PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programa de Pós-Graduação em Informática

André Dias dos Santos Júnior

LOCALIZAÇÃO E CLASSIFICAÇÃO DE OBJETOS USANDO REDES NEURAIS CONVOLUTIVAS

Belo Horizonte

31 de Maio de 2019

André Dias dos Santos Júnior

LOCALIZAÇÃO E CLASSIFICAÇÃO DE OBJETOS USANDO REDES NEURAIS CONVOLUTIVAS

Monografia apresentado ao Curso de Bacharelado em Engenharia de Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Dr. Zenilton Kléber Gonçalves do Patrocínio Júnior

Belo Horizonte

31 de Maio de 2019

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

André Dias dos Santos Júnior

LOCALIZAÇÃO E CLASSIFICAÇÃO DE OBJETOS USANDO REDES NEURAIS CONVOLUTIVAS

Monografia apresentado ao Curso de Bacharelado em Engenharia de Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Prof. Dr. Zenilton Kléber Gonçalves do Patrocínio Júniorr – PUC Minas

---

Prof. Dr. Membro interno – PUC Minas

Prof. Dr. Membro externo – Instituicao

Belo Horizonte, 31 de Maio de 2019.

À minha amada avó, que sempre me apoiou.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus, que me deu forças para prosseguir, mediante aos obstáculos.

Agradeço também à minha avó por ter me apoiado incondicionalmente e por ter sempre acreditado em mim.

Agradeço também à minha mãe por ser um suporte sólido e por sempre me apoiar quando precisei.

Agradeço também ao meu orientador, o professor Zenílton pelo apoio, a presença, o auxílio prestado e pelo aprendizado ao longo da minha jornada.

Agradeço ao meu pai por torcer por mim e pelas orações.

E agradeço à todos os familiares e amigos que, de alguma forma me auxiliaram ao longo deste caminho

"E fazendo que se aprende a fazer aquilo que se deve aprender a fazer."

Aristoteles

RESUMO

Atualmente há um expressivo crescimento no volume de conteúdo visual, como imagens e vídeos, disponíveis para utilização. As informações contidas em imagens e vídeos são utilizadas pelas mais variadas áreas da sociedade, como médica, industrial e, até mesmo, para fins pessoais. Todos os dias são estudados novos métodos computacionais para fazer recuperação e análise de imagens. Dois métodos que tem sido frequentemente estudados em conjunto são a localização e classificação de objetos. Arquiteturas de *Deep Learning* podem ser implementadas para fazer localização e classificação de objetos, obtendo bons resultados, porém possuem certas limitações como o tamanho dos objetos ou a resolução da imagem. Para contornar essa limitação, pretende-se acrescentar camadas de convolução ao final para fazer a localização dos objetos de diversos tamanhos de forma mais apurada. Após as camadas finais de Convolução, serão acrescentadas camadas de Deconvolução, para aumentar a escala dos mapas, e decodificar os resultados das convoluções de forma a aumentar precisão do resultado.

Palavras-chave: *Deep Learning*, Localização, Classificação, Convolução, Deconvolução.

ABSTRACT

Texto do resumo, em ingles.

Keywords: .

LISTA DE FIGURAS

FIGURA 1 – Exemplo de Classificação e Localização 9

FIGURA 2 – YOLO e SSD 20

FIGURA 3 – SSD e DSSD 21

FIGURA 4 – Módulos de predição DSSD 22

LISTA DE TABELAS

TABELA 1 – Cronograma de desenvolvimento 24

LISTA DE ABREVIATURAS E SIGLAS

CNN – *Convolutional Neural Networks* - Redes Neurais Convolucionais

COCO – *Common Objects in Context*

DBN – *Deep Belief Networks*

DenseNet – *Densely Connected Convolutional Networks* - Redes Convolutivas Densamente conectadas

DSSD – *Deconvolutional Single-Shot Detector*

FCN – *Fully Convolutional Network* - Rede Completamente Convolucional

FPS – *Frames Per Second* - Quadros Por Segundo

ILSVRC – *ImageNet Large Scale Visual Recognition Challenge*

MLP – *Multi-Layer Perceptron*

MLS – *Mean Less Squares* - erro quadrático mínimo

PASCAL VOC – *PASCAL Visual Object Classes*

Pixels – Pontos em uma imagem rasterizada ou menor elemento mostrado no dispositivo de saída de vídeo

RBM – *Restricted Boltzmann Machines*

R-CNN – *Regions with CNN features*

RNA – Rede Neural Artificial

SSD – *Single-Shot multi-box Detector*

YOLO – *You Only Look Once* - você só olha uma vez

SUMÁRIO

1 INTRODUÇÃO **9**
1.1 Motivação 11
1.2 Objetivos 12
1.3 Justificativa 12
1.4 Organização do texto 13

2 REVISÃO BIBLIOGRÁFICA **14**
2.1 *Machine Learning* 14
2.1.1 Regressão 14
2.1.2 Classificação 15
2.2 Redes Neurais Artificiais 16
2.2.1 Multi-Layer Perceptron 16
2.3 *Deep Learning* 17
2.3.1 Redes Neurais Convolucionais 18
2.4 Imagem 19

3 TRABALHOS RELACIONADOS **20**
3.1 SSD: *Single-Shot Multibox Detector* 20
3.2 DSSD: *Deconvolution Single-Shot Detector* 21

4 PROPOSTA TÉCNICA **23**
4.1 Metodologia 23
4.1.1 Levantamento Bibliográfico 23
4.1.2 Testes e avaliações com métodos da literatura 23
4.1.3 Desenvolvimento do protótipo inicial 23
4.1.4 Desenvolvimento da arquitetura final usando deconvolução 24
4.1.5 Avaliação dos resultados e comparação com o Estado da Arte .. 24
4.1.6 Monografia 24
4.2 Cronograma 24

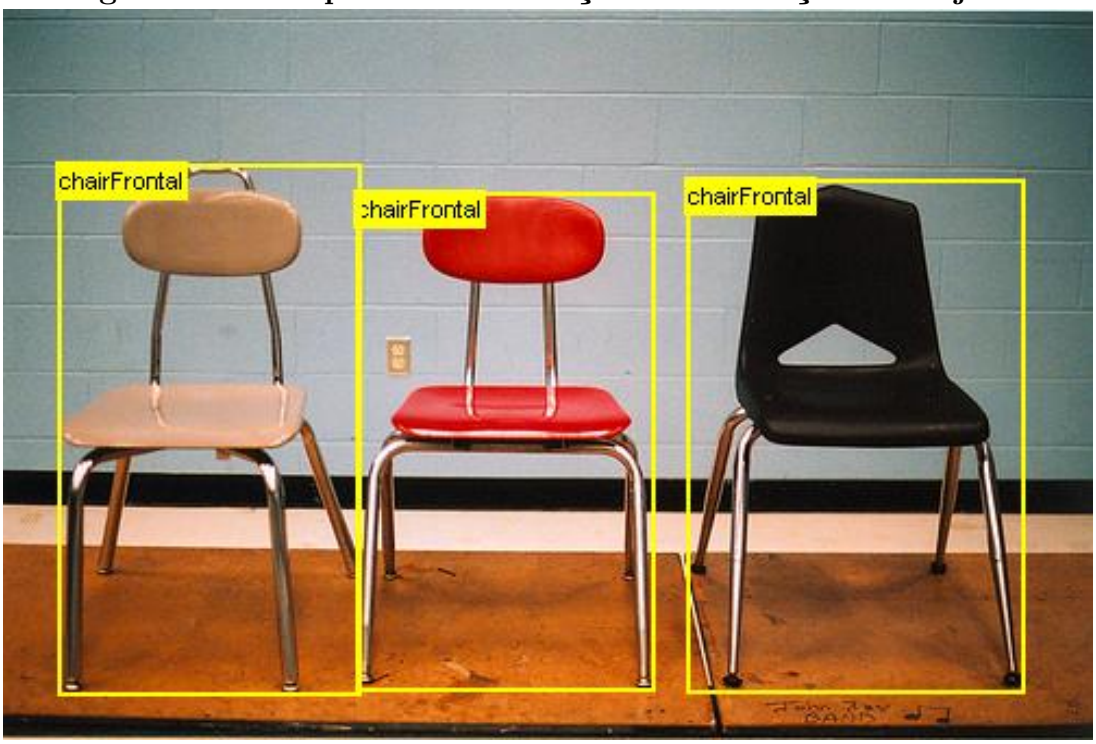
REFERÊNCIAS **25**

1 INTRODUÇÃO

Atualmente há um expressivo crescimento no volume de conteúdo visual, como imagens e vídeos, disponíveis para utilização. Podemos vincular este fato à popularização das tecnologias produtoras destes tipos de conteúdo, como celulares com câmeras, bem como a expansão da Internet e seus canais de comunicação, que se utilizam de imagens e vídeos para divulgação de informações. As informações contidas em imagens e vídeos são utilizadas pelas mais variadas áreas da sociedade, como médica, industrial e, até mesmo, para fins pessoais e de entretenimento.

O processamento digital de imagens contribui para a descoberta de informação visual contida em imagens e vídeos. Um dos problemas de processamento digital de imagens amplamente explorados na literatura é o problema de classificação e localização de objetos em imagens. Everingham et al. (2015) definem que o problema de classificação consiste em responder para cada classe de objeto se existe ou não uma ou mais instâncias daquele objeto na imagem e, o problema de localização consiste em dizer onde na imagem estão as instâncias dos objetos reconhecidos pelo classificador.

Figura 1 – Exemplo de Classificação e Localização de Objetos



Fonte: Everingham et al. (2015).

A Figura 1 mostra exemplos de como devem ser as saídas de um algoritmo de classificação e localização de objetos. Os retângulos destacados em torno dos objetos são resultados do algoritmo de localização que determina que dentro daquela região existe um objeto de interesse e os rótulos destacados em cima dos retângulos são os resultados do algoritmo de classificação, que determina que o objeto contido dentro da região pertence àquela classe.

O uso das *Convolutional Neural Networks* - Redes Neurais Convolucionais (CNN) tem se tornado cada vez mais populares, principalmente em alguns dos problemas clássicos de processamento de imagens. A primeira abordagem desse tipo foi proposta por Fukushima (1980) para fazer reconhecimento de caracteres escritos a mão. Mais tarde, Lecun et al. (1998) desenvolveram uma arquitetura de CNN para a mesma tarefa e obtiveram uma acurácia de $0,7\%$, sendo esse muito superior aos resultados obtidos por qualquer outro classificador utilizado até então.

Contudo, as CNN só passaram a ser utilizadas com mais frequência algum tempo depois. Deng et al. (2009) lançaram o ImageNet, uma base de dados de larga escala com mais de 14 milhões de imagens anotadas manualmente divididas em mais de 22 mil classes diferentes. Essa base de imagens pode ser usada para as tarefas de classificação, detecção e localização de objetos. Além disso, Deng et al. (2009) também lançaram o *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), uma competição global anual, aonde os competidores são avaliados nas tarefas de classificação de imagens, detecção de objetos e localização de objetos. Krizhevsky, Sutskever e Hinton (2012) alcançaram o melhor resultado no ILSVRC de 2012 usando a AlexNet - uma CNN - e desde então, as CNNs obtém sempre os melhores resultados nos desafios.

Nas tarefas de localização de objetos as CNNs também têm se destacado com bons resultados. Redmon et al. (2015) propuseram o *You Only Look Once* - você só olha uma vez (YOLO) e obtiveram um bom resultado fazendo localização e classificação de objetos em tempo real, conseguindo processar 45 *Frames Per Second* - Quadros Por Segundo (FPS) com uma acurácia de 63,4%. Ren et al. (2017) propuseram o *Regions with CNN features* (R-CNN) e conseguiram obter resultados significativamente melhores para as tarefas de localização e classificação chegando a uma acurácia de até 78,8%. Numa implementação alternativa com o enfoque no processamento em tempo real, a acurácia cai um pouco, chegando a 73,2% porém, ele processa apenas 7 FPS.

Liu et al. (2015) propuseram o *Single-Shot multi-box Detector* (SSD), uma abordagem que além de obter uma acurácia elevada (74,3%), supera a velocidade alcançada no YOLO atingindo 59 FPS. Esses resultados são especialmente relevantes, pois, além de serem competitivos com o estado da arte, trabalham com imagens menores - enquanto os quadros processados pelo YOLO têm dimensões 448×448 e os processados pelo R-CNN

têm 1000×600, o SSD processa quadros de dimensões 300×300 - consumindo assim menos memória sem comprometer a acurácia do método.

Uma outra área impactada pelo uso das CNNs é a área de segmentação semântica. Long, Shelhamer e Darrell (2014) propuseram o uso de uma *Fully Convolutional Network* - Rede Completamente Convolucional (FCN) para fazer segmentação semântica e conseguiram obter resultados superiores ao estado da arte, utilizando camadas de deconvolução para aumentar a resolução dos mapas de filtros gerados na saída da CNN. Noh, Hong e Han (2015) propuseram um trabalho mais avançado de forma a tratar algumas limitações encontradas por Long, Shelhamer e Darrell (2014). Essa arquitetura funciona com a estrutura de um *autoencoder*, usando as camadas de deconvolução que não servem para aumentar a resolução, mas para recuperar as informações da imagem de forma mais refinada, de forma a melhorar os resultados da segmentação.

Por fim, Fu et al. (2017) propôs uma abordagem alternativa na resolução dos problemas de localização e classificação de objetos utilizando camadas de deconvolução a *Deconvolutional Single-Shot Detector* (DSSD). A DSSD recebe essa sigla, pois é uma extensão do SSD (LIU et al., 2015). A diferença é que, ao final, ele acrescenta camadas de deconvolução e, com os resultados das deconvoluções ele faz a classificação e a localização. Os resultados obtidos superam os apresentados pelo SSD em acurácia, alcançando 81,5% embora, o DSSD perde na velocidade (13,6 FPS).

1.1 Motivação

Estudos envolvendo redes neurais convolucionais têm sido realizados com muita frequência tanto em áreas gerais, quanto aplicados à áreas específicas. Estudos vem sendo feitos para classificação de imagens (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; SIMONYAN; ZISSERMAN, 2014), classificação e detecção de objetos Fu et al. (2017), Lin et al. (2014), segmentação de imagens Long, Shelhamer e Darrell (2014), Noh, Hong e Han (2015), segmentação de objetos em vídeos Caelles et al. (2017), Voigtlaender e Leibe (2017). Isso se deve aos bons resultados obtidos pelos métodos aplicados nas respectivas áreas.

Além disso, o problema de classificação e detecção de objetos tem sido aproveitado como solução para problemas mais específicos, como detecção facial (Yang; Jiachun, 2018), contagem de pessoas (Ren; Fang; Djahel, 2017) e detecção de pedestres (Lan et al., 2018).

Por fim o uso das camadas de deconvolução no método proposto traz boas expectativas, uma vez que já há resultados na literatura que mostram a sua eficiência (Noh; Hong; Han, 2015; FU et al., 2017). Em suma, pode-se dizer que o tema ainda tem muito que ser explorado, que a proposta é promissora, e que bons resultados podem trazer contribuições

para trabalhos futuros.

1.2 Objetivos

O objetivo do trabalho é utilizar uma arquitetura de CNN modificada com camadas de deconvolução para fazer detecção e classificação de objetos. Para isso, será necessário:

- Adaptar a CNN *Densely Connected Convolutional Networks* - Redes Convolutivas Densamente conectadas (DenseNet) (Huang et al., 2017) para o problema de classificação de objetos em imagens;
- Adaptar a rede para fazer a localização e classificação de forma similar à SSD;
- Alterar a arquitetura da rede inicial para receber camadas de deconvolução de forma a melhorar os resultados da classificação e da Localização;
- Avaliar os resultados obtidos comparando-os com a literatura.

1.3 Justificativa

Com os avanços tecnológicos alcançados ao longo das últimas décadas, a produção de conteúdo multimídia tem crescido consideravelmente e tem sido amplamente explorada pelos mais diversos setores da sociedade. Todos os dias mais de 95 milhões de fotos e vídeos são postadas no Instagram*. Esse imenso volume de dados que são produzidos e acessados em multimídia inviabiliza a manipulação deste conteúdo por meio de ação humana; criando assim, a necessidade de automatizar a recuperação e análise de informações relevantes contidas nas mesmas.

Nesse sentido, todos os dias são estudados novos algoritmos e novas técnicas para fazer recuperação de informação multimídia. E uma das formas amplamente utilizadas é a de localização e classificação de objetos em imagens. Everingham et al. (2015) definem que o problema de classificação consiste em responder para cada classe de objeto se existe ou não uma ou mais instâncias daquele objeto na imagem e, o problema de localização consiste em dizer onde na imagem estão as instâncias dos objetos reconhecidos pelo classificador.

É válido estabelecer que "os mecanismos sensitivos dos seres humanos (como visão e audição) sugerem a necessidade de uma arquitetura profunda para extrair a sua estrutura complexa" (DENG; YU, 2014). Porém, de acordo com Caelles et al. (2017), uma grande limitação ao utilizar arquiteturas de Deep Learning é a necessidade de treinamento com

*https://www.instagram.com

um grande volume de dados. Essa limitação torna o processo mais custoso em termos de tempo de processamento e outros recursos computacionais(como memória). Além disso, a base de dados deve ter os resultados da classificação e localização esperados feitos de forma manual, o que aumenta o esforço humano.

Tendo isso em vista, a proposta de utilizar a DenseNet (Huang et al., 2017) pré-treinada em classificação de imagens visa diminuir o custo por meio do *Transfer Learning*. Além disso, utilizando as camadas adicionais de deconvolução como proposto por Fu et al. (2017), espera-se obter resultados ainda melhores. Esses resultados melhores repercutiriam de forma positiva em outras áreas relacionadas à localização e classificação de objetos.

1.4 Organização do texto

O Capítulo 2 traz os principais conceitos relacionados à *Deep Learning*, redes neurais artificiais, classificação de objetos em imagens e detecção de objetos em imagens, necessários à compreensão do trabalho. O Capítulo 3 contém a metodologia a ser seguida para o desenvolvimento do trabalho. Além disso contém um cronograma a ser seguido e a definição das métricas a ser utilizadas para avaliar os resultados obtidos.

2 REVISÃO BIBLIOGRÁFICA

2.1 *Machine Learning*

Machine Learning ou Aprendizado de Máquina é um campo de estudo cada vez mais recorrente na área de tecnologia. Em uma definição alternativa, pode-se dizer que "Aprendizado de Máquina é uma área de Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática" (MONARD; BARANAUSKAS, 2003). Entre as principais aplicações de Aprendizado de máquina, pode se destacar o reconhecimento de padrões, classificação de imagens e a mineração de dados.

Monard e Baranauskas (2003) define ainda que o aprendizado de máquina pode ser supervisionado ou não-supervisionado. Quando falamos do primeiro caso, queremos dizer que o algoritmo recebe um conjunto de dados na entrada, processa esses dados e retorna uma saída. Essa saída é comparada com um valor previamente associado à entrada, comumente conhecido como rótulo. Já no aprendizado não-supervisionado, as entradas não possuem rótulo algum, cabendo ao algoritmo fazer a distinção das diversas entradas. Geralmente nesses casos, é necessária uma análise posterior à execução do algoritmo para se entender os resultados obtidos.

O aprendizado supervisionado se divide em dois grupos menores: classificação e regressão. Santos (2012) define que quando o problema possui um conjunto discreto de saídas e o objetivo é atribuir a qual dessas saídas a entrada pertence, se trata de um problema de classificação. Santos (2012) define ainda que quando o objetivo é prever uma saída de valor contínuo pra uma entrada, se trata de um problema de regressão.

As seções 2.1.1 e 2.1.2 irão definir melhor esses conceitos.

2.1.1 Regressão

Como definido anteriormente, o problema de regressão consiste em calcular para cada entrada uma saída que possua um valor contínuo. Dosualdo (2003) define que a regressão consiste em fazer uma relação entre os atributos $X = x_1, x_2, ..., x_n$ - onde X é o conjunto de entrada e cada x_i é um atributo numérico ou quantitativo - e Y, tal que Y é um atributo ou um conjunto de atributos meta. Apté e Weiss (1997) define essa relação através da equação 2.1:

$$Y = f(x_1, x_2, ..., x_n) \tag{2.1}$$

A relação entre o(s) atributo(s) X e o(s) atributo(s)-meta Y alcançada como resultado de um algoritmo de regressão é chamado de modelo. Após a definição do modelo é importante fazer uma avaliação para dizer o quão confiável será a predição gerada pelo modelo. Existem diversas funções usadas na avaliação dos algoritmos de regressão, e uma das mais comuns é o *Mean Less Squares* - erro quadrático mínimo (MLS) que é definido pela equação 2.2:

$$E = \sum_{i=0}^{N} (Y_i - f(X)_i)^2 \tag{2.2}$$

Onde E é o erro, N é o tamanho do conjunto de amostras, Y_i é o valor esperado do atributo-meta e $f(X_i)$ é o resultado do modelo para a entrada X_i.

Alguns exemplos de problemas de regressão são:

1. Predizer o percentual de gordura que uma pessoa possui no corpo, recebendo como entrada atributos como altura, idade, peso, sexo (DOSUALDO, 2003).

2. Predizer o preço de um imóvel baseado em atributos como o tamanho, número de cômodos, número de quartos, etc. (PEREIRA; GARSON; ARAÚJO, 2012).

Uma grande limitação nos métodos de regressão é que eles em sua maioria solucionam problemas que possuem apenas um atributo-meta. Porém, os métodos de regressão baseados em Rede Neural Artificial (RNA) podem retornar múltiplos resultados. Esse termo será definido na Sessão 2.2.

2.1.2 Classificação

Como definido anteriormente, o problema de classificação consiste em um problema de predição quando as saídas são discretas. A ideia dos algoritmos de classificação é definir para cada entrada, um rótulo ou classe no meio de um conjunto finito de rótulos. Um método estatístico muito utilizado nos problemas de classificação é a regressão logística. A regressão logística é bem similar à regressão linear, com a diferença que o resultado é um número entre zero e um, configurando na verdade, a probabilidade de, dado uma entrada X tal que $X = x_1, x_2, ..., x_n$ gerar uma saída verdadeira para a hipótese Y. Normalmente, os modelos de regressão logística aplica a função sigmóide, descrita pela equação 2.3:

$$Y = \frac{1}{1 + e^{-g(x)}} \tag{2.3}$$

2.2 Redes Neurais Artificiais

As RNA são uma das principais técnicas de aprendizagem de máquina e podem ser implementadas tanto para problemas supervisionados, quanto não-supervisionados. Jost (2015) definiu as RNAs da seguinte forma:

> As RNAs possuem inspiração nas redes neurais biológicas, constituídas de neurônios separados por camadas, que processam informações e estão conectados via pesos sinápticos, sendo na maioria das vezes sistemas adaptativos que modificam sua estrutura através de informações, que fluem pela rede durante a etapa de aprendizado (JOST, 2015).

Além disso, é importante mencionar que em uma rede neural, existem duas camadas em particular que são muito importantes: a primeira camada, que é a camada de entrada de dados, e a última camada que é a camada de saída. Existem dois tipos de RNAs: as redes *feed forward*, que processamento sempre flui da entrada para a saída, e as redes recorrentes, que os dados fluem nos dois sentidos.

A RNA *feed forward* mais básica que existe é a perceptron, composta apenas por um conjunto de neurônios de entrada e um único neurônio de saída. A Equação 2.4 representa a fórmula de um perceptron, onde n é o número de entradas, W_i são os respectivos pesos de cada entrada, X_i são as entradas e B é o Bias do neurônio.

$$Y = \left(\sum_{i=1}^{n} (W_i \times X_i) \right) - B \tag{2.4}$$

Uma rede percepron não é uma arquitetura de redes neurais artificiais muito poderosa, mas deu base para outras arquiteturas mais robustas. A principal delas, é o Perceptron Multi-Camadas.

2.2.1 Multi-Layer Perceptron

A rede Perceptron Multicamadas ou *Multi-Layer Perceptron* (MLP) são baseadas nos Perceptrons simples, como mencionado anteriormente. A diferença, porém, é que elas trabalham com mais camadas do que simplesmente as camadas de entrada e saída. Geralmente elas possuem uma ou duas camadas intermediárias às camadas externas. Essas camadas intermediárias, também são chamadas de camadas ocultas e servem para conferir uma robustez maior ao método.

Uma outra diferença das MLPs para as perceptrons comuns é a equação para calcular a saída de cada neurônio. Nas redes MLPs, a equação utilizada é a função sigmóide. A Equação 2.4 descreve a fórmula da função sigmóide.

$$Y = \frac{1}{1 + e^z} \tag{2.5}$$

Onde z é descrito pela Equação 2.6:

$$Z_{(i+1)j} = \Big(\sum_{k=1}^{n} (W_{ik} \times X_{ik}) \Big) - B_j \tag{2.6}$$

Onde i é o número da camada, j o neurônio de destino, k o neurônio de origem, W é o peso e X a entrada.

Como foi dito anteriormente, a MLP trabalha com aprendizado supervisionado, isso quer dizer que os dados que ela opera são rotulados, e tem uma saída esperada. Quando as entradas são processadas e os resultados das saídas são calculados, eles são comparados com os valores esperados pelos rótulos e é gerado a função de erro quadrático, definida pela Equação 2.7:

$$E = \frac{1}{2} \times \sum_{i=1}^{n} (Y_i - O_i)^2 \tag{2.7}$$

Onde n é o número de saídas, i é qual saída calculada, Y são as saídas da rede e O são as saídas esperadas.

Sabendo-se disso, o objetivo para melhorar a precisão da MLP é minimizar o valor da função erro, ou seja, tornar as saídas da rede o mais próximo possível das saídas esperadas. Para tanto, é utilizado o método de retropropagação de erro que reajusta os pesos da rede de acordo com os valores obtidos usando a descida de gradiente.

Arnold et al. (2011) define que aumentar o número de camadas em um MLP não garante uma melhoria dos resultados, pois a descida de gradiente pode chegar a um mínimo local. Além disso, o aumento do número de camadas implica em um tempo muito maior de processamento. Para lidar com esse problema surge o Deep Learning, uma arquitetura avançada com múltiplas camadas, que soluciona a dificuldade que as redes neurais possuem ao lidar com dados de alta dimensionalidade (ARNOLD et al., 2011).

2.3 *Deep Learning*

A principio não parecia ser viável manipular arquiteturas profundas de redes neurais. Porém, de acordo com Deng e Yu (2014) surgiu um algoritmo de aprendizado não-supervisionado que conseguiu aliviar empiricamente as dificuldades de otimização em arquiteturas profundas. Esse algoritmo é a *Deep Belief Networks* (DBN), um modelo generativo profundo composto de uma camada visível e várias camadas ocultas compostas

por uma pilha de *Restricted Boltzmann Machines* (RBM)s.

Deng e Yu (2014) definem ainda que o aprendizado nas DBNs é feito por um algoritmo guloso que ajustas os pesos camada-por-camada com uma complexidade linear ao tamanho e à profundidade da rede. E uma relação inesperada entre as DBNs e as MLPs surgiu quando descobriu-se que ao utilizar os pesos de uma DBN com arquitetura correspondente, você consegue inicializar os pesos de uma MLP de forma a produzir melhores resultados do que utilizando pesos aleatórios.

Uma segunda alternativa para trabalhar com aumento de camadas é o empilhamento de auto-codificadores. O empilhamento de auto-codificadores consiste basicamente em inserir na saída de uma rede neural uma segunda rede neural que pra cada entrada produz uma saída específica e retreinar a nova rede neural utilizando o algoritmo de retropropagação de erro (DENG; YU, 2014).

2.3.1 Redes Neurais Convolucionais

As CNN são um modelo específico de *Deep Learning* muito utilizados em aplicações de visão computacional e aplicações de reconhecimento de imagens. De acordo com Ferreira (2017), as CNNs utilizam matrizes para processar as entradas de dados, sendo essas matrizes unidimensionais para sinais e sequências, bidimensionais para imagens e tridimensionais para imagens volumétricas e vídeos.

Ao contrário das RNAs tradicionais (como MLP), as CNNs não necessariamente ligam todos os neurônios na camada de origem a todos os neurônios da camada de destino. Ferreira (2017) define que existem três tipos comuns de camadas utilizadas nas redes neurais convolucionais: as camadas de convolução, camadas de pooling e camadas completamente conexas.

A ideia da camada de convolução, é de que cada neurônio recebe como entrada neurônios próximos e tem por objetivo criar mapas e filtros. Ferreira (2017) define que em aplicações de reconhecimento de objetos, por exemplo, é comum as primeiras camadas de convolução detectarem bordas ou manchas que seriam as características mais básicas da imagem. As camadas de convolução mais profundas, detectam outras características mais específicas.

As camadas de pooling são utilizadas para reduzir a dimensão de dados vindo das camadas de convolução, consequentemente reduzindo o custo computacional (FERREIRA, 2017). Um exemplo seria receber como entrada uma matriz 4x4 e enviar para a próxima camada uma matriz 2x2 escolhendo um representante de cada quatro da matriz de entrada para a matriz de saída, sendo que esse representante é determinado por alguma propriedade específica. As propriedades mais utilizadas são a de máximo e de

média (FERREIRA, 2017).

Por fim, as camadas completamente conexas são exatamente iguais às redes MLP e é comum utilizá-las no final da rede para conectar à saída. Seu acréscimo no final é importante pois faz a ligação de todos os filtros (FERREIRA, 2017). As arquiteturas de CNNs geralmente intercalam algumas camadas de convolução com uma camada de pooling e com duas camadas completamente conexas ao final.

2.4 Imagem

Torres e Falcão (2006) definem uma imagem Î como sendo um par $(D_i,\vec{I})$ onde:

- D_i é um conjunto finito de Pontos em uma imagem rasterizada ou menor elemento mostrado no dispositivo de saída de vídeo (Pixels);
- $\vec{I}$:$D_I \mapsto R^n$ é uma função que atribui para cada *pixel* p em D_i um vetor $\vec{I} \in R^n$ (quando uma cor RGB é atribuída a cada *pixel*, por exemplo, podemos dizer que $\vec{I} \in R^3$).

Uma imagem digital pode ser considerada como a representação numérica da luz refletida em um determinado ponto representado no espaço. No caso de uma imagem em tons de cinza, a luz é amostrada em um ponto *(X,Y)* e quantizada para um valor inteiro. Sendo colorida, a luz é quantizada para os valores de vermelho ou *red*(R), verde ou *green*(G) e azul ou *blue*(B), tendo então três componentes. Cada elemento amostrado é considerado um pixel da imagem. Sendo assim, D_i representa a posição de cada ponto amostrado enquanto o vetor $\vec{I}$, é uma função que mapeia cada pixel p da imagem, em um valor real para todas as suas componentes, no caso R, G e B.

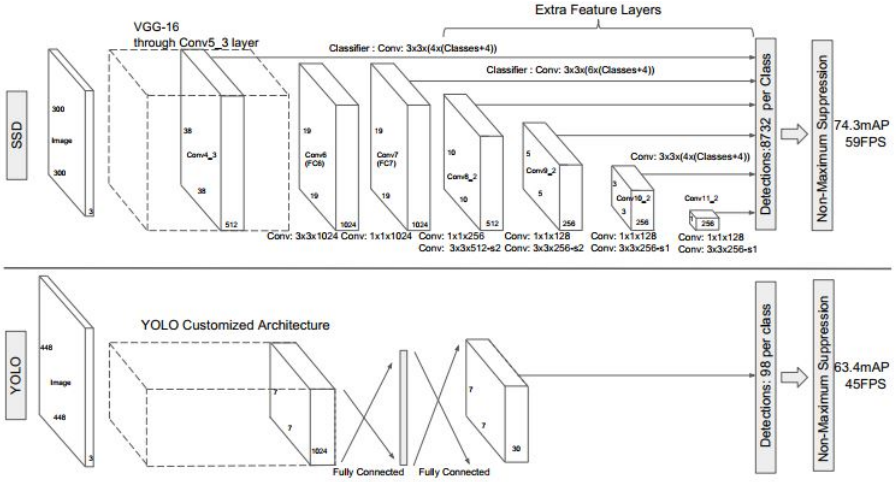
3 TRABALHOS RELACIONADOS

3.1 SSD: *Single-Shot Multibox Detector*

Liu et al. (2015) propuseram um método a base de redes neurais convolucionais para fazer a localização e detecção de objetos. A proposta deles melhorou significativamente os resultados apresentados no estado da arte. Isso se deve ao fato de que eles não só conseguiram propor um modelo que faz a localização e classificação de forma eficiente (chegando a 74,3% de acurácia), como conseguiram obter esse resultado fazendo classificação e localização em tempo real, com uma velocidade de 59 FPS. Uma outra vantagem obtida por esse método é que ele consegue fazer a localização e classificação em imagens significativamente menores, uma vez que os quadros processados pelo YOLO têm dimensões 448×448 e os processados pelo R-CNN têm 1000×600, o SSD processa quadros de dimensões 300×300.

A abordagem consiste em utilizar a rede VGG16 (SIMONYAN; ZISSERMAN, 2014) como arquitetura base, substituir as camadas completamente conectadas fc6 e fc7 por camadas convolucionais, alterar o filtro pool5 de $2 \times 2 - s2$ para $3 \times 3 - s1$, e usaram o algoritmo *à trous*(HOLSCHNEIDER et al., 1990) para preencher os espaços vazios. Além disso, eles removeram todas as camadas de dropout e a última camada completamente conectada. Por fim, as camadas de convolução geram os resultados de mais de 8000 localizações e classificações, as quais são filtradas em um passo final de supressão de não-máximos, que elimina todos os resultados com confiança abaixo de $0,5$.

Figura 2 – Comparação entre YOLO e SSD



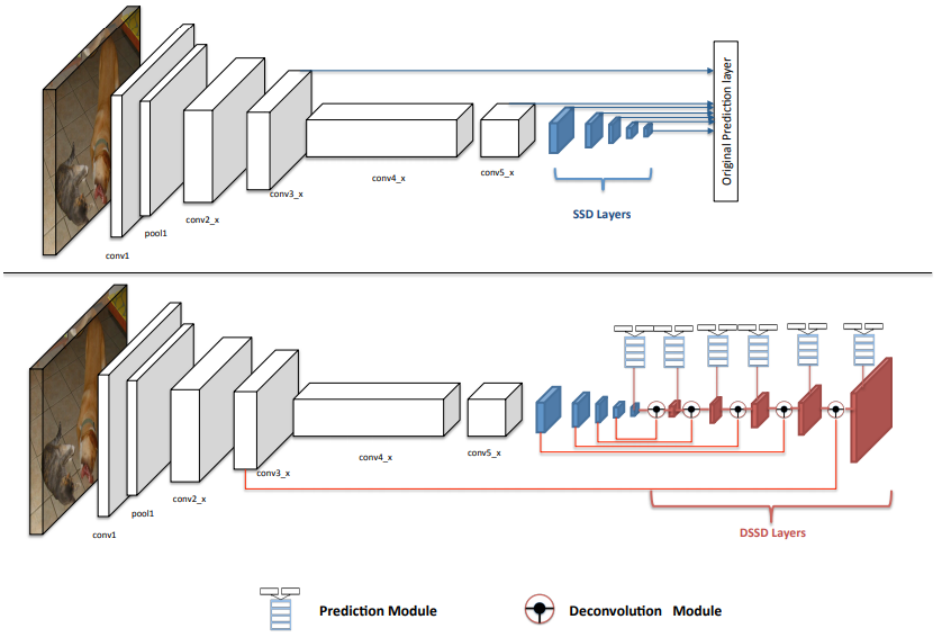
Fonte: Liu et al. (2015).

A Figura 2 mostra as diferenças entre as arquiteturas YOLO e SSD. Enquanto Redmon et al. (2015) usaram uma camada completamente conectada intermediária para fazer a localização dos objetos, ao passo que Liu et al. (2015) usaram camadas de convolução sobre mapas de múltiplos tamanhos. Além disso, o algoritmo trabalha com *bounding-boxes* de tamanhos padrões $\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$. Os filtros de convolução adicionais, os tamanhos padrões de *bounding-boxes* e o uso de *data-augmentation* foram cruciais na obtenção dos bons resultados.

3.2 DSSD: *Deconvolution Single-Shot Detector*

Fu et al. (2017) propuseram uma extensão do SSD. Depois dos resultados obtidos pelo SSD ao fazer localizção e classificação de objetos com uma acurácia de $79,5\%$, eles propuseram uma abordagem alternativa, usando camadas de deconvolução ao final da rede. As camadas de deconvolução tem entre seus resultados o aumento de resolução do mapa de entrada. A abordagem visou explorar esse efeito com o intuito de aumentar a acurácia da classificação e localização de objetos, e, com isso, atingir uma acurácia de $81,5\%$. Embora essa abordagem tenha uma acurácia maior do que a obtida por Liu et al. (2015), ela não é rápida o bastante pra fazer localização e classificação em tempo real.

Figura 3 – Comparação entre SSD e DSSD

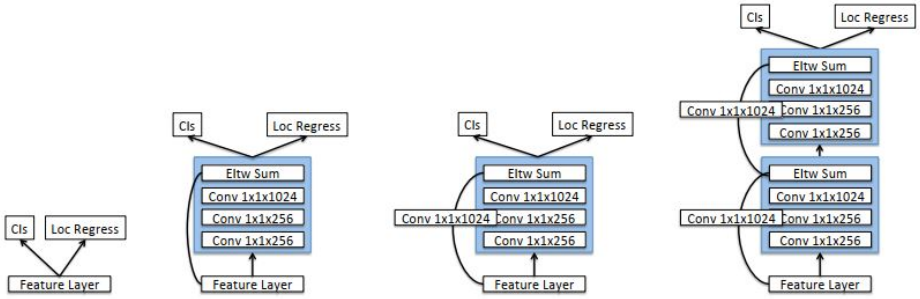


Fonte: Fu et al. (2017).

Nesse trabalho, foram propostas duas alterações principais no modelo SSD. A primeira delas foi a utilização da rede neural ResNet-101 He et al. (2016) e a segunda foi a utilização das camadas adicionais de deconvolução. Como mostra a Figura 3, a

arquitetura agora também utiliza de módulos de predição individuais, onde a saída da última camada de convolução e a saída de cada camada de deconvolução gera seus próprios resultados para a classificação e localização. A Figura 4 as variações dos módulos de predição da DSSD.

Figura 4 – Módulos de predição DSSD



Fonte: Fu et al. (2017).

Como mencionado anteriormente, embora o DSSD tenha melhorado a acurácia do SSD, ele não é aplicável para fazer a localização e classificação em tempo real. Com uma acurácia de $81,5\%$ ele consegue processar apenas $6,6$ FPS. Isso se deve ao uso das deconvoluções, da ResNet 101 - que possui mais camadas, e, portanto toma mais tempo de processamento - e também ao aumento no número de *bounding boxes* geradas(43688 vs. 17080), fazendo assim com que a supressão de não-máximos leve mais tempo.

4 PROPOSTA TÉCNICA

4.1 Metodologia

A seguir, serão definidos os passos necessários para a elaboração do trabalho proposto.

4.1.1 Levantamento Bibliográfico

Nesta fase será feito um levantamento na literatura de tudo o que é necessário para realizar o trabalho proposto. Será feito um estudo para compreensão da arquitetura DenseNet (Huang et al., 2017). Além disso, serão feitos estudos com métodos da literatura como YOLO (REDMON et al., 2015), SSD (LIU et al., 2015) e DSSD (FU et al., 2017). Serão feitos também estudos para analisar o conteúdo das bases de dados para testes. São essas: *PASCAL Visual Object Classes* (PASCAL VOC) (EVERINGHAM et al., 2015) e *Common Objects in Context* (COCO) (LIN et al., 2014). Além disso, nessa etapa serão levantadas as formas de avaliar os resultados dos algoritmos.

4.1.2 Testes e avaliações com métodos da literatura

Nesta etapa serão feitos os testes com os métodos já implementados na literatura. O objetivo é compreender o funcionamento, levantar os principais obstáculos nas respectivas implementações e selecionar as melhores tecnologias para a realização do projeto proposto. Nesta etapa também serão testados os algoritmos para avaliar os resultados obtidos, propostos por Everingham et al. (2015) e por Lin et al. (2014).

4.1.3 Desenvolvimento do protótipo inicial

A proposta é desenvolver um protótipo inicial usando a arquitetura da DenseNet121 apresentada por Huang et al. (2017) com alterações. As alterações a ser feitas são a remoção da camada final de classificação e a inserção de camadas intermediárias de convolução para realizar a localização dos objetos em diferentes escalas, como apresentado por Liu et al. (2015). A essa altura, a arquitetura já será capaz de fazer a localização e classificação dos objetos nas imagens.

4.1.4 Desenvolvimento da arquitetura final usando deconvolução

Depois de alterar a arquitetura para fazer a detecção em múltiplas escalas usando convolução, será feita uma nova modificação da arquitetura, acrescentando camadas de deconvolução. Para cada camada de deconvolução é acrescentado um módulo de predição, que fará a localização e a classificação dos objetos. O acréscimo das camadas de deconvolução e dos novos módulos de predição seguem o modelo proposto por Fu et al. (2017).

4.1.5 Avaliação dos resultados e comparação com o Estado da Arte

Por fim, erá feita a avaliação dos resultados dos métodos propostos e a comparação dos mesmos com os resultados do estado da arte. Essas avaliações serão feitas com base nas métricas que serão apresentadas na Sub-seção 4.1.1. O objetivo é dizer o quão efetivo foi a utilização de uma arquitetura de CNN mais moderna e avançada com métodos já conhecidos na resolução do problema de localzação e classificação.

4.1.6 Monografia

Por fim será confeccionado uma monografia apresentando e descrevendo a implementação do método proposto, os resultados obtidos, uma avaliação dos resultados e uma comparação com o estado da arte.

4.2 Cronograma

O projeto será desenvolvido ao longo do ano 2019 seguindo o cronograma apresentado na Tabela 1. O período está dividido em bimestres, sendo o primeiro bimestre equivalente aos meses de janeiro e fevereiro de 2019, e assim sucessivamente.

Tabela 1 – Cronograma de desenvolvimento do projeto

Atividades	Bimestre					
	1	2	3	4	5	6
Levantamento Bibliográfico	x	x	x			
Testes com métodos da literatura			x	x		
Desenvolvimento da arquitetura inicial			x	x		
Desenvolvimento da arquitetura final				x	x	
Avaliação dos resultados					x	x
Elaborar Monografia					x	x

Fonte: Elaborado pelo autor

REFERÊNCIAS

APTÉ, C.; WEISS, S. Data mining with decision trees and decision rules. Future Generation Computer Systems, v. 13, n. 2, p. 197 – 210, 1997. ISSN 0167-739X. Tradução nossa. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167739X97000216>>.

ARNOLD, L. et al. An introduction to deep learning. In: European Symposium on Artificial Neural Networks (ESANN). [S.l.: s.n.], 2011. Tradução nossa.

Caelles, S. et al. One-shot video object segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2017. p. 5320–5329. ISSN 1063-6919. Tradução nossa.

Deng, J. et al. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2009. p. 248–255. ISSN 1063-6919. Tradução nossa.

DENG, L.; YU, D. Deep learning: Methods and applications. Foundations and Trends® in Signal Processing, v. 7, n. 3–4, p. 197–387, 2014. Tradução nossa.

DOSUALDO, D. G. Investigação de regressão no processo de mineração de dados. Dissertação de Mestrado — Universidade de São Paulo, Maio 2003.

EVERINGHAM, M. et al. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, v. 111, n. 1, p. 98–136, jan 2015. Tradução nossa.

FERREIRA, A. dos S. Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja. Dissertação (Dissertação (Mestrado)) — Universidade Federal do Mato Grosso do Sul, Pos-Graduação em Ciências da Computação, 2017.

FU, C. et al. DSSD : Deconvolutional single shot detector. CoRR, abs/1701.06659, 2017. Tradução nossa. Disponível em: <<http://arxiv.org/abs/1701.06659>>.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, v. 36, n. 4, p. 193–202, Apr 1980. ISSN 1432-0770. Tradução nossa. Disponível em: <<https://doi.org/10.1007/BF00344251>>.

He, K. et al. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016. p. 770–778. ISSN 1063-6919. Tradução nossa.

HOLSCHNEIDER, M. et al. A real-time algorithm for signal analysis with the help of the wavelet transform. In: COMBES, J.-M.; GROSSMANN, A.; TCHAMITCHIAN, P. (Ed.). Wavelets. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990. p. 286–297. ISBN 978-3-642-75988-8.

Huang, G. et al. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2017. p. 2261–2269. ISSN 1063-6919. Tradução nossa.

JOST, I. Aplicação de *Deep Learning* em Dados Refinados para Mineração de Opiniões. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo., 2015.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. USA: Curran Associates Inc., 2012. (NIPS'12), p. 1097–1105. Tradução nossa. Disponível em: <<http://dl.acm.org/citation.cfm?id=2999134.2999257>>.

Lan, W. et al. Pedestrian detection based on yolo network model. In: 2018 IEEE International Conference on Mechatronics and Automation (ICMA). [S.l.: s.n.], 2018. p. 1547–1551. ISSN 2152-744X. Tradução nossa.

Lecun, Y. et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219. Tradução nossa.

LIN, T. et al. Microsoft COCO: common objects in context. CoRR, abs/1405.0312, 2014. Tradução nossa. Disponível em: <<http://arxiv.org/abs/1405.0312>>.

LIU, W. et al. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015. Tradução nossa. Disponível em: <<http://arxiv.org/abs/1512.02325>>.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014. Tradução nossa. Disponível em: <<http://arxiv.org/abs/1411.4038>>.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. Sistemas Inteligentes-Fundamentos e Aplicações, v. 1, n. 1, 2003.

Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In: 2015 IEEE International Conference on Computer Vision (ICCV). [S.l.: s.n.], 2015. p. 1520–1528. ISSN 2380-7504. Tradução nossa.

PEREIRA, J. C.; GARSON, S.; ARAÚJO, E. G. Construção de um modelo para o preço de venda de casas residenciais na cidade de sorocaba-sp. Revista GEPROS, n. 4, p. 153, 2012.

REDMON, J. et al. You only look once: Unified, real-time object detection. CoRR, abs/1506.02640, 2015. Tradução nossa. Disponível em: <<http://arxiv.org/abs/1506.02640>>.

Ren, P.; Fang, W.; Djahel, S. A novel yolo-based real-time people counting approach. In: 2017 International Smart Cities Conference (ISC2). [S.l.: s.n.], 2017. p. 1–2. Tradução nossa.

Ren, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 39, n. 6, p. 1137–1149, June 2017. ISSN 0162-8828.

SANTOS, A. de M. Tese de Doutorado, Investigando a combinação de técnicas de aprendizado semissupervisionado e classificação hierárquica multirrótulo. [S.l.]: Universidade Federal do Rio Grande do Norte, 2012.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. Tradução nossa.

TORRES, R. D. S.; FALCÃO, A. X. Content-based image retrieval: Theory and applications. Revista de Informática Teórica e Aplicada, v. 13, p. 161–185, 2006. Tradução nossa.

VOIGTLAENDER, P.; LEIBE, B. Online adaptation of convolutional neural networks for video object segmentation. CoRR, abs/1706.09364, 2017. Tradução nossa. Disponível em: <<http://arxiv.org/abs/1706.09364>>.

Yang, W.; Jiachun, Z. Real-time face detection based on yolo. In: 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII). [S.l.: s.n.], 2018. p. 221–224. Tradução nossa.