. **Cardiologia**. 2019. Disponível em: https://www.sanarmed.com/cardiologia-.... Acesso em: 18 ago. 2020.

SHEROUSE, G. W.; NOVINS, K.; CHANEY, E. L. Computation of digitally reconstructed radiographs for use in radiotherapy treatment design. **International Journal of Radiation Oncology*Biology*Physics**, v. 18, n. 3, p. 651-658, 1990.

SONG, Chen; SONG, Jiaru; HUANG, Qixing. HybridPose: 6D Object Pose Estimation Under Hybrid Representations. *In*: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 19876476., 2020, Seattle. **Proceedings** […]. Seattle: IEEE/CVF, 2020. p. 428-437.

SUNDERMEYER, M.; MARTON Z. C.; DURNER, M.; BRUCKER, M.; TRIEBEL, R. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. *In*: EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV), 15., 2018, Munich. **Proceedings** […]. Munich: Springer International Publishing, 2018. v. 11210. p. 712-729.

TJADEN, H.; SCHWANECKE, U.; SCHÖMER, E. Real-Time Monocular Pose Estimation of 3D Objects Using Temporally Consistent Local Color Histograms. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2017, Venice. **Proceedings** […]. Venice: IEEE, 2017. p. 124-132.

WANG, S.; CLARK, R.; WEN, H.; TRIGONI, N. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. *In:* IEEE International Conference on Robotics and Automation (ICRA), 2017. **Proceedings** […]. IEEE ICRA, 2017, p. 2043-2050.

WANG, C.; XU, D.; ZHU, Y.; MARTÍN-MARTÍN, R.; LU, C.; FEI-FEI, L.; SAVARESE, S. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. *In*: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2019, Long Beach. **Proceedings** […]. Long Beach: IEEE, 2019. p. 3338-3347.

XIANG, Yu; SCHMIDT, Tanner; NARAYANAN, Venkatraman; FOX, Dieter. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *In*: ROBOTICS: SCIENCE AND SYSTEMS, XIV, 2018, Pittsburgh. **Proceedings** […]. Pittsburgh: Carnegie Mellon University, 2018. Disponível em: http://www.roboticsproceedings.org/rss14/p19.htmlt. Acesso em: 12/10/2020.

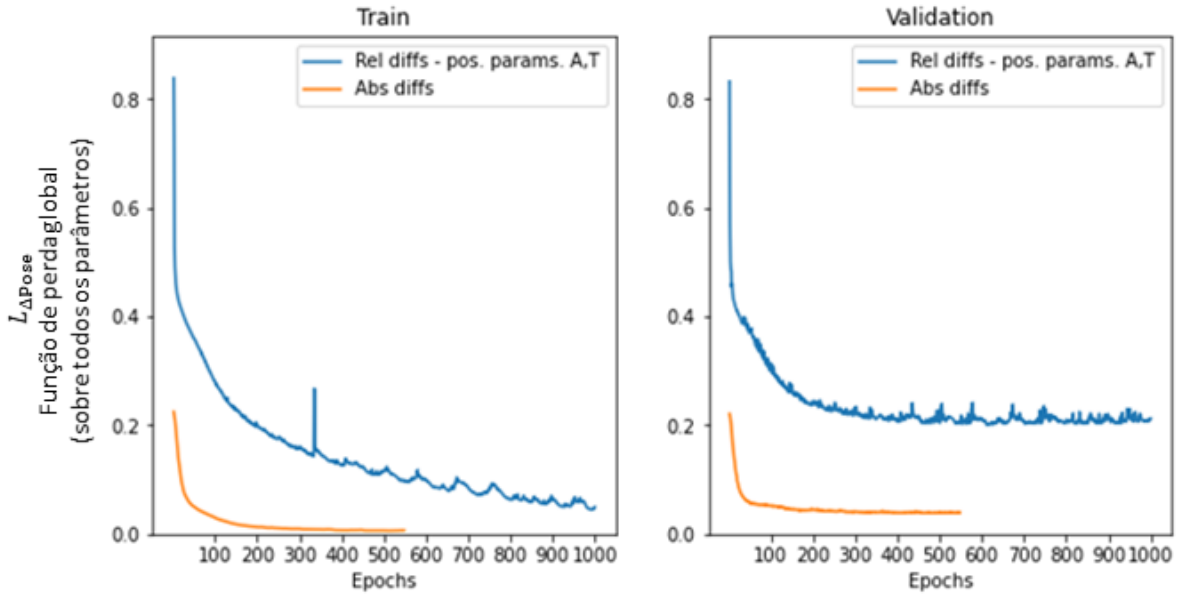
ZEILER, M. D.; TAYLOR, G. W.; FERGUS, R. Adaptive deconvolutional networks for mid and high level feature learning. *In:* International Conference on Computer Vision, 2011. **Proceedings** […]. ICCV, 2011, p. 2018-2025.

APÊNDICE A – EXPERIMENTOS COM DELTAPOSENET UTILIZANDO DIFERENÇA DE POSE RELATIVA

Foi analisada uma versão da DeltaPoseNet para prever a diferença relativa (ou seja, com sinal) entre as imagens de entrada. Para estes experimentos, a *ground-truth* tinha valores normalizados na faixa $[-1,1]$ (seguindo o mesmo procedimento de normalização descrito na Seção 2.3.1.2).

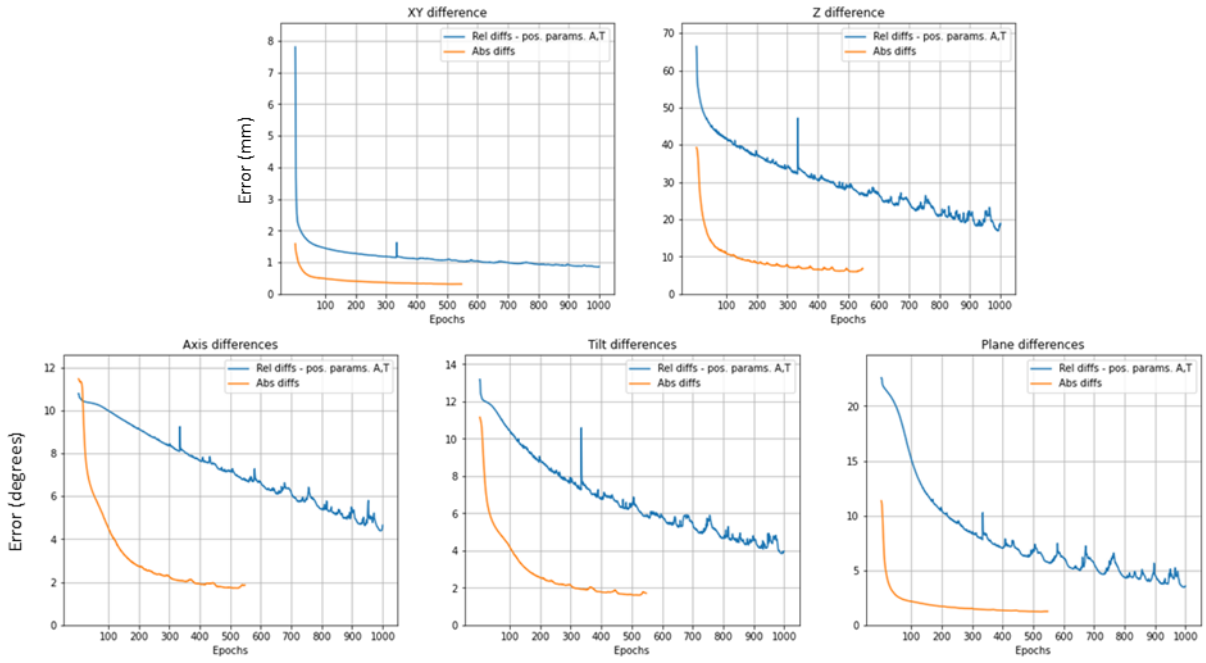
A partir da Figura 32, da Figura 33 e da Figura 34, pode-se ver que o desempenho da versão absoluta foi consideravelmente melhor, e é por isso que optou-se por prosseguir com esta versão. Uma dificuldade básica enfrentada pela versão relativa é que o domínio de cada parâmetro é duplicado. Investigar formas de melhorar o desempenho da versão relativa seria interessante em trabalhos futuros, mas para o escopo deste projeto, ter um estimador de pose absoluta já forneceu resultados e análises interessantes.

Figura 32 – Curvas da função de perda de treino e teste para DeltaPoseNet treinada considerando a diferença relativa de pose entre as imagens de entrada (azul) e considerando a diferença de pose absoluta (laranja) (conjunto de dados contendo todos os modelos do MitraClip)



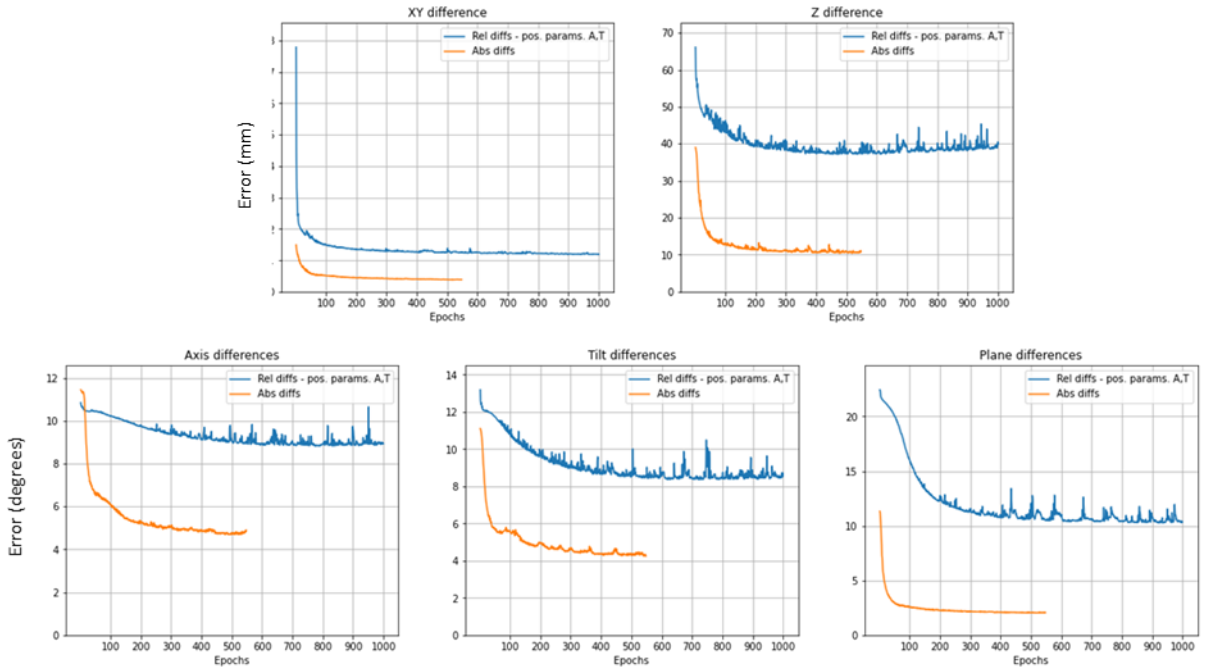
Fonte: Produção da própria autora.

Figura 33 – Erros durante o treinamento para os diferentes parâmetros de pose sobre o conjunto de **treinamento**, comparando as versões absolutas e relativas da DeltaPoseNet



Fonte: Produção da própria autora.

Figura 34 - Erros durante o treinamento para os diferentes parâmetros de pose sobre o conjunto de **teste**, comparando as versões absolutas e relativas da DeltaPoseNet



Fonte: Produção da própria autora.