

INSTITUTO FEDERAL DO ESPÍRITO SANTO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

ANTÔNIO CARLOS DURÃES DA SILVA

**UMA COMPARAÇÃO DE ARQUITETURAS BASEADAS EM U-NET NA
ESTIMATIVA DE PROFUNDIDADE MONOCULAR**

Serra
2024

ANTÔNIO CARLOS DURÃES DA SILVA

**UMA COMPARAÇÃO DE ARQUITETURAS BASEADAS EM U-NET NA
ESTIMATIVA DE PROFUNDIDADE MONOCULAR**

Dissertação de Mestrado apresentada ao Programa
de Pós-Graduação em Computação Aplicada do
Instituto Federal do Espírito Santo para obtenção
do Título de Mestre em Computação Aplicada.

Orientador(a): Prof^a Dra. Kelly Assis De Souza
Gazolli

Serra
2024

Ao meu querido pai, Gирnaldo Alexandre da Silva (in memoriam), que em vida me incentivou, vibrou e se alegrou junto comigo com minhas conquistas acadêmicas e, após sua partida, tornou-se minha principal motivação para concluir esse ciclo de minha vida estudantil.

AGRADECIMENTOS

Aos meus pais, Gирnaldo e Marlete, por, à sua maneira, me incentivarem a buscar formação acadêmica, valorizar os estudos e por sempre terem lutado para que eu pudesse me dedicar.

À minha orientadora, Prof.^a Dr.^a Kelly Assis de Souza Gazolli, não só por me fornecer todo apoio necessário para o desenvolvimento desse trabalho, mas também por acreditar que eu seria capaz de chegar até essa etapa de minha vida acadêmica e por não me deixar perder a esperança de chegar ao fim desse ciclo.

Ao meu amigo Jonathan Ribeiro, por compartilhar seus momentos de alegrias e adversidades, seja no âmbito acadêmico, quanto no pessoal e profissional. Essa presença foi um dos pilares ao qual me apoiei para não desistir do programa de mestrado.

Resumo

A estimativa de profundidade monocular é um problema de visão computacional que possui diversas aplicações que vão desde realidade aumentada até procedimentos cirúrgicos. Dada a semelhança entre as tarefas de segmentação e estimativa de profundidade monocular, além do bom desempenho da rede U-net e suas variações na tarefa de segmentação, este estudo tem como objetivo comparar o desempenho de variações das arquiteturas U-Net e UNet++, por meio do uso de diferentes redes como codificador, e da arquitetura TransUnet na estimativa de profundidade monocular. Os resultados alcançados no conjunto de dados NYU Depth V2 mostram que U-Net usando a rede CoaT-Lite (M) como codificador supera todas as outras abordagens avaliadas.

Palavras-chave: Estimativa de profundidade monocular. U-Net. UNet++. Transformers.

ABSTRACT

Monocular depth estimation is a computer vision problem which has diverse applications ranging from augmented reality to surgical procedures. Given the similarity between the segmentation and monocular depth estimation tasks, in addition to the good performance of the U-net network and its variations in the segmentation task, this study aims to compare the performance of variations of U-Net and UNet++ architectures, each one adopting a different network as encoder, and the TransUnet architecture in monocular depth estimation. The results achieved on the NYU Depth V2 dataset show that U-Net using the CoaT-Lite (M) network as encoder outperforms all other evaluated approaches.

Keywords: *Monocular depth estimation. U-Net. UNet++. Transformers.*

LISTA DE FIGURAS

Figura 1 – Exemplo de imagens do conjunto NYU v2	9
Figura 2 – Representação da aplicação de uma operação de Convolução	12
Figura 3 – Arquitetura da rede VGG-19 BN	13
Figura 4 – Arquitetura simplificada da rede Inception-ResNet V2 e seu STEM . .	14
Figura 5 – Arquitetura do blocos “Inception A” e módulos de redução	15
Figura 6 – Arquitetura dos blocos “Inception-ResNet”	16
Figura 7 – Arquitetura da rede Xception	17
Figura 8 – Arquitetura da rede U-Net	18
Figura 9 – Arquitetura da rede UNet++	19
Figura 10 – Etapas de pré-processamento de uma entrada <i>Transformer</i>	20
Figura 11 – Arquitetura da rede <i>Transformer</i>	21
Figura 12 – Arquitetura da rede Vision Transformer	22
Figura 13 – Arquitetura geral dos codificadores Mixed Transformer	23
Figura 14 – Arquitetura geral das redes CoAtNets	25
Figura 15 – Arquitetura geral das redes CoaT	27
Figura 16 – Arquitetura da rede TransUnet	29
Figura 17 – Arquitetura geral da rede U-Net com codificador	35
Figura 18 – Arquitetura geral da rede UNet++ com codificador	36
Figura 19 – Resultados qualitativos de modelos U-Net com codificadores apenas CNN no conjunto NYU v2	43
Figura 20 – Mapas de calor de erro de modelos U-Net com codificadores apenas CNN no conjunto NYU v2	43
Figura 21 – Resultados qualitativos de modelos UNet++ com codificadores apenas CNN no conjunto NYU v2	44
Figura 22 – Mapas de calor de erro de modelos UNet++ com codificadores apenas CNN no conjunto NYU v2	44
Figura 23 – Resultados qualitativos de modelos U-Net com codificadores <i>Transformers</i> no conjunto NYU v2	45
Figura 24 – Mapas de calor de erro de modelos U-Net com codificadores <i>Transformer</i> no conjunto NYU v2	45
Figura 25 – Resultados qualitativos de modelos UNet++ com codificadores <i>Transformers</i> no conjunto NYU v2	46
Figura 26 – Mapas de calor de erro de modelos UNet++ com codificadores <i>Transformer</i> no conjunto NYU v2	46
Figura 27 – Resultados qualitativos dos melhores modelos no conjunto NYU v2 . .	47
Figura 28 – Mapas de calor de erro dos melhores modelos no conjunto NYU v2 . .	47

LISTA DE SIGLAS

BN	- Batch Normalization
CNN	- Convolutional Neural Network
EP	- Estimativa de profundidade
ILSVRC	- ImageNet Large Scale Visual Recognition Challenge
LIDAR	- Laser Imaging Detection and Ranging
MLP	- Multi-Layer Perceptron
RMSE	- Root Mean Square Error
RNN	- Recurrent Neural Network
ReLU	- Rectified Linear Unit
SSIM	- Structural Similarity Index Measure
VGG	- Visual Geometry Group

SUMÁRIO

1	INTRODUÇÃO	8
1.1	PROPOSTA DE TRABALHO	9
1.2	OBJETIVO GERAL	10
1.3	OBJETIVOS ESPECÍFICOS	10
1.4	PUBLICAÇÃO ASSOCIADA	10
1.5	ORGANIZAÇÃO DO TRABALHO	10
2	REFERENCIAL TEÓRICO	11
2.1	Redes Neurais Profundas	11
2.1.1	Redes Neurais Convolucionais	11
2.1.1.1	VGG-19 BN	12
2.1.1.2	Inception-ResNet V2	14
2.1.1.3	Xception	16
2.1.1.4	U-Net	18
2.1.1.5	UNet++	19
2.1.2	Redes Neurais Transformers	19
2.1.2.1	Mixed Transformer (B2)	22
2.1.3	Arquiteturas Híbridas	23
2.1.3.1	CoAtNet	23
2.1.3.2	CoaT	25
2.1.3.3	TransUnet	28
2.2	Trabalhos relacionados	29
3	METODOLOGIA	33
3.1	Arquiteturas Adotadas	33
3.2	Adaptação dos codificadores nas redes U-Net e UNet++	33
3.3	Arquitetura U-Net Adaptada	34
3.4	Arquitetura UNet++ Adaptada	35
3.5	Função de perda	36
3.6	Uso de pesos e limitação de resolução de entrada	37
4	EXPERIMENTOS, RESULTADOS E DISCUSSÕES	38
4.1	Base de dados	38
4.2	Detalhes de implementação	38
4.3	Avaliação	39
4.4	Resultados	39
4.4.1	Análise qualitativa – Resolução 224x224	42
4.4.2	Análise qualitativa e quantitativa – Resoluções maiores	46
5	CONCLUSÕES	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

A tarefa de estimativa de profundidade (EP) consiste em calcular um valor numérico que representa a distância entre um *pixel* de uma imagem e o seu observador (MERTAN; DUFF; UNAL, 2022). Essa atividade desempenha um importante papel em diversas aplicações, tais como, realidade aumentada (HUANG; SUN, 2020), identificação de proximidade com elementos em guiagem autônoma (XIAO et al., 2022) e até mesmo em procedimentos cirúrgicos auxiliados por computador (ITOH et al., 2021; LIU et al., 2020).

Existem dispositivos capazes de realizar a estimativa de profundidade, um dos mais utilizados é o sensor LIDAR (do inglês, *Laser Imaging Detection and Ranging*), juntamente com as câmeras estéreo (POGGI et al., 2020). No entanto, tais dispositivos são propensos a problemas mecânicos e complicações relacionadas à natureza das superfícies do ambiente, além do alto custo de aquisição (POGGI et al., 2020). Paralelamente às questões que envolvem o uso de dispositivos dedicados, há a oportunidade de construir aplicações de realidade aumentada para dispositivos móveis, sendo essa uma das principais motivações para estimativa de profundidade usando imagens obtidas por câmeras de baixo custo, que capturam fotografias monoculares (XIE et al., 2019).

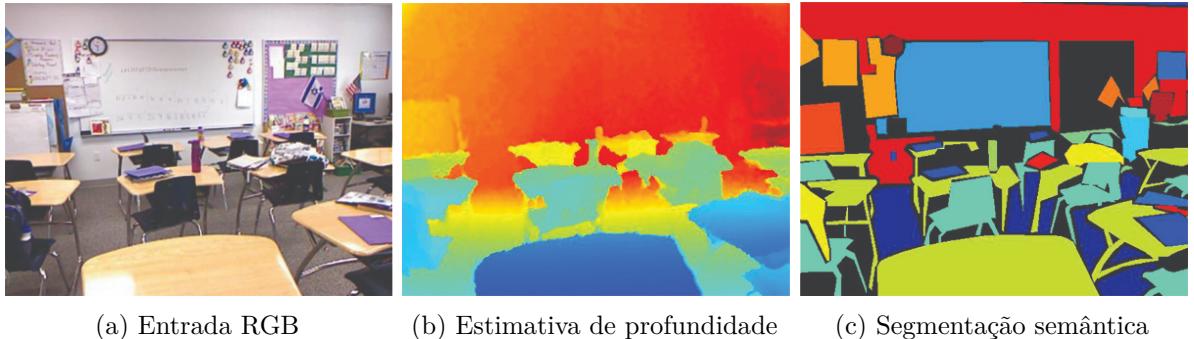
Nesse contexto, foram desenvolvidas diversas abordagens que realizam a estimativa de profundidade monocular. De acordo com Ming et al. (2021), inicialmente, essa tarefa era executada por meio de técnicas primitivas que se baseavam em pistas de profundidade, como foco, sombra e formas (JUNG et al., 2009; HAN; HONG, 2011), posteriormente por modelos probabilísticos baseados em aprendizado de máquina (YAN et al., 2019; TSENG; LAI, 2011) e por fim, por métodos baseados em aprendizado profundo, como redes neurais artificiais.

Assim como na tarefa de estimativa de profundidade, a segmentação semântica vem sendo aprimorada por meio do emprego de redes neurais artificiais (ULKU; AKAGÜNDÜZ, 2022), sendo a arquitetura U-Net e suas variantes frequentemente aplicadas (ZHOU et al., 2020; HUANG et al., 2020; SIDDIQUE et al., 2021). Inicialmente projetada com foco na segmentação de imagens médicas (RONNEBERGER; FISCHER; BROX, 2015), a rede U-Net, que é do tipo codificador-decodificador, recebeu diversas variações (PUNN; AGARWAL, 2022) e passou a ser utilizada na solução de outros problemas de visão computacional (HE; FANG; PLAZA, 2020; CAO; ZHANG, 2020).

Segundo Ming et al. (2021), tanto a segmentação semântica quanto a estimativa de profundidade são classificações no nível de *pixel*, o que possibilita o compartilhamento das características extraídas da imagem entre ambas tarefas, sendo necessários apenas dois módulos de saída distintos, um para cada finalidade. Na Figura 1 estão exemplos de saídas de ambas tarefas, sendo que a estimativa de profundidade (b) demarca com cores

frias as superfícies mais próximas da câmera, e a segmentação (c) demarca cada classe (cadeira, mesa, quadro, parede) de objetos com uma cor única sem oferecer informações sobre a distância.

Figura 1 – Exemplo de entrada, saída de estimativa de profundidade e segmentação semântica do conjunto de dados de ambientes internos (*indoor*) NYU Depth V2



Fonte: Extraído de Silberman et al. (2012)

O aprendizado autossupervisionado têm sido frequentemente aplicado na estimativa de profundidade, alcançando resultados consideráveis. No entanto, os métodos existentes utilizam redes de arquiteturas complexas, que dependem de módulos para reconstrução e discriminação da entrada ou para estimativa de pose (GODARD et al., 2019; SHU et al., 2020; PILLAI; AMBRUŞ; GAIDON, 2019). Além disso, essa abordagem de aprendizado necessita de uma etapa de extração de padrões das entradas para gerar uma saída esperada a ser usada como profundidade verdadeira, uma vez que esse tipo de aprendizado dispõe apenas de nenhum dado rotulado ou de uma base parcialmente rotulada.

1.1 PROPOSTA DE TRABALHO

Dada a semelhança entre as tarefas de segmentação e de estimativa de profundidade, este trabalho tem como principal objetivo verificar o desempenho de combinações de arquiteturas baseadas em U-Net aplicadas à estimativa de profundidade de imagens monoculares. Foram selecionados codificadores bem estabelecidos na literatura para extração de características de imagens, são eles: CoAtNet, CoAT, Mixed Transformer B2 (abreviado como MiT-B2), Inception ResNet V2, VGG-19, Xception. Por fim, além da U-Net original, foram selecionadas duas de suas variantes: TrasUnet e UNet++.

Os experimentos foram realizados utilizando o aprendizado supervisionado e a base de dados NYU Depth V2 (SILBERMAN et al., 2012), mais especificamente um subconjunto de 50 mil imagens, conforme proposto por Alhashim e Wonka (2018).

1.2 OBJETIVO GERAL

O objetivo principal deste trabalho é verificar a aplicabilidade de modelos de redes neurais baseados em U-Net e aprendizado supervisionado na estimativa de profundidades em imagens monoculares.

1.3 OBJETIVOS ESPECÍFICOS

1. Implementar U-Net e UNet++ de modo a suportar os codificadores a serem utilizados.
2. Treinar modelos com a partição de treinamento da base NYU Depth V2.
3. Aplicar modelos treinados para avaliar partição de teste da base NYU Depth V2.
4. Coletar resultados na partição de teste.
5. Comparar os resultados obtidos com os de trabalhos semelhantes.

1.4 PUBLICAÇÃO ASSOCIADA

Artigo publicado nos anais do Workshop de Visão Computacional (WVC 2023).

SILVA, Antônio Carlos Durães da; GAZOLLI, Kelly Assis de Souza. Comparing U-Net based architectures in monocular depth estimation. In: WORKSHOP DE VISÃO COMPUTACIONAL (WVC), 18. , 2023, São Bernardo do Campo/SP. Anais do XVIII Workshop de Visão Computacional (WVC 2023), 2023. p. 48-53.

Link para acesso à publicação:

<https://sol.sbc.org.br/index.php/wvc/article/view/27531>

1.5 ORGANIZAÇÃO DO TRABALHO

Os próximos capítulos deste trabalho seguem a seguinte organização:

No Capítulo 2 são abordados conceitos relativos a redes neurais profundas e trabalhos relacionados. No Capítulo 3 é apresentada a metodologia utilizada, detalhando-se os modelos (codificadores e decodificadores) e a base de dados utilizada. O Capítulo 4 descreve a execução dos experimentos e apresenta os resultados alcançados, comparando-os com trabalhos semelhantes. Por fim, no Capítulo 5, são apresentados a conclusão e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo são abordadas algumas arquiteturas de redes neurais profundas utilizadas em problemas de visão computacional, além dos trabalhos relacionados.

2.1 Redes Neurais Profundas

Nos últimos anos, a aprendizagem profunda tem se destacado como uma abordagem computacional popular no campo do aprendizado de máquina, alcançando resultados excepcionais em uma variedade de tarefas cognitivas complexas (TAYE, 2023). As redes neurais profundas são caracterizadas por possuírem múltiplas camadas ocultas entre as camadas de entrada e de saída. Ao longo do tempo, várias configurações de redes neurais profundas foram propostas, tais como, as Redes Neurais Convolucionais e as Redes Neurais Transformers.

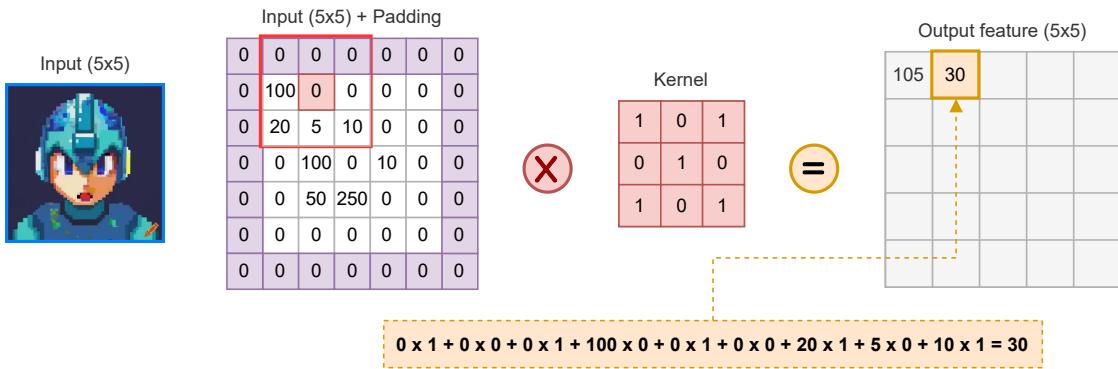
2.1.1 Redes Neurais Convolucionais

As redes neurais convolucionais (*Convolutional Neural Networks - CNNs ou ConvNets*) são um tipo específico de rede neural artificial originalmente projetada para tarefas de visão computacional, como detecção de objetos, classificação e segmentação de imagens (LECUN; BENGIO; HINTON, 2015). Segundo LeCun, Bengio e Hinton (2015), as CNNs são capazes de processar dados representados por matrizes não apenas bidimensionais, como imagens e espectogramas de áudio, mas também unidimensionais, como textos, e tridimensionais, como vídeos.

A principal operação para esse tipo de rede neural é a convolução, que utiliza uma matriz filtro (também chamada de *kernel*) aplicada sobre uma matriz de entrada. Por exemplo, a entrada para a convolução pode ser uma matriz 2D de dimensões NxM, com valores variando de 0 a 255, representando um dos canais de uma imagem com largura N e altura M *pixels*. A convolução é executada ao aplicar o filtro sobre todos os valores da matriz de entrada com um determinado preenchimento (chamado de *padding*) e um valor de passo (*stride*).

Na Figura 2 é ilustrada a aplicação de uma convolução 2D em uma imagem de 5x5 *pixels*, com um filtro 3x3 (em vermelho), passo 1 e preenchimento de 1 *pixel* (blocos roxos). Cada valor da entrada e seus valores adjacentes são multiplicados pelo valor de mesma posição no filtro. Por fim, é realizada a soma desses resultados para formar os valores do mapa de característica de saída.

Figura 2 – Representação da aplicação de uma operação de Convolução



Fonte: Elaborado pelo autor (2024), sendo a imagem de entrada gerada no site Craiyon, de IA generativa, com o *prompt* “megaman pixel art”

Outro componente tão crucial para as CNNs quanto as convoluções são as funções de ativação, que permitem às redes neurais agregar não linearidade e identificar padrões mais complexos (LECUN; BENGIO; HINTON, 2015). Cada função de ativação transforma os valores de entrada de acordo com sua equação específica. Um exemplo é a função ReLU (*Rectified Linear Unit*), descrita na Equação 1, que retorna o valor de entrada apenas se for maior que zero; caso contrário, retorna zero (XU et al., 2020).

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (1)$$

2.1.1.1 VGG-19 BN

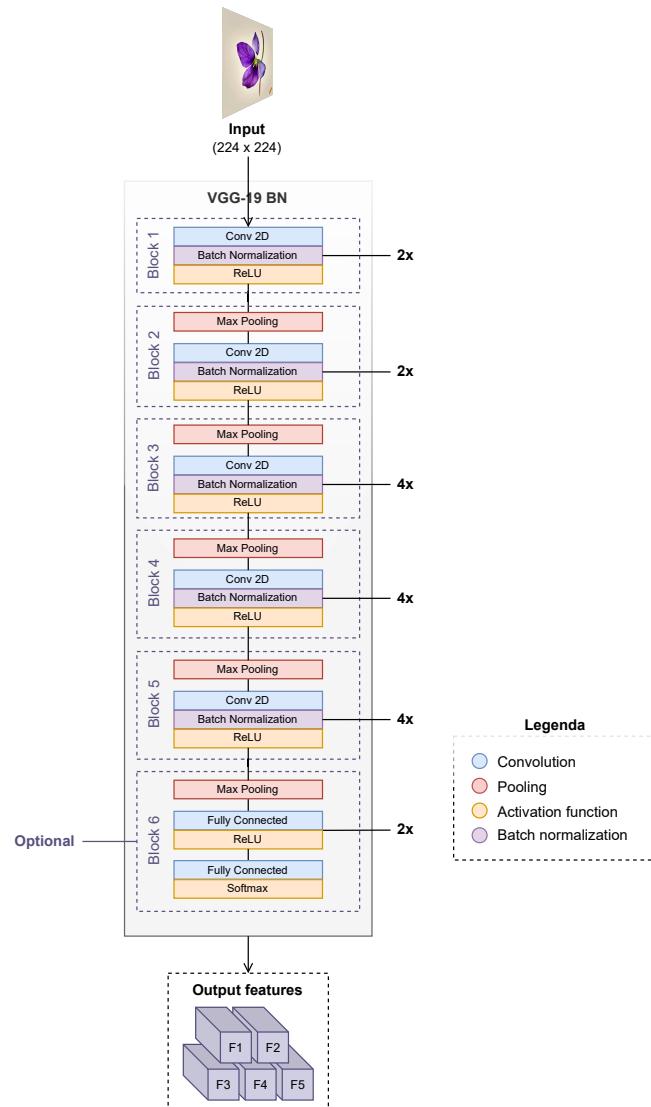
Desenvolvida pelo Grupo de Geometria Visual (*Visual Geometry Group - VGG*) da Universidade de Oxford, a VGG-19 é uma rede neural convolucional composta por 19 camadas, das quais 16 são de convolução 2D e 3 são camadas densas, ou seja, totalmente conectadas (SIMONYAN; ZISSERMAN, 2015). A VGG-19 é uma variação da popular VGG-16, que é amplamente reconhecida pelo sucesso em tarefas de detecção de objetos e classificação de imagens, tendo vencido o *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) (SIMONYAN; ZISSERMAN, 2015).

A Figura 3 apresenta a representação gráfica da arquitetura VGG-19. A rede recebe como entrada imagens com dimensões de 224 *pixels* de largura e altura, que são processadas por uma sequência de camadas de convolução com filtros de dimensão 3x3. Cada sequência de convolução é seguida por uma operação de agrupamento máximo (*max pooling*), que reduz a resolução, aumentando a invariância a detalhes da entrada e diminuindo o número de parâmetros a serem processados pelas camadas subsequentes. Ao final, a rede utiliza três

camadas totalmente conectadas para combinar as características extraídas nas camadas anteriores, sendo as duas primeiras compostas por 4096 canais e a última por 1000 canais, correspondentes ao número de classes presentes no desafio ILSVRC. Como a rede foi projetada para tarefas de classificação, a última camada é uma função *softmax*, que converte a entrada em uma distribuição de probabilidades para cada uma das mil classes.

A principal diferença da arquitetura VGG-19 BN em relação à versão tradicional da VGG-16 é a adição de uma sequência de convolução 2D seguida de ReLU nos blocos 3 a 5, além da aplicação da normalização em lote após cada operação de convolução 2D, justificando a sigla BN (*batch normalization*) em seu nome.

Figura 3 – Arquitetura da rede VGG-19 BN



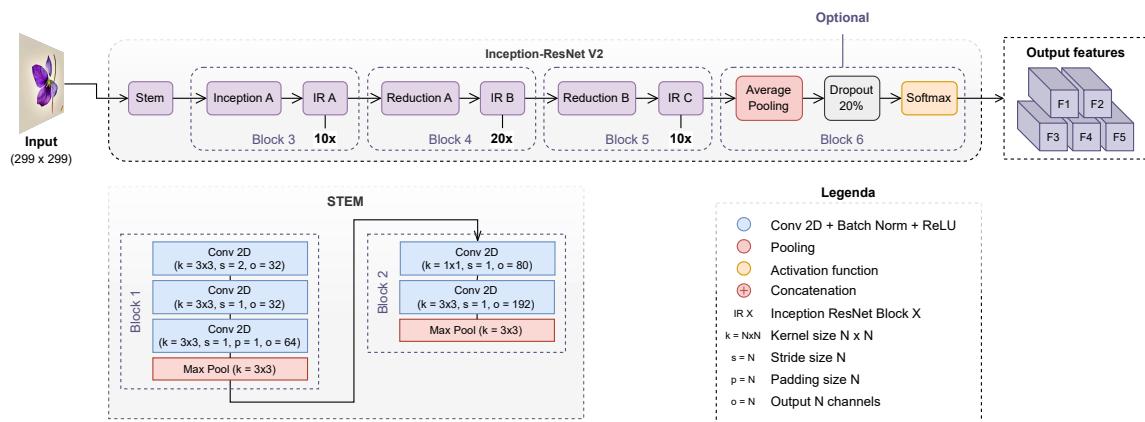
Fonte: Elaborado pelo autor (2024)

2.1.1.2 Inception-ResNet V2

O desempenho significativo e semelhante das redes Inception V3, proposta pelos autores da Google (SZEGEDY et al., 2015) e ResNet, proposta por pesquisadores da Microsoft (HE et al., 2016), no desafio ILSVRC 2015, fez com que os pesquisadores da Google cogitassem os possíveis benefícios de combinar conexões residuais com a arquitetura Inception (SZEGEDY et al., 2017). Assim, surgiram as arquiteturas de rede Inception-ResNet-v1 e v2, que diferem na configuração de hiper-parâmetros, no módulo inicial e na convergência mais rápida da versão 2.

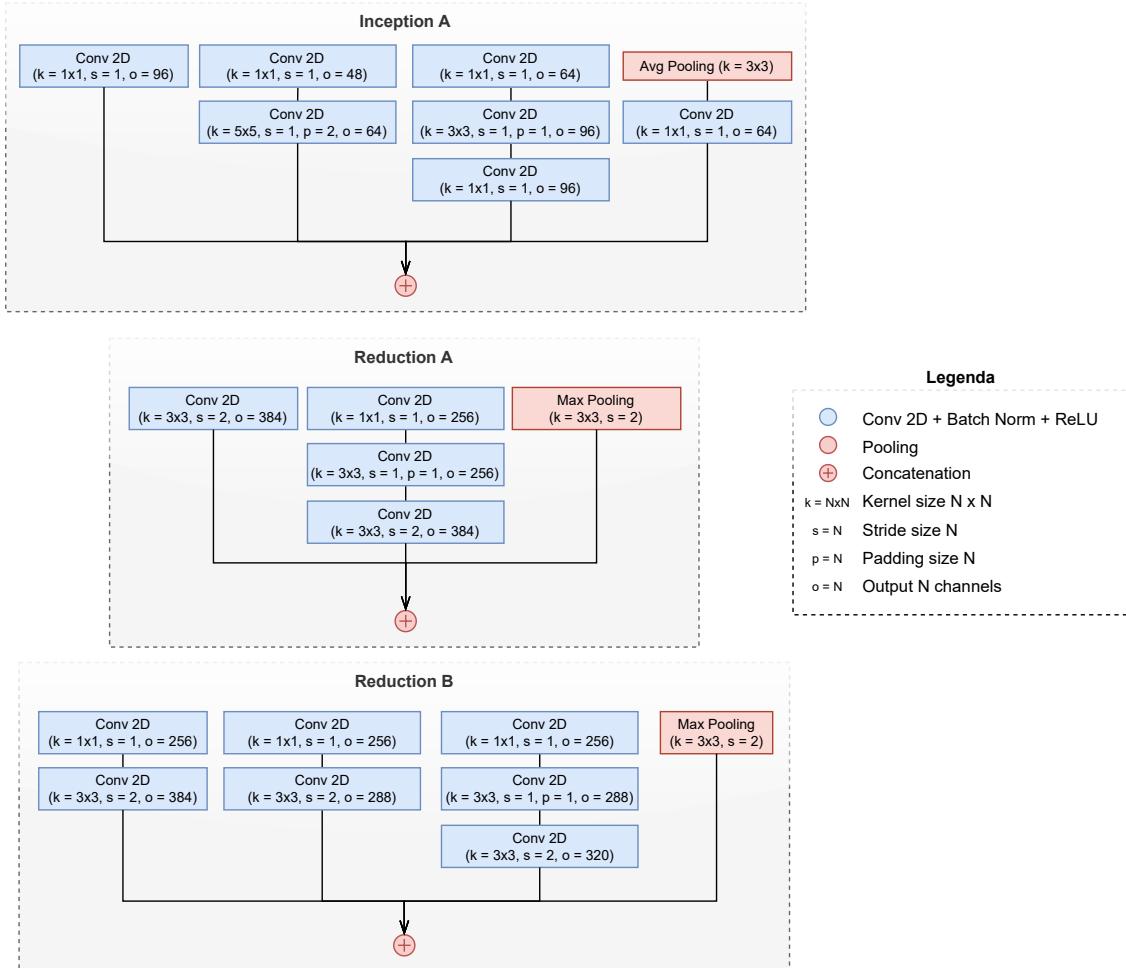
A Figura 4 mostra a arquitetura simplificada com todos os módulos da Inception-ResNet. Após receber a entrada de 299 *pixels* de largura e altura, a rede encaminha a imagem para o módulo inicial, Stem, encarregado por aplicar uma sequência de convoluções iniciais antes de repassar a saída para os módulos Inception-ResNet e Reduction (módulo de redução). Em seguida, o módulo Inception, representado na Figura 5, concatena a saída de convoluções com tamanhos distintos de filtros. Esse processo possibilita que a rede extraia representações da imagem em diferentes escalas, de características mais refinadas às mais amplas. A saída de cada módulo Inception-ResNet, representados na Figura 6, é recebida por um módulo de redução, responsável por aumentar o número de canais e reduzir a dimensão da entrada. Por fim, uma operação de *dropout* elimina aleatoriamente 20% dos neurônios da rede para regularizá-la antes de aplicar a função final de ativação.

Figura 4 – Arquitetura simplificada da rede Inception-ResNet V2 e seu STEM



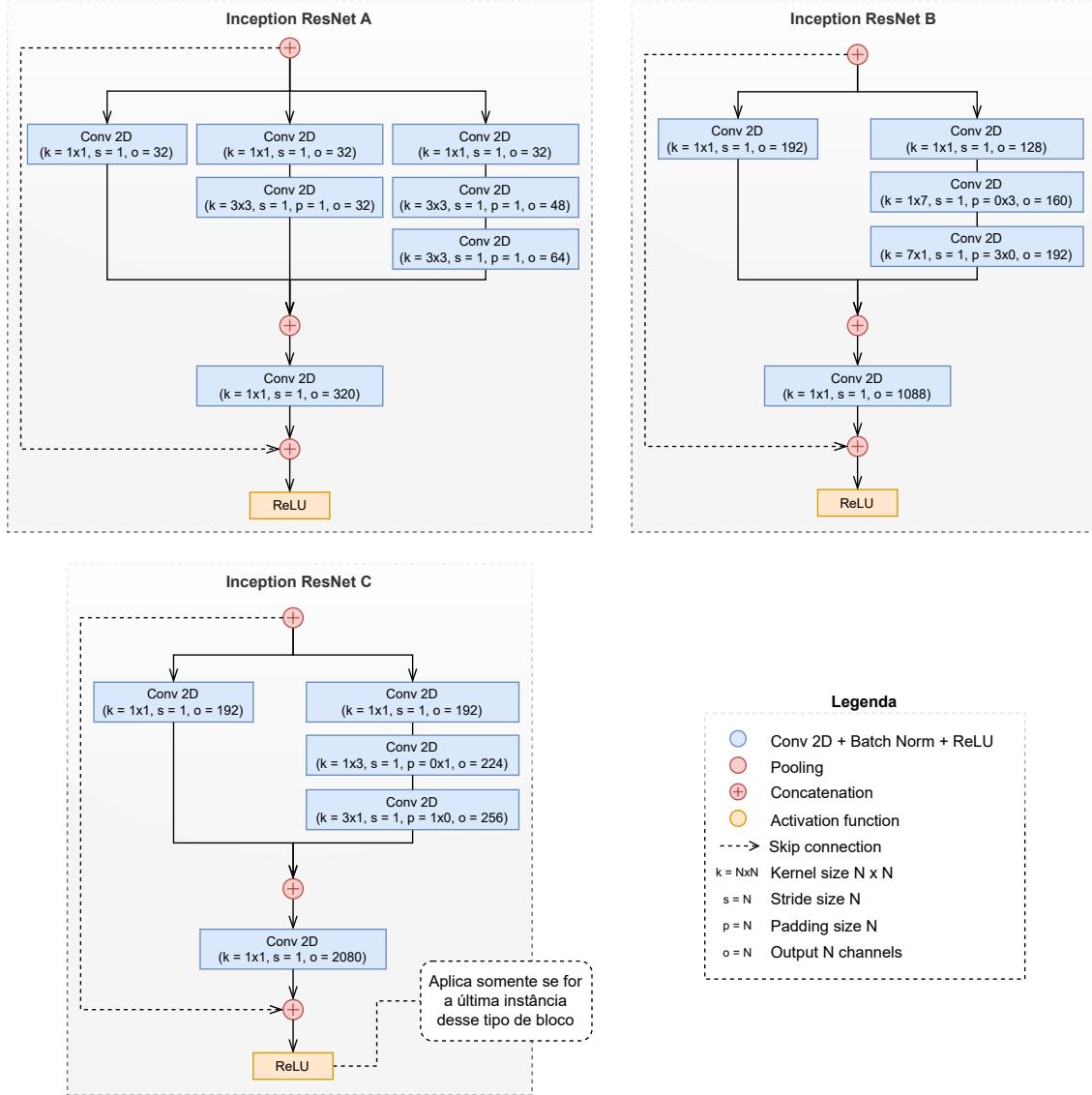
Fonte: Elaborado pelo autor (2024)

Figura 5 – Arquitetura do blocos “Inception A” e módulos de redução



Fonte: Elaborado pelo autor (2024)

Figura 6 – Arquitetura dos blocos “Inception-ResNet”



Fonte: Elaborado pelo autor (2024)

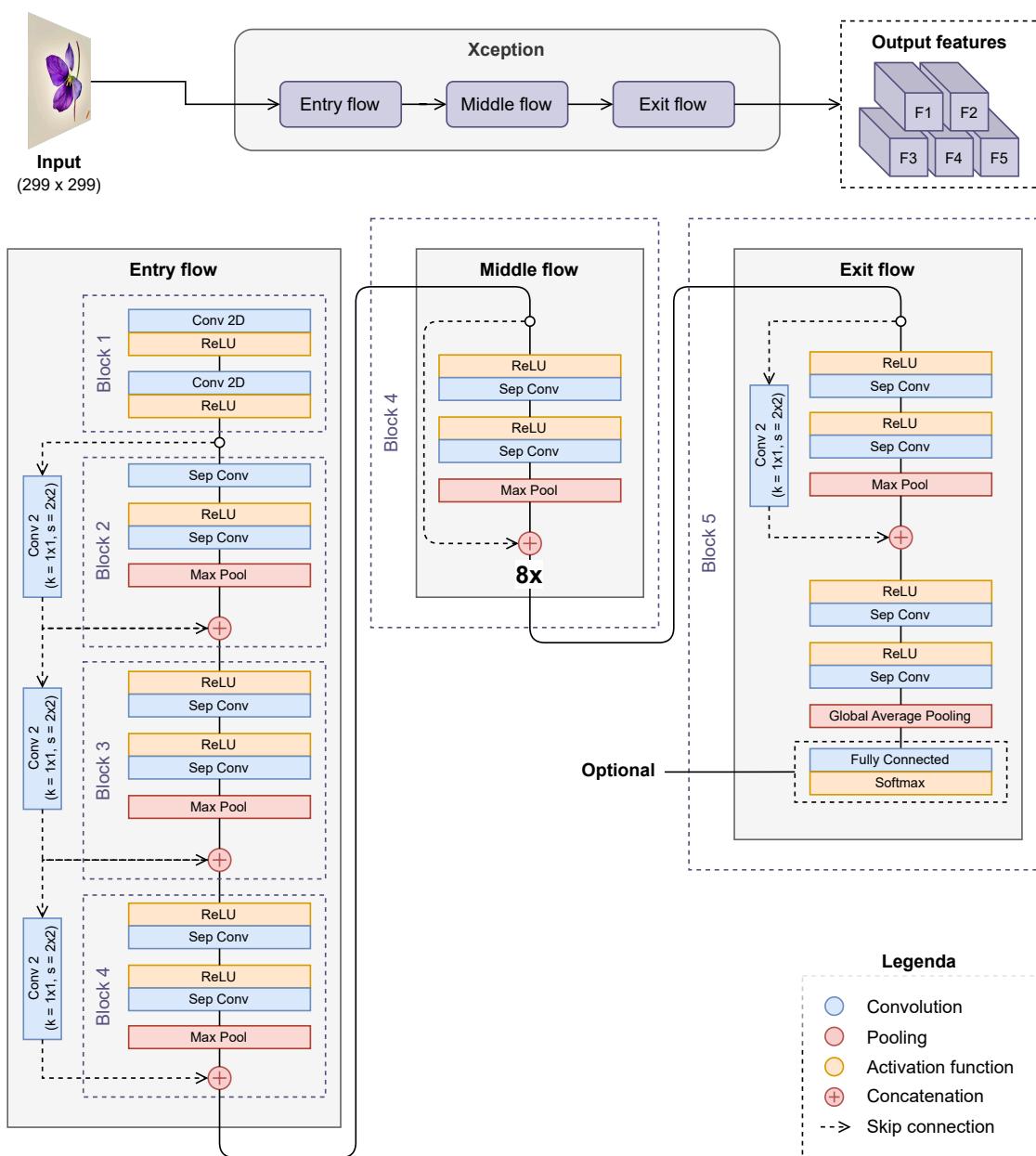
2.1.1.3 Xception

Inspirado nas melhorias do módulo Inception ao longo de suas três primeiras versões, o trabalho de Chollet (2017) traz como principal contribuição a arquitetura Xception, uma rede baseada em convoluções separáveis em profundidade (do inglês, “depthwise separable convolution”). A arquitetura Xception é estruturada em três partes: fluxo de entrada, intermediário (ou central) e o de saída, conforme a representação da Figura 7.

O fluxo de entrada utiliza duas sequências de convoluções tradicionais seguidas por ReLU para extrair as características de baixo nível e reduzir a dimensão da entrada, uma imagem colorida com 299 pixels de largura e altura. A saída resultante é então processada por

três blocos compostos por convolução separável 1x1, convolução separável 3x3, ReLU e agrupamento máximo antes de chegar ao fluxo intermediário. O fluxo intermediário aplica uma sequência de 8 blocos com ReLU, convolução separável 3x3 com 728 filtros e concatenação, a fim de extrair características de alto nível da entrada. Finalmente, para extrair características mais complexas, o fluxo de saída executa convoluções separáveis com números maiores de filtros (1024, 1536, 2048). Como a rede foi projetada para classificação de imagens, uma operação de agrupamento global e uma camada totalmente conectada convertem a saída para uma distribuição de probabilidade.

Figura 7 – Arquitetura da rede Xception



Fonte: Elaborado pelo autor (2024)

2.1.1.4 U-Net

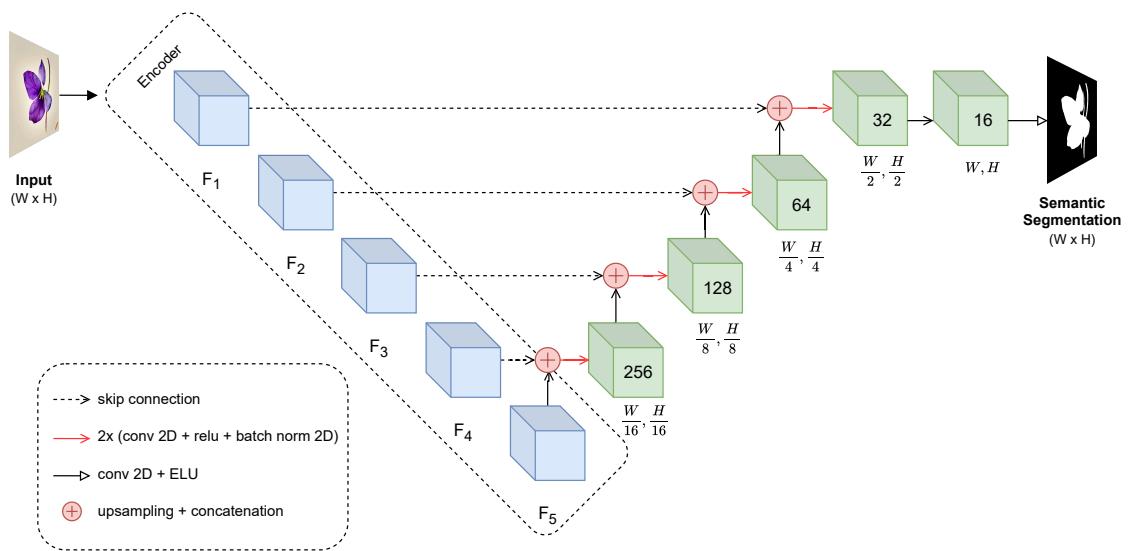
Ronneberger, Fischer e Brox (2015) propuseram a rede U-Net, uma arquitetura do tipo codificador-decodificador, para a segmentação semântica de imagens biomédicas. Uma das principais contribuições dos autores é a proposta de unir uma sequência de contração (usada para extrair informações contextuais) com uma sequência de expansão (utilizada para capturar informações de localização) por meio de conexões de salto (*skip connections*).

Na Figura 8 está a representação da arquitetura U-Net. O processo se inicia com o codificador, que recebe a imagem de entrada e extrai os mapas de características (representadas pelos blocos azuis) com diferentes dimensões e número de canais.

A concatenação dos dois mapas de características com maior número de canais (F_4 e F_5) passa por duas sequências de convolução 2D, função de ativação e normalização em lote, formando assim o primeiro bloco do decodificador (representado na cor verde).

Com exceção do primeiro bloco do decodificador, que é alimentado por duas saídas do codificador, todos os outros blocos subsequentes são alimentados pela concatenação da saída do bloco decodificador anterior com o mapa de características de um nível acima. À medida que o número de canais do decodificador é reduzido pela metade, a resolução dos mapas de características é dobrada, garantindo que, ao final, a saída da rede tenha a mesma dimensão da entrada.

Figura 8 – Arquitetura da rede U-Net

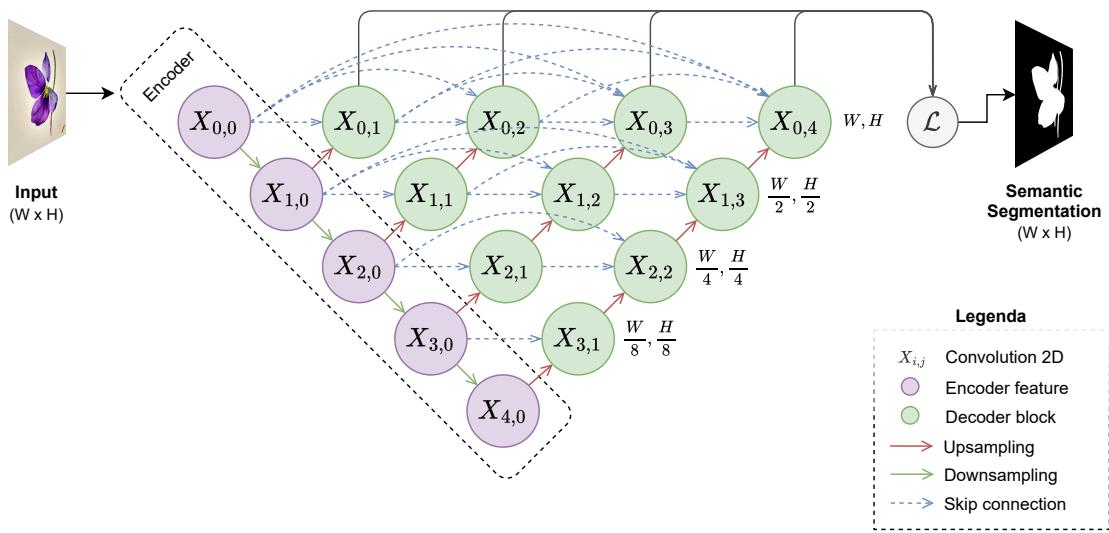


Fonte: Elaborado pelo autor (2024).

2.1.1.5 UNet++

O trabalho de Zhou et al. (2020) redesenhou as conexões de salto da U-Net, incluindo blocos densos de convolução a fim de melhorar o compartilhamento de informações entre codificador e decodificador por meio da concatenação de características de dimensões distintas. Além das mudanças na arquitetura, os autores propuseram o uso de um mecanismo de supervisão profunda (*deep supervision*) para podar camadas da rede treinada durante a inferência de dados e reduzir o tempo necessário para realizar essa tarefa. Essa nova abordagem foi denominada de UNet++.

Figura 9 – Arquitetura da rede UNet++



Fonte: Adaptado de Zhou et al. (2020)

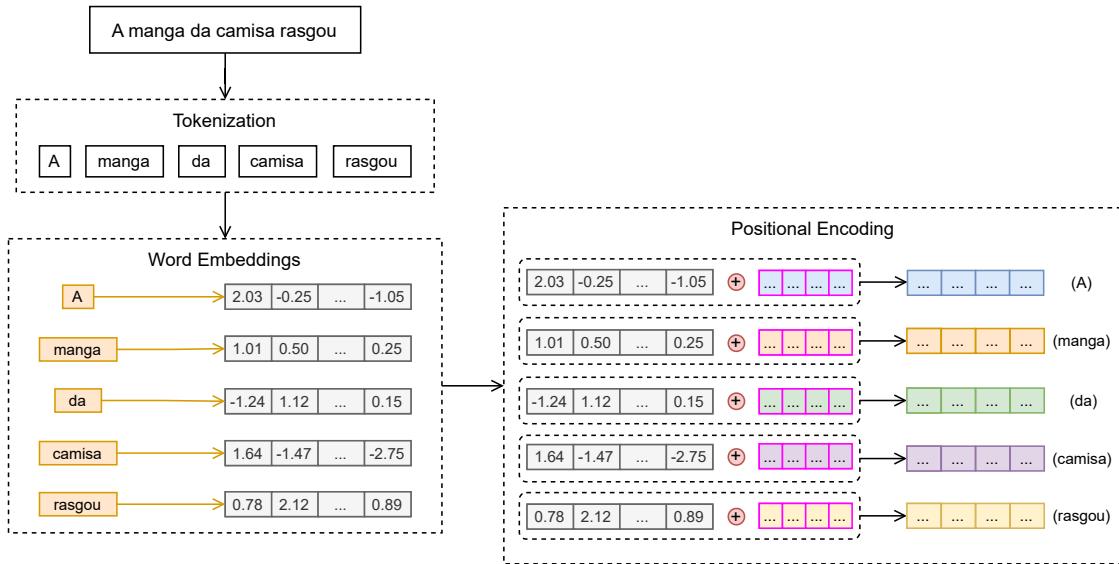
2.1.2 Redes Neurais Transformers

Apresentada por Vaswani et al. (2017), a arquitetura *Transformer* é uma rede neural proposta para tarefas do tipo sequência para sequência (*sequence-to-sequence* ou seq-2-seq), como tradução e sumarização de texto. Segundo os autores, um dos intuiitos para desenvolver esse tipo de arquitetura foi superar os desafios enfrentados pelas redes neurais recorrentes (*Recurrent Neural Network* - RNN), outro tipo de rede responsável por realizar as tarefas seq-2-seq.

Na Figura 10 estão ilustradas as etapas de preparo da entrada da rede *Transformer*: Tokenização (*tokenization*), *word embedding* e codificação posicional (*positional encoding*). A etapa de tokenização é responsável por converter o texto em um conjunto de *tokens*, usando algum separador como espaço em branco ou vírgula (WELBERS; ATTEVELDT; BENOIT, 2017). O segundo processo é utilizado para converter cada *token* em um vetor numérico, de forma a preservar as semelhanças semânticas e sintáticas entre eles

(GHANNAY et al., 2016). A etapa de codificação posicional tem o papel de manter a ordem de entrada nos *embeddings* (VASWANI et al., 2017), esse processo é crucial para que o modelo *Transformer* possa realizar o processamento dos *tokens* em paralelo sem perder as informações contextuais e semânticas fornecidas pela ordem das palavras em relação às outras.

Figura 10 – Etapas de pré-processamento de uma entrada *Transformer*

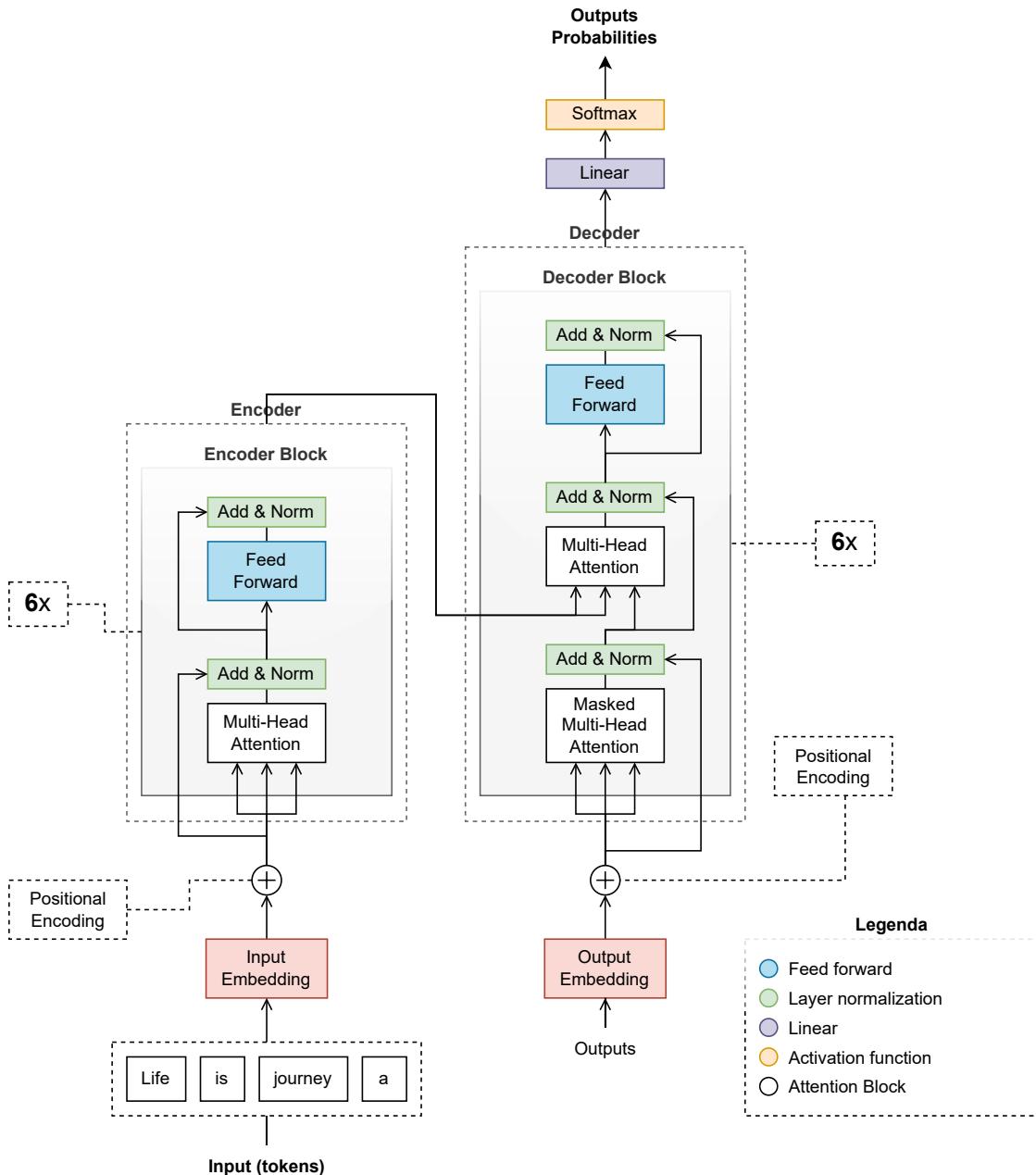


Fonte: Elaborador pelo autor (2024).

O principal componente dessa rede é o mecanismo de autoattenção (*self-attention*), recurso responsável por utilizar o contexto para atribuir significado a cada palavra de acordo com as outras em seu entorno (VASWANI et al., 2017). Por possuir esse mecanismo o modelo é capaz de distinguir o significado de frases com palavras homônimas, como no exemplo “Comi uma manga ontem” e “A manga da camisa rasgou”, em que a palavra “manga” assume significados distintos de acordo com o restante das palavras que compõem cada frase.

Na Figura 11 está a representação da arquitetura proposta por Vaswani et al. (2017). Seu codificador é composto por 6 blocos iguais e seu decodificador também possui o mesmo número de blocos.

Figura 11 – Arquitetura da rede *Transformer*



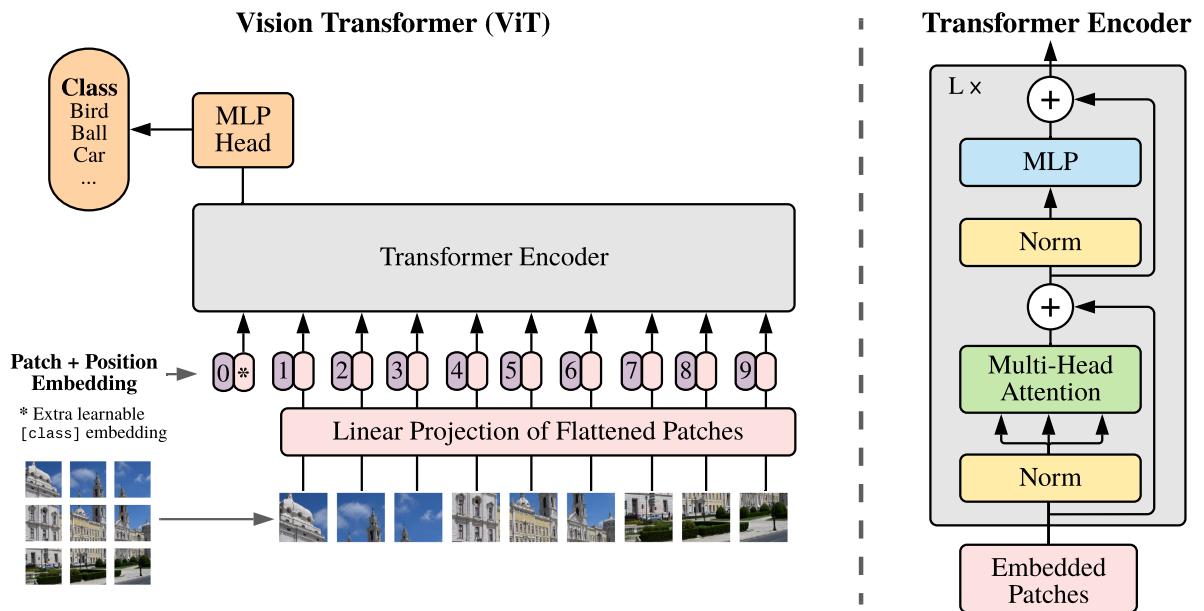
Fonte: Adaptado de Vaswani et al. (2017).

Dosovitskiy et al. (2021) foram responsáveis por adaptarem a arquitetura *Transformer* às aplicações da área de Visão Computacional. Por buscarem uma arquitetura para classificação de imagem, os pesquisadores descartaram o decodificador da arquitetura original, uma vez que seu uso era voltado para tradução. Além disso, o modelo *Vision Transformer* representa a imagem de entrada como um conjunto de pedaços (*patches*) de largura igual à altura, de forma semelhante aos *embeddings* de palavras utilizados pela

arquitetura original (DOSOVITSKIY et al., 2021).

Na Figura 12 está a representação simplificada da arquitetura e do fluxo da imagem de entrada. A entrada é dividida em 9 *patches* de dimensão NxN *pixels* (os autores utilizam o tamanho 16x16). Os *patches* são transformados para se tornarem vetores e os vetores resultantes passam pela codificação posicional antes de irem para o codificador *Transformer*. Após passar por todas as camadas de Atenção para entender a relevância de cada *patch* para os outros e para a tarefa, as informações de saída alimentam a cabeça de classificação multicamadas perceptron (*Multi-Layer Perceptron* - MLP) que retornará a classe que a rede identificou como a mais provável da imagem pertencer.

Figura 12 – Arquitetura da rede Vision Transformer



Fonte: Dosovitskiy et al. (2021).

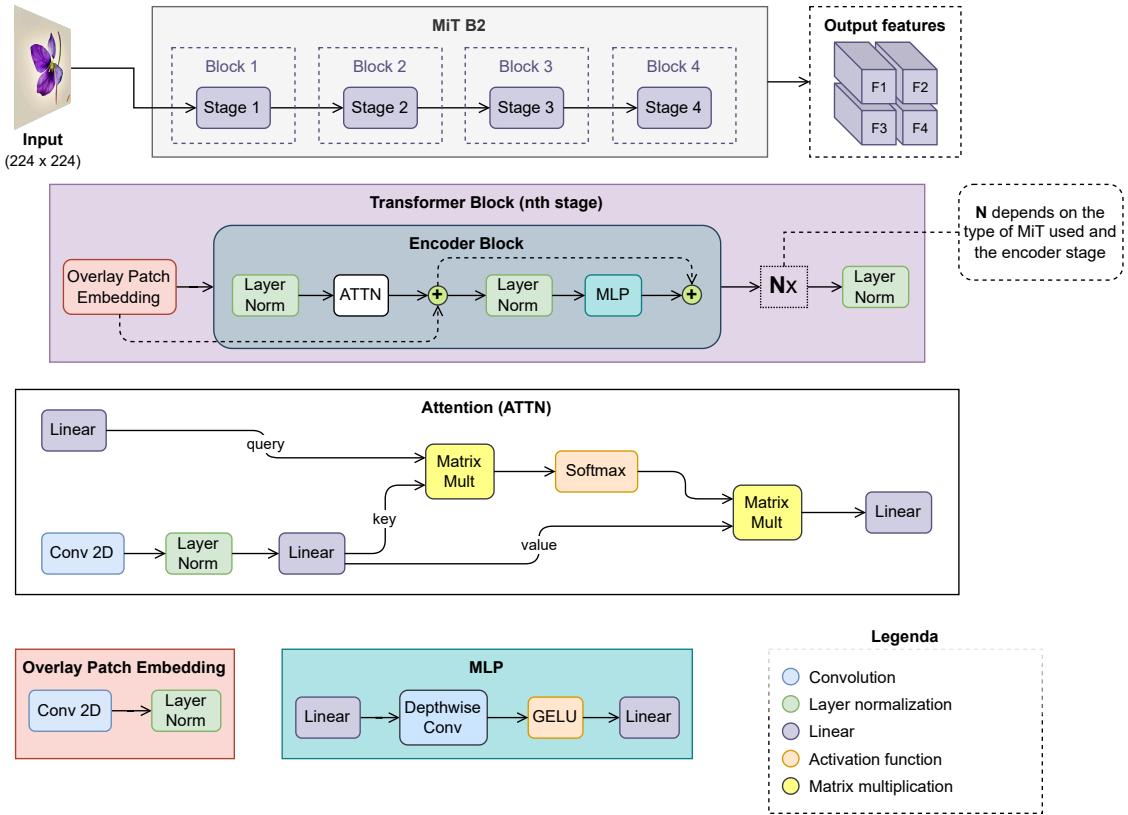
2.1.2.1 Mixed Transformer (B2)

Xie et al. (2021), autores da rede SegFormer, um *framework* de segmentação semântica, inspirados pela arquitetura *Transformer*, desenvolveram uma família de codificadores que faz uso de mecanismos de atenção, o conjunto Mixed Transformer. A família de codificadores é composta por seis versões, da B0 a B5, sendo a primeira mais simples, com menos parâmetros, e a última, mais complexa e com melhores resultados nos conjuntos de dados testados pelos autores (XIE et al., 2021).

A Figura 13 ilustra uma visão geral da família de codificadores da rede SegFormer. Qualquer variação do codificador é composta por 4 blocos *Transformers* (representados na cor roxa dentro do bloco *encoder*), cada um contém um módulo customizado de autoatenção, responsável por identificar a relação e dependência entre partes (*patches* ou *pixels*) da

imagem de entrada. Para usufruir de características locais e globais, a cada um desses blocos, a resolução da entrada é reduzida por um fator de 4, 8, 16 e 32.

Figura 13 – Arquitetura geral dos codificadores Mixed Transformer



Fonte: Elaborado pelo autor (2024)

2.1.3 Arquiteturas Híbridas

De acordo com Guo et al. (2022), as arquiteturas híbridas, que combinam *Transformers* e CNNs, conseguem integrar a capacidade dos mecanismos de atenção de capturar dependências de longo alcance com a habilidade das redes convolucionais de extrair informações locais. Neste trabalho, três redes se enquadram nessa categoria e serão detalhadas em sub-seções a seguir: CoAtNet, CoaT e TransUnet.

2.1.3.1 CoAtNet

Com o objetivo de aproveitar a capacidade das CNNs para extrair características locais por meio de convoluções e a habilidade das redes *Transformers* de capturar dependências globais e serem escaláveis para quantidades consideráveis de dados, Dai et al. (2021) propuseram a arquitetura CoAtNet. O nome da rede originou da junção de *convolution* (“Co”) e *self-attention* (“At”).

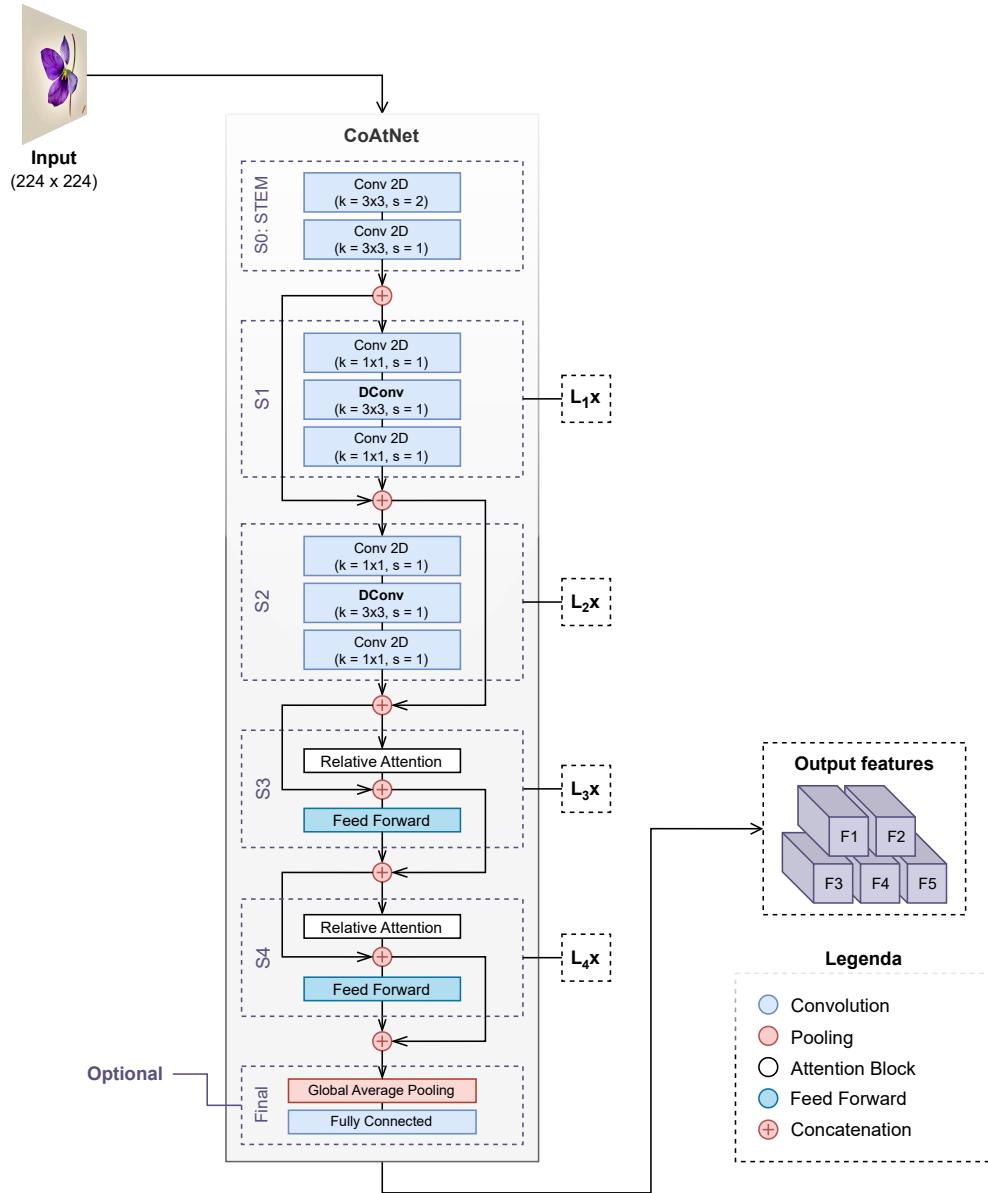
Apesar de o desenvolvimento de uma arquitetura voltada para dispositivos móveis não fosse uma prioridade para os autores, uma contribuição importante foi a incorporação do bloco MBConv (*Mobile Inverted Bottleneck Convolution*), originalmente descritos na arquitetura MobileNetV2 (SANDLER et al., 2019). Esse tipo de bloco convolucional tornou a rede mais eficiente, tanto no consumo de recursos computacionais quanto no tempo de processamento. A MBConv aproveita a redução de operações oferecida pelas convoluções separáveis em profundidade, que são mais econômicas em comparação às convoluções tradicionais.

Na Figura 14 está apresentada a arquitetura geral da rede CoAtNet. A rede é organizada em etapas (os autores chamam de *stages*), as quais são apresentadas nos blocos roxos pontilhados e variam de S_0 a S_4 . A primeira etapa é composta por uma sequência de convoluções 2D tradicionais de 3x3, com passo (*stride*) 2 e 1, respectivamente. As próximas duas etapas, S_1 e S_2 , são compostas por uma sequência de convolução 2D 1x1, convolução separável em profundidade 3x3 e convolução 2D 1x1. Essas três primeiras etapas são responsáveis, principalmente, pela extração de informações locais da entrada. As etapas S_3 e S_4 são um agrupamento de Atenção Relativa 2D (*2D Relative Attention*) e FFN (*Feed Forward Network*), combinação responsável por capturar informações globais da entrada. Quando utilizado para classificação de imagens, a rede inclui as camadas finais de agrupamento global e totalmente conectadas, que são dispensáveis caso o objetivo seja utilizar a rede apenas como codificador.

Como apresentado na Figura 14, a rede faz uso de conexões residuais, por isso a saída de cada $etapa_i$ é concatenada com a saída da etapa $etapa_{i+1}$ e esse ciclo se repete até a camada de *pooling* global da rede. Além das conexões residuais entre as etapas também há aplicação dessas conexões entre as camadas de Atenção Relativa e FFN. O uso desse tipo de conexão auxilia a mitigar a degradação de gradiente ao longo das camadas da CNN, além de ajudar a preservar características relevantes extraídas em camadas anteriores (HE et al., 2016).

A rede possui versões numeradas de CoAtNet-0 a CoAtNet-7, cada uma com uma configuração específica de largura e número de filtros de saída. Com exceção da primeira etapa (S_0), todas as outras são repetidas várias vezes, conforme indicado pelo marcador $L_i \times$, que varia de acordo com a versão da CoAtNet. Por exemplo, nas versões CoAtNet-2 e CoAtNet-3, os valores de repetição são $L_1 \times = 2$, $L_2 \times = 6$, $L_3 \times = 14$ e $L_4 \times = 2$ para ambas as versões da rede.

Figura 14 – Arquitetura geral das redes CoAtNets



Fonte: Adaptado de Dai et al. (2021)

Embora os autores tenham focado em apresentar o desempenho da rede na classificação de imagens, os próprios relatam que a arquitetura proposta pode apresentar potencial para detecção de objetos e segmentação de imagens.

2.1.3.2 CoaT

Com estrutura arquitetural semelhante à CoAtNet, a CoaT (*Co-scale conv-attentional image Transformers*), foi proposta por Xu et al. (2021) com intuito de desfrutar dos benefícios de operações convolucionais e mecanismos de atenção herdados da arquitetura

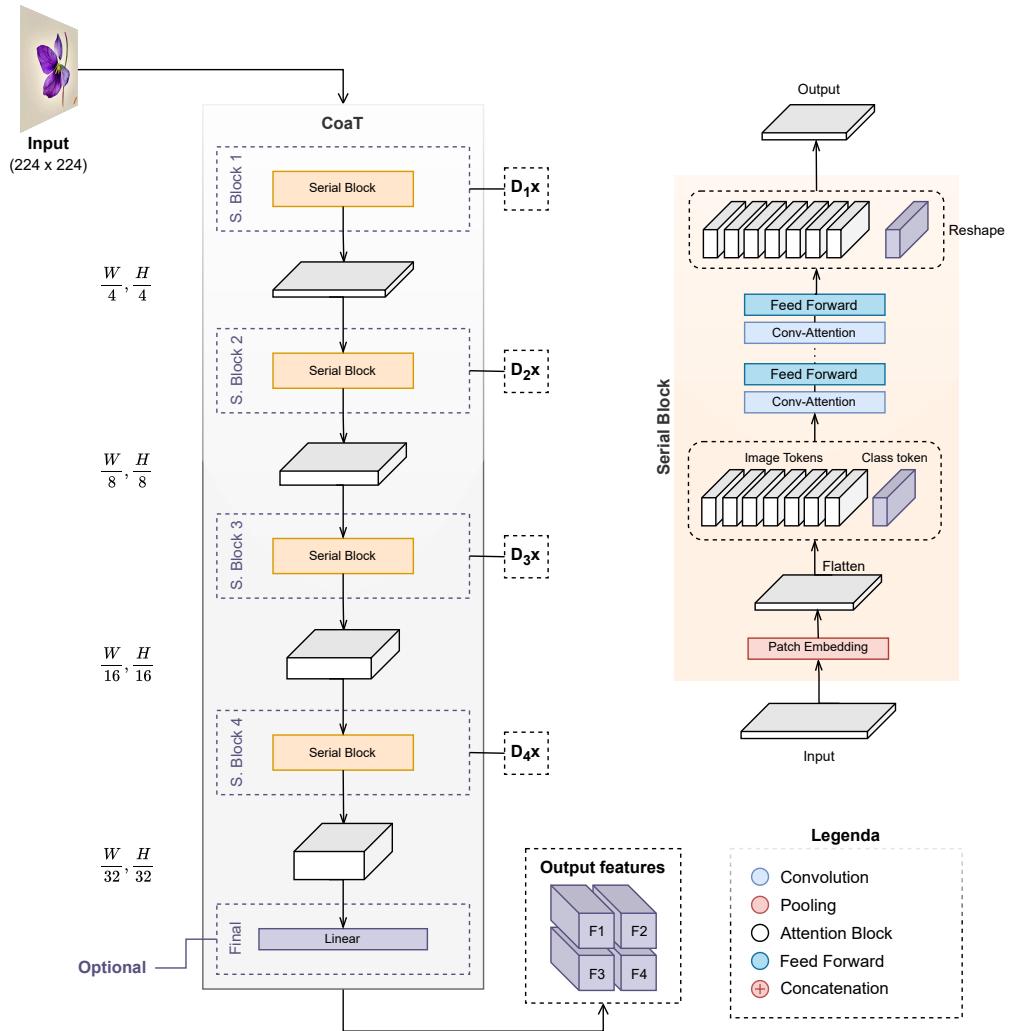
Transformers. Os autores relataram desempenho satisfatório de sua rede nas tarefas de classificação de imagem, detecção de objetos e segmentação de instância.

Um dos principais recursos dessa arquitetura é o Mecanismo de Co-Escala (*Co-Scale Mechanism*), que permite à rede capturar simultaneamente informações locais e globais. Diferente de outros mecanismos multiescala, como o módulo *Atrous Spatial Pyramid Pooling* (ASPP) usado por Chen et al. (2017), que gera representações isoladas da imagem e as concatena no final, o mecanismo proposto por Xu et al. (2021) promove a comunicação e integração contínuas entre as representações em diferentes escalas ao longo do aprendizado. A arquitetura CoaT possui dois grupos de configurações: CoaT-Lite, uma versão mais simples com menos camadas, e CoaT, que tem um número semelhante de parâmetros, porém mais camadas. Os blocos responsáveis pela Co-Escala, chamados de blocos paralelos (*parallel blocks*), estão presentes no final da rede nas variações CoaT-Lite, enquanto nas variações CoaT aparecem em todos os blocos, exceto no primeiro.

Embora os autores não tenham explicitamente mencionado o objetivo de criar uma arquitetura voltada para dispositivos com recursos computacionais limitados, as diferentes versões da rede CoaT apresentam um baixo número de parâmetros em comparação com outras redes que alcançam resultados semelhantes ou são projetadas para a tarefa de classificação de imagem e detecção de objeto, conforme relatam Xu et al. (2021). A versão mais econômica, CoaT-Lite Tiny, possui cerca de 5.7 milhões de parâmetros, enquanto a versão mais robusta, CoaT-Lite Medium, conta com aproximadamente 45 milhões de parâmetros.

A Figura 15 apresenta a arquitetura geral da rede CoaT e suas variações. Todas as versões possuem quatro blocos seriais sequenciais, responsáveis por reduzir a resolução da entrada, converter a imagem em *tokens* e compreender as relações entre eles. Para capturar essas relações, a rede utiliza uma sequência de blocos de convolução com mecanismos de atenção, chamados pelos autores de *conv-attentional blocks*. Uma contribuição importante dessa arquitetura é justamente a integração da convolução com a atenção, que ocorre de forma intercalada e integrada, ao invés de sequencial ou isolada, como em outras abordagens que visam combinar convolução e mecanismos de atenção.

Figura 15 – Arquitetura geral das redes CoaT



Fonte: Adaptado de Xu et al. (2021)

Ao comparar a arquitetura da rede CoAtNet e a da CoaT é possível notar que a primeira (Figura 14) faz uso isolado das operações de convolução, concentrando-as nas três primeiras etapas da arquitetura e o mecanismo de atenção nas últimas etapas, enquanto que a segunda rede (Figura 15) combina convolução com mecanismo de atenção diversas vezes em um mesmo bloco serial, reforçando a integração entre esses dois recursos.

Essa organização dos blocos convolucionais e de atenção indica que a rede CoAtNet inicia com a captura de informações locais e finaliza com a extração de informações globais, enquanto que a rede CoaT faz a combinação de forma contínua dessas duas hierarquias de informação. Outra diferença entre as duas arquiteturas é a presença de um mecanismo para capturar informações de representações em escalas distintas da entrada, recurso presente apenas nas variações CoaT.

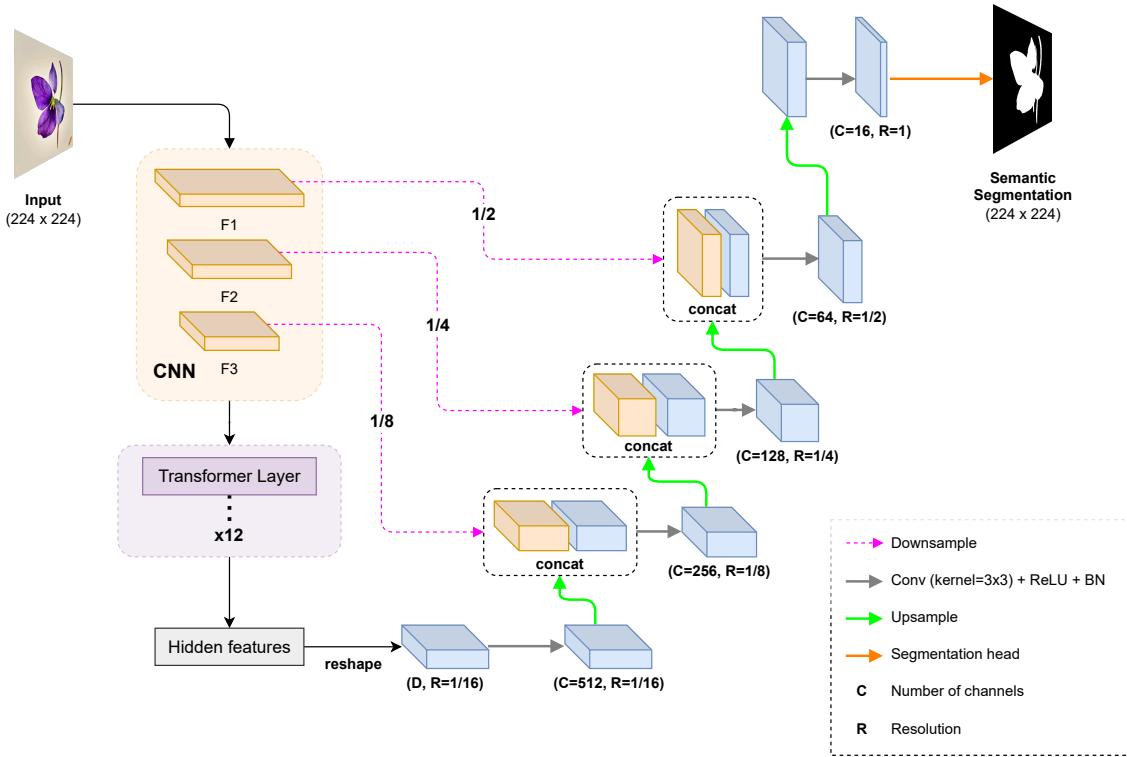
2.1.3.3 TransUnet

Considerando o uso promissor de modelos *Transformers*, originalmente empregados em tarefas de processamento de linguagem natural (VASWANI et al., 2017), na área de visão computacional (KHAN et al., 2022; CARION et al., 2020; HONG et al., 2022), Chen et al. (2021) propuseram uma solução híbrida que combina uma rede CNN com camadas *Transformers*. Essa arquitetura foi desenvolvida com base na estrutura da U-Net e chamada de TransUnet.

A Figura 16 apresenta o fluxo de dados pela rede e a comunicação entre o módulo CNN, o módulo *Transformers* e o decodificador. O codificador da arquitetura é composto pelo módulo CNN (que faz uso de uma ResNet-50) e de uma sequência de 12 camadas *Transformers*. O módulo CNN recebe a imagem de entrada, extrai suas características e reduz pela metade a dimensão das informações a cada bloco convolucional por uma sequência de 3 blocos, entregando características com 3 dimensões distintas. A saída do módulo CNN passa por um processo de *embedding* e seu resultado é processado pela sequência de camadas *Transformers*.

Seu decodificador é inicializado com a concatenação de características extraídas pela sequência de camadas *Transformers* e as características de menor dimensão obtidas pelo módulo CNN. Com exceção da primeira camada do decodificador, as próximas são resultados da concatenação da saída de sua respectiva camada do codificador com a camada anterior do decodificador após uma operação de *upsampling* para dobrar sua resolução, assim como na U-Net.

Figura 16 – Arquitetura da rede TransUnet



Fonte: Elaborado pelo autor (2023)

2.2 Trabalhos relacionados

Eigen, Puhrsch e Fergus (2014) propuseram uma abordagem que combina duas CNNs, uma de escala global (*Global Coarse-Scale Network*) e uma de escala local (*Local Fine-Scale Network*). A rede de escala global é responsável por extrair informações mais gerais e amplas da imagem, esse comportamento favorece a estimativa de profundidade em superfícies extensas ou fundos de paisagens, por exemplo. Já a rede local, é responsável por extrair informações associadas a regiões específicas da imagem, favorecendo a estimativa em estruturas e superfícies com mudanças pouco abruptas de profundidade, como bordas e pequenas texturas. Os autores também contribuíram apresentando uma função de perda que combina as perdas em escala global e local.

O trabalho de He, Wang e Hu (2018) utilizou uma CNN baseada na arquitetura VGG com aprendizado supervisionado para extrair características da imagem e estimar a profundidade. A principal contribuição dos autores foi a inclusão da distância focal da câmera como entrada da CNN, permitindo que essa informação fosse utilizada para ajustar a previsão do mapa de profundidade. Essa inclusão aperfeiçoou a capacidade de generalização do modelo, especialmente ao lidar com imagens do mesmo ambiente, mas com diferentes distâncias focais da câmera.

Fu et al. (2018) propuseram uma nova abordagem para a estimativa de profundidade, reformulando o problema como uma tarefa de regressão ordinal, em vez de regressão contínua. Em vez de prever valores exatos de profundidade para cada *pixel*, o modelo estima uma classe que representa um intervalo de valores, facilitando o aprendizado com dados de menor variabilidade. Os autores desenvolveram a *Deep Ordinal Regression Network* (DORN), uma rede neural profunda com dois componentes principais: (1) um codificador CNN, como ResNet ou VGG, pré-treinado em outra tarefa de visão computacional, que extrai características da imagem, e (2) uma camada de classificação ordinal que recebe essas características e classifica cada *pixel* em uma das classes ordinais de profundidade.

Qi et al. (2018) propuseram a arquitetura GeoNet, uma solução composta por duas CNNs: uma para estimativa de profundidade e outra para estimativa de normais de superfície. A principal contribuição dos autores foi explorar a interdependência entre essas duas tarefas para melhorar os resultados em ambas. A estimativa de profundidade determina a distância do observador até cada ponto da superfície, enquanto a estimativa de normais de superfície calcula os vetores de orientação dessas superfícies. A solução proposta utiliza aprendizado supervisionado e é composta por três módulos: (1) *Depth-to-Normal Network*, que estima as normais de superfície a partir dos mapas de profundidade; (2) *Normal-to-Depth Network*, que ajusta os mapas de profundidade com base nas normais estimadas; e (3) *Refinement Module*, que refina as estimativas após o processamento pelos módulos anteriores.

Assim como a GeoNet, a *Task-Recursive Learning* (TRL) Network, proposta por Zhang et al. (2018), combina a resolução de duas tarefas simultâneas: segmentação semântica e estimativa de profundidade. A arquitetura é uma CNN do tipo codificador-decodificador, baseada em ResNet, que alterna entre as duas tarefas, utilizando a saída de uma como parte da entrada da outra. Essa troca de informações entre as tarefas, juntamente com o ciclo recursivo de refinamento permite que as previsões se aprimorem a cada iteração, até convergirem para um resultado mais preciso. Além do ciclo recursivo, os autores propuseram uma função de perda combinada, que não apenas soma as perdas de cada tarefa, mas também avalia a consistência entre as previsões de ambas.

Huynh et al. (2020) apresentaram uma solução que combina um mecanismo de atenção baseado em profundidade com uma arquitetura codificador-decodificador. O componente principal da arquitetura proposta é o módulo *Depth-Attention Volume* (DAV), projetado para capturar dependências geométricas não locais entre os objetos da imagem. Esse módulo aprende, de forma implícita, restrições de coplanaridade e utiliza essas informações para guiar a estimativa de profundidade em superfícies planas de imagens similares. Segundo os autores, o modelo apresenta um desempenho superior em conjuntos de dados de ambientes internos, como o conjunto de dados NYU Depth V2.

He et al. (2021) propuseram a *Semantic Object Segmentation and Depth Estimation*

Network, abreviada como SOSD-Net, uma rede que, assim como a apresentada por Zhang et al. (2018), utiliza as interdependências entre as tarefas de segmentação semântica e estimativa de profundidade monocular para aprimorar o desempenho em ambas as tarefas. O codificador da arquitetura proposta faz uso da rede Xception juntamente com o ASPP para extrair as características da entrada. O decodificador possui três fluxos: um para a segmentação semântica, um para a estimativa de profundidade e outro para uma representação comum das características. Após o decodificador gerar as saídas desses fluxos, as três representações são utilizadas para melhorar o resultado da tarefa-alvo.

Tang et al. (2021) propuseram uma estrutura codificador-decodificador que utiliza a Res2Net-50 para extrair características da imagem de entrada e um decodificador com pirâmide de características para capturar e integrar mapas de profundidade em múltiplas escalas. O componente de pirâmide utilizado é semelhante ao proposto por Lin et al. (2017) para detecção de objetos, o uso desse recurso possibilita a combinação de detalhes finos com informações mais amplas, resultando em mapas de profundidade mais precisos ao aproveitar características de diferentes resoluções.

Ramamonjisoa et al. (2021) desenvolveram uma arquitetura codificador-decodificador para a estimativa de profundidade em imagens monoculares, utilizando decomposição em *wavelets*. Essa abordagem permite a previsão de coeficientes de *wavelet* esparsos, resultando em mapas de profundidade de alta fidelidade e eficiência computacional superior em comparação com outros métodos citados pelos autores. Uma das principais contribuições é a capacidade de aprender os coeficientes sem supervisão direta, supervisionando apenas o mapa de profundidade final, o que se mostra especialmente útil em situações onde dados rotulados não estão disponíveis. Além disso, o modelo requer menos da metade das operações de multiplicação e adição na rede de decodificação em relação a abordagens anteriores, sem comprometer a precisão (RAMAMONJISOA et al., 2021). Os autores mostram que a aplicação da decomposição em *wavelets* também é possível em outras redes com arquitetura codificador-decodificador.

Seguindo a abordagem de pirâmide de convolução dilatada, Lee et al. (2021) desenvolveram uma arquitetura flexível, capaz de integrar diversos codificadores para extração de características com uma etapa de decodificação que utiliza o módulo ASPP e camadas de guias locais para orientação planar. A arquitetura faz uso de quatro guias locais, cada um responsável por produzir um mapa de profundidade em uma resolução específica. Essa estrutura multi-escala possibilita que a rede capture tanto detalhes refinados quanto informações globais, mantendo um baixo número de parâmetros, que varia principalmente de acordo com o codificador utilizado.

O trabalho de Abdulwahab et al. (2023) apresenta uma arquitetura supervisionada de *autoencoder* para estimativa de profundidade. O modelo utiliza a rede de segmentação

semântica HRNet-v2 para extrair informações semânticas contextuais, que são combinadas com características menos refinadas obtidas de uma rede SENet-154, originalmente desenvolvida para classificação de imagens (HU et al., 2019). Ao mesclar as características brutas da imagem de entrada, extraídas por um codificador, às informações semânticas, essa arquitetura codificador-decodificador mantém a descontinuidade das superfícies e objetos, além de lidar eficazmente com oclusões.

Para realizar a estimativa de profundidade monocular em dispositivos com recursos limitados, como sistemas embarcados e dispositivos móveis, Guzzo e Gazolli (2023) propuseram uma arquitetura leve baseada na UNet++. Os autores adotaram a MobileNetV2 como codificador, pois suas convoluções separáveis em profundidade e blocos residuais invertidos permitem um baixo consumo de recursos e um número reduzido de parâmetros. Além disso, o codificador foi previamente treinado no conjunto de dados ImageNet, otimizando seu desempenho.

3 METODOLOGIA

Neste capítulo são apresentadas as arquiteturas utilizadas para a estimativa de profundidade e a motivação que fundamenta a escolha das redes utilizadas como codificadores, além da descrição da função de perda utilizada nos experimentos computacionais.

3.1 Arquiteturas Adotadas

Para a realização dos experimentos computacionais, foram utilizadas variações das arquiteturas U-Net e UNet++, nas quais os codificadores originais foram substituídos pelas seguintes redes: VGG-19 BN, Inception-ResNet-v2, Xception, Mixed Transformer, CoAtNet e CoaT, mantendo-se os decodificadores originais. Além disso, a rede TransUnet foi utilizada em sua forma original, conforme proposta por Chen et al. (2021).

A seleção das redes supracitadas como codificadores teve como norte o *benchmark* de CNNs para classificação de imagens elaborado por Bianco et al. (2018), no qual são apresentados não apenas a acurácia das redes neurais, mas também o número de parâmetros e o consumo de memória.

Para acelerar a etapa de aprendizado e prover melhores resultados, todos os codificadores utilizados, inclusive o da rede TransUnet, foram pré-treinados no conjunto de classificação de imagens ImageNet.

3.2 Adaptação dos codificadores nas redes U-Net e UNet++

Dado que as redes U-Net e UNet++ são do tipo codificador-decodificador, não foi necessário realizar modificações específicas para a utilização dos codificadores propostos, pois essas arquiteturas permitem a integração com qualquer codificador. Portanto, foi possível adaptar uma implementação geral para todas as variações de codificadores utilizadas. O principal ajuste consistiu em definir como cada codificador forneceria os cinco conjuntos de características (*features*) extraídas que serão utilizadas como entrada para os decodificadores.

As Figuras 3, 4, 7, 13, 14 e 15 representam a arquitetura original das redes VGG-19 BN, Inception-ResNet-v2, Xception, MiT-B2, CoAtNet e CoaT, respectivamente. Como essas redes possuem diversas camadas e blocos, e as redes U-Net e UNet++ recebem cinco grupo de características de entrada, foi necessário definir cinco “blocos lógicos”, responsáveis por fornecer cada conjunto de características. Esses blocos lógicos foram escolhidos com base em suas características e estão destacados com retângulos roxos pontilhados.

Para a utilização da rede neural VGG-19 BN (Figura 3) como codificador, o último bloco (*Block 6*) dessa rede foi dispensado. Isso se deve ao fato de que a U-Net e a UNet++

utilizam no máximo cinco saídas do codificador, e de que o *Block 6* contém apenas camadas totalmente conectadas, destinadas à classificação e não à extração de características, o que não atende ao objetivo do presente trabalho.

A definição dos dois primeiros blocos lógicos na Inception-ResNet-v2 (Figura 4) se baseia no fato de que esses dois conjuntos de camadas fornecem características com maior resolução (altura e largura) e menor profundidade (número de canais). Já a combinação dos blocos Reduction-A com os Inception-ResNet-B e Reduction-B com os Inception-ResNet-C foi realizada porque cada um desses pares possui o mesmo número de canais, resolução de saída e retornam características com baixa resolução e alta profundidade.

Dos seis conjuntos de características entregues pelo codificador Xception (Figura 7), apenas os cinco primeiros são utilizados. Os primeiros conjuntos são características mais gerais e primitivas, como bordas, cantos e cores da entrada, enquanto os últimos tendem a apresentar informações globais e específicas do domínio do problema, como objetos inteiros e suas formas (ZEILER; FERGUS, 2014).

Como as redes das famílias Mix Transformer e CoAtNet, Figura 13 e 14, respectivamente, geram apenas quatro grupos de características de saída, foi necessário adicionar dois tensores: um com a mesma resolução da entrada e outro com metade dessa resolução. Para solucionar esse problema de implementação, optou-se por utilizar um tensor de características vazio, com zero canais, e reaproveitar o tensor de entrada. Essa abordagem garantiu que o processo de decodificação ocorresse sem erros, preservando as características extraídas. No caso da rede CoaT, Figura 15, o mesmo ajuste foi necessário, mas apenas a adição de um tensor foi suficiente, já que a rede retorna cinco grupos de características.

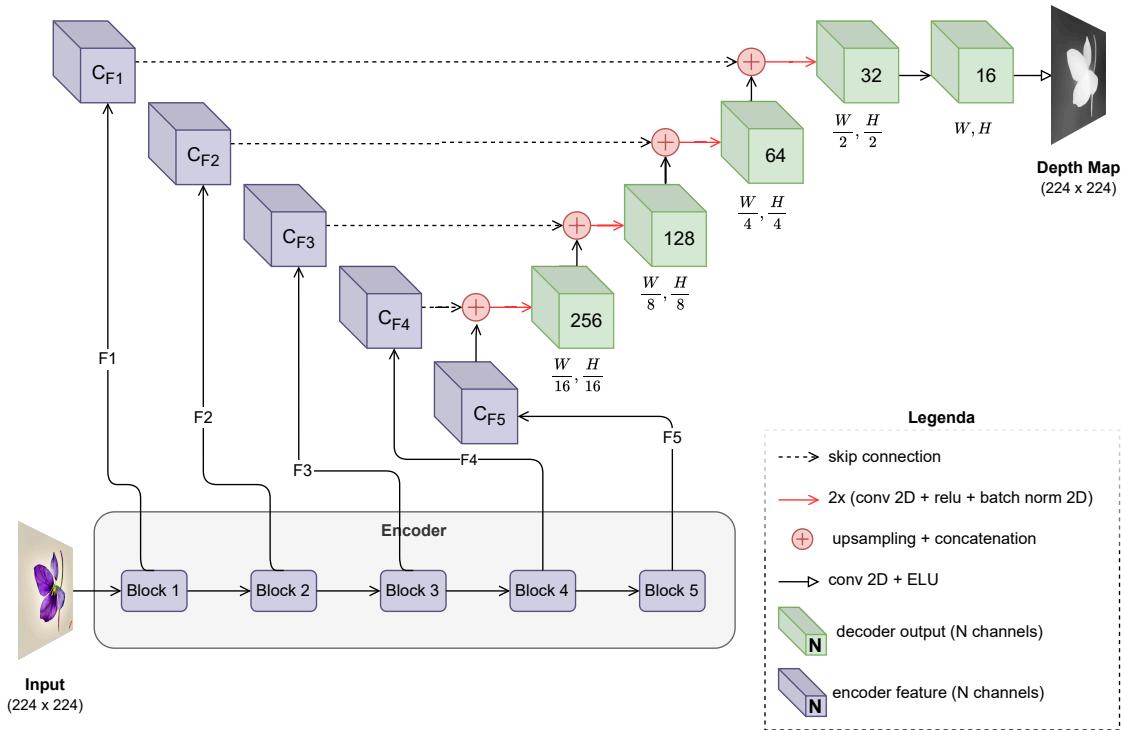
3.3 Arquitetura U-Net Adaptada

Na Figura 17 está representada a arquitetura geral utilizada da U-Net. Inicialmente a entrada, uma imagem RGB, é recebida pelo codificador (bloco cinza, *Encoder*). Cada bloco do codificador (bloco roxo arredondado) é responsável por entregar um conjunto específico de características da entrada. Os primeiros blocos do decodificador são construídos a partir das características extraídas dos últimos blocos do codificador. Exemplo: A construção do primeiro bloco (cubo verde, com 256 canais) do decodificador se dá a partir da concatenação das características extraídas pelos dois últimos blocos do codificador, blocos 4 e 5. Os próximos blocos do decodificador são construídos a partir da concatenação de seu bloco anterior com o próximo bloco codificador ainda não processado.

As características dos últimos blocos do codificador são utilizadas primeiro por serem as que possuem maior número de canais e menor resolução (altura, largura), assim como os primeiros blocos do decodificador. Por isso, a primeira *feature* (F1) entregue pelo

codificador é utilizada para construir o penúltimo bloco do decodificador, com 32 canais. Por fim, o último bloco do decodificador entrega as características com 16 canais para uma cabeça de segmentação (*segmentation head*) que aplicará uma convolução 2D, a função de ativação ELU e retornará o mapa de profundidade.

Figura 17 – Arquitetura geral da rede U-Net com codificador



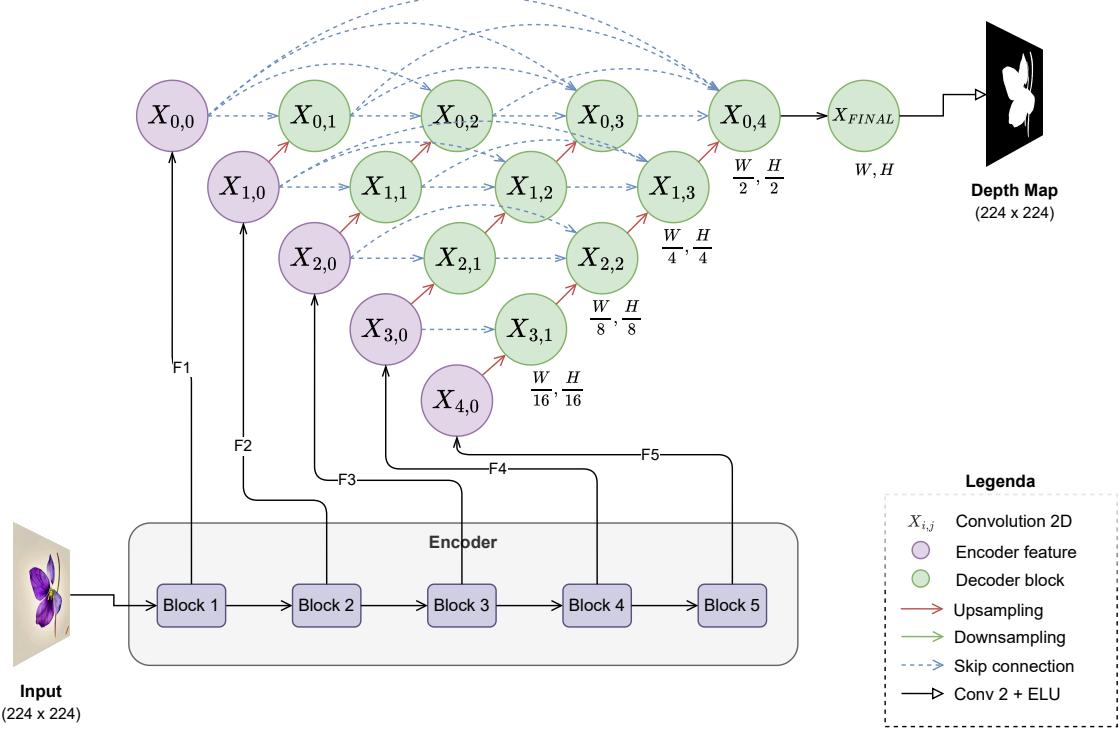
Fonte: Elaborado pelo autor (2024)

3.4 Arquitetura UNet++ Adaptada

A arquitetura geral da UNet++ adaptada para um novo codificador é representada na Figura 18. Os círculos roxos ($X_{0,0}, X_{1,0}, X_{2,0}, X_{3,0}, X_{4,0}$) representam as *features* extraídas pelos blocos do codificador. Do recebimento da entrada RGB à produção do mapa de profundidade, o fluxo de processamento é semelhante ao da U-Net discutido previamente. A principal distinção entre as duas arquiteturas é o maior consumo de recursos como memória e tempo de processamento, devido às conexões densas da UNet++.

Devido à complexidade computacional e às limitações de *hardware*, não foi possível implementar a UNet++ com codificadores da família Mixed Transformer.

Figura 18 – Arquitetura geral da rede UNet++ com codificador



Fonte: Elaborado pelo autor (2024)

3.5 Função de perda

Para avaliar o aprendizado das redes analisadas, foi utilizada uma abordagem que envolve a soma ponderada de outras três funções de perdas, conforme proposto por Alhashim e Wonka (2018): erro médio absoluto (L_{depth}), perda L1 sobre os gradientes dos mapas de profundidade (L_{grad}) e medida do índice de similaridade estrutural (L_{SIM}), apresentadas nas Equações 2,3, 4, respectivamente, nas quais y representa o mapa de estimativa de profundidade verdadeiro e \hat{y} , o mapa de estimativa de profundidade predito.

$$L_{\text{depth}}(y, \hat{y}) = \frac{1}{n} \sum_p^n |y_p - \hat{y}_p|. \quad (2)$$

$$L_{\text{grad}}(y, \hat{y}) = \frac{1}{n} \sum_p^n |\mathbf{g_x}(y_p, \hat{y}_p)| + |\mathbf{g_y}(y_p, \hat{y}_p)| \quad (3)$$

$$L_{\text{SIM}}(y, \hat{y}) = \frac{1 - \text{SSIM}(y, \hat{y})}{2} \quad (4)$$

A função final, resultante da soma das três funções de perda, é apresentada na Equação 5:

$$L(y, \hat{y}) = \lambda L_{\text{depth}}(y, \hat{y}) + L_{\text{grad}}(y, \hat{y}) + L_{\text{SSIM}}(y, \hat{y}) \quad (5)$$

3.6 Uso de pesos e limitação de resolução de entrada

Os codificadores CoAtNet-2^{1,2}, CoAtNet-3³ e CoaT-Lite Medium^{4,5} tiveram seus pesos pré-treinados no ImageNet carregados a partir da biblioteca *PyTorch Image Models* (conhecida como “timm”), para evitar a necessidade de realizar o pré-treinamento completo dessas redes no conjunto de dados de classificação de imagens citado. No entanto, esses codificadores possuem detalhes de implementação que limitam o treinamento com resoluções diferentes das utilizadas durante o pré-treinamento no ImageNet. Assim, os treinamentos com o CoAtNet-2 ficaram restritos às resoluções de 224x224 e 384x384, enquanto o CoAtNet-3 só aceitou entradas de 224x224, o que também ocorre com a rede TransUnet. Já o codificador CoaT-Lite Medium permitiu o treinamento com outras resoluções, mas isso exigiu modificações no código-fonte de sua implementação na biblioteca timm.

Para gerar mapas de profundidade e mapas de calor dos erros com maior qualidade visual, as entradas de inferência foram processadas na resolução original de 480x640 *pixels*. No entanto, devido às restrições de resolução mencionadas, as entradas para as redes CoAtNet-3 e TransUnet foram redimensionadas para 224x224 antes da predição, e após a inferência, o redimensionamento para 480x640 foi feito utilizando interpolação bilinear. Dessa forma, é esperado que as saídas dessas redes apresentem algum nível de serrilhado ou qualidade de imagem inferior, enquanto seus mapas de calor de erros tendem a subestimar os erros reais, devido ao redimensionamento aplicado aos mapas de profundidade.

¹ <https://huggingface.co/timm/coatnet_rmlp_2_rw_224.sw_in1k>

² <https://huggingface.co/timm/coatnet_rmlp_2_rw_384.sw_in12k_ft_in1k>

³ <https://huggingface.co/timm/coatnet_3_rw_224.sw_in12k>

⁴ <https://huggingface.co/timm/coat_lite_medium.in1k>

⁵ <https://huggingface.co/timm/coat_lite_medium_384.in1k>

4 EXPERIMENTOS, RESULTADOS E DISCUSSÕES

Neste capítulo os experimentos realizados para avaliar a eficácia das arquiteturas propostas para a estimativa de profundidade monocular são descritos, ademais os resultados obtidos são discutidos.

4.1 Base de dados

Os experimentos foram executados utilizando-se a base de imagens coloridas e *indoor* NYU Depth V2 (SILBERMAN et al., 2012). Embora o conjunto original seja composto por mais de 100 mil imagens, cada uma com uma resolução de 480x640, optou-se por empregar o subconjunto de 50 mil amostras, conforme estabelecido por Alhashim e Wonka (2018), para a fase de treinamento das redes. Para a avaliação das arquiteturas foi utilizado o subconjunto oficial de imagens e mapas de profundidade NYU V2 dedicados à etapa de teste, com 654 pares de imagens e mapas de profundidade.

Ainda na etapa de treinamento, a resolução das imagens foi reduzida para 224x224, utilizando a técnica de interpolação bilinear com o propósito de mitigar distorções significativas do processo de redimensionamento. Com intuito de desconsiderar bordas vazias das imagens, durante a etapa de teste, tanto as imagens coloridas quanto os mapas de profundidade foram recortados utilizando os valores propostos por Eigen, Puhrsch e Fergus (2014).

Visando aprimorar a capacidade de aprendizado da rede ao lidar com novos dados (generalização) e mitigar o problema de sobre-ajuste, foram incluídas as duas transformações (50% de chance de espelhar horizontalmente a imagem e 25% de chance de permutar seus canais de cores) utilizadas por Alhashim e Wonka (2018) para manipular os dados do conjunto de treinamento. Ao carregar cada par de imagem e mapa de profundidade, a estratégia de aumento de dados (*Data Augmentation*) pode ser aplicada ao par, atuando sobre o conjunto de treino sem acrescentar novos dados, apenas transformando os dados existentes antes de processá-los.

4.2 Detalhes de implementação

Todos os experimentos foram executados em máquinas que seguem a configuração: Placa gráfica NVIDIA GeForce RTX 3060 (12GB), AMD Ryzen 5 PRO 4650GE 3.3GHz, 16GB RAM e Ubuntu 24.04 LTS como sistema operacional.

Em todas as arquiteturas, foram utilizadas as versões pré-treinadas no conjunto de dados para classificação de imagens ImageNet de seus codificadores. Durante a etapa de treinamento, não foi realizado o congelamento de qualquer camada de nenhum dos codificadores.

4.3 Avaliação

Para verificar o desempenho dos modelos, foram utilizadas as seguintes métricas de estimativa de profundidade, propostas por Eigen, Puhrsch e Fergus (2014) e apresentadas nas Equações 6,7, 8, 9:

- *Threshold* (δ_j):

$$\delta_j : \% \text{ of } \hat{i} \text{ s.t. } \max \left(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i} \right) < 1.25^j, j \in \{1, 2, 3\} \quad (6)$$

- *Root Mean Square Error* (RMSE):

$$RMSE : \sqrt{\frac{1}{P} \sum_{i=1}^P (y_i - \hat{y}_i)^2} \quad (7)$$

- *Scale invariant error* (\log_{10}):

$$\log_{10} = \frac{1}{P} \sum_{i=1}^P |\log_{10}(y_i) - \log_{10}(\hat{y}_i)| \quad (8)$$

- *Absolute Relative Difference* (rel):

$$rel = \frac{1}{P} \sum_{i=1}^P \frac{|y_i - \hat{y}_i|}{y_i} \quad (9)$$

4.4 Resultados

Na Tabela 1 são apresentados os resultados quantitativos obtidos pelos modelos avaliados, arquiteturas U-Net e UNet++ com diferentes codificadores, e TransUnet, bem como por outras arquiteturas reconhecidas por sua aplicação na estimativa de profundidade monociliar. Ao analisar os resultados quantitativos é possível notar que os codificadores que alcançaram os melhores resultados são baseados em arquiteturas mais recentes.

Os modelos híbridos, majoritariamente, apresentam os melhores resultados nas métricas analisadas, com destaque para a U-Net com CoaT-Lite Medium, que obtém os maiores valores de acurácia e os menores de erro. A segunda combinação com melhores valores foi a combinação de UNet++ com o mesmo codificador. Para superar as limitações das redes puramente convolucionais em preservar características ao longo das camadas e capturar informações globais da entrada, redes projetadas com mecanismos de atenção, como as variantes CoAtNet, CoaT, MiT e TransUnet, podem apresentar melhor manutenção das características extraídas desde o início do processamento. Além do uso de mecanismos de atenção, as arquiteturas TransUnet (CHEN et al., 2021), a Mixed Transformer B2

(XIE et al., 2021) e a CoaT-Lite (XU et al., 2021) foram projetadas e testadas para a tarefa de segmentação semântica, o que pode ter proporcionado uma vantagem sobre redes inicialmente pensadas apenas para tarefas de classificação de imagens.

A relação entre complexidade do modelo em quantidade de parâmetros e resultado é um ponto relevante. Com exceção da CoAtNet-3, as três redes com maiores números de parâmetros não apresentam resultados que acompanhem seu tamanho em nenhuma métrica, assim como as três arquiteturas com os menores números de parâmetros não apresentam nenhum dos três melhores valores para qualquer uma das métricas utilizadas. Há fatores que dificultam estabelecer essa relação entre complexidade e eficácia como o período de desenvolvimento da arquitetura, a utilização de mecanismos como atenção, conexões de salto e outros recursos arquiteturais, que podem exercer papéis cruciais no desempenho.

Os resultados apresentados indicam que as arquiteturas propostas superam as outras abordagens, demonstrando desempenho superior em todos os limiares (δ_1 a δ_3) e no RMSE, mesmo em comparação com modelos mais complexos. Isso inclui redes que combinam segmentação e estimativa de profundidade (HE et al., 2021; ZHANG et al., 2018), utilizam informações de distância focal (HE; WANG; HU, 2018), reformulam o problema como regressão ordinal (FU et al., 2018), ou combinam redes para estimativa de diferentes superfícies (QI et al., 2018). Esses resultados destacam a eficiência dos nossos modelos para estimativa de profundidade, sem necessidade de modificações estruturais e arquiteturais complexas para entregar um desempenho satisfatório.

Tabela 1 – Comparação quantitativa considerando as arquiteturas propostas e trabalhos relacionados aplicados ao conjunto de dados NYU Depth V2

Method	Encoder	Params (M)	RMSE ↓	rel ↓	log10 ↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
Eigen, Puhrsch e Fergus (2014)	-	141,1	0,641	0,158	-	0,769	0,950	0,988
He, Wang e Hu (2018)	VGG	-	0,572	0,151	0,064	0,789	0,948	0,986
Fu et al. (2018)	-	110,0	0,509	<u>0,115</u>	0,051	0,828	0,965	0,992
Qi et al. (2018)	ResNet-50	67,2	0,569	0,128	0,057	0,834	0,960	0,990
Zhang et al. (2018)	ResNet-50	-	0,501	0,144	-	0,815	0,962	0,992
He et al. (2021)	Xception + ASPP	24,0	0,514	0,145	0,062	0,805	0,962	0,992
Tang et al. (2021)	Res2-50	-	0,579	0,132	0,056	0,826	0,963	0,992
Ramamonjisoa et al. (2021)	Monodepth	48+	0,551	0,126	0,054	0,845	0,968	0,992
Abdulwahab et al. (2023)	HRNet-V2	45,2+	0,523	0,121	0,053	0,852	0,974	0,994
TransUnet (CHEN et al., 2021)	ResNet-50	105,3	0,478	0,134	0,055	0,844	0,965	0,988
U-Net [224x224] (proposed)	I. ResNet V2	62,0	0,479	0,128	0,055	0,846	0,967	0,991
	VGG-19 (BN)	29,1	0,481	0,129	0,055	0,843	0,967	0,992
	Xception	28,8	0,501	0,133	0,057	0,838	0,963	0,988
	MiT (B2)	27,5	0,461	0,129	0,054	0,851	0,970	0,992
	CoAtNet-2	78,0	0,457	0,125	0,052	0,857	0,972	0,991
	CoAtNet-3	170,7	0,434	0,120	0,050	0,868	0,974	0,992
	CoaT-Lite	88,8	0,436	<u>0,115</u>	0,049	0,875	0,977	0,993
U-Net [384x384] (proposed)	MiT (B2)	27,5	0,436	0,118	0,050	0,869	0,976	<u>0,993</u>
	CoAtNet-2	78,0	0,446	0,119	0,051	0,866	0,974	<u>0,993</u>
	CoaT-Lite	88,8	0,419	0,111	0,047	0,884	0,979	0,994
UNet++ [224x224] (proposed)	I. ResNet V2	71,1	0,489	0,130	0,055	0,844	0,964	0,989
	VGG-19 (BN)	44,7	0,499	0,140	0,059	0,821	0,964	0,991
	Xception	34,0	0,492	0,132	0,058	0,843	0,965	0,988
	CoAtNet-2	84,0	0,444	0,120	0,051	0,867	0,974	0,992
	CoAtNet-3	183,8	0,443	0,120	0,051	0,866	0,971	0,992
	CoaT-Lite	92,5	0,439	<u>0,115</u>	0,050	0,873	0,975	0,993
UNet++ [384x384] (proposed)	CoAtNet-2	78,0	0,497	0,131	0,057	0,832	0,965	0,991
	CoaT-Lite	88,8	<u>0,426</u>	0,111	<u>0,048</u>	<u>0,883</u>	<u>0,978</u>	0,994

Para quantificar a diferença entre as implementações U-Net e UNet++, foi utilizada a Equação 10, que calcula a diferença percentual relativa entre os valores das métricas obtidas pelas duas redes. Os resultados dessa equação foram apresentados na Tabela 2, facilitando a visualização das diferenças. Na tabela, métricas de erro (RMSE, rel e log10) com valores negativos indicam uma vantagem da UNet++, enquanto valores positivos favorecem a U-Net. Já para os indicadores de acurácia (δ_1 a δ_3), valores negativos indicam uma vantagem para a U-Net, e valores positivos para a UNet++.

Ao analisar as métricas de erro, a UNet++ se destaca, obtendo 2 dos 6 melhores valores para RMSE, 2 dos 6 para rel e 1 dos 6 para log10. Essa mesma proporção se aplica aos indicadores de acurácia δ_1 , δ_2 e δ_3 . Os melhores valores obtidos pela UNet++ são 2,84% em RMSE, 4% em rel e 1,92% em log10. Já a U-Net apresenta como melhores valores 3,74%, 8,53% e 7,27%, respectivamente, para essas métricas. Em relação aos indicadores de acurácia (δ_1 , δ_2 e δ_3), os melhores valores para a UNet++ são 1,17%, 0,21% e 0,1%, enquanto para a U-Net são 2,61%, 0,31% e 0,2%.

Em resumo, a Tabela 2 denota que, para todas as métricas analisadas, os melhores resultados foram obtidos pela versão simples da rede codificador-decodificador. Os melhores valores da UNet++ foram registrados principalmente com os codificadores Xception e CoAtNet-2. No entanto, vale ressaltar que as implementações da UNet++ demandam uma carga computacional maior devido às suas convoluções densas e aninhadas. Esse desempenho sugere que, para explorar plenamente o potencial do aninhamento dos blocos, ajustes arquiteturais mais refinados podem ser necessários.

$$\text{Porcentagem de Diferença} = \frac{\text{Valor de U-Net} - \text{Valor de UNet}++}{\text{Valor de U-Net}} \times 100 \quad (10)$$

Tabela 2 – Diferença percentual relativa entre U-Net e UNet++ ao treinar com imagens 224x224 para diferentes codificadores

Encoder	RMSE (%) ↓	rel (%) ↓	log10 (%) ↓	δ_1 (%) ↑	δ_2 (%) ↑	δ_3 (%) ↑
ResNet V2	2,09	1,56	0	-0,24	-0,31	-0,20
VGG-19	3,74	8,53	7,27	-2,61	-0,31	-0,10
Xception	-1,80	-0,75	1,75	0,60	0,21	0
CoAtNet-2	-2,84	-4	-1,92	1,17	0,21	0,10
CoAtNet-3	2,07	0	2	-0,23	-0,31	0
CoaT-Lite	0,69	0	2,04	-0,23	-0,20	0

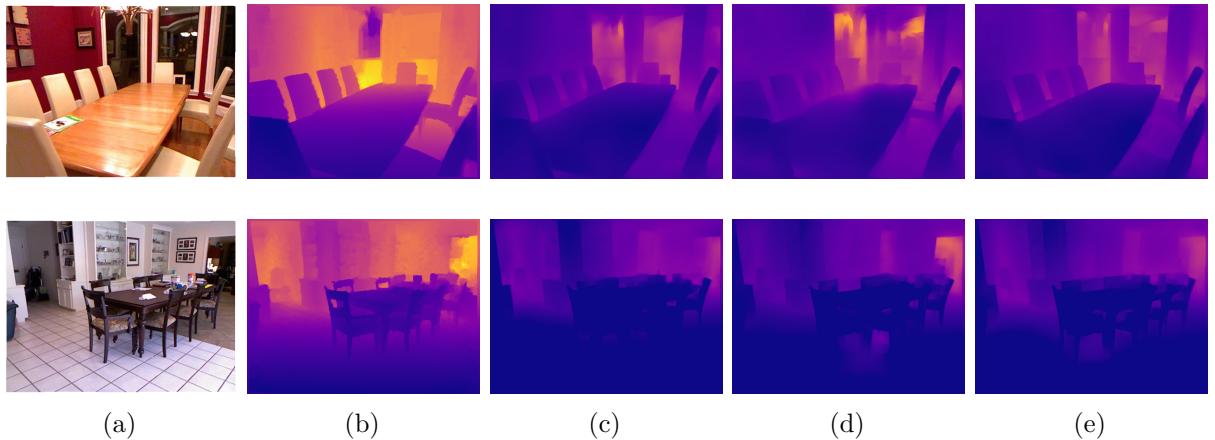
4.4.1 Análise qualitativa – Resolução 224x224

Além dos resultados quantitativos, também foram realizadas comparações qualitativas entre as saídas de cada rede implementada, para isso foi realizada a inferência para as imagens de ID “01399” e “01406” do conjunto de teste do NYU Depth V2. As saídas foram pré-processadas com a biblioteca OpenCV, aplicando um esquema de cores que utiliza tons frios para as superfícies mais próximas e tons quentes para as mais distantes. O mapa de cores “Plasma” foi usado para essas saídas. Já para destacar os erros entre a saída prevista e o mapa esperado, foi utilizado o mapa de cores “Jet”, que realça os *pixels* com maior erro em cores quentes e aqueles com menor erro em cores frias. Com intuito de apresentar a comparação qualitativa com melhor resolução e evitar o excesso de imagens agrupadas, optou-se por separar as saídas entre redes U-Net e UNet++, bem como isolar as saídas de codificadores baseados em *Transformers* dos puramente convolucionais.

Ao analisar as Figuras 19 e 20, correspondentes às saídas e mapa de calor de erros das redes U-Net com codificadores convolucionais, respectivamente, é possível notar que todas as redes apresentam dificuldade para estimar superfícies mais distantes, como as paredes, parte da mesa e cadeiras mais afastadas. Na segunda linha, as superfícies mais detalhadas da imagem de entrada estão distantes da câmera, o que torna essa imagem um desafio

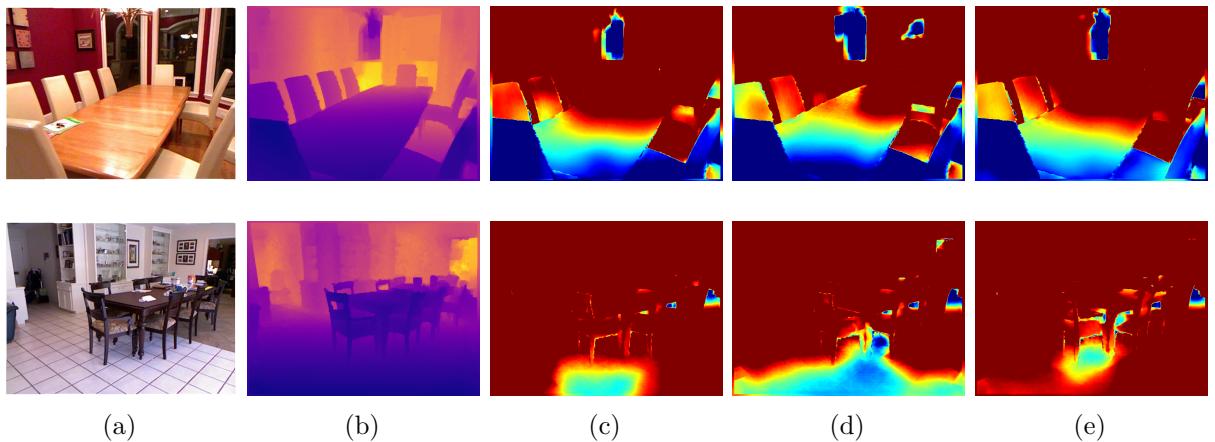
maior para as redes do que a primeira. A distância mencionada resulta em mapas de estimativa com objetos menos definidos, mais esfumaçados e agrupados, como pode ser visto na segunda linha e coluna c) da Figura 19.

Figura 19 – Resultados qualitativos de redes U-Net com codificadores sem mecanismos de atenção. (a) Entrada. (b) *Ground truth* (saída esperada). (c) Saída Inception-ResNet V2. (d) Saída VGG-19. (e) Saída Xception.



Fonte: Elaborado pelo autor (2024)

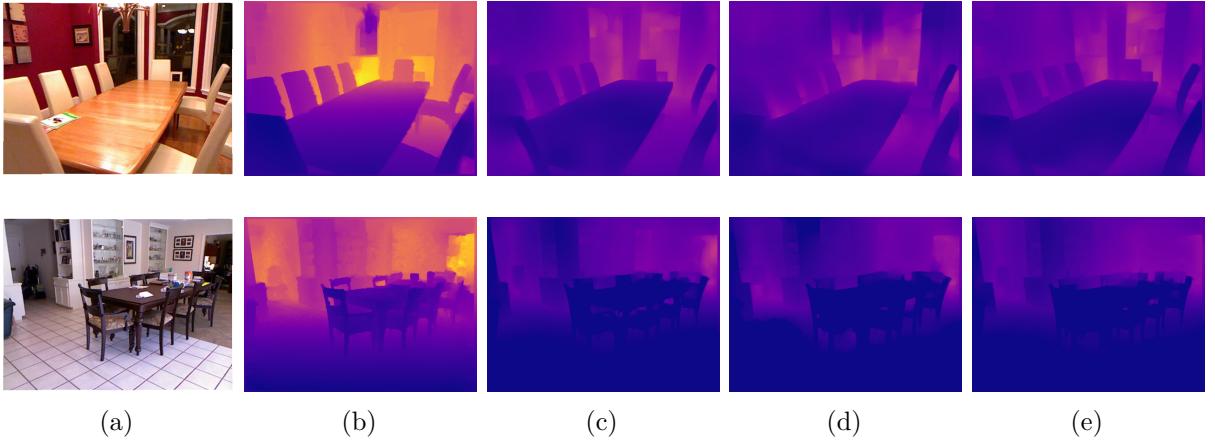
Figura 20 – Mapas de calor de erro de redes U-Net com codificadores sem mecanismos de atenção. (a) Entrada. (b) *Ground truth* (saída esperada). (c) Saída Inception-ResNet V2. (d) Saída VGG-19. (e) Saída Xception.



Fonte: Elaborado pelo autor (2024)

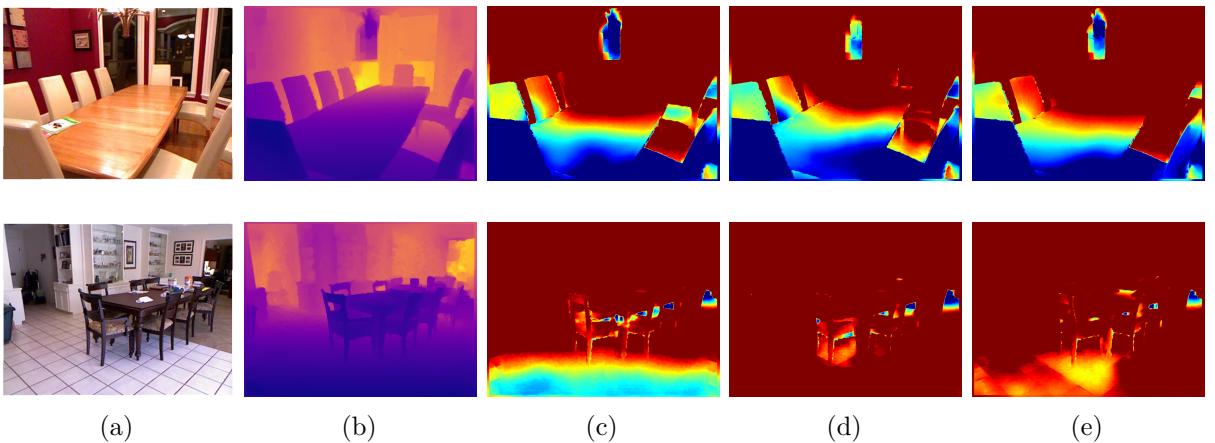
As Figuras 21 e 22 indicam que as redes UNet++ podem sofrer dos mesmos problemas que sua versão U-Net com os mesmos codificadores. Os mapas de calor com os erros apresentam grande semelhança com os mapas da Figura 20. A falta de detalhes em objetos, como as cadeiras, por exemplo, também é um problema compartilhado entre as U-Net e UNet++ propostas.

Figura 21 – Resultados qualitativos de redes UNet++ com codificadores sem mecanismos de atenção.
 (a) Entrada. (b) *Ground truth* (saída esperada). (c) Saída Inception-ResNet V2. (d) Saída VGG-19. (e) Saída Xception.



Fonte: Elaborado pelo autor (2024)

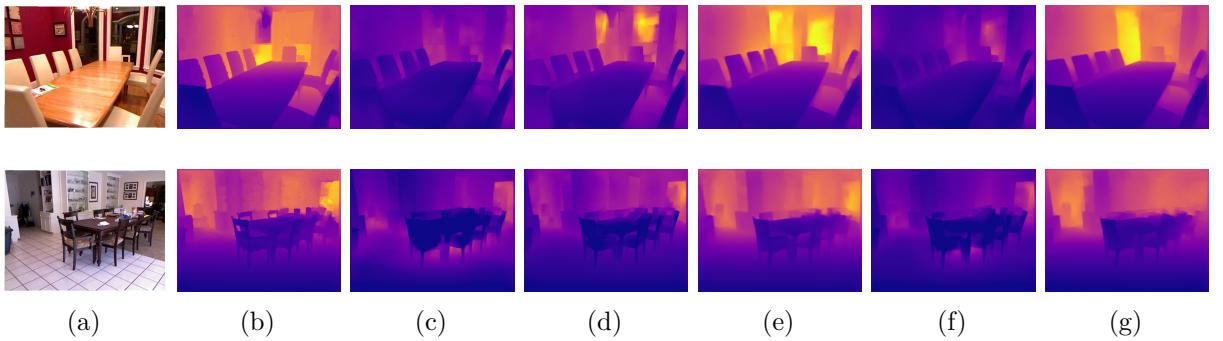
Figura 22 – Mapas de calor de erro de redes UNet++ com codificadores sem mecanismos de atenção.
 (a) Entrada. (b) *Ground truth* (saída esperada). (c) Saída Inception-ResNet V2. (d) Saída VGG-19. (e) Saída Xception.



Fonte: Elaborado pelo autor (2024)

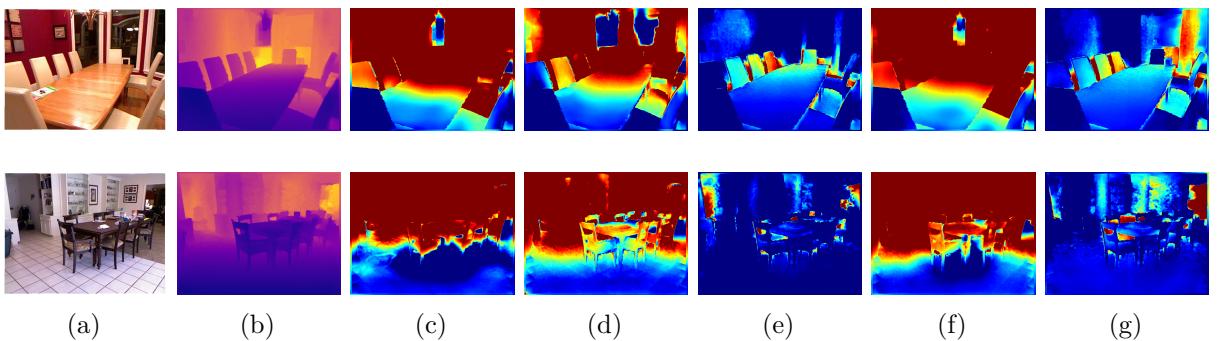
As amostras apresentadas nas Figuras 23 e 24 sugerem que a inclusão de codificadores com mecanismos de atenção, independentemente do tipo, aprimora a capacidade das redes U-Net de capturar detalhes. Essa melhoria é particularmente evidente na coluna d) da Figura 23 (saída da rede U-Net com CoAtNet-2), onde as cadeiras, tanto na primeira quanto na segunda imagem, mantêm seus detalhes preservados e apresentam menos objetos aglomerados como se fossem algo único. Além disso, os mapas de calor mostram que os codificadores CoAtNet-2 e MiT-B2 foram mais eficazes na estimativa da profundidade do piso. Os mapas de calor das redes com os codificadores CoAtNet-3 e TransUnet demonstram uma subestimação dos erros, um efeito colateral do redimensionamento dos mapas de profundidade, conforme discutido previamente no Capítulo 3.

Figura 23 – Resultados qualitativos de redes U-Net com codificadores com mecanismos de atenção. (a) Entrada. (b) *Ground truth* (saída esperada). (c) CoaT-Lite Medium. (d) CoAtNet-2. (e) CoAtNet-3. (f) MiT-B2. (g) TransUnet.



Fonte: Elaborado pelo autor (2024)

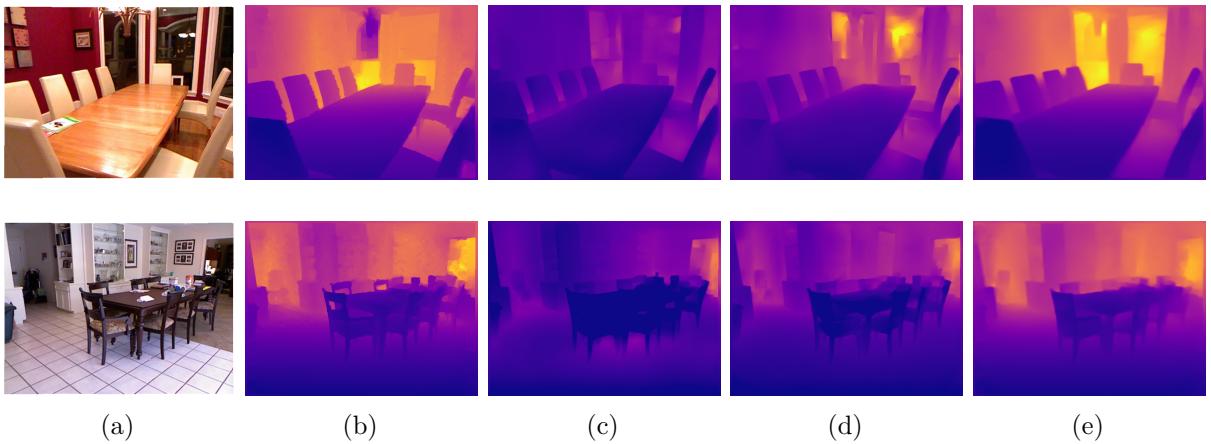
Figura 24 – Mapas de calor de erro de redes U-Net com codificadores com mecanismos de atenção. (a) Entrada. (b) *Ground truth* (saída esperada). (c) CoaT-Lite Medium. (d) CoAtNet-2. (e) CoAtNet-3. (f) MiT-B2. (g) TransUnet.



Fonte: Elaborado pelo autor (2024)

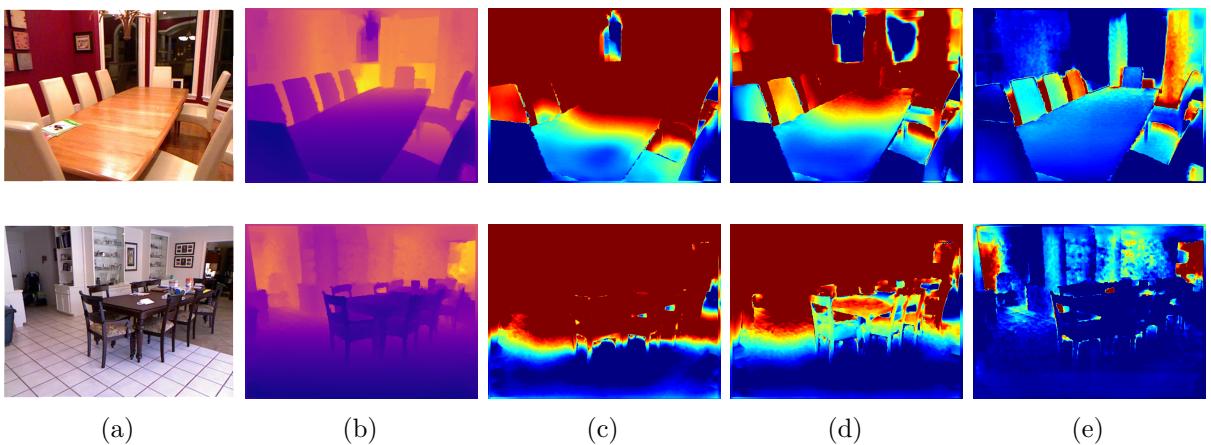
As Figuras 25 e 26 mostram as saídas das redes UNet++ com codificadores *Transformer*, exceto o MiT-B2, como discutido em capítulos anteriores. O codificador CoAtNet-2, mais uma vez, apresenta resultados com maior detalhamento e melhor separação dos objetos em ambas as imagens. Embora o CoAtNet-3 consiga separar as cadeiras na segunda imagem, suas saídas são mais grosseiras. Por fim, o codificador CoaT-Lite Medium gera saídas menos refinadas e com mais erros de estimativa em comparação ao CoAtNet-2, mas ainda é capaz de separar os objetos na primeira imagem.

Figura 25 – Resultados qualitativos de redes UNet++ com codificadores com mecanismos de atenção. (a) Entrada. (b) *Ground truth* (saída esperada). (c) CoaT-Lite Medium. (d) CoAtNet-2. (e) CoAtNet-3.



Fonte: Elaborado pelo autor (2024)

Figura 26 – Mapas de calor de erro de redes UNet++ com codificadores com mecanismos de atenção. (a) Entrada. (b) *Ground truth* (saída esperada). (c) CoaT-Lite Medium. (d) CoAtNet-2. (e) CoAtNet-3.



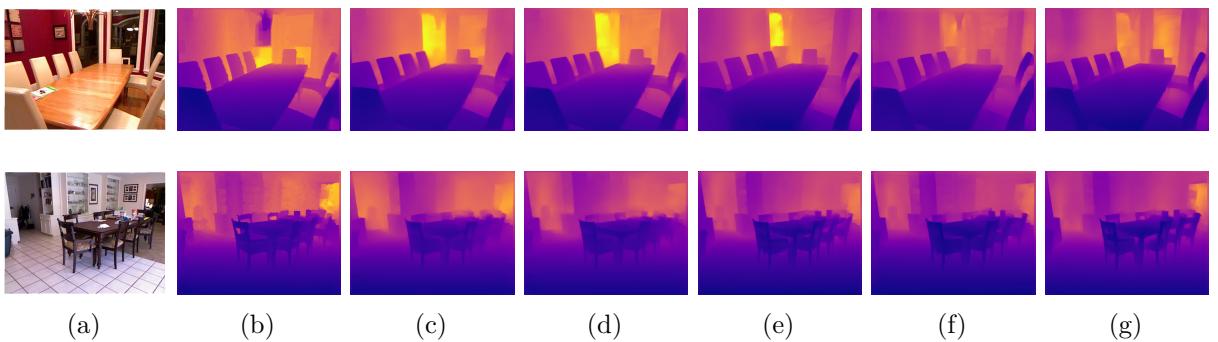
Fonte: Elaborado pelo autor (2024)

4.4.2 Análise qualitativa e quantitativa – Resoluções maiores

Os estudos de Lee et al. (2021) e Rudolph et al. (2022) sugerem que o tamanho das imagens utilizadas no treinamento pode influenciar significativamente o desempenho das redes neurais. Os autores relatam que resoluções mais altas geraram melhorias em todas as métricas de estimativa de profundidade. As Figuras 27 e 28 ilustram uma comparação qualitativa entre as redes com os três codificadores que apresentaram os maiores valores na métrica δ_1 . A primeira figura exibe os mapas de profundidade gerados por cada rede, enquanto a segunda apresenta os mapas de calor dos erros. As redes com os codificadores CoaT-Lite Medium foram treinadas com imagens de 288x384 pixels (altura x largura), enquanto as redes com CoAtNet-2 e MiT-B2 utilizaram imagens de 384x384 pixels.

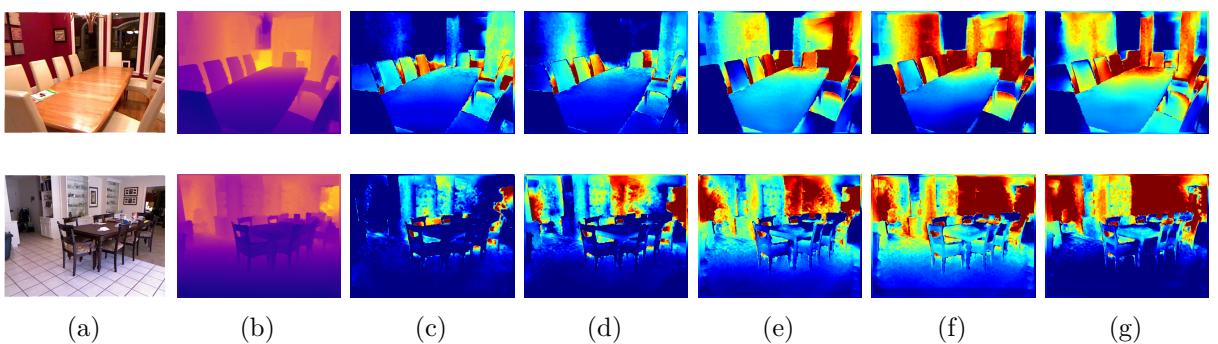
Comparando as saídas das Figuras 27 e 28 com suas versões anteriores, treinadas com entradas de 224x224 *pixels*, nota-se um maior refinamento dos objetos, superfícies mais bem definidas e maior precisão na estimativa de elementos como o piso, mesas e cadeiras em ambas as imagens. Essa melhoria evidencia que o treinamento com imagens de maior resolução contribui positivamente para a qualidade dos mapas de profundidade estimados. No entanto, mesmo com o aumento da resolução, as redes ainda enfrentam dificuldades para estimar superfícies mais distantes. O principal indicativo dessa limitação é o erro na estimativa das paredes, embora a imprecisão ao estimar as mesas também ressalte essa fraqueza em ambas as imagens. Visualmente, as redes que se destacaram apresentando erros menos significativos foram a U-Net CoAtNet-2 (e) e U-Net MiT-B2 (g).

Figura 27 – Resultados qualitativos dos melhores modelos. (a) Entrada. (b) *Ground truth* (saída esperada). (c) U-Net c/ CoaT-Lite Medium. (d) UNet++ c/ CoaT-Lite Medium. (e) U-Net c/ CoAtNet-2. (f) UNet++ c/ CoAtNet-2. (g) U-Net c/ MiT-B2.



Fonte: Elaborado pelo autor (2024)

Figura 28 – Mapas de calor de erro dos melhores modelos. (a) Entrada. (b) *Ground truth* (saída esperada). (c) U-Net c/ CoaT-Lite Medium. (d) UNet++ c/ CoaT-Lite Medium. (e) U-Net c/ CoAtNet-2. (f) UNet++ c/ CoAtNet-2. (g) U-Net c/ MiT-B2.



Fonte: Elaborado pelo autor (2024)

Para além das comparações visuais, a Tabela 3 apresenta uma comparação quantitativa dos resultados dos experimentos com tamanhos de entrada 224x224 e 384x384 *pixels* utilizando os três codificadores com maior valor de δ_1 , os quais foram CoaT-Lite, CoAtNet-2 e MiT-B2. O principal objetivo é identificar quais codificadores e versões da U-Net sofreram maior impacto pelo aumento de resolução das imagens durante o treinamento.

Ao utilizar o tamanho de entrada 224x224 era possível observar desde a Tabela 1 que a UNet++ obtinha melhores valores com o codificador CoAtNet-2 do que a versão simples da U-Net. Esse cenário não se manteve ao utilizar a resolução 384x384, a UNet++ não apenas não aproveitou o ganho de resolução, como também teve um desempenho inferior ao observado com a resolução menor. Essa queda pode ser atribuída à necessidade de reduzir o tamanho do lote, de 8 para 4 imagens, devido o aumento da complexidade do modelo e do consumo de memória.

Tabela 3 – Comparaçāo quantitativa dos trēs codificadores com melhor δ_1 treinados com maior resolução de entrada

Method	Encoder	Resolution	Batch	Params (M)	RMSE ↓	rel ↓	log10 ↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
U-Net	MiT (B2)	224 × 224	8	27,48	0,461	0,129	0,054	0,851	0,970	0,992
	CoAtNet-2	224 × 224	8	77,98	0,457	0,125	0,052	0,857	0,972	0,991
	CoaT-Lite	224 × 224	8	88,78	0,436	<u>0,115</u>	0,049	0,875	0,977	<u>0,993</u>
U-Net++	CoAtNet-2	224 × 224	8	83,99	0,444	0,120	0,051	0,867	0,974	0,992
	CoaT-Lite	224 × 224	8	92,48	0,439	<u>0,115</u>	0,050	0,873	0,975	<u>0,993</u>
U-Net	MiT (B2)	384 × 384	12	27,48	0,436	0,118	0,050	0,869	0,976	<u>0,993</u>
	CoAtNet-2	384 × 384	6	77,98	0,446	0,119	0,051	0,866	0,974	<u>0,993</u>
	CoaT-Lite	384 × 384	8	88,78	0,419	0,111	0,047	0,884	0,979	0,994
U-Net++	CoAtNet-2	384 × 384	4	83,99	0,497	0,131	0,057	0,832	0,965	0,991
	CoaT-Lite	384 × 384	8	92,48	<u>0,426</u>	0,111	<u>0,048</u>	<u>0,883</u>	<u>0,978</u>	0,994

A porcentagem de impacto do aumento de resolução pode ser conferido na Tabela 4, nela estão os valores de diferença percentual relativa dos valores obtidos pela U-Net com a resolução 384x384 e seus valores para resolução 224x224, o mesmo funcionamento se aplica a UNet++. Para métricas de erro, quanto menor o valor, melhor, para os valores de δ , quanto maior, melhor. Embora a UNet++ com CoaT-Lite tenha alcançado o segundo melhor desempenho nos trēs indicadores δ e mostrado melhorias nas métricas de erro, foi a U-Net que mais se beneficiou do aumento de resolução. Isso é evidenciado pelos seus resultados superiores com o codificador CoaT-Lite e pelo fato de que, com o MiT-B2, a U-Net mostrou ganhos mais expressivos com o aumento de resolução. Esses resultados sugerem que a U-Net aproveitou melhor as resoluções mais altas, obtendo os melhores valores absolutos nas métricas.

Tabela 4 – Diferença percentual relativa entre o desempenho de modelos U-Net e UNET++ com tamanhos 384x384 e 224x224, no dataset NYU Depth v2

Method (%)	Encoder (%)	RMSE (%) ↓	rel (%) ↓	log10 (%) ↓	$\delta_1(\%) \uparrow$	$\delta_2(\%) \uparrow$	$\delta_3(\%) \uparrow$
U-Net	MiT (B2)	-5.423	-8.527	-7.407	2.115	0.619	<u>0.101</u>
	CoAtNet-2	-2.407	<u>-4.800</u>	-1.923	1.050	0.206	0.202
	CoaT-Lite	<u>-3.899</u>	-3.478	<u>-4.082</u>	1.029	0.205	<u>0.101</u>
U-Net++	CoAtNet-2	11.937	9.167	11.765	-4.037	-0.924	-0.101
	CoaT-Lite	-2.961	-3.478	-4.000	<u>1.145</u>	<u>0.308</u>	<u>0.101</u>

5 CONCLUSÕES

Este trabalho explorou a semelhança entre as tarefas de estimativa de profundidade monocular e segmentação semântica, avaliando arquiteturas originalmente desenvolvidas para segmentação em sua aplicação à estimativa de profundidade. A reutilização de redes neurais artificiais já estabelecidas para compor uma arquitetura voltada para a estimativa de profundidade demonstra ser valiosa, especialmente ao considerar a economia de tempo, esforço humano e recursos, em comparação com a criação de uma arquitetura dedicada exclusivamente a essa tarefa.

Para compor as redes utilizadas neste trabalho, a arquitetura *encoder-decoder* U-Net e sua versão com conexões aninhadas (UNet++) foram combinadas com redes amplamente consolidadas na literatura, projetadas para extração de características em imagens, seja para tarefas de classificação ou segmentação semântica. Essas combinações permitiram avaliar o desempenho de redes convolucionais e de redes baseadas em *Transformers*, abrangendo desde arquiteturas clássicas até modelos mais recentes, com diferentes níveis de complexidade e número de parâmetros. A análise quantitativa de diversas arquiteturas do tipo codificador-decodificador indica que a escolha do codificador tem influência significativa no desempenho da estimativa de profundidade.

Os resultados obtidos pela combinação da arquitetura U-Net com a rede CoaT-Lite (M), utilizada como codificador e originalmente proposta para segmentação, são promissores e superam os das outras abordagens avaliadas. Vale ressaltar que essa análise se mantém válida mesmo quando comparada com soluções mais complexas, como as que combinam múltiplas redes ou que são multitarefas (segmentação e estimativa de profundidade, por exemplo). Como discutido no Capítulo 4, uma das possíveis hipóteses para esse destaque é a capacidade das redes com mecanismos de atenção de capturarem informações globais da entrada e manter as características ao longo das camadas da rede com menos perdas que as redes puramente CNN.

Desse modo, os resultados obtidos neste trabalho reforçam a hipótese da similaridade entre segmentação semântica e estimativa de profundidade monocular, fornecendo pistas para o desenvolvimento de soluções mais eficazes e adequadas para casos onde há restrições de recursos computacionais, uma vez que o aprendizado supervisionado foi utilizado. Além disso, os resultados indicam que redes neurais desenvolvidas para segmentação de imagens podem ser aplicadas à estimativa de profundidade monocular, fornecendo resultados satisfatórios. O avanço dessa abordagem pode reduzir a necessidade de criar arquiteturas complexas do zero para a tarefa de estimativa de profundidade.

Para trabalhos futuros, algumas possibilidades podem ser exploradas. Entre elas, realizar experimentos com redes mais complexas da família Mixed Transformer, pré-treinar os

modelos em conjuntos de dados de segmentação semântica antes de aplicá-los à estimativa de profundidade, usar dados adicionais de estimativa de profundidade antes do treinamento final, aumentar o número de imagens por lote e utilizar resoluções maiores no treinamento. Além desses ajustes, a utilização de outras redes baseadas em *Transformers* que possam ser usadas como codificadores também é um ponto relevante para explorar a viabilidade de reutilizar redes já estabelecidas na construção de uma arquitetura eficaz para estimativa de profundidade.

REFERÊNCIAS

- ABDULWAHAB, Saddam et al. Deep monocular depth estimation based on content and contextual features. *Sensors*, MDPI AG, v. 23, n. 6, p. 2919, mar. 2023. ISSN 1424-8220. Disponível em: <<http://dx.doi.org/10.3390/s23062919>>.
- ALHASHIM, Ibraheem; WONKA, Peter. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941, 2018. Disponível em: <<https://arxiv.org/abs/1812.11941>>.
- BIANCO, Simone et al. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, v. 6, p. 64270–64277, 2018.
- CAO, Kaili; ZHANG, Xiaoli. An improved res-unet model for tree species classification using airborne high-resolution images. *Remote Sensing*, v. 12, n. 7, 2020. ISSN 2072-4292. Disponível em: <<https://www.mdpi.com/2072-4292/12/7/1128>>.
- CARION, Nicolas et al. End-to-end object detection with transformers. In: *Computer Vision – ECCV 2020*. Springer International Publishing, 2020. p. 213–229. Disponível em: <https://doi.org/10.1007/978-3-030-58452-8_13>.
- CHEN, Jieneng et al. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. 2021.
- CHEN, Liang-Chieh et al. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2017. Disponível em: <<https://arxiv.org/abs/1606.00915>>.
- CHOLLET, Francois. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017.
- DAI, Zihang et al. *CoAtNet: Marrying Convolution and Attention for All Data Sizes*. 2021. Disponível em: <<https://arxiv.org/abs/2106.04803>>.
- DOSOVITSKIY, Alexey et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. Disponível em: <<https://arxiv.org/abs/2010.11929>>.
- EIGEN, David; PUHRSCH, Christian; FERGUS, Rob. *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network*. 2014.
- FU, Huan et al. *Deep Ordinal Regression Network for Monocular Depth Estimation*. 2018.
- GHANNAY, Sahar et al. Word embedding evaluation and combination. In: CALZOLARI, Nicoletta et al. (Ed.). *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), 2016. p. 300–305. Disponível em: <<https://aclanthology.org/L16-1046>>.
- GODARD, Clement et al. Digging into self-supervised monocular depth estimation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019.
- GUO, Jianyuan et al. *CMT: Convolutional Neural Networks Meet Vision Transformers*. 2022. Disponível em: <<https://arxiv.org/abs/2107.06263>>.

- GUZZO, Luiz; GAZOLLI, Kelly. Utilizando a arquitetura unet++ na estimativa de profundidade monocular. In: *Anais do L Seminário Integrado de Software e Hardware*. Porto Alegre, RS, Brasil: SBC, 2023. p. 131–142. ISSN 2595-6205. Disponível em: <<https://sol.sbc.org.br/index.php/semish/article/view/25068>>.
- HAN, Kyuseo; HONG, Kihyun. Geometric and texture cue based depth-map estimation for 2d to 3d image conversion. In: *2011 IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.: s.n.], 2011. p. 651–652.
- HE, Kaiming et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.
- HE, Lei et al. SOSD-net: Joint semantic object segmentation and depth estimation from monocular images. *Neurocomputing*, Elsevier BV, v. 440, p. 251–263, jun. 2021. Disponível em: <<https://doi.org/10.1016/j.neucom.2021.01.126>>.
- HE, Lei; WANG, Guanghui; HU, Zhanyi. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, v. 27, n. 9, p. 4676–4689, 2018.
- HE, Nanjun; FANG, Leyuan; PLAZA, Antonio. Hybrid first and second order attention unet for building segmentation in remote sensing images. *Science China Information Sciences*, Springer Science and Business Media LLC, v. 63, n. 4, mar. 2020. Disponível em: <<https://doi.org/10.1007/s11432-019-2791-7>>.
- HONG, Danfeng et al. Spectralformer: Rethinking hyperspectral image classification with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, v. 60, p. 1–15, 2022.
- HU, Jie et al. *Squeeze-and-Excitation Networks*. 2019. Disponível em: <<https://arxiv.org/abs/1709.01507>>.
- HUANG, Huimin et al. *UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation*. 2020.
- HUANG, Renyong; SUN, Mingyi. Network algorithm real-time depth image 3d human recognition for augmented reality. *Journal of Real-Time Image Processing*, Springer Science and Business Media LLC, v. 18, n. 2, p. 307–319, nov. 2020. Disponível em: <<https://doi.org/10.1007/s11554-020-01045-z>>.
- HUYNH, Lam et al. Guiding monocular depth estimation using depth-attention volume. In: _____. *Computer Vision – ECCV 2020*. Springer International Publishing, 2020. p. 581–597. ISBN 9783030585747. Disponível em: <http://dx.doi.org/10.1007/978-3-030-58574-7_35>.
- ITOH, Hayato et al. Unsupervised colonoscopic depth estimation by domain translations with a lambertian-reflection keeping auxiliary task. *International Journal of Computer Assisted Radiology and Surgery*, Springer Science and Business Media LLC, v. 16, n. 6, p. 989–1001, maio 2021. Disponível em: <<https://doi.org/10.1007/s11548-021-02398-x>>.
- JUNG, Yong Ju et al. A novel 2d-to-3d conversion technique based on relative height-depth cue. In: WOODS, Andrew J.; HOLLIMAN, Nicolas S.; MERRITT, John O. (Ed.). *SPIE Proceedings*. SPIE, 2009. Disponível em: <<https://doi.org/10.1117/12.806058>>.

- KHAN, Salman et al. Transformers in vision: A survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 54, n. 10s, sep 2022. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/3505244>>.
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *Nature*, Springer Science and Business Media LLC, v. 521, n. 7553, p. 436–444, maio 2015. ISSN 1476-4687. Disponível em: <<http://dx.doi.org/10.1038/nature14539>>.
- LEE, Jin Han et al. *From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation*. 2021. Disponível em: <<https://arxiv.org/abs/1907.10326>>.
- LIN, Tsung-Yi et al. *Feature Pyramid Networks for Object Detection*. 2017. Disponível em: <<https://arxiv.org/abs/1612.03144>>.
- LIU, Xingtong et al. Dense depth estimation in monocular endoscopy with self-supervised learning methods. *IEEE Transactions on Medical Imaging*, Institute of Electrical and Electronics Engineers (IEEE), v. 39, n. 5, p. 1438–1447, maio 2020. Disponível em: <<https://doi.org/10.1109/tmi.2019.2950936>>.
- MERTAN, Alican; DUFF, Damien Jade; UNAL, Gozde. Single image depth estimation: An overview. *Digital Signal Processing*, v. 123, p. 103441, 2022. ISSN 1051-2004. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1051200422000586>>.
- MING, Yue et al. Deep learning for monocular depth estimation: A review. *Neurocomputing*, Elsevier BV, v. 438, p. 14–33, maio 2021. Disponível em: <<https://doi.org/10.1016/j.neucom.2020.12.089>>.
- PILLAI, Sudeep; AMBRUŞ, Rareş; GAIDON, Adrien. Superdepth: Self-supervised, super-resolved monocular depth estimation. In: *2019 International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2019. p. 9250–9256.
- POGGI, Matteo et al. *On the Synergies between Machine Learning and Binocular Stereo for Depth Estimation from Images: a Survey*. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2004.08566>>.
- PUNN, Narinder Singh; AGARWAL, Sonali. Modality specific u-net variants for biomedical image segmentation: a survey. *Artificial Intelligence Review*, Springer Science and Business Media LLC, v. 55, n. 7, p. 5845–5889, mar. 2022. Disponível em: <<https://doi.org/10.1007/s10462-022-10152-1>>.
- QI, Xiaojuan et al. Geonet: Geometric neural network for joint depth and surface normal estimation. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 283–291.
- RAMAMONJISOA, Michaël et al. *Single Image Depth Prediction with Wavelet Decomposition*. 2021. Disponível em: <<https://arxiv.org/abs/2106.02022>>.
- RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-net: Convolutional networks for biomedical image segmentation. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015. p. 234–241. Disponível em: <https://doi.org/10.1007/978-3-319-24574-4_28>.

- RUDOLPH, Michael et al. Lightweight monocular depth estimation through guided decoding. In: *2022 International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2022. p. 2344–2350.
- SANDLER, Mark et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. Disponível em: <<https://arxiv.org/abs/1801.04381>>.
- SHU, Chang et al. Feature-metric loss for self-supervised learning of depth and egomotion. In: *Computer Vision – ECCV 2020*. Springer International Publishing, 2020. p. 572–588. Disponível em: <https://doi.org/10.1007/978-3-030-58529-7_34>.
- SIDDIQUE, Nahian et al. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, v. 9, p. 82031–82057, 2021.
- SILBERMAN, Nathan et al. Indoor segmentation and support inference from RGBD images. In: *Computer Vision – ECCV 2012*. Springer Berlin Heidelberg, 2012. p. 746–760. Disponível em: <https://doi.org/10.1007/978-3-642-33715-4_54>.
- SIMONYAN, Karen; ZISSERMAN, Andrew. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.
- SZEGEDY, Christian et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2017. (AAAI'17), p. 4278–4284.
- SZEGEDY, Christian et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. Disponível em: <<https://arxiv.org/abs/1512.00567>>.
- TANG, Mengxia et al. Encoder-decoder structure with the feature pyramid for depth estimation from a single image. *IEEE Access*, v. 9, p. 22640–22650, 2021.
- TAYE, Mohammad Mustafa. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers*, v. 12, n. 5, 2023. ISSN 2073-431X.
- TSENG, Sheng-Po; LAI, Shang-Hong. Accurate depth map estimation from video via mrf optimization. In: *2011 Visual Communications and Image Processing (VCIP)*. [S.l.: s.n.], 2011. p. 1–4.
- ULKU, Irem; AKAGÜNDÜZ, Erdem. A survey on deep learning-based architectures for semantic segmentation on 2d images. *Applied Artificial Intelligence*, Taylor & Francis, v. 36, n. 1, p. 2032924, 2022. Disponível em: <<https://doi.org/10.1080/08839514.2022.2032924>>.
- VASWANI, Ashish et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- WELBERS, Kasper; ATTEVELDT, Wouter Van; BENOIT, Kenneth. Text analysis in r. *Communication Methods and Measures*, Informa UK Limited, v. 11, n. 4, p. 245–265, out. 2017. ISSN 1931-2466. Disponível em: <<http://dx.doi.org/10.1080/19312458.2017.1387238>>.

- XIAO, Yi et al. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers (IEEE), v. 23, n. 1, p. 537–547, jan. 2022. Disponível em: <<https://doi.org/10.1109/tits.2020.3013234>>.
- XIE, Enze et al. Segformer: Simple and efficient design for semantic segmentation with transformers. In: BEYGELZIMER, A. et al. (Ed.). *Advances in Neural Information Processing Systems*. [s.n.], 2021. Disponível em: <<https://openreview.net/forum?id=OG18MI5TRL>>.
- XIE, Jiaxin et al. *Video Depth Estimation by Fusing Flow-to-Depth Proposals*. arXiv, 2019. Disponível em: <<https://arxiv.org/abs/1912.12874>>.
- XU, Jin et al. Reluplex made more practical: Leaky relu. In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020. Disponível em: <<http://dx.doi.org/10.1109/ISCC50000.2020.9219587>>.
- XU, Weijian et al. *Co-Scale Conv-Attentional Image Transformers*. 2021. Disponível em: <<https://arxiv.org/abs/2104.06399>>.
- YAN, Han et al. Single image depth estimation with normal guided scale invariant deep convolutional fields. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 29, n. 1, p. 80–92, 2019.
- ZEILER, Matthew D.; FERGUS, Rob. Visualizing and understanding convolutional networks. In: _____. *Computer Vision – ECCV 2014*. Springer International Publishing, 2014. p. 818–833. ISBN 9783319105901. Disponível em: <http://dx.doi.org/10.1007/978-3-319-10590-1_53>.
- ZHANG, Zhenyu et al. Joint task-recursive learning for semantic segmentation and depth estimation. In: *Computer Vision – ECCV 2018*. Springer International Publishing, 2018. p. 238–255. Disponível em: <https://doi.org/10.1007/978-3-030-01249-6_15>.
- ZHOU, Zongwei et al. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, Institute of Electrical and Electronics Engineers (IEEE), v. 39, n. 6, p. 1856–1867, jun. 2020. Disponível em: <<https://doi.org/10.1109/tmi.2019.2959609>>.