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

Matheus Coutinho Cavalcante

SEGMENTAÇÃO DE IMAGENS DE LESÕES DE PELE USANDO A REDE NEURAL CONVOLUCIONAL U-NET

Vitória-ES

Julho/2019

Matheus Coutinho Cavalcante

SEGMENTAÇÃO DE IMAGENS DE LESÕES DE PELE USANDO A REDE NEURAL CONVOLUCIONAL U-NET

Parte manuscrita do Projeto de Graduação do aluno Matheus Coutinho Cavalcante, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Vitória-ES

Julho/2019

Matheus Coutinho Cavalcante

SEGMENTAÇÃO DE IMAGENS DE LESÕES DE PELE USANDO A REDE NEURAL CONVOLUCIONAL U-NET

Parte manuscrita do Projeto de Graduação do aluno Matheus Coutinho Cavalcante, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovado em 19 de julho de 2019.

COMISSÃO EXAMINADORA:

Prof. Dr. Jorge Leonid Aching Samatelo
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Patrick Marques Ciarelli
Universidade Federal do Espírito Santo
Examinador

Prof. Msc. Philipe Rangel Demuth
Universidade Federal do Espírito Santo
Examinador

Vitória-ES

Julho/2019

AGRADECIMENTOS

Aos meus pais, Pedro e Marilis pelo amor e pelo apoio dados a mim durante todo o período da graduação.

Ao meu irmão Lucas por todo apoio emocional e por me servir de inspiração.

Ao meu orientador Jorge pela ajuda dada durante a realização deste trabalho e por despertar meu interesse por esse tema fascinante.

À banca examinadora pela aceitação do convite e pelo tempo investido para leitura e avaliação desse trabalho.

RESUMO

No Brasil, estima-se que o câncer de pele é responsável por volta de 33% de todos os diagnósticos de câncer, sendo registrado pelo Instituto Nacional do Câncer (INCA) cerca de 180 mil novos casos somente em 2018. Destes casos, 6.260 foram causados por melanoma, que é o mais letal entre os cânceres de pele. Desta forma, a detecção em estágios iniciais é de extrema importância. Sendo assim, para aumentar a rapidez e a eficiência dos diagnósticos, a utilização de métodos automáticos de diagnóstico se faz necessário. No entanto, os algoritmos automatizados não superam muito os diagnósticos mais comuns. Atualmente, existem métodos que trouxeram grandes avanços na detecção de melanoma em imagens dermatoscópicas.Este trabalho faz uso de redes convolucionais para solucionar o problema de segmentação de lesões de pele utilizando o banco de dados do desafio ISIC 2018. Foram utilizadas técnicas de pré-processamento de imagens para auxiliar na tarefa de segmentação semântica, assim como uso de *transfer learning* para que haja uma convergência mais rápida dos resultados, utilizando a U-Net, rede bastante consolidada no problema de segmentação semântica. Na etapa de avaliação, foi obtido um resultado de 0.772 com o índice de Jaccard limiarizado, estimando-se na décima quinta colocação no desafio ISIC 2018.

Palavras-chave: *Deep-learning*; Redes neurais convolucionais; Segmentação Semântica; *Transfer-learning*; Melanoma; Câncer de Pele.

ABSTRACT

In Brazil, it is estimated that skin cancer accounts for around 33 % of all diagnoses of cancer, being registered by the National Cancer Institute (INCA) around of 180,000 new cases only in 2018. Of these cases, 6,260 were caused by melanoma, which is the most lethal among skin cancers. Thus, detection in the early stages is extremely important. Therefore, to increase the speed and efficiency of diagnostics, the use of automatic diagnostic methods becomes necessary. However, automated algorithms do not far outweigh the most common diagnostics. Currently, there are methods that have made great strides in the detection of melanoma in dermatoscopic images. This work makes use of convolutional networks to solve the problem of segmentation of skin lesions using the ISIC 2018 challenge database. Image preprocessing techniques were used to help in the task of semantic segmentation, as well as the usage of transfer learning to achieve a faster convergence of results, using U-Net, a network strongly consolidated in the semantic segmentation problem. In the evaluation stage, a result of 0.772 was obtained with the threshold Jaccard index, being estimated in the fifteenth place in the ISIC 2018 challenge.

Keywords: Deep-learning; Convolutional Neural Networks; Transfer Learning; Semantic Segmentation; Skin Cancer; Melanoma.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Program Interface*
BCE	*Binary Cross Entropy*
BN	*Batch Normalization*
CNN	*Convolutional Neural Network*
DL	*Deep Learning*
GPU	*Graphics Processing Unit*
GT	*Ground Truth*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
IoU	*Intersection over Union*
ISIC	*International Skin Imaging Collaboration*
ML	*Machine Learning*
ReLU	*Rectified Linear Unit*
SGD	*Stochastic Gradient Descent*
SOL	*Structured Output Learning*
TF	*Transfer Learning*
UFES	Universidade Federal do Espírito Santo

SUMÁRIO

1 INTRODUÇÃO

1.1 Apresentação e Objeto de Pesquisa

No Brasil, estima-se que o câncer de pele é responsável por volta de 33% de todos os diagnósticos de câncer, sendo registrado pelo Instituto Nacional do Câncer (INCA) cerca de 180 mil novos casos somente em 2018. Destes casos, 6.260 foram causados por melanoma (INCA, 2017), que é o mais letal entre os cânceres de pele e é responsável por mais de 70% das mortes no Brasil (DIMATOS et al., 2009). Estima-se que 1.794 mortes por melanoma cutâneo em 2015 no país, sendo 1.012 homens e 782 mulheres (INCA, 2017).

A inspeção visual é a forma inicial de diagnóstico de câncer de pele e pode ser aprimorada por análise dermatoscópica. A dermatoscopia, também conhecida como microscopia de superfície amplificada, é uma técnica que utiliza um amplificador óptico para ampliar a captura da imagem de uma lesão, o que facilita a inspeção visual da superfície da pele, podendo melhor detectar estruturas como pigmentos, pontos, glóbulos e entre outras estruturas de pele (BRAUN et al., 2005). No entanto, a distinção entre melanoma e outras estruturas de pele, como nevos melanocíticos (pintas), pode ser difícil em estágios iniciais, podendo confundir o especialista em uma análise inicial. Sem auxílio de dermatoscopia, a acurácia de um diagnóstico por inspeção visual é aproximadamente 60%, podendo aumentar para 75%-84% com auxílio de dermatoscopia por profissionais treinados (KITTLER et al., 2017) (JAIN; JAGTAP; PISE, 2015).

Uma detecção rápida é de extrema importância para o tratamento de melanoma cutâneo, visto que a taxas de sobrevivência decai de 99% se detectados em estágios iniciais para 14% em estágios mais avançados (ESTEVA et al., 2017). Dessa forma, para aumentar a rapidez do diagnóstico, a utilização de métodos para diagnósticos automáticos se faz necessário na melhoria na detecção e na cura da doença, visto que um computador pode extrair informações que podem não ser facilmente percebidos por humanos, como assimetria ou características da textura, com rapidez e precisão no diagnóstico.

De forma geral, os métodos de diagnósticos automatizados seguem os seguintes processamentos: (*i*) aquisição da imagem de lesão de pele; (*ii*) segmentação da região da pele em que a lesão se encontra; (*iii*) extração de características da mancha da lesão e (*iv*) classificação das características obtidas. A segmentação é o processo que separa a região da lesão das demais regiões da pele e, dentre os procedimentos de diagnósticos automatizados, pode ser considerado fundamental, pois as etapas posteriores dependem do resultado da segmentação. Algumas técnicas de segmentação de imagens podem ser classificadas como: (*i*) limiarização de histograma de intensidade, que envolvem a determinação de um

valor limiar de intensidade que ira separar objetos do fundo; (*ii*) *clustering*, que utilizam algoritmos de *clustering* para particionar um espaço de cores em regiões homogêneas; (*iii*) morfológicos, que envolvem detecção de contorno de objetos usando *watershed* e entre outras (CELEBI et al., 2009).

Outra abordagem esta baseada em *Machine Learning* - ML (tradução livre, aprendizado de máquinas), que objetiva a elaboração de algoritmos que tem a capacidade de adquirir conhecimento por si mesmo ao extrair certos padrões a partir de dados e efetuar uma tarefa de predição (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma subárea de ML que está sendo utilizada em diversas aplicações é *Deep Learning* - DL (tradução livre, aprendizado profundo). Em DL, são usados modelos organizados em camadas, onde cada camada utiliza os dados de saída da camada anterior como dados de entrada. Dessa forma, uma camada posterior na rede é resultado de transformações lineares e não lineares da camada anteriores (LECUN; BENGIO; HINTON, 2015). Para a aplicações em processamento de imagem e visão computacional, as arquiteturas de redes neurais profundas de maior uso são as *Convolutional Neural Networks* – CNN (tradução, redes neurais convolucionais), propostas em 1998 por Yann LeCun (LECUN et al., 1998). No caso particular da tarefa de segmentação de imagens, existem diversas aplicações que utilizam redes CNN em diversos contextos, como: direção autônoma (LI et al., 2018), separação de beterraba de plantas (Milioto; Lottes; Stachniss, 2018), detecção de objetos salientes (JI; ZHANG; WU, 2018) e entre outros. Os avanços recentes das CNN abriram um ramo de possibilidades para a aplicação de redes neurais em diagnósticos médicos através de imagens com desempenhos razoáveis. Por exemplo, a aplicação de CNN em diagnósticos médicos chegou a ultrapassar o desempenho de médicos competentes na detecção de retinopatia diabética em imagens da retina (GULSHAN; PENG; AL., 2016). Com estes resultados promissores, a possibilidade de desenvolvimentos cada vez melhores utilizando CNN são grandes. Sendo assim, é proposto para este trabalho a utilização de CNN como elemento fundamental em técnicas de segmentação de imagens dermatoscópicas, como a U-Net, que é usada fortemente em aplicações em áreas médicas.

1.2 Justificativa

Existem vários estudos na literatura sobre a construção de um sistema automático baseado em DL capaz de segmentar as regiões de pele as lesões em imagens dermatoscópicas, alguns dos quais, por sua relevância, serão apresentados a seguir.

Em ***Survey on semantic segmentation using deep learning techniques*** (LATEEF; RUICHEK, 2019), os autores descrevem 10 diferentes métodos de arquiteturas de

redes neurais para realizar tarefa de segmentação semântica, como métodos baseados em *encoder*, método baseado em região e entre outros. Os autores também discutem sobre diferentes arquiteturas de CNN, como VGGNet, ResNet e entre outras, e apresentam um resumo de diferentes banco de dados disponíveis publicamente. A análise dos métodos é feita de forma detalhada e é focada nas arquiteturas e como elas alcançaram as devidas performances. O trabalho em questão serviu de inspiração na escolha das redes VGGNet e ResNet como redes classificadoras na proposta deste trabalho.

Em ***Skin lesion segmentation in clinical images using deep learning*** (Jafari et al., 2016), os autores propuseram um método baseado em DL para segmentação de lesões de pele utilizando uma CNN. Este método utiliza um container de pixels (*image patch*) para a extração de características locais e globais a nível de pixel e alimenta estes dados separadamente em duas CNN em paralelo. No final, utilizam a camada totalmente conectada para a classificação de cada pixel e são efetuados pós-processamento utilizando operações morfológicas para preenchimento de buracos na imagem segmentada.

Em ***ISIC 2017 - Skin Lesion Analysis Towards Melanoma Detection*** (BERSETH, 2017), o autor apresenta sua solução para o desafio ISIC 2017. O autor utilizou uma rede baseada em U-net com pré-processamento de redimensionamento de imagens para 192×192. Também foi utilizado o processo de *data augmentation* para aplicação de rotação, aproximação (*zoom*) e espelhamento das imagens. O autor utilizou a *cross-validation* para treinar o modelo e o índice de Jaccard para a medição no conjunto de validação.

Em ***Skin Lesion Analysis towards Melanoma Detection Using Deep Learning Network*** (LI; SHEN, 2018), os autores apresentam uma solução para a segmentação de lesão de imagens utilizando um *emsemble* de duas *fully convolutional residual networks* - FCRN (em tradução livre, redes residuais totalmente convolicionais) treinadas em diferentes base de dados. Utilizou-se um redimensionamento para cada FCRN separadamente, nos quais foram 500×375 e 300×225 respectivamente e também foi utilizado *data augmentation* com a operação de rotação.

Em ***Deep-Learning Ensembles for Skin-Lesion Segmentation, Analysis, Classification: RECOD Titans at ISIC Challenge 2018*** (BISSOTO et al., 2018), os autores apresentaram uma solução para o desafio ISIC 2018, utilizando um *emsemble* de uma FusionNet e uma U-Net com uma VGG-16 com *transfer learning*. Os modelos foram treinados em banco de dados diferentes. Foi utilizado um pós-processamento de imagens para preencher buracos usando operações morfológicas. A função de custo utilizada foi uma entropia cruzada binária em conjunto com o índice de Jaccard.

1.3 Objetivos

Objetivo Geral

- O desenvolvimento de uma técnica de segmentação de imagens dermatoscópicas com lesões de pele baseada em arquiteturas CNN consolidadas da literatura especializadas para a tarefa de segmentação semântica.
- Validar e testar a técnica implementada;

1.4 Estrutura do Texto

O presente trabalho está estruturado da seguinte maneira:

- **Introdução**: este capítulo inicial tem como objetivo contextualizar o problema estudado, apresentando o problema em si e as suas soluções;
- **Embasamento Teórico**: aqui serão apresentados os assuntos tratados neste trabalho;
- **Proposta**: neste capítulo é apresentada a técnica proposta para solucionar o problema de estudo;
- **Resultados**: neste capítulo são apresentados os resultados dos experimentos, assim como uma comparação com os trabalhos da literatura;
- **Conclusão**: no capítulo final deste trabalho são apresentadas as discussões sobre os resultados, além de propor melhorias para estudos futuros.

2 EMBASAMENTO TEÓRICO

2.1 Introdução

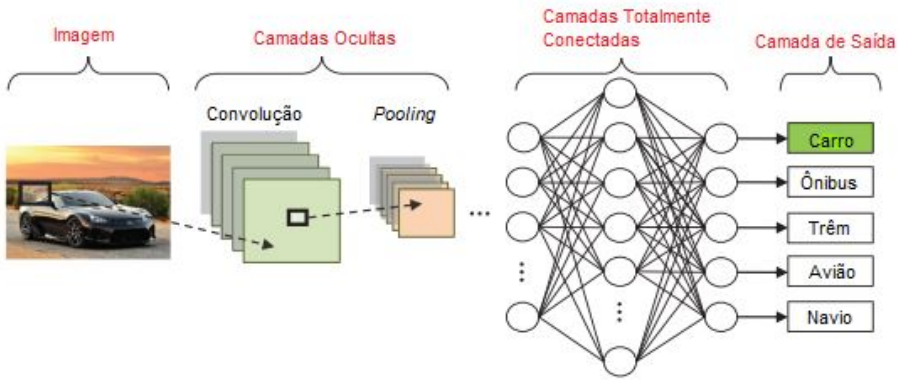
Este capítulo tem por finalidade estabelecer os conceitos teóricos necessários usados no trabalho. O capítulo se inicia com a descrição de uma Rede Neural Convolucional e como é caracterizada. Em seguida é feita uma apresentação de arquiteturas CNN bem consolidadas na literatura e do paradigma de aprendizagem *transfer learning*. Por final, é explicitado o problema de segmentação semântica e a rede neural U-Net.

2.2 Redes Neurais Convolucionais

As CNN são redes neurais profundas utilizadas no âmbito de visão computacional, as quais utilizam a convolução em pelo menos uma de suas camadas, diferentemente de outros tipos de redes, que utilizam multiplicações por matrizes ao invés da convolução (LECUN et al., 1989) (GOODFELLOW; BENGIO; COURVILLE, 2016).

De acordo com (GU et al., 2017) e (RAWAT; WANG, 2017), os componentes básicos mais comuns nas arquiteturas de CNN são: (*i*) camadas convolucionais, (*ii*) camadas de *pooling* (ou *sub-sampling*), e (*iii*) camadas totalmente conectadas. Na continuação é descrita brevemente cada uma delas. A Figura 1 ilustra a arquitetura de uma CNN para classificação de imagens.

Figura 1 – Representação visual de uma arquitetura CNN.

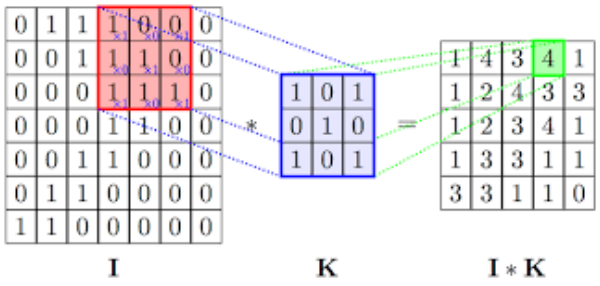


Fonte: (RAWAT; WANG, 2017).

- **A camada de Convolução**. A camada de convolução é o elemento central de uma CNN. Os parâmetros desta camada consistem de um conjunto de filtros ou *kernels*.

Os filtros convolucionais devem se deslocar ao longo dos eixos x e y da matriz na entrada por meio de passos ou *strides*. Os *strides* especificam a quantidade de passos a ser deslocada a cada operação de convolução. Cada filtro é convolvido com a entrada, produzindo mapa de características, que é a matriz de saída da camada de convolução, relacionado com o filtro. Uma explicação mais visual pode ser notada na Figura 2. O empilhamento dos diferentes mapas de características produzidos por todos filtros ao serem aplicados na entrada formam a saída da camada de convolução. Cada ponto na saída pode ser interpretado como uma saída de um neurônio que está conectado a uma região na entrada e que compartilha os parâmetros com os neurônios no mesmo mapa de ativação. Após gerados os mapas de características, aplica-se uma função de ativação não-linear.

Figura 2 – Representação visual da camada convolucional.



Fonte: (RAWAT; WANG, 2017).

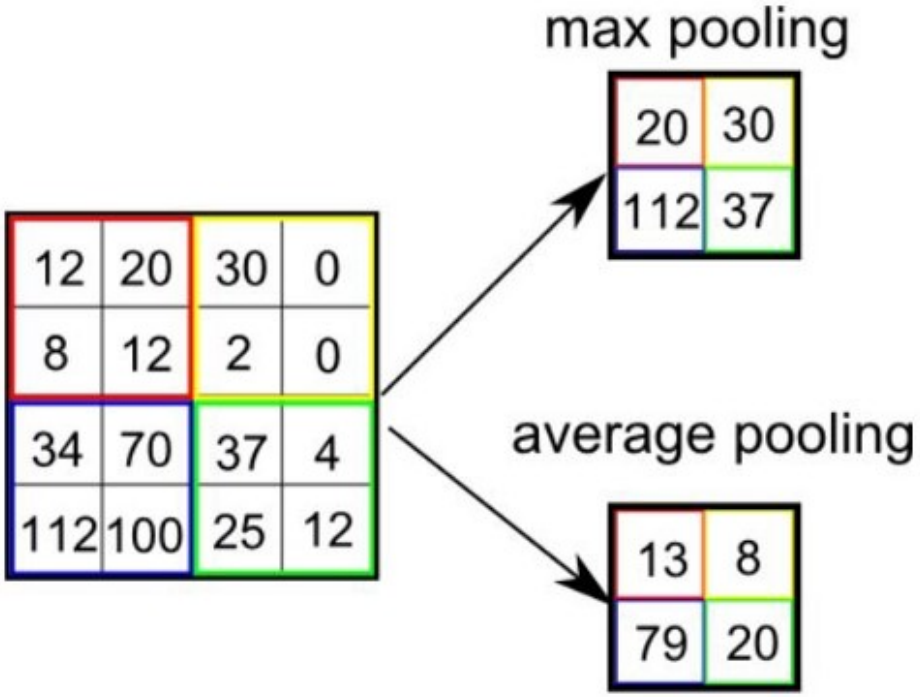
- **A camada de *Pooling***. A camada de *pooling* (tradução livre, agrupamento) comumente é utilizada entre duas camadas de convolução. A finalidade da camada de *pooling* é simplificar e reduzir a resolução espacial da informação gerada pela camada de convolução.

  Uma operação típica de *pooling* é *max pooling*, onde é retornada a ativação máxima de uma região de entrada (NIELSEN, 2015). Um exemplo de *max-pooling* pode ser observado na Figura 3, no qual tem-se uma ativação máxima de uma unidade de *pooling* considerando uma região de 2×2, assim redimensionando o mapa de características pela metade.

  A operação de *pooling* facilita a tarefa de classificação ao reter apenas ativações mais importantes, entretanto a informação espacial que está contida no campo receptivo é perdida, assim podendo gerar uma predição incorreta do objeto a ser segmentado (NOH; HONG; HAN, 2015).

- **Camada de *Unpooling***. Ao contrário da camada de *pooling*, a camada de *unpooling* gera um mapa de características com uma resolução espacial maior proveniente de um mapa de características com resolução menor. A implementação da camada de *unpooling* mitiga a redução da informação espacial que foi perdida pela operação de

Figura 3 – Representação visual da camada de *pooling*



.

Fonte: (RAWAT; WANG, 2017).

pooling em camadas anteriores por meio da reconstrução dos mapas de características para seus tamanhos originais.

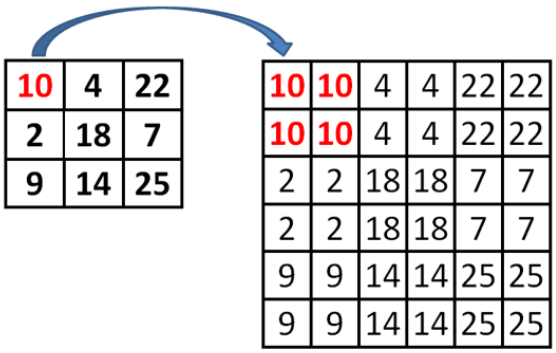
Existem diversos métodos a ser utilizados para aumentar a resolução espacial dos mapas de características. Em (LONG; SHELHAMER; DARRELL, 2015), os autores utilizam a convolução com *stride* fracionário como forma de aumentar a resolução. Esta é uma convolução que utiliza um passo fracionário, fazendo com o que o resultado da operação tenha uma resolução espacial maior que a entrada. Em (NOH; HONG; HAN, 2015), os autores guardam as localizações dos maiores elementos após as operações de *max-pooling* e, na camada de *unpooling*, inserem-os de volta na posição salva, preenchendo as outras posições com valores nulos. O método utilizado para aumentar a resolução espacial dentro da camada de *unpooling* para este trabalho é o de interpolação por vizinho mais próximo. A interpolação por vizinho mais próximo atribui o valor do vizinho mais próximo no mapa de características original a cada nova posição do novo mapa de características, de forma a aumentar sua resolução espacial e evitar o preenchimento de valores nulos em posições. A representação visual da interpolação por vizinho mais próximo pode ser observada na Figura 4.

- **A camada Totalmente conectada**. Após várias camadas convolucionais e de *pooling* empilhadas uma no topo da outra, pode-se extrair cada vez mais características de maiores níveis de abstração. A camada totalmente conectada é a camada que

toma todos os neurônios na camada anterior e os conectam a cada neurônio da camada posterior.

Por fim, posteriormente às camadas totalmente conectadas, está a camada de saída. A camada de saída é capaz de interpretar essas características e realizar funções de raciocínios de alto nível como classificação de objetos em uma imagem (RAWAT; WANG, 2017). Para a classificação, utiliza-se a função de ativação *softmax* (RUSSAKOVSKY et al., 2014). Além disso, a camada de saída possui a mesma quantidade de neurônios que o número de classes.

Figura 4 – Exemplo de *un-pooling* por interpolação por vizinho mais próximo.



Fonte: (ANGEL, 2018)

Para o treinamento das redes CNN, o algoritmo de aprendizagem mais usado é o de *backpropagation* (RUMELHART et al., 1985) (LECUN et al., 1989). Este algoritmo calcula o gradiente de uma função objetivo de modo a minimizar os erros que afetam a desempenho na tarefa da rede, ajustando os parâmetros da rede, como vetores de pesos e as bias, a cada iteração do algoritmo. Ao minimizar a função de custo, pode-se encontrar o melhor conjunto de parâmetros da rede.

2.3 Arquitetura de redes CNN

Existem na literatura arquiteturas de redes CNN que foram propostas em estudos da área, e desde então se tornaram populares graças ao desempenho comprovadamente superior em algum quesito de interesse. Essas arquiteturas foram e são aplicadas desde então para trabalhar com todo tipo de problemas de visão computacional e processamento de imagens, sendo necessário somente que sejam feitos os ajustes para preparar a rede neural para receber e processar corretamente as entradas específicas de cada problema.

Muitas das arquiteturas de sucesso, incluindo as que serão aqui utilizadas, surgem como submissões a competições *online*. Possivelmente a mais famosa competição nessa área de estudo é o *ImageNet Large Scale Visual Recognition Challenge* - ILSVRC, uma competição que ocorre anualmente desde 2010, e que premia os melhores trabalhos em tarefas de visão computacional. O ILSVRC é uma competição no campo de visão computacional que tem como objetivo promover o desenvolvimento de técnicas para detecção de objetos, localização de objetos e classificação de imagens, em imagens de larga escala. O desafio possui um banco de imagens de mais de quatorze milhões de imagens rotuladas por humanos e com mais de vinte e uma mil classes[1]. Os primeiros anos do desafio foram um marco para o campo de visão computacional, onde foram melhores sucedidas as soluções baseadas em DL, assim melhorando drasticamente o progresso no reconhecimento de objetos.

A seguir há um detalhamento das diferenciações trazidas pelas redes VGG, ResNet e DenseNet, que serão utilizadas no presente trabalho.

2.3.1 VGG

A rede VGG, proposta por (SIMONYAN; ZISSERMAN, 2014), possui uma arquitetura conformada por uma sequência de camadas de convolução com filtros de dimensão 3×3 seguidas de uma camada de *max-pooling*. A rede VGG se tornou popular na área, pois foi a rede campeã na tarefa de localização e ficou na segunda posição da tarefa de classificação da competição ILSVRC 2014.

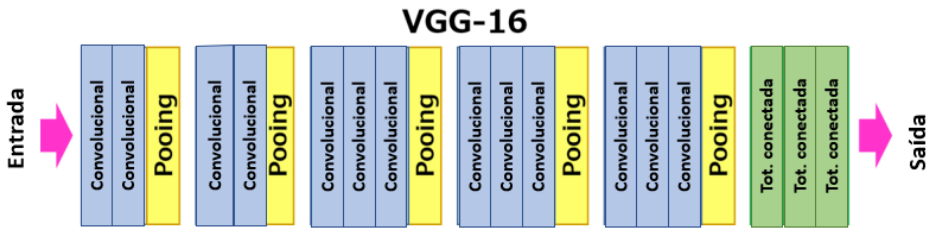
O ano em que a rede VGG venceu a competição na tarefa de localização foi o primeiro ano em que os modelos baseados em DL atingiram taxas de erro menores que 10%. Existem vários modelos atualmente que são baseados na ideia da VGG de efetuar convoluções consecutivas com filtros de 3 x 3, como por exemplo ResNet e DenseNet. A VGG pode ser visualizada na Figura 5.

Existe vários trabalhos na literatura que utilizam a VGG, como nos citados a seguir:

- Em (SUN et al., 2015), os autores utilizaram a VGG para reconhecimento facial de uma pessoa presente em uma imagem.
- Em (KE et al., 2018), é utilizada a VGG para identificar padrões de sincronização em eletroencefalogramas em pessoas com epilepsia.

[1] <http://www.image-net.org/about-stats>

Figura 5 – Representação visual da VGG-16.



Fonte: Produção do próprio autor.

- Em (Sercu et al., 2016), os autores utilizaram a VGG para reconhecimento de fala em arquivos de áudio.

Tabela 1 – Estrutura da rede VGG-16. A primeira linha indica a primeira camada da rede, enquanto que a ultima linha representa a ultima camada.

Camada	Filtros	Dimensão da saída
Convolucional	$3 \times 3 \times 64$	$224 \times 224 \times 64$
Convolucional	$3 \times 3 \times 64$	$224 \times 224 \times 64$
Max Pooling	$2 \times 2 \times 64$	$112 \times 112 \times 64$
Convolucional	$3 \times 3 \times 128$	$112 \times 112 \times 128$
Convolucional	$3 \times 3 \times 128$	$112 \times 112 \times 128$
Max Pooling	$2 \times 2 \times 128$	$56 \times 56 \times 128$
Convolucional	$3 \times 3 \times 256$	$56 \times 56 \times 256$
Convolucional	$3 \times 3 \times 256$	$56 \times 56 \times 256$
Convolucional	$3 \times 3 \times 256$	$56 \times 56 \times 256$
Max Pooling	$2 \times 2 \times 256$	$28 \times 28 \times 256$
Convolucional	$3 \times 3 \times 256$	$28 \times 28 \times 512$
Convolucional	$3 \times 3 \times 256$	$28 \times 28 \times 512$
Convolucional	$3 \times 3 \times 256$	$28 \times 28 \times 512$
Max Pooling	$2 \times 2 \times 512$	$14 \times 14 \times 512$
Convolucional	$3 \times 3 \times 512$	$14 \times 14 \times 512$
Convolucional	$3 \times 3 \times 512$	$14 \times 14 \times 512$
Convolucional	$3 \times 3 \times 512$	$14 \times 14 \times 512$
Max Pooling	$2 \times 2 \times 512$	$7 \times 7 \times 512$
Totalmente conectada	$1 \times 1 \times 4096$	$1 \times 1 \times 4096$
Totalmente conectada	$1 \times 1 \times 4096$	$1 \times 1 \times 4096$
Totalmente conectada	$1 \times 1 \times 4096$	$1 \times 1 \times 1000$

2.3.2 ResNet

Modelos de redes convolucionais com altas profundidades no que corresponde à quantidade de camadas convolucionais empilhadas, como VGG (SIMONYAN; ZISSERMAN, 2014)

e Inception (SZEGEDY et al., 2015), podem levar à conclusão de que quanto maior a profundidade da rede, melhor será o resultado. No entanto, a partir de uma certa profundidade, o rendimento da rede diminui por causa do problema de *vanishing gradient* (tradução livre, desvanecimento do gradiente). Quando a profundidade da rede é muito alta, os gradientes da função de custo calculados para os pesos na rede via *backpropagation* tendem a ficar cada vez menores nos pontos das camadas mais baixas da rede, impedindo o aprendizado da rede.

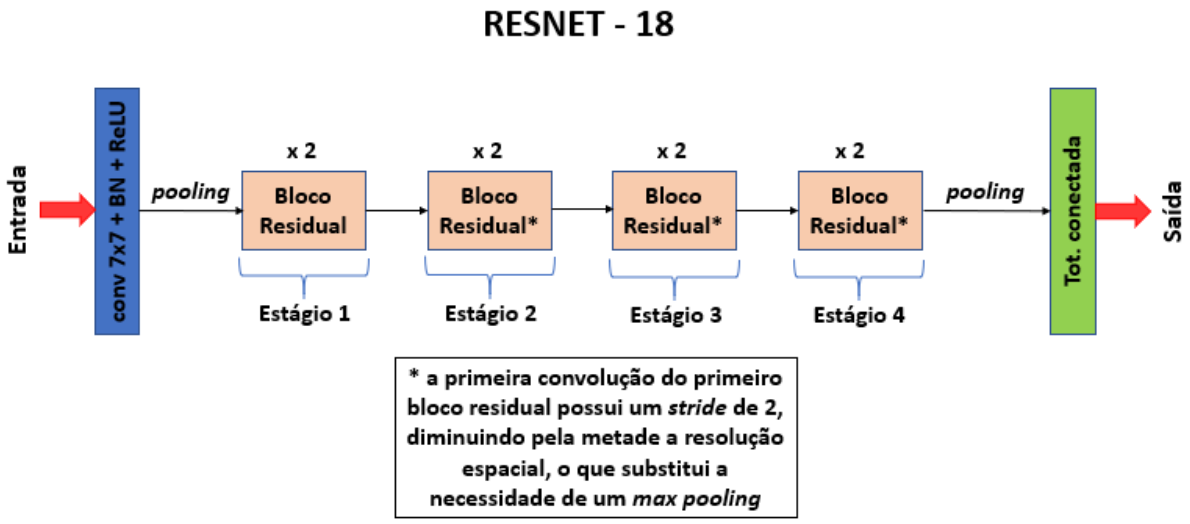
A rede ResNet, proposta por (HE et al., 2016), utiliza blocos residuais na sua arquitetura. Dentro destes blocos, existem *skip connections* (tradução livre, conexões de pulo). A arquitetura da rede completa está apresentada na Figura 6. Nos *skip connections*, o mapeamento feito é via operação de identidade e conecta a saída da camada anterior para a camada posterior, enquanto que no mapeamento original é efetuada as operações de convolução e função de ativação ReLU. Ao final de cada bloco residual, o mapeamento feito pelas convoluções e pela função de ativação ReLU é somado junto ao mapeamento por identidade. A saída do bloco residual pode ser representada por: $\mathcal{F}(x) + x = \mathcal{H}(x)$, onde $\mathcal{F}(x)$ a função que denota as operações de convolução seguida da função de ativação ReLU, x a entrada contendo a matriz com os mapas de características e $\mathcal{H}(x)$ a saída de cada bloco residual. Este procedimento pode ser visualizado na Figura 7. Desta forma, os gradientes da função de custo conseguem fluir para as camadas iniciais via *skip connections*. É importante notar que a primeira convolução dos estágios 2, 3 e 4 da Figura 6 possuem um *stride* de 2, o que faz com que a resolução espacial diminua pela metade, substituindo assim a camada de *max-pooling*.

Existem arquiteturas da ResNet com diferentes profundidades. Desde modelos com 18 camadas de profundidade até 152 camadas.

Existe vários trabalhos na literatura que utilizam a ResNet, como nos citados a seguir:

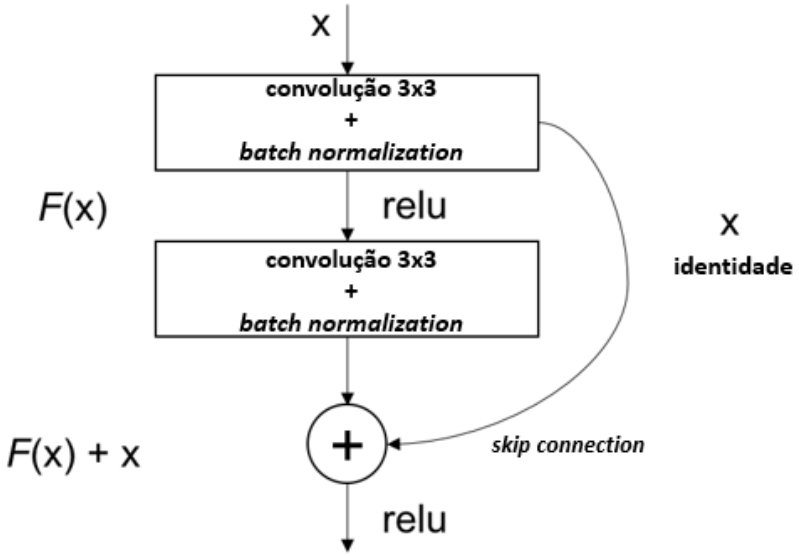
- Em (AKIBA; SUZUKI; FUKUDA, 2017), os autores utilizaram a ResNet-50 para treinar na base de dados ImageNet em apenas 15 minutos com um tamanho de *batch* de 32000.
- Em (CHEN et al., 2017), é utilizada a ResNet para identificar golpes em ligações telefônicas, detectando falsas gravações de áudio.
- Em (JUNG et al., 2017), os autores utilizaram a ResNet para classificação e localização de veículos presentes em uma imagem para sistemas de vigilância de tráfico.

Figura 6 – Representação gráfica da ResNet-18.



Fonte: Produção do próprio autor.

Figura 7 – Representação visual do bloco residual da ResNet-18.



Fonte: Adaptado de (HE et al., 2016).

2.3.3 DenseNet

A rede DenseNet, desenvolvida por (HUANG et al., 2017), representa o próximo passo no desenvolvimento de redes convolucionais ainda mais profundas do que as ResNets. Como descrito anteriormente, quanto maior a profundidade da rede convolucional, maior é o caminho para que a informação contida na camada de entrada flua para a camada de saída, fazendo com o que os gradientes, que fluem em direção oposta, sumam quando estiverem

calculando os pesos em camadas iniciais. A ResNet, descrita na seção anterior, contorna o sinal de uma camada para outra através da conexão usando a operação identidade. A DenseNet foi desenvolvida tendo como característica principal a criação de caminhos curtos que permitem o fluxo máximo de informação entre as camadas na rede, conectando todas as camadas (com mesmas as dimensões de mapas de características) diretamente cada uma com as outras.

Em redes convolucionais tradicionais, a saída da camada l é conectada na entrada da camada $(l-1)$, em que cada camada possui uma composição de operações, sendo elas: convolução 3×3 ou *max-pooling*, *batch normalization* e uma função de ativação ReLU. A equação possui a seguinte representação: $x_l = H_l(x_{l-1})$. onde $H_l(x_l)$ representa a composição de operações na camada l e x_l é a saída da camada l. As ResNets extendem este comportamento utilizando os *skip connections* com a função de identidade. Reformulando a equação, obtem-se: $x_l = H_l(x_{l-1}) + x_{l-1}$, No entanto, a soma entre a função de identidade com a saída de H_l pode influenciar negativamente no fluxo de informações na rede. Para aprimorar ainda mais o fluxo de informações na rede, utiliza-se a concatenação das camadas anteriores ao invés de somatório. Finalmente, a equação referente à um bloco de operações da rede DenseNet fica da forma:

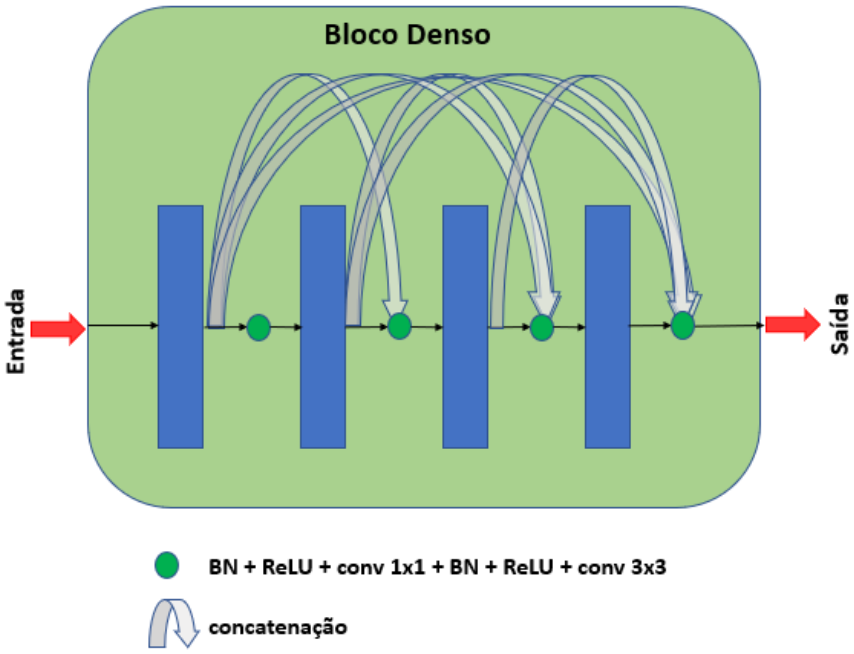
$$x_l = H_l([x_0, x_1, ..., x_{l-1}])$$

Semelhantemente à ResNet, esta concatenação de mapas de características devem ter dimensões iguais dentro de um mesmo bloco chamado de bloco denso. Entre dois blocos densos tem a camada de transição, responsável pelas diminuição do tamanho dos mapas de características. Nas camadas de transição, ocorrem uma convolução 1×1 e uma operação de *pooling*. Os blocos denso e de transição e a DenseNet podem ser visualizados nas Figuras 8, 9 e 10, respectivamente.

Existe vários trabalhos na literatura que utilizam a DenseNet, como nos citados a seguir:

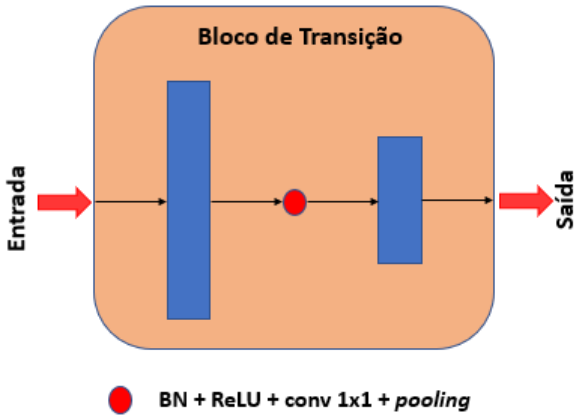
- Em (ZHANG et al., 2018), os autores utilizaram a DenseNet para reconstrução de imagens de tomografia computadorizada esparsas.
- Em (LIANG et al., 2018), os autores utilizam a DenseNet para prever o genótipo isocitrato desidrogenase em imagens de tumores.

Figura 8 – Representação visual do bloco denso da DenseNet com 4 camadas.



Fonte: Produção do próprio autor.

Figura 9 – Representação visual do bloco de transição da DenseNet.

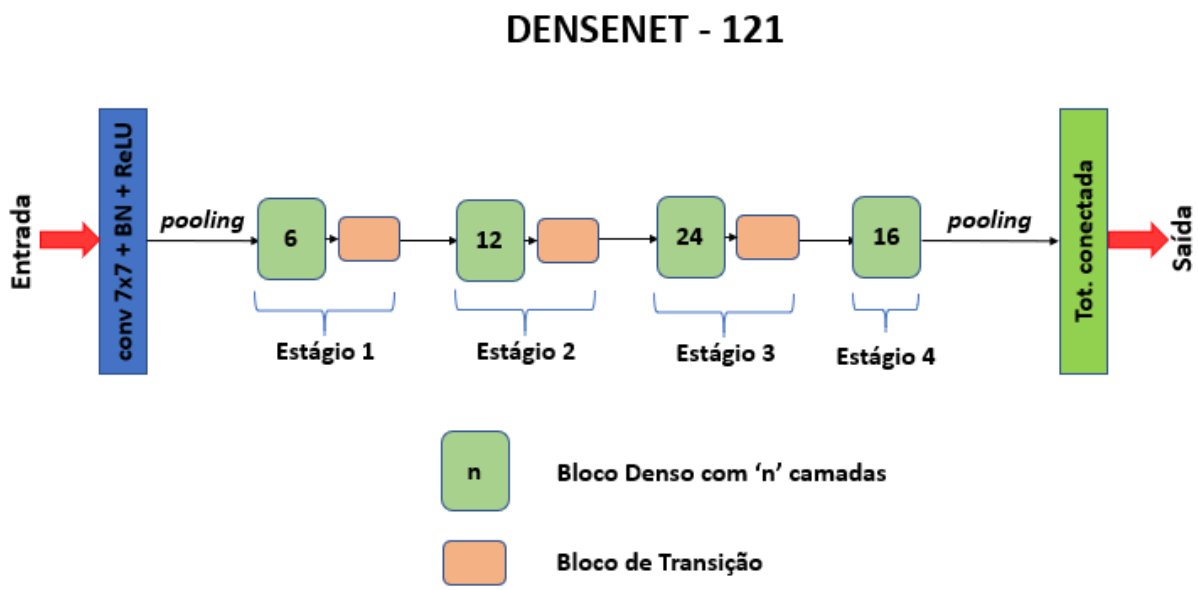


Fonte: Produção do próprio autor.

2.4 Segmentação Semântica

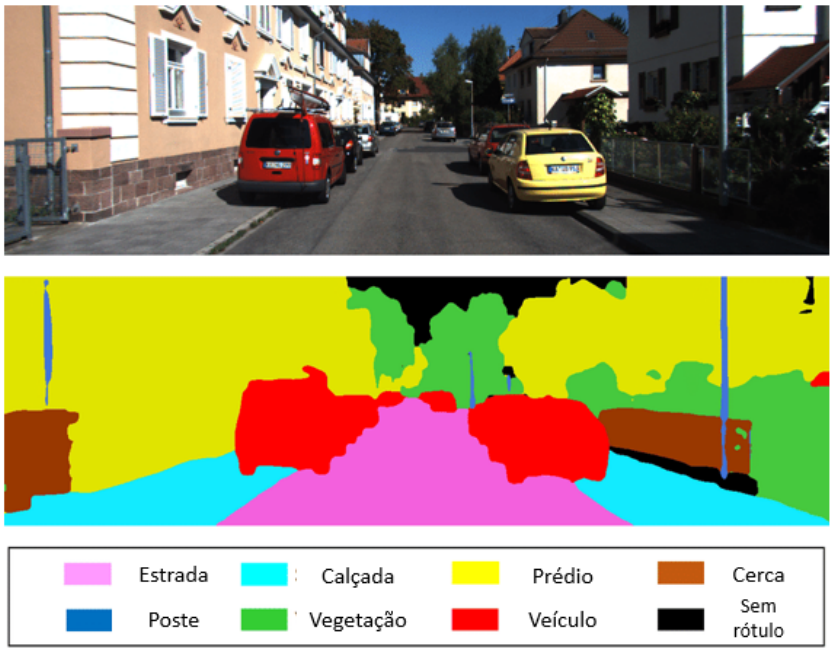
A segmentação semântica é o processo de atribuição de uma classe de objeto para cada pixel de uma imagem, como pode ser visto na Figura 11. Isto faz com que a tarefa de segmentação seja complexa e uma das poucas áreas da visão computacional em que o desempenho das redes neurais profundas estão aquém da performance humana (PLANCHE; ANDRES, 2019).

Figura 10 – Representação gráfica da DenseNet-121.



Fonte: Produção do próprio autor.

Figura 11 – Exemplo de Segmentação Semântica.



Fonte: Traduzido de (JEONG; YOON; PARK, 2018).

Para o caso da segmentação semântica, existem dois desafios de importância para a avaliação de diferentes algoritmos orientados a solucionar o problema: (*i*) *PASCAL Visual Object Classes - VOC* (EVERINGHAM et al., 2015) com 21 classes, e (*ii*) *Microsoft Common Objects in Context - COCO* (LIN et al., 2014) com mais de 80 classes diferentes.

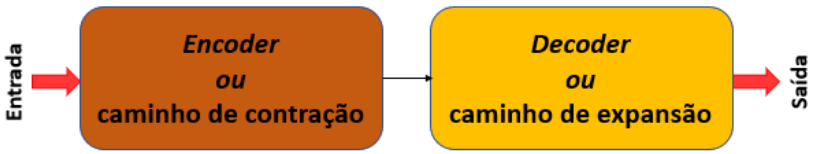
2.5 U-Net

A abordagem de uma *fully convolutional network* (em tradução livre, rede totalmente convolucional) foi introduzida por (LONG; SHELHAMER; DARRELL, 2015), em que os autores propuseram uma adaptação de redes popularizadas já existentes, que são utilizadas para classificação de imagens, transformando as camadas totalmente conectadas em camadas de convolução. Isto permite a geração de mapas de características de segmentação para cada imagem, podendo ser de qualquer tamanho.

- **Encoder**. Os mapas de características gerados possuem resolução espacial bem menores do que a imagem original. Este processo de redução da resolução espacial da imagem de entrada através de uma série de convoluções ocorre no caminho chamado de caminho de contração, ou, também, chamado de *encoder*.
- **Decoder**. Na saída do *encoder*, é feita um aumento da resolução espacial através de uma série de deconvoluções dos mapas de características para a resolução inicial da imagem de entrada. O caminho onde este processo de expansão da resolução espacial ocorre é chamado de *decoder* ou caminho de expansão.

A arquitetura de uma rede *Encoder-Decoder*, no qual a U-Net faz parte, pode ser visualizada na Figura 12.

Figura 12 – Arquitetura de uma rede *Encoder-Decoder*.



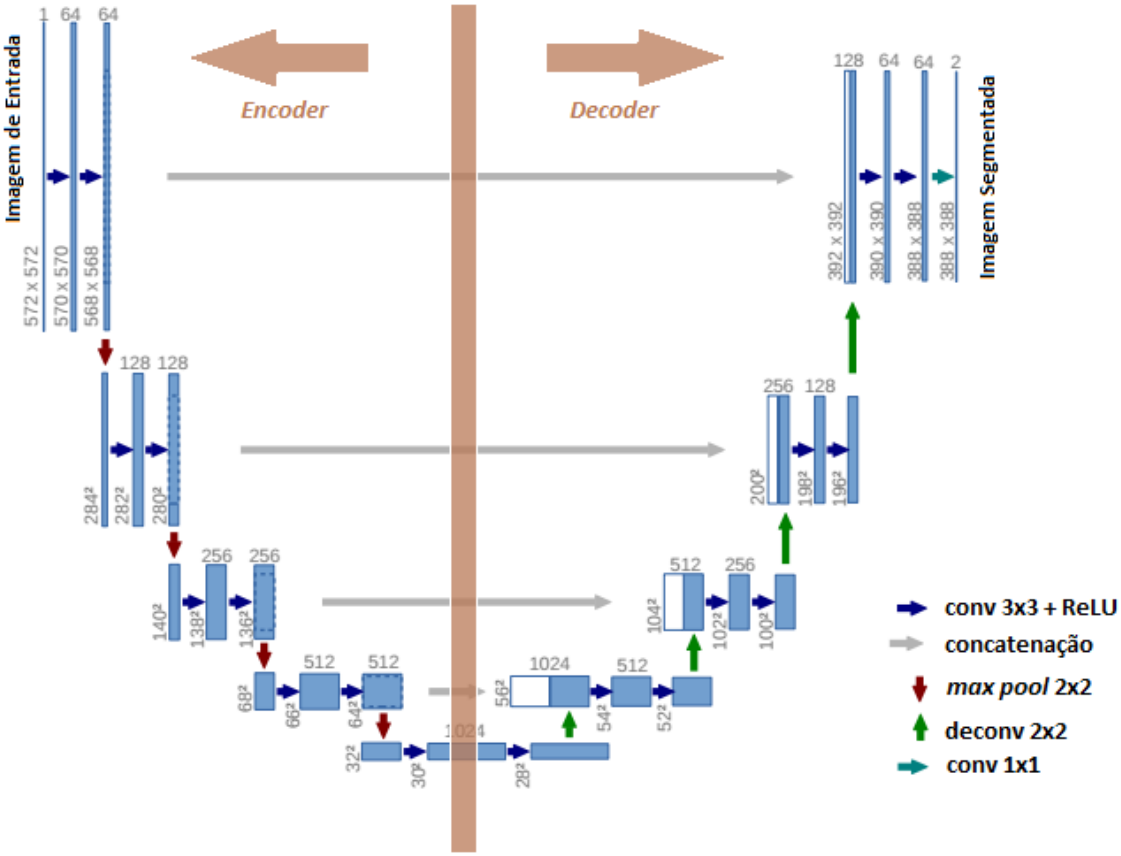
Fonte: Produção do próprio autor.

A U-Net, rede proposta por (RONNEBERGER; FISCHER; BROX, 2015), é uma versão extendida das redes totalmente convolucionais (FCN), de forma a proporcionar segmentações mais precisas com conjuntos de treinos pequenos. A principal diferença em relação à FCN se faz na camada de *decoder*, onde há um número grande de mapas de características, tendo como consequência uma simetria em relação à camada de *encoder*, assim tomando um formato de "U", como a Figura 13 demonstra. A forma adotada para aumentar o número de mapas de características no *decoder* é concatenar a saída de uma camada de *unpooling* com a saída da camada de convolução do *encoder* correspondente. Desta forma,

a rede aprende o conteúdo da informação da imagem e onde este conteúdo está localizado na imagem.

Em relação à tarefa de segmentação, é necessário que a rede neural seja capaz de combinar a informação da localização com a informação contextual dos mapas de características da imagem a ser predita. A U-Net é capaz de juntar as informações contextuais obtidas do caminho de contração (*encoder*) com as informações de localização adquiridas do caminho de expansão (*decoder*) com um bom desempenho. Sem necessidade de pré ou pós-processamento, os autores da U-Net atingiu o melhor resultado do desafio de segmentação EM (RONNEBERGER; FISCHER; BROX, 2015).

Figura 13 – Arquitetura da U-net.



Fonte: Adaptado de (RONNEBERGER; FISCHER; BROX, 2015)

Na literatura existem diversos tipos de trabalhos que utilizaram a rede U-Net, como os citados a seguir:

- Em (JANSSON et al., 2017), os autores utilizaram a U-Net para separação de diferentes fontes de áudio dado uma gravação de áudio de baixa qualidade.
- Em (ZHANG; LIU; WANG, 2018), os autores utilizaram a U-Net para segmentação semântica de estradas em imagens coletadas de trânsito de veículos.
- Em (DONG et al., 2017), os autores utilizaram a U-Net para segmentar tumores cerebrais em imagens de ressonância magnética 3D.

2.6 Resumo

Deste capítulo pode-se notar que: a segmentação de imagens foi objeto de vários estudos utilizando CNN. A utilização da U-Net com outros classificadores para a tarefa de segmentação demonstra que o campo de pesquisa de DL tem avanços contínuos, com diferentes tipos de aplicações. A seguir é apresentada a real execução do trabalho, que se utiliza a U-Net com diferentes classificadores e as etapas de pré-processamento. Portanto, os seguintes capítulos estão dedicados especificamente às etapas de processamento e segmentação.

3 PROPOSTA

3.1 Introdução

Neste capítulo a solução implementada para a tarefa de segmentação de lesões de pele é apresentada. Esta solução está subdividida em duas etapas:

- **Geração de Dados**. Nesta etapa, as imagens de lesões de pele são redimensionadas e são aplicadas transformações nos dados de forma a gerar novos dados a partir de dados antigos (*data augmentation*);
- **Segmentação Semântica via U-Net**: Aqui são descritas as modificações feitas na arquitetura da U-Net via o conceito de *backbones* e, a seguir, também é descrita a função de custo usada.

3.2 Geração de Dados

O objetivo desta etapa é a geração e preparo das imagens dermatoscópicas do conjunto de dados a um tamanho determinado. A geração de imagens consiste em aplicar operações nas imagens já existentes, como rotação e translação, de forma a gerar novas imagens dermatoscópicas e aumentar o conjunto de dados. Esta técnica é chamada de *data augmentation* (em tradução livre, incremento de dados) é essencial para melhorar o rendimento de redes CNN. A Figura 14 apresenta as operações de *data augmentation* efetuadas no conjunto de treinamento.

Após da aplicação das operação de *data augmentation*, cada imagem dermatoscópica é redimensionada a um tamanho específico. O tamanho escolhido foi de 576×576, pois esta é a dimensão original de menor tamanho das imagens contidas no conjunto de dados em estudo. Desta forma, não é necessário efetuar nenhuma expansão de imagens, que inserem dados que não estavam presentes nas imagens originais, assim podendo inserir imprecisões nos dados. Sendo assim, todas as imagens dos conjuntos de treinamento, validação e teste foram redimensionadas para este tamanho. Mais detalhes serão expostos na seção 4.2 do capitulo 4.

Figura 14 – Imagens do conjunto de treinamento com as respectivas referências antes e após a aplicação das seguintes operações de *data augmentation*: (a) imagem original; (b) rotação; (c) deslocamento; (d) *zoom*.

Fonte: Banco de Dados ISIC *Archive*

3.3 Segmentação Semântica via U-Net

O problema de segmentação semântica pode ser formulado como um problema de *Structured Output Learning* - SOL (tradução livre, aprendizado de saídas estruturadas). Para o caso em estudo, o problema de segmentação de uma imagem dermatoscópica pode ser formulado como segue: seja, $\mathcal{T} = \{(\mathbf{I}_1, \mathbf{B}_1), \cdots, (\mathbf{I}_T, \mathbf{B}_T)\}$, o conjunto de treinamento, onde $\mathbf{I}_i \in [0, 255]^{N \times M \times 3}$ é a i-ésima imagem dermatoscópica de entrada, $\mathbf{B}_i \in [0, 1]^{N \times M}$ é a imagem binária correspondente $\mathbf{I}_i$. Suponha-se que existe uma função objetivo $\mathbf{f} : [0, 255]^{N \times M \times 3} \rightarrow [0, 1]^{N \times M}$ parametrizada por $\boldsymbol{\theta}$ que relacionam cada imagem $\mathbf{I}$ com sua correspondente representação binária $\mathbf{B}$. Então, tomando em conta o conjunto de treinamento $\mathcal{T}$, deseja-se determinar os parâmetros $\boldsymbol{\theta}$ que minimizem uma função de custo L, tal como é indicado na Equação (3.1).

$$\boldsymbol{\theta}_{min} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\mathcal{T}, \boldsymbol{\theta}). \tag{3.1}$$

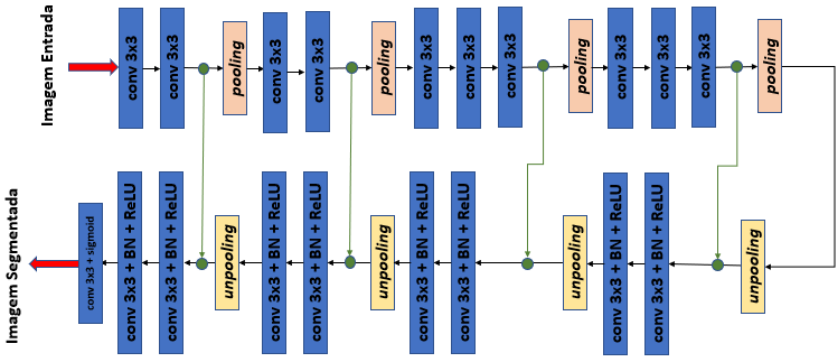
Nesta proposta, foi considerado que:

Em relação à função objetivo f. Ela vem definida pela estrutura da rede U-Net modificada. A U-Net permite combinar a arquitetura original da rede com arquiteturas de CNN orientadas para classificação, objetivando uma melhor extração de características das imagens de entrada. Tal adaptação faz uso do conceito de *Backbone* (tradução livre, espinha dorsal) na etapa do *encoder* da U-net. Entende-se por *Backbone* a arquitetura principal de uma CNN orientada a classificação excluindo-se a camada de entrada e a camada de saída. Para o caso deste trabalho, os *backbones* usados foram:

- A VGGNet com 16 camadas. A U-net modificada via a VGGNet é mostrada na Figura 15.

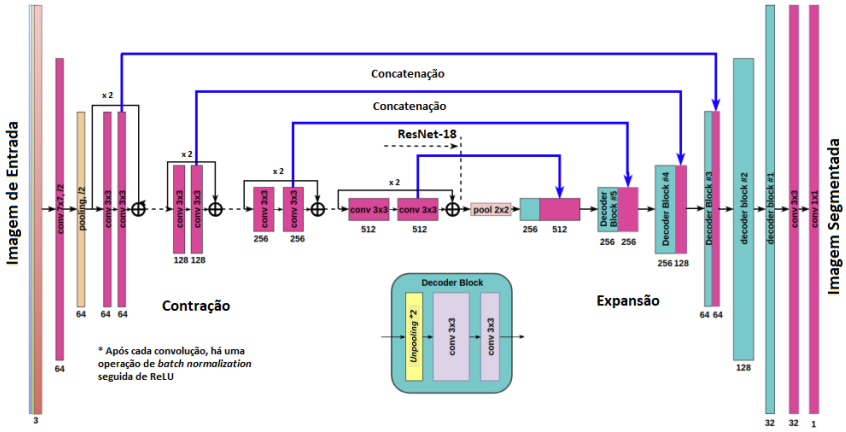
- A ResNet, com 18 camadas. A U-net modificada via a ResNet é mostrada na Figura 16.
- A DenseNet, com 121 camadas. A U-net modificada via a DenseNet é mostrada na Figura 17

Figura 15 – Arquitetura da U-net com *backbone* de VGG-16.



Fonte: Produção do próprio autor

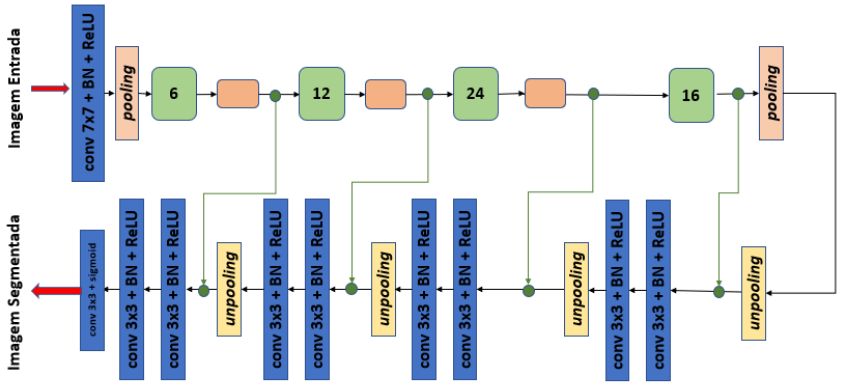
Figura 16 – Arquitetura da U-net com *backbone* de ResNet-18.



Fonte: Adaptado de (BUSLAEV et al., 2018)

Em relação à função de custo a minimizar L. Nas imagens dermatoscópicas, as lesões de pele geralmente ocupam uma parte relativamente pequena de toda a imagem, o que gera um desbalanceamento entre o número de pixels do fundo e do objeto de interesse, tal situação, geralmente faz com que o modelo $\mathbf{f}$ na etapa de treinamento fique preso em um mínimo local da função de perda L. Nesse contexto, a saída predita por $\mathbf{f}$, correspondente à saída do *decoder* da U-Net modificada, geralmente tem uma tendencia a ser valores relacionados com a classe fundo. Para contornar o problema do desbalanceamento das

Figura 17 – Arquitetura da U-net com *backbone* de DenseNet-121.



Fonte: Produção do próprio autor

classes, foram tomadas as recomendações indicadas em (ZHU et al., 2019; JIANG et al., 2019), onde, a função de perda foi definida como:

$$L(\mathbf{B}, \hat{\mathbf{O}}) = L_{BCE}(\mathbf{B}, \hat{\mathbf{O}}) + L_{IoUp}(\mathbf{B}, \hat{\mathbf{O}}), \tag{3.2}$$

onde:

$$L_{BCE}(\mathbf{B}, \hat{\mathbf{O}}) = -\frac{1}{NM}\text{sum}(\mathbf{B} \odot \ln(\hat{\mathbf{O}}) + (1 - \mathbf{B}) \odot \ln(1 - \hat{\mathbf{O}})), \tag{3.3}$$

$$L_{IoUp}(\mathbf{B}, \hat{\mathbf{O}}) = 1 - \frac{\text{sum}(\mathbf{B} \odot \hat{\mathbf{O}})}{\text{sum}(\mathbf{B} + \hat{\mathbf{O}} - \mathbf{B} \odot \hat{\mathbf{O}})}. \tag{3.4}$$

Aqui: L_{BCE} é a função de custo *Binary Cross Entropy* - BCE (tradução livre, Entropia Cruzada Binária); L_{IoUp} é a função de custo *Intersection over Union* Aproximado - IoUp (tradução livre, interseção sobre união); $\mathbf{B}$ é a imagem binária esperada; $\hat{\mathbf{O}}$ é a imagem predita (saída do *decoder* da rede U-net modificada); $\odot$ representa a multiplicação elemento a elemento (produto Hadamard); sum($\bullet$) é uma função que soma todos os elementos de um matriz; ln($\bullet$) representa a função logaritmo neperiano aplicada a cada elemento de uma matriz.

Alguns comentários:

- A função de custo L_{BCE} mede a distância entre a distribuição probabilística da saída da rede e a real distribuição da saída esperada.

- A função de custo L_{IoUp} é uma aproximação do Índice de Jaccard (JI), o qual mede a similaridade entre a representação binaria de $\hat{\mathbf{O}}$, denotada como $\hat{\mathbf{B}}$, e $\mathbf{B}$, tal que, quando $\hat{\mathbf{B}}$ e $\mathbf{B}$ são similares $JI \to 1$, caso contrario $JI \to 0$. A tendência inversa é seguida por L_{IoUp}.

- L_{IoUp} se preocupa com penalizar uma baixa intersecção entre o predito e o esperado, ou seja, é um indicador de natureza global, implicando que sua minimização permita aumentar a acurácia do modelo. enquanto, L_{BCE} trata o problema de segmentação como se fosse um problema de classificação a nível de pixels, penalizando cada pixel classificado incorretamente, implicando que sua minimização permita aumentar a precisão do modelo.

- A intensão de definir a função de perda como a soma de L_{IoUp} e L_{BCE} é poder penalizar erros de segmentação a nível global como local, com a intensão de aumentar tanto a acurácia como a precisão do modelo simultaneamente.

3.4 Resumo

Neste capítulo foi descrita a solução do problema de segmentação de lesões de pele de imagens dermatoscópicas utilizando a U-Net com diferentes classificadores e as etapas de pré-processamento, assim como diferentes parâmetros da rede utilizada. No capítulo 4 está descrito a implementação da solução, assim como os recursos utilizados e os resultados gerados a partir das métricas.

4 RESULTADOS

4.1 Introdução

Neste capítulo serão apresentados os resultados obtidos através da técnica proposta no capítulo 3. O capítulo inicia com uma descrição detalhada do banco de dados, assim como detalhes da implementação. Serão apresentadas, também, as métricas utilizadas para avaliar o desempenho da rede, assim como o resultado do experimento. Finalmente, são mostrados os melhores resultados do desafio ISIC 2018, assim como uma comparação dos resultados dos mesmos com o método proposto neste trabalho.

4.2 Base de Dados ISIC 2018

O projeto *International Skin Imaging Collaboration* - ISIC [1] é uma parceria acadêmica internacional que tem como objetivo facilitar o desenvolvimento de técnicas para a melhora no diagnóstico de melanoma. O ISIC *Archive* [2], desenvolvido pelo ISIC, é um repositório internacional de imagens dermatoscópicas de lesões de pele e possui a maior coleção de imagens de dermatocópicas disposta publicamente. Esta coleção de imagens foi avaliada e rotulada por profissionais na área e e foram coletadas dos principais centros clínicos do mundo. Em 2017, o ISIC criou o desafio *Skin Lesion Analysis Towards Melanoma Detection* (tradução livre, Análise de Lesão de Pele para Detecção de Melanoma), que é um desafio onde os participantes desenvolvem sistemas automáticos de análise de imagens para dar suporte ao diagnóstico de melanomas e, desde então, este desafio foi realizado anualmente (CODELLA et al., 2017; CODELLA et al., 2019). Este desafio é dividido em três etapas: (*i*) segmentação de lesões; (*ii*) detecção de atributos de lesões; (*iii*) classificação da doença.

A tarefa de segmentação de lesões de pele do desafio ISIC 2018 conta com uma conjunto de dados formado por 2.594 imagens de entrada para treino junto com o mesmo número imagens de referência segmentadas pela própria ISIC, ou também chamadas de *GroundTruth* - GT, com 100 imagens de entrada fornecidas para validação e com 1.000 imagens de entrada para testes. Todas as imagens de entrada possuem um formato JPEG e as imagens de referência possuem formato PNG. As imagens de lesões contém uma lesão principal e, em alguns casos, pequenas lesões secundárias e algumas marcações tais como réguas métricas.

[1] <https://isdis.org/isic-project/>
[2] <https://www.isic-archive.com>

As imagens GT são máscaras binárias com 2 níveis em escala de cinza e representam a localização das lesões das imagens de entrada. Desta forma, elas possuem exatamente as mesmas dimensões das imagens de entrada. Cada pixel das imagens GT possui apenas dois valores: 0, que indica o fundo a imagem ou área complementar à área da lesão; 255, que indica a lesão principal.

Para treinar a U-Net, as imagens da base de dados foram divididas em três conjuntos: treinamento, validação e teste. Como as imagens GT dos conjuntos de validação e de teste originalmente não foram fornecidas pela ISIC, estes conjuntos não puderam ser utilizados, visto que era impossível compará-las com as imagens originais. Em vez disto, foram retiradas aleatoriamente 400 imagens do conjunto de treinamento original e separadas em 100 imagens para o conjunto de validação e 300 imagens para o conjunto de teste, sobrando assim 2.194 imagens para o conjunto de treinamento..

4.3 Recursos Computacionais

Recursos de *Software*

A implementação do projeto foi baseada na linguagem de programação Python, o que torna o código mais fácil de ser implementado. A rede rede neural foi implementada em *TensorFlow*, *software* de código aberto desenvolvido pela *Google brain Team*[3], e *Keras*, biblioteca *open-source* de redes neurais escrita em Python. *Keras* fornece uma maneira conveniente de definir e treinar quase qualquer tipo de modelo de DL e foi inicialmente desenvolvido para pesquisadores, com o objetivo de permitir uma rápida experimentação.

Recursos de *Hardware*

O projeto foi inteiramente implementado na plataforma *online* gratuita chamada ***Google Colaboratory***. Esta plataforma permite a execução e escrita de códigos em Python com processamentos em nuvem com bons recursos computacionais. Entretanto, existe uma limitação de 12 horas por sessão, sendo incapaz de treinar uma rede neural por mais tempo do que o limite. Caso o limite seja excedido, a sessão é encerrada e todo o processo é perdido. O *Colaboratory* possui suporte para geração de gráficos, integração com *TensorFlow* e com máquinas locais e permite instalação de bibliotecas externas de Python. A máquina virtual disponibilizada pelo *Google Colaboratory* e utilizada para o pré-processamento e para os treinamentos das redes neurais possui a seguinte configuração:

[3] <https://research.google.com/teams/brain/>

(*i*) sistema operacional Linux, distribuição Ubuntu *Server* 16.04; (*ii*) processador Intel(R) Xeon(R) CPU, 2.20GHz, 3.60GHz com 2 núcleos físicos; (*iii*) memória RAM de 13 GB; (*iv*) unidade de armazenamento de 360GB (disco rígido); (*v*) placa de vídeo NVIDIA Tesla T4, com 15 GB de memória dedicada.

4.4 Detalhes de implementação

A seguir são descritas os principais detalhes da implementação da rede neural proposta para a tarefa de segmentação de imagens da base de dados ISIC 2018.

Inicialmente, durante os testes, percebeu-se que a GPU do *Google Colaboratory* não possui memória suficiente para carregar todas imagens do conjunto de dados de treinamento, mesmo após o redimensionamento das imagens. Isto ocorre devido ao grande tamanho dos arquivos do conjunto de dados. A fim de solucionar este problema, a linguagem Python possui um tipo de função que funciona como gerador de dados. O gerador de dados funciona como um iterador que retorna uma certa quantidade de dados, ou um *batch*, conforme ele é chamado. O *batch* é dito como a quantidade de imagens carregadas na memória de vez para a etapa de treino ou teste. O tamanho do *batch* é importante em treinamento de redes neurais, pois ele possui um efeito na velocidade e nos requerimentos de recursos de sistemas, permitindo o treinamento de conjuntos de dados relativamente grandes em sistemas com recursos computacionais que não sejam do topo de linha. Para o caso deste trabalho, foi utilizado um tamanho de *batch* de oito imagens de treinamento por vez.

O API *Keras* possui um gerador de dados em que é possível aplicar a técnica de *data augmentation* ao mesmo tempo da geração de dados chamado de *ImageDataGenerator*[4]. As transformações utilizadas no *data augmentation* podem ser visualizadas na Tabela 2. O *ImageDataGenerator* é capaz de efetuar transformações das imagens como deslocamento aleatório horizontal e vertical, inversões aleatórias ao longo dos eixos horizontal e vertical, rotações aleatórias, *zooms* aleatórios e entre outros. As transformações sobre o conjunto de treinamento utilizadas neste trabalho foram as operações de rotação, deslocamento e *zoom*.

Implementação da U-Net. A implementação da U-Net feita em *Tensorflow* e *Keras* utilizou uma biblioteca disponível em *GitHub* [5]. A biblioteca disponibilizada por (YAKUBOVSKIY, 2019), possui a implementação da U-Net com *backbones* de diversas redes classificadoras, como VGG, ResNet, Inception, DenseNet e entre outros. O modelo também

[4] <https://keras.io/preprocessing/image/>
[5] <https://github.com/qubvel/segmentation_models>

Tabela 2 – Parâmetros do *data augmentation*.

Transformação	**Valor**
Rotação	90º
Deslocamento	0,2
Zoom	0,2

Fonte – Produção do próprio autor

utiliza a técnica de *transfer learning*, com os pesos das redes treinadas com a base de dados da ImageNet 2012 [6].

As implementações dos treinamentos das redes U-Net com os diferentes *backbones* e dos testes foram feitas inteiramente em *Keras*.

Descrição dos valores dos hiperparâmetros da rede. Os hiperparâmetros que foram utilizados na rede U-Net com os *backbones* de VGG16, de ResNet-18 e de DenseNet-121 estão listados na Tabela 3. Devido à limitação de tempo imposta pela plataforma *Google Colaboratory*, a quantidade de épocas utilizadas com cada arquitetura foi diferente. Na rede VGG-16 e na ResNet-18, o treino ocorreu por 40 épocas, enquanto na DenseNet-121 foram 30 épocas. Pela Tabela 3, observa-se que o tamanho de *batch* para a DenseNet-121 foi menor, devido à limitação de memória GPU. Para todos os casos, a taxa de aprendizagem foi decaindo pela metade cada vez que o aprendizado da rede parasse de melhorar, fazendo com que tenha uma melhor convergência.

Tabela 3 – Hiper-parâmetros para o treinamento das redes U-net com diferentes *backbones*.

Hiperparâmetro	**VGG-16**	**ResNet-18**	**DenseNet-121**
Tamanho de *Batch*	8	8	4
Otimizador	*Adam*	*Adam*	*Adam*
Taxa de aprendizagem	$0,0001$	$0,0001$	$0,0001$

Fonte – Produção do próprio autor

Tempos de treinamento. Os treinos realizaram duraram certa de 10 horas para a U-Net com VGG-16, 9 horas para a ResNet-18 e 11 horas para a DenseNet-121.

[6] <http://www.image-net.org/challenges/LSVRC/2012/>

4.5 Experimentos

4.5.1 Métricas

No desafio ISIC 2018 (CODELLA et al., 2019), a métrica principal utilizada para a avaliação dos participantes na tarefa de segmentação de lesões de pele foi o índice de Jaccard limiarizado (JL). O índice de Jaccard, cuja representação pode ser visualizada na Figura 4.1, é calculado em relação a cada imagem binária predita com a respectiva imagem de referência esperada utilizando a Equação (4.1), porém sua pontuação será registrado de acordo com um limiar. A Equação (4.2) representa o índice de Jaccard limiarizado aplicado em cada imagem binária ($\mathbf{B}$) predita em relação à imagem binária esperada ($\hat{\mathbf{B}}$).

$$JI = \frac{|\mathbf{B} \cap \hat{\mathbf{B}}|}{|\mathbf{B} \cup \hat{\mathbf{B}}|} \tag{4.1}$$

$$JL = \begin{cases} 0 & JI < 0,65 \\ JI & JI \geq 0,65 \end{cases} \tag{4.2}$$

Figura 18 – Representação do índice de Jaccard.

Fonte: Produção do próprio autor

Aqui, $|\bullet|$ é o operador de cardinalidade. Este índice representa a interseção das imagens preditas com o GT sobre a união das imagens preditas com o GT. O limiar escolhido para o índice de Jaccard foi o mesmo utilizado no desafio ISIC 2018. Além do índice de Jaccard limiarizado, outras métricas foram utilizadas para comparação de resultados. entre elas tem-se: (i) acurácia (Ac), que avalia a capacidade da técnica de classificar corretamente os pixels como objeto ou fundo; (ii) sensibilidade (Se), que avalia a capacidade da técnica em classificar corretamente os pixels como objeto; (iii) especificidade (Es), que avalia a capacidade da técnica em classificar corretamente os pixels como fundo; e (iv) coeficiente de Dice ($DICE$), que é a relação de similaridade entre imagem predita e a imagem esperada. As relações que definem essas métricas são apresentadas nas Equações (4.3) – (4.6),

onde: VP são os verdadeiros positivos (número de pixels corretamente classificados pela técnica como objeto), VN são os verdadeiros negativos (número de pixels corretamente classificados pela técnica como fundo), FP são os falsos positivos (número de pixels classificados erroneamente pela técnica como objeto), e FN são falsos negativos (número de pixels classificados erroneamente pela técnica como fundo).

$$Ac = \frac{VP + VN}{VP + VN + FP + FN}.100 \tag{4.3}$$

$$Se = \frac{VP}{VP + FN}.100 \tag{4.4}$$

$$Es = \frac{VP}{VP + FP}.100 \tag{4.5}$$

$$DICE = \frac{2VP}{2VP + FP + FN}. \tag{4.6}$$

4.5.2 Avaliação no ISIC 2018

No experimento realizado, o seguinte procedimento foi utilizado:

- **Divisão do banco de dados**. O banco de dados é dividido em um conjunto de treino e teste tal como é indicado na seção 4.2.
- **Definição da arquitetura**. É selecionado um especifico *backbone* pre-treinado, o qual é adicionado no etapa de *encode* da U-net.
- **Etapa de treinamento**. A rede U-net com o correspondente *backbone* é treinado, aplicando-se a técnica de *data augmentation* para o incremento do conjunto de dados de treino.
- **Etapa de inferência**. A rede U-net modificada, já treinada, e avaliada usando o conjunto de teste, calculando-se as métricas explicitadas na seção 4.5.1.

Na Tabela 4 é possível visualizar os resultados obtidos. Nas Figuras 19, 20 e 21, pode-se observas os gráficos de perda dos treinamentos da U-Net com as *backbones* da VGG-16, ResNet-18 e DenseNet-121 respectivamente.

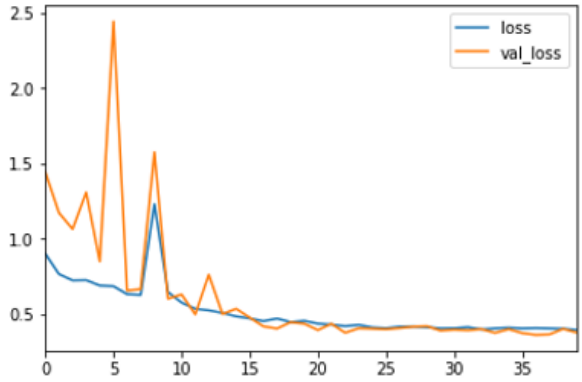
Observando a Tabela 4, percebe-se que:

- o melhor resultado obtido foi usando o modelo da U-net baseado na DenseNet-121, que atingiu um índice de Jaccard médio limiarizado de **0.772**;

Tabela 4 – Resultados do desempenho na segmentação de imagens de lesão de pele para cada modelo de U-net utilizado.

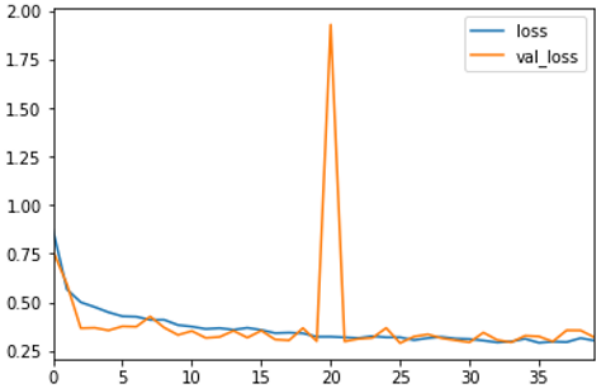
Metodologia	JL	$Ac(\%)$	$Se(\%)$	$Es(\%)$	$DICE$
U-Net + VGG-16	0,694	94,92	85,89	96,79	0,846
U-Net + ResNet-18	0,741	96,09	90,18	97,11	0,895
U-Net + DenseNet-121	**0,772**	**96,24**	**90,30**	**96,56**	**0,903**

Figura 19 – Gráfico de perdas do treinamento da U-Net com a VGG-16. Eixo das abscissas quantifica as épocas de treino, enquanto o eixo das ordenadas quantifica as perdas.



Fonte: Produção do próprio autor

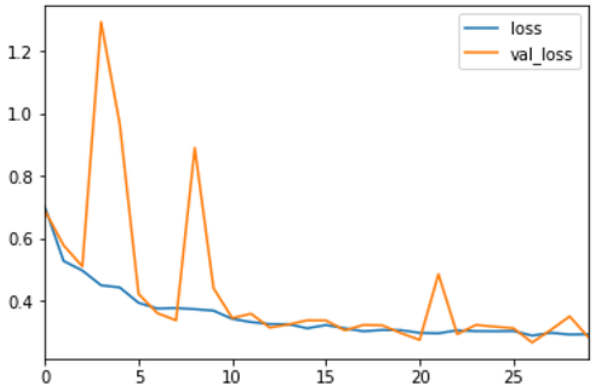
Figura 20 – Gráfico de perdas do treinamento da U-Net com a ResNet-18. Eixo das abscissas quantifica as épocas de treino, enquanto o eixo das ordenadas quantifica as perdas.



Fonte: Produção do próprio autor

- a rede baseada em ResNet-18 ficou pouco abaixo em relação ao anterior, o que era esperado, visto que a DenseNet possui mais conexões de corta-caminho, levando a uma melhor otimização do gradiente;
- a VGG-16 é a que possui o pior desempenho comparado às outras duas redes, uma possivel explicação pode-se dever a que sua profundidade é inferior as outras arquiteturas, implicando que, as características extraídas por esta rede sejam insuficientes para o problema em estudo.

Figura 21 – Gráfico de perdas do treinamento da U-Net com a DenseNet-121. Eixo das abscissas quantifica as épocas de treino, enquanto o eixo das ordenadas quantifica as perdas.



Fonte: Produção do próprio autor

Foram observadas algumas dificuldades na geração das imagens segmentadas do conjunto de teste via U-Net, visto na Figura 22. De forma geral, a rede tende a gerar máscaras semelhantes às máscaras originais, como pode se observar na Figura 22a. No entanto, existem imagens em que a lesão possui tons similares aos tons de pele, o que dificultou em partes na segmentação da lesão. Nesse caso, a imagem predita teve uma área maior que a imagem original, como pode-se observar na Figura 22b. Todavia, nas imagens em que se aparenta ter duas lesões principais como a Figura 22c, a rede gerou uma máscara com duas segmentações na mesma imagem, o que não está indicado na máscara original. Na Figura 22d, a máscara gerada pela rede possui alguns pontos fora da lesão principal, pois há uma aparente lesão secundária na imagem, o que confunde a rede. É interessante notar que na Figura 22e onde há presença de adesivos de pele a rede ignorou por completo os adesivos. Em imagens onde há presença de pelo que cruza a região da lesão e do fundo, como na Figura 22f, a imagem predita teve uma área menor que a imagem original. Por fim, em imagens como a da Figura 22g em que a lesão possui um tom mais escuro e que possui ao redor tons mais claros, a rede classificou erroneamente como fundo a área com tons mais claros.

4.5.2.1 Comparação de resultados

Para obter uma noção da qualidade da U-Net treinada com os diferentes *backbones*, elas foram comparadas com os 5 melhores resultados obtidos pelas equipes durante o desafio ISIC 2018 (ver Tabela 5). A pontuação das dez melhores equipes colocadas pode ser visualizada na Tabela 6. Nota-se que uma equipe pode enviar um número ilimitado de vezes. A classificação geral é obtida através da média do índice de Jaccard limiarizado das imagens do conjunto de testes. É importante notar que não é possível fazer uma

comparação exata dos resultados deste trabalho com as submissões feitas no desafio, visto que as imagens testadas não são iguais. Se comparado com a lista de participantes completa, a solução proposta atingiria a posição 15º no desafio ISIC 2018.

Tabela 5 – Métodos das 5 melhores equipes do desafio ISIC 2018 na tarefa de segmentação de lesão de pele .

Equipe	**Entrada**	**Rede Neural**	***Emsemble***	**TL**
MT	512×512	DeepLab e PSPNet com ResNet101	Sim	Não
Holidayburned	192×256	DeepLab, DenseNet, U-Net, VGG	Sim	Sim
imsight	512×512	Baseada em ResNet34	Não	Não
Tencent Youtu Lab	224×224	U-Net, VGG, DeepLabV3	Sim	Sim
NMN_team	384×576	U-Net, ResNet152, DenseNet169, DeepLabV3	Sim	Sim

Fonte – Produção do próprio autor

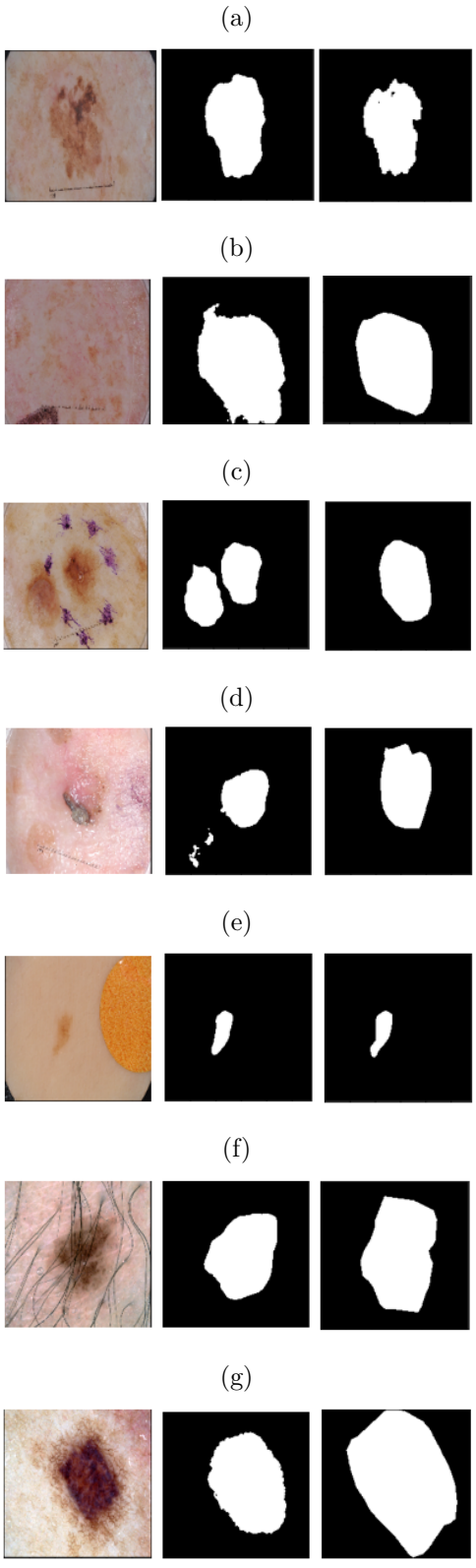
Tabela 6 – *Ranking* do desafio ISIC 2018 na tarefa de segmentação de lesão de pele.

Posição	**Equipe**	**Pontuação**
1	MT	0,802
2	Holidayburned	0,799
3	imsight	0,799
4	Tencent Youtu Lab	0,798
5	NMN_team	0,796
6	MT	0,794
7	Holidayburned	0,794
8	Holidayburned	0,794
9	GPM-UC3M	0,788
10	NMN_team	0,784

Fonte – Produção do próprio autor

É interessante observar que os melhores classificados utilizaram *ensembles* de diferentes redes para um único resultado, *transfer learning* e *data augmentation* para alcançar melhores resultados. Tendo isso considerado, as equipes possuíam bastante poder computacional, chegando a 8 GPUs para servidores (Tesla v100).

Figura 22 – Entradas e saídas com diferentes condições. As imagens originais estão a esquerda. No centro estão as imagens preditas e a direita estão as imagens de referência originais.



Fonte: Banco de Dados ISIC *Archive*

5 CONCLUSÕES E PROJETOS FUTUROS

5.1 Conclusões

O objetivo principal deste trabalho foi propor uma técnica de segmentação automática de lesões de pele utilizando redes neurais convolucionais profundas. Para tal objetivo, foi utilizado a rede neural U-Net com as redes classificadoras VGG-16, ResNet-18 e DenseNet-121 como *encoder* da U-Net. O banco de dados fornecido pelo desafio ISIC 2018 foi utilizado para avaliar a técnica proposta, atingindo um resultado de 0.772 no índice de Jaccard limiarizado, se colocando na posição 15º no desafio. Com a realização deste trabalho, foi permitido comprovar a importância da U-Net na tarefa de segmentação de imagens, atingindo um resultado satisfatório tendo em vista a dificuldade do problema a ser resolvido.

5.2 Temas a serem pesquisados

Os temas a serem seguidos em trabalhos futuros com o objetivo de aprimorar os resultados:

- Devem ser pesquisadas técnicas de *emsemble*, visto que redes neurais consolidadas na literatura estão sendo utilizadas em conjunto para gerar melhores resultados.
- Utilizar mais técnicas de *data augmentation* para que seja permitido estender mais os treinos sem que o *overfitting* seja causado, melhorando assim o desempenho da rede na tarefa de segmentação de imagens.
- Utilizar a técnica de *cross-validation* no conjunto de teste.

REFERÊNCIAS

AKIBA, T.; SUZUKI, S.; FUKUDA, K. Extremely large minibatch sgd: training resnet-50 on imagenet in 15 minutes. arXiv preprint arXiv:1711.04325, 2017. Citado na página 21.

ANGEL, A. Nearest Neighbor Interpolation. 2018. Disponível em: <https://www.imageeprocessing.com/2017/11/nearest-neighbor-interpolation.html>. Acesso em: 12 julho 2019. Citado na página 18.

BERSETH, M. ISIC 2017 - skin lesion analysis towards melanoma detection. CoRR, abs/1703.00523, 2017. Disponível em: <http://arxiv.org/abs/1703.00523>. Citado na página 13.

BISSOTO, A.; PEREZ, F.; RIBEIRO, V.; FORNACIALI, M.; AVILA, S.; VALLE, E. Deep-learning ensembles for skin-lesion segmentation, analysis, classification: RECOD titans at ISIC challenge 2018. CoRR, abs/1808.08480, 2018. Disponível em: <http://arxiv.org/abs/1808.08480>. Citado na página 13.

BRAUN, R. P.; RABINOVITZ, H. S.; OLIVIERO, M.; KOPF, A. W.; SAURAT, J.-H. Dermoscopy of pigmented skin lesions. Journal of the American Academy of Dermatology, Elsevier, v. 52, n. 1, p. 109–121, 2005. Citado na página 11.

BUSLAEV, A.; SEFERBEKOV, S. S.; IGLOVIKOV, V.; SHVETS, A. Fully convolutional network for automatic road extraction from satellite imagery. In: CVPR Workshops. [S.l.: s.n.], 2018. p. 207–210. Citado na página 31.

CELEBI, M. E.; IYATOMI, H.; SCHAEFER, G.; STOECKER, W. V. Lesion border detection in dermoscopy images. Computerized medical imaging and graphics, Elsevier, v. 33, n. 2, p. 148–153, 2009. Citado na página 12.

CHEN, Z.; XIE, Z.; ZHANG, W.; XU, X. Resnet and model fusion for automatic spoofing detection. In: INTERSPEECH. [S.l.: s.n.], 2017. p. 102–106. Citado na página 21.

CODELLA, N. C. F.; GUTMAN, D.; CELEBI, M. E.; HELBA, B.; MARCHETTI, M. A.; DUSZA, S. W.; KALLOO, A.; LIOPYRIS, K.; MISHRA, N.; KITTLER, H.; HALPERN, A. Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). CoRR, p. 2–6, 2017. Citado na página 34.

CODELLA, N. C. F.; ROTEMBERG, V.; TSCHANDL, P.; CELEBI, M. E.; DUSZA, S. W.; GUTMAN, D.; HELBA, B.; KALLOO, A.; LIOPYRIS, K.; MARCHETTI, M. A.; KITTLER, H.; HALPERN, A. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). CoRR, abs/1902.03368, 2019. Disponível em: <http://arxiv.org/abs/1902.03368>. Citado 2 vezes nas páginas 34 e 38.

DIMATOS, D. C.; DUARTE, F. O.; MACHADO, R. S.; VIEIRA, V. J.; VASCONCELLOS, Z. A. A. de; BINS-ELY, J.; NEVES, R. d'Éça. Melanoma cutâneo no Brasil. Arquivos Catarinenses de Medicina, v. 38, n. Suplemento 01, p. 14, 2009. Citado na página 11.

DONG, H.; YANG, G.; LIU, F.; MO, Y.; GUO, Y. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In: SPRINGER. annual conference on medical image understanding and analysis. [S.l.], 2017. p. 506–517. Citado na página 28.

ESTEVA, A.; KUPREL, B.; NOVOA, R. A.; KO, J.; SWETTER, S. M.; BLAU, H. M.; THRUN, S. Dermatologist-level classification of skin cancer with deep neural networks. Nature, Macmillan Publishers Limited, part of Springer Nature. All rights reserved., v. 542, p. 115, jan 2017. Citado na página 11.

EVERINGHAM, M.; ESLAMI, S. A.; GOOL, L. V.; WILLIAMS, C. K.; WINN, J.; ZISSERMAN, A. The pascal visual object classes challenge: A retrospective. International journal of computer vision, Springer, v. 111, n. 1, p. 98–136, 2015. Citado na página 25.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 2 vezes nas páginas 12 e 15.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J. et al. Recent advances in convolutional neural networks. Pattern Recognition, Elsevier, 2017. Citado na página 15.

GULSHAN, V.; PENG, L.; AL., E. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA: Journal of the American Medical Association, v. 316, n. 22, p. 2402–2410, dec 2016. ISSN 0098-7484. Citado na página 12.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 770–778. Citado 2 vezes nas páginas 21 e 22.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2017. p. 4700–4708. Citado na página 22.

INCA. Estimativa 2018: Incidência de câncer no Brasil. [S.l.]: Ministério da Saúde, 2017. Citado na página 11.

Jafari, M. H.; Karimi, N.; Nasr-Esfahani, E.; Samavi, S.; Soroushmehr, S. M. R.; Ward, K.; Najarian, K. Skin lesion segmentation in clinical images using deep learning. In: 2016 23rd International Conference on Pattern Recognition (ICPR). [S.l.: s.n.], 2016. p. 337–342. Citado na página 13.

JAIN, S.; JAGTAP, V.; PISE, N. Computer Aided Melanoma Skin Cancer Detection Using Image Processing. Procedia Computer Science, v. 48, p. 736–741, dec 2015. Citado na página 11.

JANSSON, A.; HUMPHREY, E.; MONTECCHIO, N.; BITTNER, R.; KUMAR, A.; WEYDE, T. Singing voice separation with deep u-net convolutional networks. 2017. Citado na página 28.

JEONG, J.; YOON, T. S.; PARK, J. Towards a meaningful 3d map using a 3d lidar and a camera. Sensors, v. 18, p. 2571, 08 2018. Citado na página 25.

JI, Y.; ZHANG, H.; WU, Q. J. Salient object detection via multi-scale attention cnn. Neurocomputing, v. 322, p. 130 – 140, 2018. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231218311342>. Citado na página 12.

JIANG, H.; CHEN, X.; SHI, F.; MA, Y.; XIANG, D.; YE, L.; SU, J.; LI, Z.; CHEN, Q.; HUA, Y.; XU, X.; ZHU, W.; FAN, Y. Improved cgan based linear lesion segmentation in high myopia icga images. Biomed. Opt. Express, OSA, v. 10, n. 5, p. 2355–2366, May 2019. Citado na página 32.

JUNG, H.; CHOI, M.-K.; JUNG, J.; LEE, J.-H.; KWON, S.; JUNG, W. Y. Resnet-based vehicle classification and localization in traffic surveillance systems. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. [S.l.: s.n.], 2017. Citado na página 21.

KE, H.; CHEN, D.; LI, X.; TANG, Y.; SHAH, T.; RANJAN, R. Towards brain big data classification: epileptic eeg identification with a lightweight vggnet on global mic. IEEE Access, IEEE, v. 6, p. 14722–14733, 2018. Citado na página 19.

KITTLER, H.; PEHAMBERGER, H.; WOLFF, K.; BINDER, M. Diagnostic accuracy of dermoscopy. The Lancet Oncology, Elsevier, v. 3, n. 3, p. 159–165, dec 2017. ISSN 1470-2045. Citado na página 11.

LATEEF, F.; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. Neurocomputing, Elsevier, v. 338, p. 321–348, 2019. Citado na página 12.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, v. 521, n. 7553, p. 436–444, 2015. ISSN 14764687. Citado na página 12.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Comput., MIT Press, Cambridge, MA, USA, v. 1, n. 4, p. 541–551, 1989. ISSN 0899-7667. Citado na página 18.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, v. 86, n. 11, p. 2278–2323, 1998. ISSN 00189219. Citado na página 12.

LECUN, Y. et al. Generalization and network design strategies. In: Connectionism in perspective. [S.l.]: Citeseer, 1989. v. 19. Citado na página 15.

LI, B.; LIU, S.; XU, W.; QIU, W. Real-time object detection and semantic segmentation for autonomous driving. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. MIPPR 2017: Automatic Target Recognition and Navigation. [S.l.], 2018. v. 10608, p. 106080P. Citado na página 12.

LI, Y.; SHEN, L. Skin lesion analysis towards melanoma detection using deep learning network. Sensors, Multidisciplinary Digital Publishing Institute, v. 18, n. 2, p. 556, 2018. Citado na página 13.

LIANG, S.; ZHANG, R.; LIANG, D.; SONG, T.; AI, T.; XIA, C.; XIA, L.; WANG, Y. Multimodal 3d densenet for idh genotype prediction in gliomas. Genes, Multidisciplinary Digital Publishing Institute, v. 9, n. 8, p. 382, 2018. Citado na página 23.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: SPRINGER. European conference on computer vision. [S.l.], 2014. p. 740–755. Citado na página 25.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2015. p. 3431–3440. Citado 2 vezes nas páginas 17 e 26.

Milioto, A.; Lottes, P.; Stachniss, C. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). [S.l.: s.n.], 2018. p. 2229–2235. ISSN 2577-087X. Citado na página 12.

NIELSEN, M. Neural Networks and Deep Learning. 2015. Disponível em: <http://neuralnetworksanddeeplearning.com/chap6.html>. Acesso em: 11 maio 2018. Citado na página 16.

NIERADZIK, L. Losses for Image Segmentation. 2018. Disponível em: <https://lars76.github.io/neural-networks/object-detection/losses-for-segmentation/>. Acesso em: 17 julho 2019. Citado na página 51.

NOH, H.; HONG, S.; HAN, B. Learning deconvolution network for semantic segmentation. In: The IEEE International Conference on Computer Vision (ICCV). [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 16 e 17.

PLANCHE, B.; ANDRES, E. Hands-On Computer Vision with TensorFlow 2. [S.l.]: Packt Publishing Ltd, 2019. ISBN 978-1788830645. Citado na página 24.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, MIT Press, v. 29, n. 9, p. 2352–2449, 2017. Citado 4 vezes nas páginas 15, 16, 17 e 18.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. International Conference on Medical image computing and computer-assisted intervention. [S.l.], 2015. p. 234–241. Citado 2 vezes nas páginas 26 e 27.

RUMELHART, D.; HINTON, G.; WILLIAMS, R.; CALIFORNIA, S. D. I. f. C. S. University of. Learning Internal Representations by Error Propagation. [S.l.]: Institute for Cognitive Science, University of California, San Diego, 1985. (ICS report). Citado na página 18.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M. S.; BERG, A. C.; LI, F.-F. ImageNet Large Scale Visual Recognition Challenge. CoRR, abs/1409.0575, 2014. Citado na página 18.

Sercu, T.; Puhrsch, C.; Kingsbury, B.; LeCun, Y. Very deep multilingual convolutional neural networks for lvcsr. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], 2016. p. 4955–4959. ISSN 2379-190X. Citado na página 20.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. Citado 2 vezes nas páginas 19 e 20.

SUN, Y.; LIANG, D.; WANG, X.; TANG, X. Deepid3: Face recognition with very deep neural networks. CoRR, abs/1502.00873, 2015. Disponível em: <http://arxiv.org/abs/1502.00873>. Citado na página 19.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2015. p. 1–9. Citado na página 21.

YAKUBOVSKIY, P. Segmentation Models. [S.l.]: GitHub, 2019. <https://github.com/qubvel/segmentation_models>. Citado na página 36.

ZHANG, Z.; LIANG, X.; DONG, X.; XIE, Y.; CAO, G. A sparse-view ct reconstruction method based on combination of densenet and deconvolution. IEEE transactions on medical imaging, IEEE, v. 37, n. 6, p. 1407–1417, 2018. Citado na página 23.

ZHANG, Z.; LIU, Q.; WANG, Y. Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters, IEEE, v. 15, n. 5, p. 749–753, 2018. Citado na página 28.

ZHU, W.; HUANG, Y.; ZENG, L.; CHEN, X.; LIU, Y.; QIAN, Z.; DU, N.; FAN, W.; XIE, X. Anatomynet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. Medical Physics, v. 46, n. 2, p. 576–589, 2019. Citado na página 32.

Anexos

ANEXO A – FUNÇÕES DE PERDA PARA SEGMENTAÇÃO DE IMAGENS

Neste anexo, serão implementadas as funções de custo mais comuns para segmentação de imagens em *Keras*/*Tensorflow*. A referência para este anexo está em (NIERADZIK, 2018).

A.1 Entropia Cruzada

Seja $P(Y = 0) = p$ e $P(Y = 1) = 1 - p$. As predições são dadas pela função de sigmoid

$$P(\hat{Y} = 0) = \frac{1}{1 + e^{-x}} = \hat{p}, \tag{A.1}$$

$$P(\hat{Y} = 1) = 1 - \frac{1}{1 + e^{-x}} = 1 - \hat{p} \tag{A.2}$$

A Entropia Cruzada (CE) é definida como:

$$CE(p, \hat{p}) = -(p \log(\hat{p}) + (1 - p) \log(1 - \hat{p})), \tag{A.3}$$

$$\tag{A.4}$$

Em *Keras*, a função de custo é *binary_crossentropy(y_true, y_pred)*.

A.1.1 Entropia Cruzada Ponderada

A Entropia Cruzada Ponderada (WCE) é uma variante da Entropia Cruzada onde todos exemplos positivos são ponderados por um coeficiente. É bastante utilizado em caso de desbalanceamento de classes.

WCE está definida a seguir:

$$WCE(p, \hat{p}) = -(\beta p \log(\hat{p}) + (1 - p) \log(1 - \hat{p})) \tag{A.5}$$

Para diminuir a quantidade de falsos negativos, escolha $\beta > 1$. Para diminuir a quantidade de falsos positivos, escolha $\beta < 1$.

A.2 Medidas de Sobreposição

A.2.1 Perda de Dice e Índice de Jaccard

O Coeficiente de Dice é similar ao Índice de Jaccard (Interseção sobre União):

$$DC = \frac{2VP}{2VP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{A.6}$$

$$IoU = \frac{VP}{VP + FP + FN} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}, \tag{A.7}$$

onde VP são os verdadeiros positivos, FP falsos positivos e FN falsos negativos. Pode-se observar que $DC \geq IoU$.