PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

LEANDRO PEREIRA DA SILVA

LEANNET: UMA ARQUITETURA QUE UTILIZA O CONTEXTO DA CENA PARA MELHORAR O RECONHECIMENTO DE OBJETOS

Porto Alegre

2018

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LEANNET: UMA ARQUITETURA QUE UTILIZA O CONTEXTO DA CENA PARA MELHORAR O RECONHECIMENTO DE OBJETOS

LEANDRO PEREIRA DA SILVA

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Duncan Dubugras Alcoba Ruiz

Porto Alegre
2018

Ficha Catalográfica

S586L Silva, Leandro Pereira da

LeanNet : uma arquitetura que utiliza o contexto da cena para melhorar o reconhecimento de objetos / Leandro Pereira da Silva . – 2018.

79 p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz.

1. Detecção de Objetos. 2. Rede Neural Convolucional. 3. Rede Neural. 4. Aprendizagem Profunda. 5. Objetos em Contexto. I. Ruiz, Duncan Dubugras Alcoba. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

Leandro Pereira da Silva

LeanNet: uma arquitetura que utiliza o contexto da cena para melhorar o reconhecimento de objetos

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciencia da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 27 de março de 2018.

BANCA EXAMINADORA:

Prof. Dr. Felipe Rech Meneguzzi (PPGCC/PUCRS)

Profa. Dra. Karin Becker (UFRGS)

Prof. Dr. Duncan Dubugras Alcoba Ruiz (PPGCC/PUCRS - Orientador)

DEDICATÓRIA

Dedico este trabalho principalmente a Deus, por me dar muita força e coragem para enfrentar as dificultades durante toda esta longa caminhada.

"E não somente isto, mas também nos gloriamos nas tribulações; sabendo que a tribulação produz a paciência, E a paciência a experiência, e a experiência a esperança. E a esperança não traz confusão, porquanto o amor de Deus está derramado em nossos corações pelo Espírito Santo que nos foi dado." (Romanos 5:3-5)

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus por tudo que fez na minha vida, pois sem ele nada disso seria possível.

Agradeço também ao meu orientador, Professor Doutor Duncan Dubugras Alcoba Ruiz, por toda a paciência e empenho com que sempre me orientou.

Agradeço aos meus colegas por toda ajuda que me deram e pelo esforço nos trabalhos que realizamos juntos.

Por último, agradeço à minha família e amigos pelo apoio durante essa jornada.

LEANNET: UMA ARQUITETURA QUE UTILIZA O CONTEXTO DA CENA PARA MELHORAR O RECONHECIMENTO DE OBJETOS

RESUMO

A visão computacional é a ciência que permite fornecer aos computadores a capacidade de verem o mundo em sua volta. Entre as tarefas, o reconhecimento de objetos pretende classificar objetos e identificar a posição onde cada objeto está em uma imagem. Como objetos costumam ocorrer em ambientes particulares, a utilização de seus contextos pode ser vantajosa para melhorar a tarefa de reconhecimento de objetos. Para utilizar o contexto na tarefa de reconhecimento de objetos, a abordagem proposta realiza a identificação do contexto da cena separadamente da identificação do objeto, fundindo ambas informações para a melhora da detecção do objeto. Para tanto, propomos uma nova arquitetura composta de duas redes neurais convolucionais em paralelo: uma para a identificação do objeto e outra para a identificação do contexto no qual o objeto está inserido. Por fim, a informação de ambas as redes é concatenada para realizar a classificação do objeto. Avaliamos a arquitetura proposta com os *datasets* públicos PASCAL VOC 2007 e o MS COCO, comparando o desempenho da abordagem proposta com abordagens que não utilizam o contexto. Os resultados mostram que nossa abordagem é capaz de aumentar a probabilidade de classificação para objetos que estão em contexto e reduzir para objetos que estão fora de contexto.

Palavras-Chave: Detecção de Objetos, Rede Neural Convolucional, Rede Neural, Aprendizagem Profunda, Objetos em Contexto.

LEANNET: AN ARCHITECTURE THAT USE SCENE CONTEXT TO IMPROVE OBJECT RECOGNITION

ABSTRACT

Computer vision is the science that aims to give computers the capability of seeing the world around them. Among its tasks, object recognition intends to classify objects and to identify where each object is in a given image. As objects tend to occur in particular environments, their contextual association can be useful to improve the object recognition task. To address the contextual awareness on object recognition task, the proposed approach performs the identification of the scene context separately from the identification of the object, fusing both information in order to improve the object detection. In order to do so, we propose a novel architecture composed of two convolutional neural networks running in parallel: one for object identification and the other to the identification of the context where the object is located. Finally, the information of the two-streams architecture is concatenated to perform the object classification. The evaluation is performed using PASCAL VOC 2007 and MS COCO public datasets, by comparing the performance of our proposed approach with architectures that do not use the scene context to perform the classification of the objects. Results show that our approach is able to raise in-context object scores, and reduces out-of-context objects scores.

Keywords: Object Detection, Convolutional Neural Network, Neural Network, Deep Learning, Object in Context.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

FC – fully-connected layer

CNN – Convolutional Neural Network

AMT – Amazon Mechanical Turk

RPN – Region Proposal Network

ROI – Region of Interest

MAP – Mean Average Precision

IOU – Intersection over Union

AP – Average Precision

SVM – support vector machine

LSTM – long shor-term memory

CRF – conditional random field

SUMÁRIO

1. INTRODUÇÃO

O cérebro humano foi criado para compreender a visão do mundo com facilidade, recebendo informações sobre os objetos, contexto e associações deles desde que nascemos. Assim, para um ser humano reconhecer objetos que pertencem à um determinado contexto é uma tarefa trivial, podendo haver maior dificuldade quando objetos estão fora do seu contexto. Conforme descrito pelo Biederman *et al.* [BMR82], a informação do contexto bem como o tamanho relativo entre os objetos e a localização são pistas importantes usadas pelos humanos para detectar objetos. De fato, quando usamos imagens em baixa resolução, os humanos podem distinguir se um objeto sobre a mesa é um prato ou um avião, uma vez que é improvável que tenha um avião sobre uma mesa.

Como apontado por Olivia e Torralba [OT07], no mundo real, objetos podem coocorrer com outros objetos em ambientes específicos, fornecendo uma fonte rica de associações contextuais a serem exploradas por um sistema visual. Estas relações entre um objeto e seus arredores são classificadas por Biederman *et al.* [BMR82] em cinco diferentes classes: interposição (*interposition*), suporte (*support*), probabilidade (*probability*), posição (*position*) e tamanho familiar (*familiar size*). Interposição e suporte são classes referentes ao espaço físico, probabilidade, posição e tamanho familiar são classes definidas como relações semânticas, uma vez que exigem acesso ao significado referencial do objeto considerado. Tais relações semânticas incluem informações detalhadas sobre interações entre os objetos na cena e são frequentemente usadas como recursos contextuais [GB10].

A tarefa de realizar a detecção de objetos usando um sistema computacional é importante para auxiliar as pessoas e melhorar a qualidade de vida. Sistemas autônomos [CKZ+16] por exemplo, utilizam a detecção de objetos para evitar acidentes de trânsito, enquanto sistemas assistivos [YTA14] ajudam as pessoas com deficiência visual nas atividades cotidianas. Esses sistemas estão em constante evolução e as informações do contexto ainda são pouco utilizadas.

Embora muito trabalho tenha sido proposto para melhorar os algoritmos de visão computacional, eles ainda estão longe da capacidade humana de reconhecer objetos [FLGC11]. Entretanto, com o avanço da tecnologia e disponibilidade de dados nos últimos anos, houve avanços significativos na detecção de objetos e classificação de imagens. Esse avanço se deve principalmente à utilização de redes neurais avançadas e de GPUs para acelerar o treinamento, bem como o crescente número de *datasets* públicos que estão sendo criados para serem usados nos treinamentos destas redes [GBC16].

Neste trabalho, abordamos o problema de reconhecimento de objetos usando características contextuais (*i.e.*, características extraídas do contexto da cena) como um indicativo da presença do objeto. Assim, a ideia principal deste trabalho é que a utilização do contexto auxilia o reconhecedor de objetos a melhorar a classificação de objetos depen-

dentes de contexto. Diferentemente de trabalhos anteriores, como o de Liu *et al.* [LGMZ16], nossa abordagem não necessita a oclusão do objeto para detectar o contexto. A abordagem proposta se baseia em uma arquitetura de rede neural profunda com duas redes neurais convolucionais (CNNs) em paralelo: uma para a identificação do contexto (Places365-CNN [ZLK+17]) e uma para a identificação de objetos (Faster R-CNN [RHGS15]). Ambas as redes são fundidas de forma a melhorar os resultados na tarefa de reconhecimento de objetos. Para esclarecer melhor o assunto, formulamos a seguinte questão de pesquisa: “É possível usar uma rede neural pré-treinada que classifica ambientes para extrair o contexto semântico e fazer a fusão com uma rede que detecta objetos, a fim de melhorar a classificação do objeto?"

O restante da dissertação está estruturada da seguinte forma: O Capitulo 2 apresenta a fundamentação teórica, abordando os tópicos que foram utilizados nesse trabalho. O Capitulo 3 descreve a arquitetura criada para o reconhecimento de objetos usando o contexto. O Capitulo 4 apresenta os *datasets* utilizados, as configurações dos experimentos e a formas de avaliação utilizadas. O Capitulo 5 relata os resultados correspondentes e apresenta uma discussão sobre os mesmos. O Capitulo 6 reporta os trabalhos relacionados que melhoram o reconhecimento de objetos usando o contexto. Finalmente, a dissertação termina com as conclusões e direções de trabalhos futuros no Capitulo 7.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma breve explicação sobre o funcionamento de redes neurais, descrevendo como é realizado o treinamento e como a rede aprende as informações. Em seguida, é apresentada a técnica de aprendizagem profunda, descrevendo as arquiteturas que tem contribuído bastante para melhorar os algoritmos de visão computacional.

2.1 Redes Neurais

Uma rede neural é formada por um conjunto de neurônios conectados em um grafo, podendo ser acíclico (*feedforward*) ou cíclico (recorrente). Em uma rede neural *feedforward* cada saída de um neurônio está ligado na entrada de um neurônio de outra camada. Enquanto em uma rede neural recorrente a saída de um neurônio pode estar ligada a entrada de um neurônio da mesma camada. Os neurônios têm a função de processar informações e ficam normalmente organizados em camadas. A figura 2.1 mostra a estrutura de um neurônio com relação a biologia, onde o dendrito é o canal responsável por receber a informação vinda de outros neurônios através da sinapse, passando ao corpo celular que contém uma função responsável pela excitação do neurônio e assim, passar a informação para o próximo neurônio através do axônio [FLGC11].

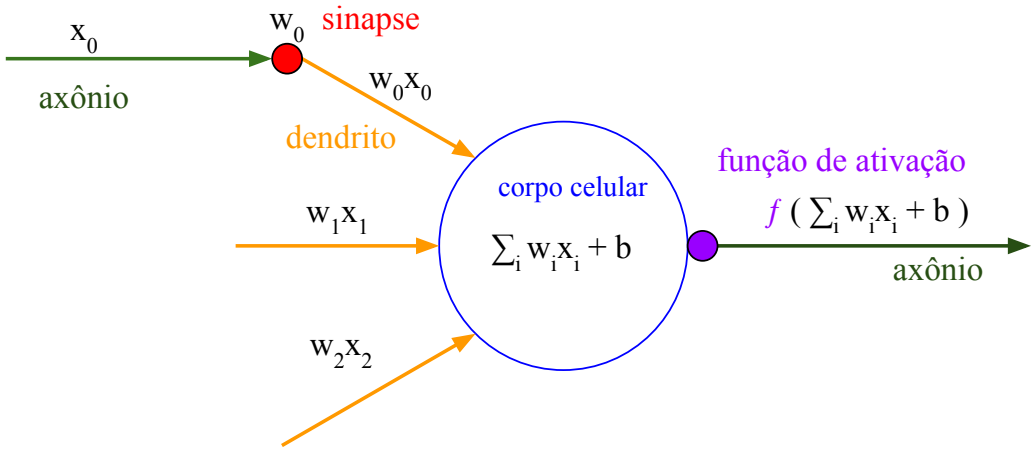


Figura 2.1 – Estrutura de um neurônio com relação a biologia (adaptado de [Uni17]).

Em uma rede neural *feedforward* a camada mais comum é a camada totalmente conectada (do inglês, *fully-connected layer*) (FC). A Figura 2.2 mostra uma rede neural com duas camadas FC (uma de entrada contendo 3 neurônios e uma de saída contendo 4 neurônios) e uma camada FC escondida contendo 6 neurônios. Os neurônios de uma mesma camada FC não fazem conexão entre si, mas somente com os neurônios da próxima

camada. Por exemplo, os neurônios da camada de entrada estão conectados com todos os neurônios da camada escondida, enquanto os neurônios da camada escondida estão conectados com todos os neurônios da camada de saída [FLGC11].

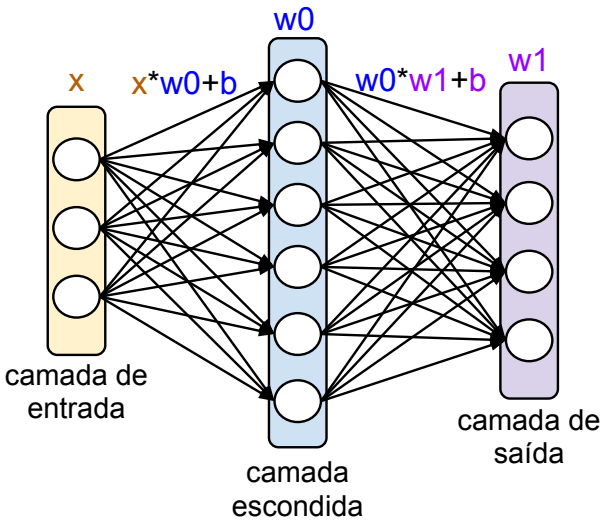


Figura 2.2 – Rede neural com duas camadas ou rede neural com uma camada escondida (adaptado de [Uni17]).

Redes neurais podem ter mais de uma camada escondida, e a quantidade de camadas assim como a quantidade de neurônios, vai depender do problema que está sendo aplicado. A camada de saída geralmente é usada para representar os escores das classes (*e.g.*, classificação) ou valores reais como as coordenadas de um objeto na imagem (*e.g.*, regressão). Por esse motivo a camada de saída da rede neural não possui função de ativação. Os pesos das conexões dos neurônios são atualizados através do algoritmo criado por Rumelhart *et al.* [RHW86] chamado *Backpropagation* [FLGC11].

2.1.1 Backpropagation

O algoritmo de *Backpropagation* calcula gradientes de forma recursiva através da regra da cadeia. A rede neural passa os dados da camada de entrada através de todas as camada até chegar na camada de saída, sendo esse processo chamado de *forward*. Quando as informações chegam na saída é utilizada uma função de custo para saber o quanto a classificação realizada pela rede se distancia da classificação correta, e assim, saber o quanto cada peso existente nas conexões dos neurônios devem ser atualizados. Por exemplo, se a rede se afastou muito da classificação correta, cada peso da conexão do neurônio será atualizado com um valor maior. Por outro lado, quando a classificação está próxima do valor correto, o valor de atualização será menor. Usando a regra da cadeia os valores de atualização são propagados de forma recursiva até a camada de entrada, sendo esse processo chamado de *backward*. Cada ciclo de *forward* e *backward* é contado como

uma iteração, e quando a rede neural visualiza todo o conjunto de dados é contado como uma época. O treinamento de uma rede neural geralmente passa por muitas épocas até a rede convergir, *i.e.*, o valor de perda (do inglês *loss*) parar de reduzir [FLGC11].

2.1.2 Funções de custo

A função de custo é uma forma de medirmos, no aprendizado supervisionado, a distância entre o valor predito e o valor real (*ground truth*) [GBC16]. As funções de custo podem ser classificadas de acordo com o tipo de rede neural. A seguir descrevemos a função de custo para problemas de classificação e regressão.

Classificação: Uma forma de calcular o quanto a rede se aproximou da classificação correta é utilizando o classificador *Softmax*, mostrado na Equação 2.1. A função recebe os escores preditos (s) não normalizados da ultima camada totalmente conectada e faz a normalização para cada classe (k). O valor dos escores de cada classe é então dividido pela soma de todos os valores normalizados (s_j), transformando o escore de cada classe em um valor de probabilidade da classe estar correta [Bis06].

$$Softmax = \frac{e^{s_k}}{\sum_j e^{s_j}} \tag{2.1}$$

Tendo o valor de *Softmax* para cada classe, é realizado o cálculo da entropia cruzada sobre a probabilidade de classe para descobrir o quanto a rede se distanciou da classificação correta, como mostra a Equação 2.2. O valor retornado pela função é utilizado pelo algoritmo de *Backpropagation* para atualizar os pesos das conexões dos neurônio da rede [GBC16].

$$L = -log(\frac{e^{s_k}}{\sum_j e^{s_j}}) \tag{2.2}$$

Regressão: é utilizada para predizer valores contínuos ou não discretos na saída da rede, como por exemplo, tamanho de uma região, preço de algum produto, *etc.*. Embora existam diversas funções de regressão, uma que é bastante utilizada nesse tipo de problema é chamada de erro quadrático médio (*L2 loss*), mostrada na Equação 2.3. Essa função realiza a soma do quadrado da diferença entre o valor real (y) e o valor predito ($\hat{y}$). Assim como na classificação, o erro resultante será propagado para as camadas anteriores pelo algoritmo de *Backpropagation*, de forma a atualizar os pesos das conexões dos neurônios [GBC16].

$$L = ||y - \hat{y}||_2^2 \tag{2.3}$$

2.1.3 Regularização

Durante o treinamento de uma rede neural pode acontecer dela se ajustar muito ao conjunto de treinamento, *i.e.*, aprender muito bem os valores do conjunto de treino, e em um conjunto de testes (*i.e.*, informações que não foram usadas no treinamento) a rede não obtém bons resultados. Esse problema é chamado de *overfitting* e faz com que a rede não consiga generalizar o problema além dos dados de treinamento. Para evitar que o *overfitting* ocorra exitem algumas técnicas de regularização que fazem com que a rede consiga generalizar melhor [GBC16].

Dropout é uma recente técnica de regularização criada por Srivastava *et al.* [SHK+14] que limita a quantidade de informações que trafega entre os neurônios durante o treinamento da rede, associando um valor de probabilidade de ativação para cada neurônio. Através disso, a rede tende a ter maior redundância de neurônios e consequentemente evita a rede ter neurônios muito específicos. A figura 2.3 ilustra um exemplo do funcionamento do *dropout* durante o treinamento, onde está sendo enviado para a rede uma imagem de um pássaro, e o *dropout* bloqueia metade dos neurônios da camada escondida, forçando os neurônios que sobraram serem mais genéricos. A cada iteração são ativados neurônios diferentes, com uma probabilidade escolhida durante o treino [SHK+14].

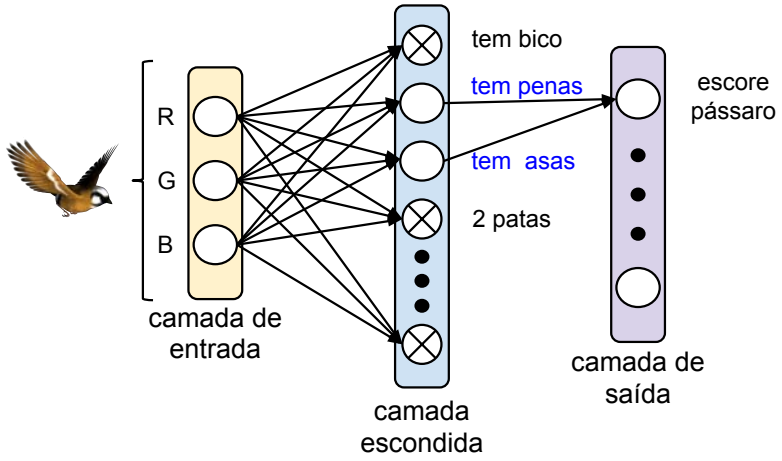


Figura 2.3 – Exemplo de utilização do *dropout* durante o treino. Durante teste o *dropout* não é utilizado. Adaptado de [Uni17].

2.2 Aprendizagem profunda

As redes neurais modernas ficaram conhecidas como Aprendizagem Profunda (do inglês *Deep Learning*) devido ao seu grande número de camadas [GBC16]. Uma rede profunda consegue obter vários níveis de abstração, melhorando o processamento de imagem,

vídeo e áudio. O método também funciona para outras abordagens como nos estudos do genoma e descobertas de remédios [LBH15]. Com o surgimento das redes neurais neurais convolucionais (do inglês *Convolutional Neural Network*) (CNN) obteve-se um grande avanço na visão computacional [LBH15].

2.2.1 Redes Neurais Convolucionais

As CNNs são redes neurais profundas contendo camadas de convolução, sendo essa a principal diferença entre uma rede neural contendo apenas camadas totalmente conectadas. A convolução processa dados em forma de múltiplas matrizes, normalmente contendo uma dimensão (1D), duas dimensões (2D) ou três dimensões (3D). Camadas contendo 1D são utilizadas para o processamento de textos, onde cada palavra é representada por um vetor de características. Camadas contendo 2D são utilizadas no processamento de imagens, visto que uma imagem contém altura e largura. Finalmente, camadas contendo 3D são utilizadas no processamento de vídeos, onde um grupo de imagens contendo 2D são agrupadas para dar o aspecto temporal de um vídeo. Cada camada de convolução aprende um tipo característica da imagem. Uma rede convolucional tipicamente usa 4 tipos de camadas: camada de convolução, camada de ativação, camada de agrupamento ou *pooling* e camada totalmente conectada (FC) [LBH15].

Camada de convolução: Essa camada tem a função de extrair características da imagem através da multiplicação da imagem de entrada por uma matriz de pesos, e como resultado obtém-se um mapa de características. Tais mapas de características permitem detectar contornos, cores e limiares [LBH15].

A convolução contém filtros ou *kernels* que percorrem toda a imagem, multiplicando seus valores pelos valores contidos na imagem. Após realizar a convolução de um filtro, o mesmo é deslocado para próxima região na horizontal ou na vertical, sendo o tamanho desse deslocamento chamado de *stride*. A convolução pode diminuir o tamanho da imagem, pois ao aplicar a convolução com um filtro sobre borda da imagem, ela vai reduzindo de tamanho. A solução para não ocorrer a redução no tamanho da imagem é a aplicação de *padding*, *i.e.*, adicionar zeros na borda da imagem. Dessa forma a convolução consegue extrair todas as características sem perdas [LBH15]. A figura 2.4 mostra 3 etapas da convolução com um filtro de 3×3, *stride*=2 e *padding*=1.

Camada de ativação: Uma das funções de ativação muito utilizadas é conhecida como *Rectified Linear Unit* (ReLU). Essa função somente é ativada quando os valores de entrada são positivos, atribuindo zero para valores negativos, como mostra a Equação 2.4. O ReLU acelera a convergência do gradiente e tem baixo custo de processamento, sendo de fácil implementação [LBH15].

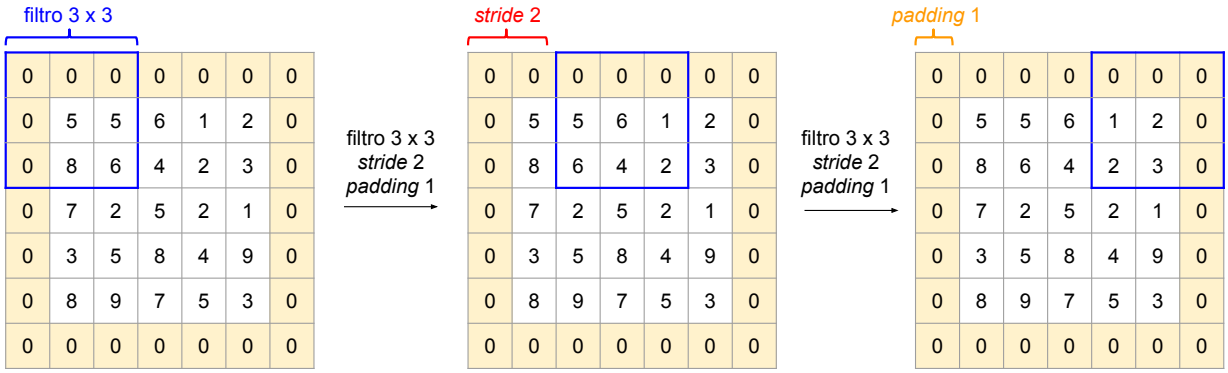


Figura 2.4 – Convolução com filtro: 3 x 3, *stride:* 2, *padding:* 1

$$f(x) = max(x, 0) \tag{2.4}$$

Camada de pooling: Essa camada reduz a dimensão da saída do mapa de características, facilitando assim o gerenciamento e reduzindo a quantidade de memória necessária para guardar as informações [LBH15]. Umas das formas utilizadas para reduzir a dimensionalidade se chama *MAX Pooling*. O algoritmo divide a matriz em subconjuntos e para cada subconjunto extrai apenas o maior valor, gerando uma matriz resultante contendo apenas os maiores valores da matriz anterior, como mostra a Figura 2.5.

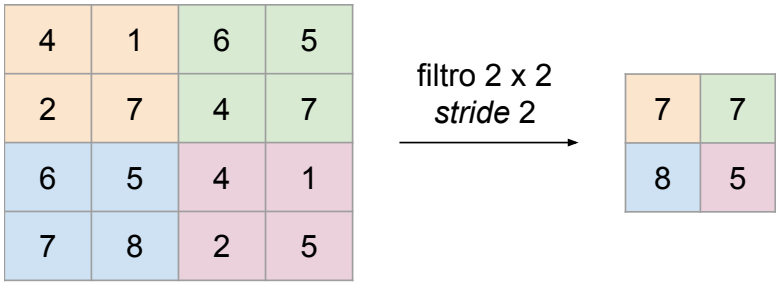


Figura 2.5 – *Max Pooling* com filtro 2×2 e *stride*=2. Adaptado de [Uni17].

Camada totalmente conectada: A camada FC, como explicado na Seção 2.1 contém ligações com todos os neurônios da camada anterior. Esse grande número de conexões torna a FC muito onerosa para o sistema, consumindo mais memória e processamento. A saída de uma FC também é diferente de uma convolução, pois enquanto uma convolução gera um mapa de características, em uma FC é gerado um vetor de características [FLGC11].

2.3 Transferência de conhecimento

De acordo com Pan e Yang [PY10], a transferência de conhecimento é a reutilização do conhecimento já adquirido durante o treinamento de um modelo. Este processo

é utilizado para reduzir o esforço necessário que se tem ao treinar um modelo do zero. Por exemplo, treinamos uma rede neural para detectar cachorros, e depois precisamos que ela reconheça também os gatos. Então utilizamos a transferência de conhecimento para ela usar o que ela já conhece de cachorro para detectar os gatos. Como os cachorros e os gatos são um pouco parecidos já que ambos tem quatro patas, focinho, pelos, *etc.*, o conhecimento adquirido irá ajudar no treinamento para detectar gatos, sem prejudicar a detecção dos cachorros. Essa transferência de aprendizado é válida principalmente nos níveis mais genéricos de características, normalmente extraídos nas primeiras camadas de convolução. Assim, ao invés de inicializarmos os pesos da rede neural aleatoriamente ou com uma distribuição Gaussiana por exemplo, inicializamos com os valores de outro modelo pré-treinado.

A transferência de conhecimento permite a utilização de todos os pesos de um modelo pré-treinado ou apenas de algumas camadas. Li e Hoiem [LH17] apresentam 3 formas de realizar a transferência de conhecimento:

- Extração de características (do inglês *feature extraction*): os pesos das camadas que foram inicializadas com um modelo pré-treinado, não sofrem alteração durante o treinamento, sendo usado somente para extração das características.
- Afinação (do inglês *fine-tuning*): ao inicializar as camadas da rede com o pesos do modelo pré-treinado, os pesos de algumas camadas são ajustados durante o treinamento. Geralmente se mantém as primeiras camadas sem alteração, usando apenas para extração de características, e as outras atualizando normalmente.
- Treinamento conjunto (do inglês *joint training*): Consiste em inicializar a rede com os pesos do modelo pré-treinado, com todas as camadas liberadas para serem ajustadas durante o treinamento do modelo.

Em nosso trabalho optamos por utilizar a extração de características, usando um modelo pré-treinado para detectar objetos e um modelo pré-treinado para classificar ambientes. A técnica de extração de característica se ajusta bem ao problema, pois não vamos precisar treinar um detector de objetos e nem um classificador de ambientes, ganhando tempo de treinamento e se beneficiando do conhecimento já adquirido por esses modelos.

2.4 Classificação de ambientes usando CNN

A classificação de imagens consiste em atribuir uma classe a uma imagem com um rótulo. Em alguns trabalhos, como no de Deng *et al.* [DDS+09], a classificação da imagem é realizada usando o objeto que mais se destaca. Em outros trabalhos, a classificação é realizada analisando apenas o fundo da imagem e rotulando-a conforme a cena, como em

Zhou *et al.* [ZLK+17]. Assim, no *dataset* ImageNet [DDS+09] as imagens são rotuladas como objetos, e nos *datasets* Places205-Standard e Places365-Standard [ZLK+17] as imagens são rotuladas como cenas. As redes neurais treinadas em *datasets* centrados nos objetos tendem a extrair as características dos objetos, enquanto as redes treinadas em *datasets* centrados nas cenas extraem características da cena.

Zhou *et al.* [ZLK+17] fez uma comparação de 3 redes neurais (AlexNet [KSH12], GoogLeNet [SLJ+15], VGG-16 [SZ15]) treinadas em diferentes *datasets* e testadas em *datasets* centrados em cenas (SUN397 [XHE+10], MIT Indoor67 [QT09], Scene15 [LSP06], SUN Attribute [PH12]) e em *datasets* centrados em objetos (Caltech101 [FFFP04], Caltech256 [GHP07], Action40 [YJK+11], Event8 [LFF07]). Podemos ver na Tabela 2.1 que as redes neurais treinadas nos *datasets* Places205-Standard e Places365-Standard obtiveram melhor resultado que as redes treinadas no ImageNet, visto que os *datasets* testados tem como foco principal o ambiente e não os objetos. Porém quando são testados em *datasets* com foco principal nos objetos, como mostrado na Tabela 2.2, as redes treinadas com o *dataset* ImageNet obteve melhor resultado do que as redes treinadas na Places205-Standard e Places365-Standard. Os *datasets* centrados em objetos não tem muita informação da cena, pois a maioria dos objetos preenchem quase toda a imagem.

Tabela 2.1 – Comparação realizada por Zhou *et al.* [ZLK+17] das abordagens de classificação de imagens usando os *datasets* SUN397 [XHE+10], MIT Indoor67 [QT09], Scene15 [LSP06], SUN Attribute [PH12] centrados na cena.

Treinamento		Teste / Acurácia			
Rede	Dataset	SUN397	MIT Indoor67	Scene15	SUN Attribute
AlexNet	Places365	**56.12%**	**70.72%**	89.25%	92.98%
	Places205	54.32%	68.24%	**89.24%**	**92.71%**
	ImageNet	42.61%	56.79%	84.05%	91.27%
GoogLeNet	Places365	**58.37%**	73.30%	**91.25%**	**92.64%**
	Places205	57.00%	**75.14%**	90.92%	92.09%
	ImageNet	43.88%	59.48%	84.95%	90.70%
VGG16	Places365	**63.24%**	76.53%	**91.97%**	**92.99%**
	Places205	61.99%	**79.76%**	91.61%	92.07%
	ImageNet	48.29%	64.87%	86.28%	91.78%

Para poder extrair característica da cena, foi utilizado nesse trabalho a Places365-CNN por ter maior acurácia na classificação de ambientes nos *datasets* centrados em cena e por possuir um modelo pré-treinado. A Places365-CNN [ZLK+17] é um modelo de rede que contém os pesos de uma CNN pré-treinada usando 1.803.460 imagens do Places365-Standard *dataset*. As imagens foram rotuladas utilizando o serviço de *crowdsourcing* oferecido pela *Amazon Mechanical Turk* (AMT) e contém uma lista de categorias de ambientes encontrados ao redor do mundo, como por exemplo quarto, plataforma de trem, centro de

Tabela 2.2 – Comparação realizada por Zhou *et al.* [ZLK+17] das abordagens de classificação de imagens usando os *datasets* Caltech101 [FFFP04], Caltech256 [GHP07], Action40 [YJK+11], Event8 [LFF07] centrados nos objetos.

Treinamento		Teste / Acurácia			
Rede	Dataset	Caltech101	Caltech256	Action40	Event8
AlexNet	Places365	66.40%	46.45%	46.82%	90.63%
	Places205	65.34%	45.30%	43.26%	94.17%
	ImageNet	**87.73%**	**66.95%**	**55.00%**	**93.71%**
GoogLeNet	Places365	61.85%	44.52%	47.52%	91.00%
	Places205	54.41%	39.27%	45.17%	92.75%
	ImageNet	**89.96%**	**75.20%**	**65.39%**	**96.13%**
VGG16	Places365	67.63%	49.20%	52.90%	90.96%
	Places205	67.58%	49.28%	53.33%	93.33%
	ImageNet	**88.42%**	**74.96%**	**66.63%**	**95.17%**

conferência, escritório de veterinário, *etc.*. No total são 365 classes de ambientes, agrupadas em 3 categorias e 16 subcategorias, como mostra a Tabela 2.3.

Tabela 2.3 – *Places365-Standard dataset*: categorias e subcategorias.

Categorias	Subcategorias
Indoor	“shopping and dining”, “workplace”, “home or hotel”, “transportation/stations”, “sport and leisure”, “Cultural”
outdoor/natural	“water, ice, snow”, “mountains, hills, desert, sky”, “forest, field, jungle”, “man-mad eelementes”
outdoor/man-made	“transportation/roads”, “cultural or historical building/place”, “sports fields, parks, leisure spaces”, “industrial and construction”, “houses, cabins, gardens and farms”, “commercial buildings. shops, markets, cities and towns”

Zhou *et al.* [ZLK+17] disponibilizou um modelo da rede pré-treinado usando 3 CNNs populares: AlexNet [KSH12], GoogLeNet [SLJ+15], e VGG-16 [SZ15]. Em nosso trabalho, utilizamos a versão com a VGG-16 pré-treinada, pois esta rede obteve melhores resultados nos testes apresentados por Zhou *et al.* [ZLK+17]. O modelo pré-treinado da VGG-16 é livremente disponível no site do *MIT CSAIL Computer Vision*[1]. A Places365-CNN na versão com a VGG-16 é uma rede neural com 16 camadas, sendo 13 camadas de convolução e 3 camadas totalmente conectadas (FC), como mostra a Figura 2.6. Nessa dissertação chamaremos o *dataset* de Places365-Standard e a CNN de Places365-CNN.

A rede tem como entrada uma imagem com 3 canais de cores: vermelho, verde e azul (RGB), tamanho fixo contendo 224 de altura e 244 de largura. As camadas de con-

[1] https://github.com/CSAILVision/places365

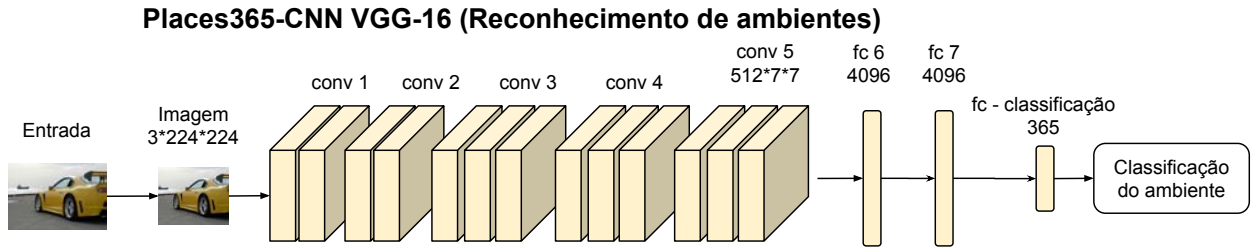


Figura 2.6 – Arquitetura da Places365-CNN na versão VGG-16 para fazer a classificação de ambientes

volução são agrupada duas a duas (*conv1* e *conv2*) e três a três (*conv3*, *conv4* e *conv5*), com uma camada de ativação ReLU depois de cada convolução e uma camada de *Pooling* depois de cada grupo. A configuração do *Pooling* na VGG-16 reduz pela metade as dimensões do mapa de características. A Figura 2.7 ilustra a *conv1*, onde ela gera 64 mapas de características com dimensões de 224×224, visto que a imagem possuí o tamanho de 224×224, a *conv1* tem 64 neurônios, filtro 3×3 e *padding*=1. Cada neurônio da convolução vai gerar um mapa de características, então o *Pooling* redimensiona o tamanho do mapa de características para 112×112.

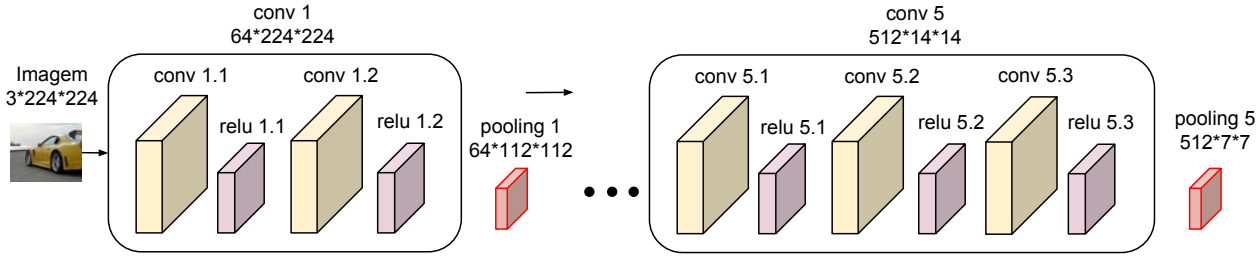


Figura 2.7 – Camadas de convolução da VGG-16 [SZ15].

Na ultima camada de convolução (*conv5.3*) a rede possuí 512 mapas de características com dimensões de 7×7 depois do *pooling 5*. Essa informação é então transformada em um vetor de características contendo 25.088 posições e enviado para a camada totalmente conectada com 4.096 neurônios, a qual multiplica os pesos por cada neurônio de entrada. Por fim, as características são enviadas para a camada de classificação que classifica entre o número de classes do *dataset*, o qual a rede foi treinada. A VGG-16 do modelo Places365-CNN por ser treinada em um *dataset* com 365 classes, possuí na FC de classificação 365 neurônios, representando os 365 ambientes. Os valores que cada neurônio da FC de classificação são transformados em probabilidades pelo *Softmax*.

2.5 Detecção de objetos usando CNN

A detecção de objetos consiste em localizar o objeto na imagem, informando as coordenadas onde ele se encontra, e classifica-lo dentre as classes conhecidas do *dataset*. Para essa tarefa é necessário que a rede neural faça uma regressão e uma classificação. Na regressão são preditas as coordenadas dos objetos (*bounding boxes*) e na classificação é predita a classe.

Redmon *et al.* [RF17] apresenta uma comparação entre as melhores arquiteturas para detecção de objetos. A comparação é realizada utilizando os *datasets* PASCAL VOC 2007 e PASCAL VOC 2012 para treinamento e o *dataset* PASCAL VOC 2007 [EVGW+10] para teste, conforme apresentado na Tabela 2.4. A arquitetura que obteve o melhor resultado no geral foi a YOLOv2 [RF17] com 78.6% mAP, porém escolhemos a Faster R-CNN [RHGS15] por ser uma das arquiteturas que possuí resultados próximos ao estado da arte, por ter um modelo pré-treinado e código disponível.

Tabela 2.4 – Comparação realizada por Redmon *et al.* [RF17] das abordagens de detecção de objetos usando o *dataset* PASCAL VOC 2007 [EVGW+10].

Arquiteturas	VOC 2007 Test / mAP
YOLOv2 [RF17]	78.6%
SSD500 [LAE+16]	76.8%
Faster R-CNN ResNet [HZRS16]	76.4%
SSD300 [LAE+16]	74.3%
Faster R-CNN VGG-16 [RHGS15]	73.2%
Fast R-CNN [Gir15]	70.0%
YOLO [RDGF16]	63.4%

Neste trabalho, propomos a LeanNet que utiliza a Faster R-CNN com a VGG-16 [RHGS15] por ser uma rede neural que obteve bons resultados, não utiliza muita memória comparado com a versão usando a ResNet [HZRS16] e tem modelo pré treinado no *dataset* PASCAL VOC 2007 [EVGW+10].

Faster R-CNN: Desenvolvida por Ren *et al.* [RHGS15], a Faster R-CNN é uma rede neural profunda (do inglês, *deep neural network*) para detectar objetos. Essa rede possui dois módulos: (a) uma rede de proposta de região (do inglês, *Region Proposal Network* (RPN)) e (B) a Fast R-CNN [Gir15].

O módulo RPN, mostrado na Figura 2.8, contém uma CNN do tipo VGG-16 [RHGS15] que recebe uma imagem como entrada e gera uma saída com um grupo de regiões propostas, *i.e.*, delimitação da área em que os objetos se encontram. Cada região proposta contém um valor de probabilidade associado representando a chance do objeto aparecer na região. O módulo produz duas saídas: uma para classificação com 2 classes (contém

objeto e não contém objeto) e uma para a regressão das 4 coordenadas do *bounding box* (x,y, altura, largura). É importante notar que a posição $x = 0$ e $y = 0$ se encontra no canto superior esquerdo da imagem.

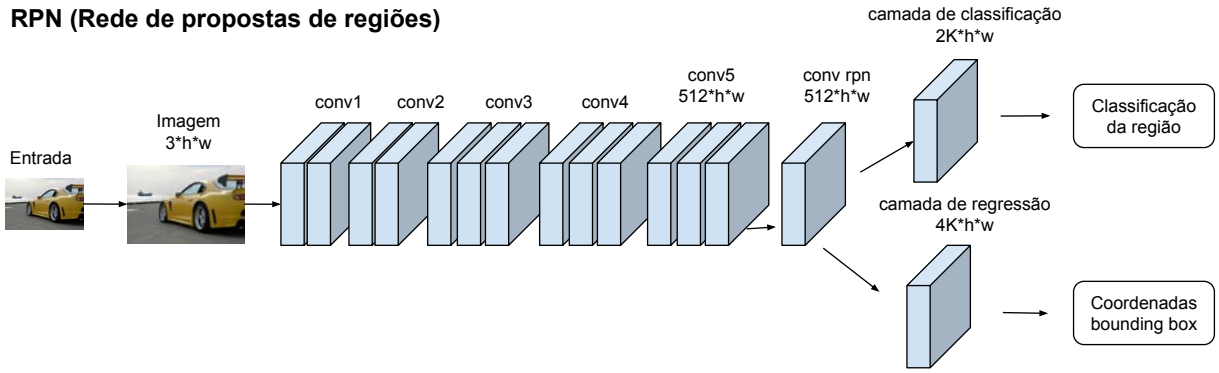


Figura 2.8 – Arquitetura da rede de propostas de regiões (RPN).

As regiões propostas são selecionadas pelo uso de uma janela deslizante sobre o último mapa de características da camada de convolução (*conv5.3*), para determinar se a região contém um objeto ou não. Cada janela deslizante pode gerar múltiplas regiões propostas, através de K âncoras. Uma âncora é uma região dentro da janela deslizante que captura os objetos e prediz as suas coordenadas através do ajuste das dimensões. A rede classifica como contendo objeto na âncora se o *intersection over union* (IoU) da região for maior que 70% sobre as regiões do *ground thuth*, e como não objeto se o escore for menor que 30%. Regiões que têm um IoU entre 30% e 70% não contribuem para o treinamento.

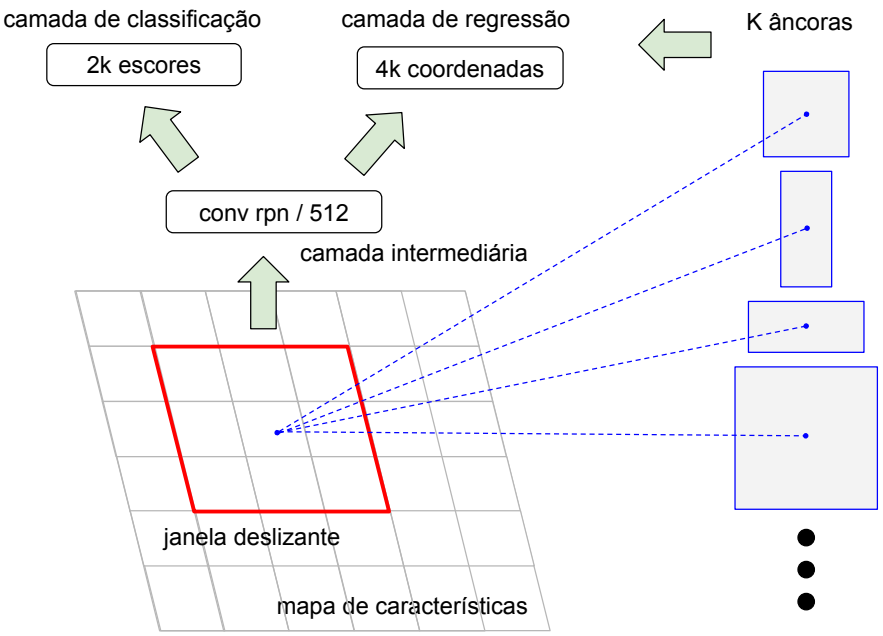


Figura 2.9 – Visualização das ancoras sobre o mapa de características da camada de convolução 5.3. Adaptado de [RHGS15].

O numero de saídas da camada de classificação é $2 \times K$ âncoras e da regressão é $4 \times K$ âncoras, como vemos na Figura 2.9.

O segundo módulo, Fast R-CNN [Gir15] usa as regiões propostas que contém objetos, através da camada de região de interesse (do inglês *Region of Interest* (RoI)). A camada RoI pega as regiões propostas projetadas no mapa de características da camada de convolução 5.3 e redimensiona para um tamanho fixo (*e.g.*, 7×7). As regiões são então agrupadas e encaminhadas para as camadas totalmente conectadas (FC) para fazer a classificação e a regressão. A FC de classificação tem a saída igual ao número de classes, adicionando uma de *background*, enquanto a FC de regressão possui na saída as 4 coordenadas dos *bounding boxes* de cada classe (cls + 1). A classe *background* é utilizada quando uma região região de interesse não é classificada entre nenhuma das classes de objetos do *dataset*.

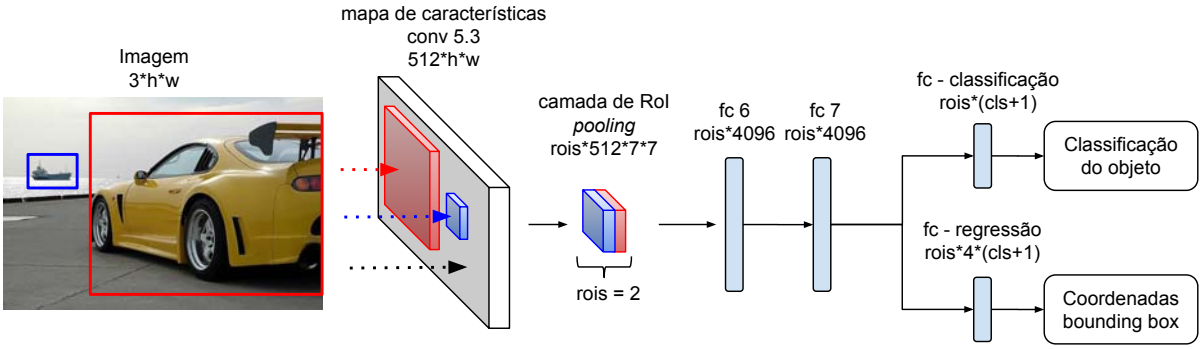


Figura 2.10 – Arquitetura da Fast R-CNN. Adaptado [Gir15].

Podemos ver na Figura 2.10 que o agrupamento realizado pela camada RoI altera a dimensão (rois) para 2 em todas as próximas camadas da rede. Esse número é variável de acordo com o número de objetos na imagem. Normalmente essa dimensão é utilizada para enviar um grupo de imagens para realizar o treinamento em uma rede neural. Porém, a Fast R-CNN a utiliza para enviar um grupo de regiões de interesse. Por esse motivo o treinamento da rede é realizado passando somente uma única imagem de cada vez, com cada imagem produzindo até N regiões de interesse, sendo N um número definido nas configurações.

Ren *et al.* [RHGS15] desenvolveu a arquitetura em dois formatos, usando a ZF-NET [ZF14] e a VGG-16 [SZ15]. No nosso trabalho utilizamos o formato da rede neural convolucional VGG-16, mostrado da Figura 2.11, para realizar os experimentos. O modelo *end-to-end* proposto treina a RPN e a Fast R-CNN em conjunto. Podemos ver que a principal diferença entre as VGG-16, utilizadas nas abordagens de detecção de objetos e na classificação de ambientes, é a adição do módulo RPN (*conv rpn*, *rpn cls escore*, *rpn bbox reg*) e a camada *RoI pooling* da Fast R-CNN [Gir15]. A quantidade de camadas de saídas também são diferentes, pois a Faster R-CNN possuí além da camada de classificação (FC - classificação) uma de regressão (FC - regressão).

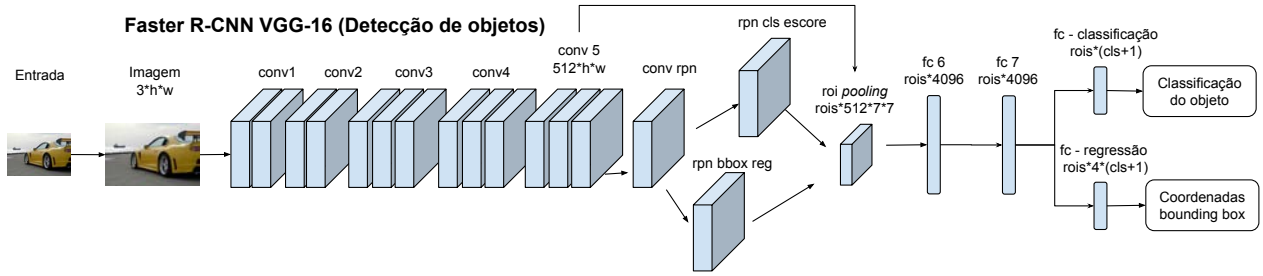


Figura 2.11 – Arquitetura da Faster R-CNN na versão VGG-16 para fazer a detecção de objetos sem o uso do contexto.

3. ARQUITETURA LEANNET

Os estudos realizados por Galleguilos e Belongie [GB10] mostram a importância de usar o contexto para obter informações sobre o reconhecimento de certos objetos. Segundo eles, o contexto pode ser explorado na visão computacional de três formas:

1. Contexto semântico: descrito pela probabilidade de um objeto presente em uma cena e não em outras cenas, *e.g.*, um computador e um teclado são mais comuns de aparecerem na mesma imagem do que um computador e um carro.

2. Contexto espacial: a probabilidade para encontrar um objeto em certa posição do que outros objetos na cena, *e.g.*, um teclado é esperado estar abaixo do monitor.

3. Contexto de escala: o tamanho do objeto em contexto com a relação de outro objeto na cena, *e.g.*, um teclado é esperado ser menor do que uma pessoa na cena.

Em nosso trabalho, exploramos o contexto semântico como a probabilidade de um objeto estar na cena e não em outras cenas. O contexto espacial não foi utilizado porque já foi exaustivamente explorado nos trabalhos citados no capitulo 6. Não utilizamos o contexto de escala por questões do escopo e ocorrer problemas como ilusão de ótica.

É importante notar que não são todos os objetos que tem relação forte com o contexto. Por exemplo, uma pessoa pode aparecer em muitos lugares como uma casa ou em uma avenida, enquanto um hidrante tende a aparecer sempre na calçada. Em nosso trabalho, a probabilidade de um objeto ser detectado em uma cena tende a aumentar quando o objeto tem uma forte relação com o contexto e a reduzir quando o objeto está fora de contexto. A fim de melhorar o reconhecimento de objetos, propomos uma abordagem que une duas redes neurais pré-treinadas para a extração de informações do objeto e do contexto separadamente. Consequentemente, a probabilidade dos objetos detectados pode ser alterada pelo contexto em que o objeto está inserido.

A nossa abordagem é separada em quatro módulos: (a) pré-processamento, (b) detector de objetos, (c) classificador de ambientes e (d) LeanNet, como ilustrado na Figura 3.1. A arquitetura recebe uma imagem como entrada e realiza o pré-processamento para ajustar a entrada para cada arquitetura (Faster R-CNN e Places365-CNN). A Faster R-CNN extrai características dos objetos enquanto a Places365-CNN extrai características do contexto da cena de uma versão da imagem pré-processada da imagem original. Finalmente, as características extraídas de ambas as redes são concatenadas na LeanNet, onde também ocorre a classificação do objeto. A predição dos *bounding boxes* ocorre na Faster R-CNN e a classificação dos ambientes na Places365-CNN.

Nossa abordagem é inspirada na redes do tipo *two-stream* [SZ14], que implementam duas redes em paralelo, contendo uma fusão antes do processo de classificação. Tais

redes tem a característica de fundirem uma rede contendo uma *stream* para imagens estáticas e outra *stream* para imagens temporais [SZ14, KTS+14, LWL+17]. Em nossa abordagem, ao invés de utilizar uma *stream* para imagens e outra para dados temporais, utilizamos uma *stream* para a detecção de objetos e outra *stream* para a detecção do contexto no qual os objetos aparecem.

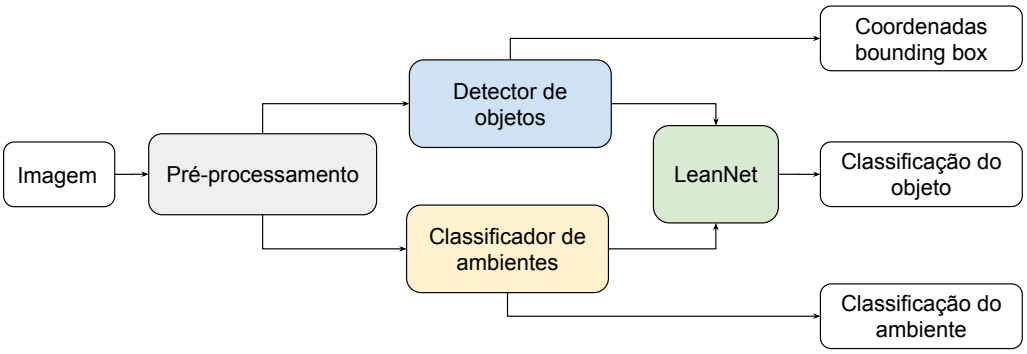


Figura 3.1 – Arquitetura da nossa abordagem LeanNet, que consiste na concatenação do detector de objetos (Faster R-CNN) e do classificador de ambientes (Places365-CNN).

3.1 Pré-processamento

O pré-processamento das imagens pretende ajustar o tamanho para a entrada da arquitetura Places365-CNN. Enquanto a Faster R-CNN recebe imagens em alta resolução com diferentes tamanhos, redimensionando-as e mantendo as proporções nos limites de 600px de altura ou 1000px de largura, a Places365-CNN trabalha com baixa resolução e tamanho fixo. Com o objetivo de atender os requisitos da Places365-CNN, primeiro transformamos a imagem de entrada na forma de um quadrado através de um corte na maior dimensão mantendo a menor dimensão, *e.g.*, uma imagem de entrada com 333×500 é cortada para 333×333. Por fim, a imagem cortada é redimensionada para 224×224, que corresponde a entrada da imagem da Places365-CNN, como mostra a Figura 3.2.

3.2 Redes pré-treinadas

Acreditamos que pelo desempenho apresentado nos artigos da Faster R-CNN [RHGS15] e da Places365-CNN [ZLK+17] o uso de redes neurais pre-treinadas é de grande contribuição para o reconhecimento dos objetos e do ambiente. A Faster R-CNN sendo especifica para a detecção dos objetos consegue extrair características mais detalhadas dos objetos, enquanto a Places365-CNN analisa toda a imagem abstraindo os objetos que aparecem na cena, capturando melhor as características do fundo da imagem. Essas duas

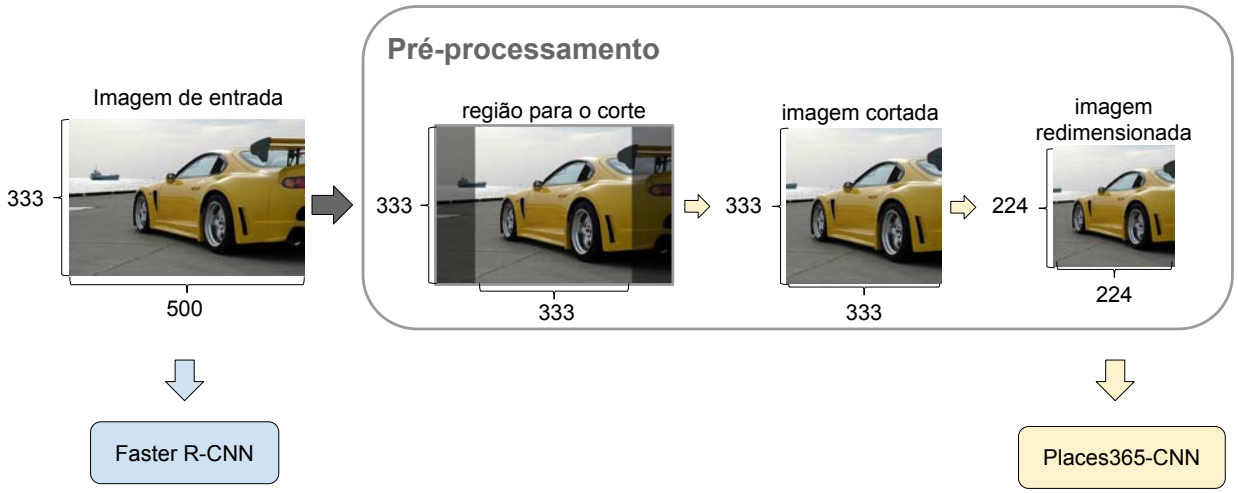


Figura 3.2 – Pré-processamento da imagem para ajustar a entrada da Places365-CNN.

informações são uteis para a rede neural aprender as relações entre os objetos e o contexto. Além disso, como as redes neurais já estão treinadas teríamos um ganho em relação ao tempo necessário para realizar o treinamento da nossa abordagem, pois só precisamos treinar uma pequena parte para aprender as relações entre os objetos e a cena.

3.3 Fusão utilizando LeanNet

Nosso método de fusão pretende unir informações do ambiente, *i.e.*, o contexto de um objeto, com a informação de cada objeto para melhorar a classificação. O método de fusão é inspirado pela abordagem proposta por Li *et al.* [LWL+17], que concatena duas camadas totalmente conectadas antes da camada que faz a classificação. A nossa abordagem LeanNet não altera os pesos das camadas da Places365-CNN e da Faster R-CNN, usando as redes apenas para a extração de características. A camada que faz a classificação do objeto é a única que tem os pesos atualizados pelo *backpropagation*, de forma a aprender as relações entre cada objeto e o ambiente.

É importante notar que a Faster R-CNN gera regiões de interesse (RoI) na saída para cada imagem de entrada, e a Places365-CNN extrai características de uma única imagem. Assim, antes da camada de concatenação é necessário multiplicar o número de características da Places365-CNN para o mesmo numero de RoIs gerados pela Faster R-CNN, mantendo assim, a saída de ambas as *streams* com a mesma dimensão. A Figura 3.3 ilustra a camada de replicação, camada “rep”, e a camada que realiza a fusão antes da classificação do objeto, camada “concat”.

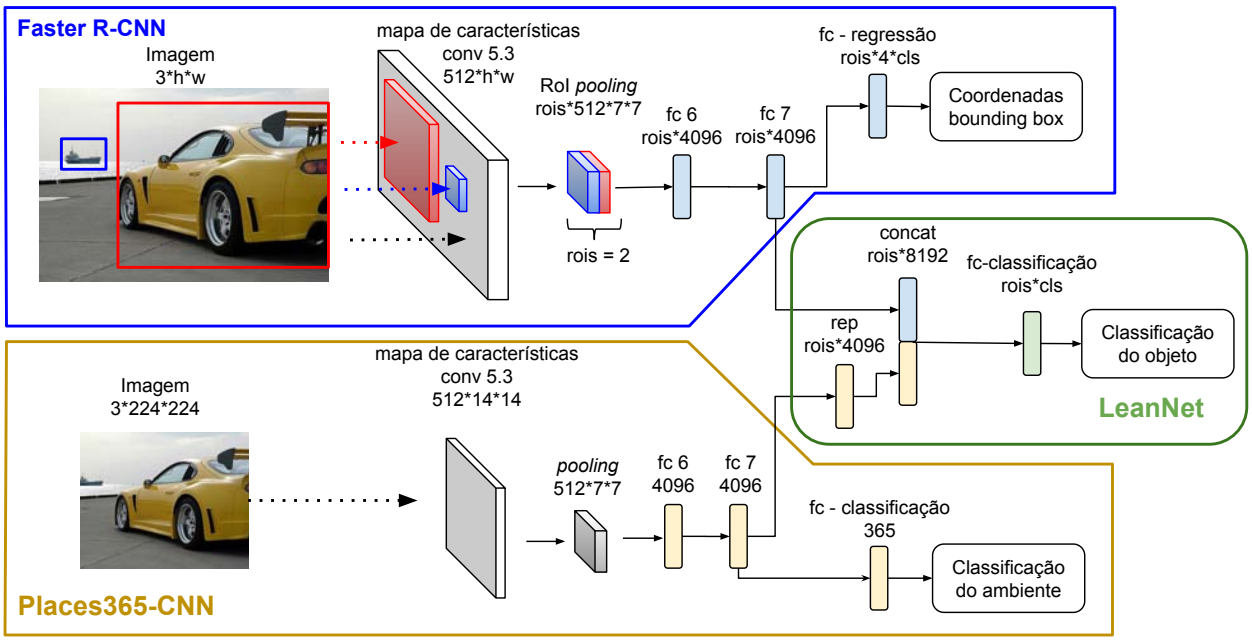


Figura 3.3 – Replicação das características da cena da Places365-CNN para o mesmo número de regiões de interesse (RoI) da Faster R-CNN.

3.3.1 Camada de replicação

A camada de replicação recebe como entrada dois vetores de características, um da rede que classifica ambientes e outro da rede que detecta objetos. O vetor de características do ambiente é replicado para um novo vetor com as mesmas dimensões do vetor de características de objetos. E para não ocupar muita memória, não realizamos uma cópia dos valores do vetor de características. Todas dimensões replicadas contém apenas o endereço de memória do vetor da rede que classifica ambientes.

3.3.2 Camada de concatenação

Recebe como entrada dois vetores com dimensões iguais e faz a concatenação das características. As características do objeto e do ambiente são colocadas uma do lado da outra. Dessa forma, a camada de classificação relaciona as características do ambiente e do objeto. A Figura 3.4 mostra um exemplo da concatenação do vetor de características das redes, com a Faster R-CNN gerando duas regiões de interesse. Podemos ver que a FC 7 da rede que classifica ambientes possui somente uma dimensão, sendo transformado para duas dimensões pela camada (Rep). A transformação para duas dimensões ocorreu porque a rede que detecta objetos gerou duas regiões de interesse, e temos que relacio-

nar cada região com uma cena. Com as duas FCs com as mesmas dimensões então é realizado a concatenação, gerando um vetor de caraterísticas com 8192 neurônios.

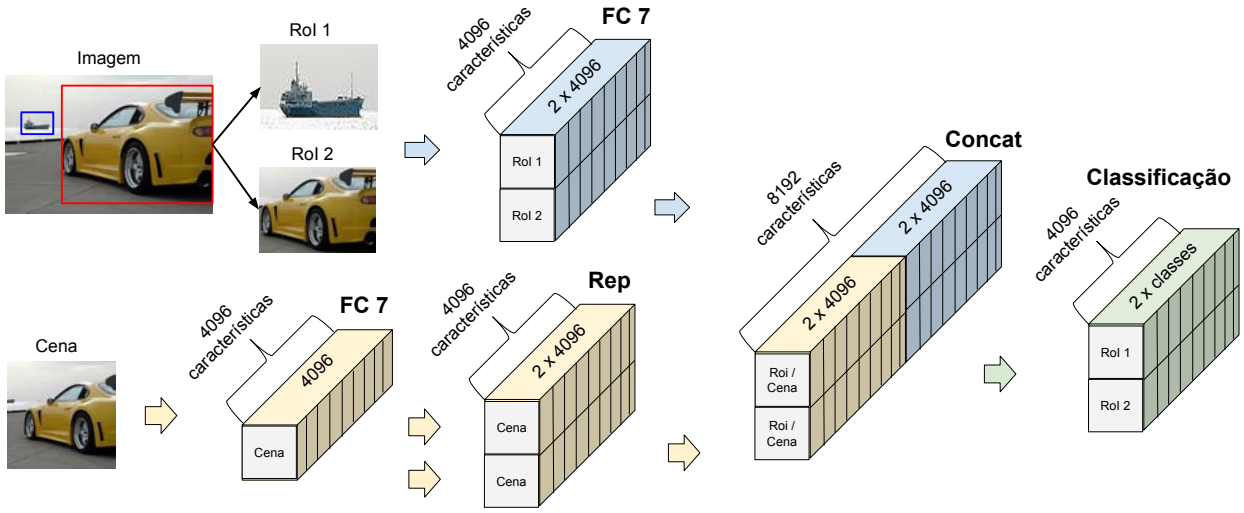


Figura 3.4 – Concatenação dos vetores de características

3.3.3 Camada de classificação

A camada de classificação da LeanNet é similar a da Faster R-CNN, ela possui o número de neurônios de saída igual o numero de classes do *dataset*, mais o *background*. A principal diferença é que ela recebe o vetor de características dos objetos e da cena, através da camada (concat). Portando, cada classe de objeto possui 8192 características, enquanto a Faster R-CNN tinha originalmente 4096.

3.4 Saída da rede

Como explicado por Li *et al.* [LWL+17], a informação do ambiente não contribui para prever as coordenadas dos *bounding boxes*. Assim, a saída para prever as coordenadas dos *bounding boxes* (FC - regressão) é conectada diretamente na camada totalmente conectada da rede que detecta objetos (FC 7), enquanto a de classificação dos objetos usando o contexto da LeanNet (FC - classificação) está conectada na camada de concatenação (concat), como ilustrado na Figura 3.3. A camada que classifica os ambientes não foi utilizada para extração das características, mas foi mantida para termos uma saída para a classificação dos ambientes e assim avaliarmos a nossa abordagem, mostrando a relação entre os objetos e o contexto. No total a LeanNet, junto com a Faster R-CNN e a Places365-CNN, possuem 3 camadas de saída, sendo uma para a predição das coordena-

das dos *bounding boxes*, uma para a classificação dos objetos usando o contexto e uma para a classificação dos ambiente.

A FC - regressão prediz para cada RoI o valor (v) das coordenadas (x, y , altura, largura) de localização dos *bounding boxes* na imagem, como mostra a Tabela 3.1. Os valores são colocados na matriz, onde as colunas são as coordenadas e as linhas as regiões de interesse. Cada linha da tabela representa um *bounding box*.

Tabela 3.1 – Formato de saída da FC - regressão dos *bounding boxes*

rois	x	y	altura	largura
roi_1	v_{10}	v_{11}	v_{12}	v_{13}
roi_2	v_{20}	v_{21}	v_{22}	v_{23}
...	...	...	...	...
roi_m	v_{m0}	v_{m1}	v_{m2}	v_{m3}

A FC - classificação dos objetos informa a probabilidade (p) que cada região de interesse (RoI) tem para cada classe de objetos (cls). Na saída é gerada uma matriz, onde cada linha corresponde a uma região de interesse (RoI) e cada coluna é uma classe de objetos, como mostra a Tabela 3.2.

Tabela 3.2 – Formato de saída da FC - classificação dos objetos

rois	*background*	cls_1	cls_2	...	cls_n
roi_1	p_{10}	p_{11}	p_{12}	...	p_{1n}
roi_2	p_{20}	p_{21}	p_{22}	...	p_{2n}
...	...	...	...	...	...
roi_m	p_{m0}	p_{m1}	p_{m2}	...	p_{mn}

A FC - classificação do ambiente gera um vetor com as probabilidades (p) para cada classe de ambiente (cena), como é mostrado na Tabela 3.3. O vetor não precisou ser replicado como a FC anterior a classificação, porque usamos a classificação de ambientes apenas para classificar a imagem e não para extração de características.

Tabela 3.3 – Formato de saída da FC - classificação dos ambientes

$cena_1$	$cena_2$	...	$cena_n$
p_1	p_2	...	p_n

A LeanNet funciona com qualquer rede que detecta objetos e ambientes, desde que sejam respeitadas as dimensões da entrada e saída da rede. O número de classes de objetos e ambientes é dinâmico, visto que não existe somente 365 tipos de ambientes e 20 ou 80 classes de objetos. Portanto, como a LeanNet utiliza como entrada as características da camada totalmente conectada antes da classificação, a quantidade de classes não

precisa ser fixa. Em nossos experimentos a Places365-CNN possui 365 classes porque usamos um modelo pré-treinado; se fossemos usar a Places205-CNN por exemplo, a rede teria 205 classes e não poderíamos usar o modelo pré-treinado da Places365-CNN e sim o da Places205-CNN.

4. EXPERIMENTOS

Nesse capitulo, descrevemos os *datasets* utilizados nos experimentos, bem como os detalhes de implementação usados em nossa abordagem e a forma de avaliação.

4.1 *Datasets*

PASCAL VOC 2007[1] [EVGW+10] é um *dataset* que consiste em imagens que representam cenas realistas, onde cada imagem tem pelo menos um objeto. Este *dataset* foi publicado através do *PASCAL Visual Object Classes Challenge 2007* e contém 20 diferentes classes de objetos, tendo um total de 9.963 imagens com 24.640 objetos anotados. As classes são agrupadas de acordo com suas características, conforme mostrado na Tabela 4.1. O *dataset* é dividido em 2.501 imagens para treino, 2.510 imagens para validação, e 4.952 imagens para teste. É importante mencionar que, como a principal intenção do PASCAL VOC 2007 é o reconhecimento de objetos, a maioria das imagens contém pouca informação sobre o contexto, com objetos cobrindo quase a totalidade da imagem.

Tabela 4.1 – Categorias e classes do *dataset* PASCAL VOC 2007.

Categorias	**Classes**
Person	person
Animal	bird, cat, cow, dog, horse, sheep
Vehicles	aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor	bottle, chair, dining table, potted plant, sofa, tv/monitor

MS COCO[2] é um *dataset* que contém imagens de cenas cotidianas e objetos em seu contexto natural. O *dataset* aborda os principais problemas de pesquisa na compreensão de cena, como a detecção de visualizações não icônicas, *i.e.*, objetos em *background*, objetos parcialmente ocluídos, e objetos em meio à desordem. A localização espacial dos objetos são anotadas usando *bounding boxes* e segmentação em nível de pixel. O *dataset* contém 80 classes para definir objetos e um total de 123.287 imagens, sendo dividido em 82.783 imagens para treinamento e 40.504 imagens para validação e teste. As classes foram agrupadas conforme mostra a Tabela 4.2. Comparando com PASCAL VOC 2007 [EVGW+10], o MS COCO contém consideravelmente mais objetos por cena, além de pequenos objetos rotulados e mais imagens.

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html
[2] http://mscoco.org/dataset/

Tabela 4.2 – Categorias e classes do *dataset* MS COCO.

Categorias	Classes
Animal	bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe
Appliance	microwave, oven, toaster, sink, refrigerator
Electronics	tvmonitor, laptop, mouse, remote, keyboard, cell phone
Food	banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake
Furniture	chair, sofa, pottedplant, bed, diningtable, toilet
Indoor objects	book, clock, vase, scissors, teddy bear, hair drier, toothbrush
Kitchenware	bottle, wine glass, cup, fork, knife, spoon, bowl
Outdoor Obj.	traffic light, fire hydrant, stop sign, parking meter, bench
Person & Acc.	person, backpack, umbrella, handbag, tie, suitcase
Sport	frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket
Vehicle	bicycle, car, motorbike, aeroplane, bus, train, truck, boat

4.2 Configuração da rede

Para realizar os experimentos, desenvolvemos nossa arquitetura com ambas *streams* contendo redes VGG-16 [SZ15]. Nossa arquitetura permite carregar dois modelos pre-treinados que executam em paralelo e uma camada totalmente conectada que faz a fusão de ambas as redes. Como não treinamos um modelo para reconhecimento da cena, um modelo pre-treinado da Places365-CNN [ZLK+17] é obrigatório para extrair características das cenas. O modelo pre-treinado na versão da VGG-16 é livremente disponível em *MIT CSAIL Computer Vision Website*[3]. Toda a arquitetura proposta foi desenvolvida utilizando o *framework* CAFFE[4].

Em nossos experimentos, utilizamos um modelo pré-treinado no PASCAL VOC 2007 [EVGW+10], bem como treinamos a partir do zero um modelo usando MS COCO [LMB+14]. A versão do modelo pré-treinado da rede VGG-16 contém pesos aprendidos usando o PASCAL VOC 2007 que é livremente disponível junto com a rede Faster R-CNN[5]. A rede treinada a partir do zero no *dataset* MS COCO contém os mesmos parâmetros descritos por Ren *et al.* [RHGS15]. Esse processo consiste na inicialização dos pesos utilizando uma distribuição Gaussiana com média zero e desvio padrão 0.01. Todas as imagens têm seus pixels subtraídos pelos valores médios de pixel por canal de todas as imagens de treinamento. Usamos uma taxa de aprendizado (do inglês, *learning rate*) de 10^{-3}, deixando cair para 10^{-4} depois de 331.132 iterações (4 épocas), e um *momentum* de 0.9 com um *weight decay* de 5×10^{-4}. Todas as convoluções usam *Rectified Linear*

[3] https://github.com/CSAILVision/places365
[4] http://caffe.berkeleyvision.org/
[5] https://github.com/rbgirshick/py-faster-rcnn

activation Units (ReLU). Fixamos o número de regiões de interesse (do inglês, *Regions of Interest - RoI*) gerado pela Faster R-CNN para 300, já que esse valor alcançou os melhores resultados no trabalho de Ren *et al.* [RHGS15]. Cada iteração envia uma única imagem e a rede termina o treinamento após 496.698 iterações (6 épocas).

4.3 Configuração da fusão

Como a saída da Faster R-CNN gera até 300 regiões de interesse para cada imagem, tivemos que replicar o número de características geradas pela rede Places365-CNN para a mesma quantidade de regiões de interesse. A camada totalmente conectada de classificação da rede é treinada por 6 épocas (496.698 mil iterações) usando o MS-COCO e por 6 épocas (70.154 mil iterações) quando usando a versão com modelo pre-treinado do PASCAL VOC. A camada totalmente conectada é treinada usando uma taxa de aprendizado de 10^{-3}, reduzindo para 10^{-4} depois de 75% do treinamento em ambos *datasets*, e um *dropout* de 50% para a rede usando o *dataset* PASCAL VOC 2007.

4.4 Avaliação

Para avaliar nossa abordagem, comparamos os resultados usando o conjunto de testes do PASCAL VOC 2007 [EVGW+10] e do MS COCO [LMB+14]. A avaliação foi dividida em 5 partes: (a) média da precisão média, (B) avaliação por categoria, (c) análise das imagens, (d) comparação com trabalhos relacionados e (e) teste estatístico.

(a) Média da precisão média: Verificamos a *mean Average Precision* (mAP) utilizando a intersecção sobre a união dos *bouding boxes* (*Intersection over Union* - IoU), *i.e.*, consideramos um *bounding box* como avaliação correta se a proporção da área de sobreposição entre o *bounding box* predito e o *ground-truth bounding box*, e a área abrangida por ambos os *bounding boxes* é maior que um determinado limiar. Variamos o limiar de IoU entre 0% e 100%, incrementando em 10% a cada passo para cada rede testada. Para comparar a adição das características do contexto com características do objeto, também testamos nossa Faster R-CNN treinada do zero com parâmetros similares ao do trabalho original apresentado por Ren *et al.* [RHGS15].

(b) Avaliação por categorias: Para avaliar em quais ambientes a nossa abordagem está melhorando o reconhecimento de objetos, e quais objetos estão sendo melhorados com a ajuda do contexto, criamos uma tabela com os *datasets* PASCAL VOC 2007 e MS COCO, selecionando os *bounding boxes* com o maior IoU em relação ao *ground thuth bounding boxes* usando a Faster R-CNN [RHGS15] e a LeanNet. Inserimos a tabela no banco de dados MySQL e agrupamos as classes de objetos e classes de ambientes, em

suas respectivas categorias. Avaliamos o total de acertos que as duas abordagens obtém, as divergências pró LeanNet (quando a LeanNet acerta e a Faster R-CNN erra), as divergências pró Faster R-CNN (quando a Faster R-CNN acerta e a LeanNet erra) e o total de erros em ambas as redes.

(c) Análise das imagens: Selecionamos algumas imagens que tiveram maiores ganhos e perdas para análise, e verificamos os objetos localizados e a relação com o contexto do ambiente da imagem que foram encontrados. Também verificamos se o ambiente classificado pela LeanNet está correto, pois não temos os rótulos dos ambientes para os *datasets* PASCAL VOC 2007 e MS COCO.

(d) Comparações com trabalhos relacionados: Para fazer a comparação com os trabalhos relacionados, selecionamos os trabalhos que avaliaram o modelo usando o mAP com IoU fixado em 50% nos *datasets* PASCAL VOC 2007 ou MS COCO. Os trabalhos que usaram outra base de dados ou outras formas de avaliações, assim como os selecionados para a comparação, são descritos no capitulo 6.

(e) Teste estatístico: Para responder a seguinte questão de pesquisa: "É possível usar uma rede neural pré-treinada que classifica ambientes para extrair o contexto semântico e fazer a fusão com uma rede que detecta objetos, a fim de melhorar a classificação dos objetos?", formulamos as seguintes hipóteses:

- Hipótese nula: Não é possível, através da junção de duas redes neurais distintas, melhorar a classificação de objetos.

- Hipótese alternativa: Sim é possível, através da junção de uma rede que classifica ambientes para extrair o contexto semântico e uma rede que detecta objetos, melhorar a classificação.

Escolhemos o teste estatístico não paramétrico de *Wilcoxon* para avaliar as nossas hipóteses calculando a diferença entre os escores de cada objeto detectado pela LeanNet e a Faster R-CNN com relação a classe do *ground thuth*. O teste de *Wilcoxon* é importante para informar se os resultados são significativos e não aleatórios. Através da diferença dos escores de cada objeto entre as duas redes é realizado um ranqueamento em ordem crescente, descartando as diferenças em zero e atribuindo maior *rank* para as maiores diferenças. Após é realizado a soma dos ranqueamentos com sinal (SR) para encontrar o valor de Z, como mostra a Equação 4.1. O valor de Z é utilizado para encontrarmos o valor de P através da distribuição normal com média igual a zero e desvio padrão igual a 1 [FLGC11].

$$Z = \frac{\sum SR_i}{\sqrt{\sum SR_i^2}} \tag{4.1}$$

Utilizamos a biblioteca *scipy.stats.wilcoxon*[6] em *Python* que recebe um vetor com as diferenças e realiza o cálculo. O nível de significância escolhido foi de 0.05, *i.e.*, rejeitamos a hipótese nula se o valor de *P* (do inglês *P-Value*) for menor que o nível de significância [FLGC11].

[6]https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.wilcoxon.html

5. RESULTADOS

Nesse capitulo mostraremos os resultados dos experimentos, fazendo uma análise nos resultados obtidos com a precisão média nos dados de teste. Visto que nossa abordagem depende fortemente de contexto para melhorar os resultados, analisamos imagens em que o contexto é importante para a classificação da imagem. Por fim, realizamos o teste estatístico não paramétrico de *Wilcoxon* para testar nossa hipótese.

5.1 Média da precisão média

Conforme explicado no capítulo 4.4, os escores da média da precisão média (mAP) são encontrados por cada abordagem variando a IoU entre 0% e 100% nos dados extraídos do *dataset* de teste. Conforme ilustra a Figura 5.1, nossa abordagem (LeanNet) obtém melhores resultados quando comparada com a Faster R-CNN [RHGS15] treinada do zero sem o uso do contexto.

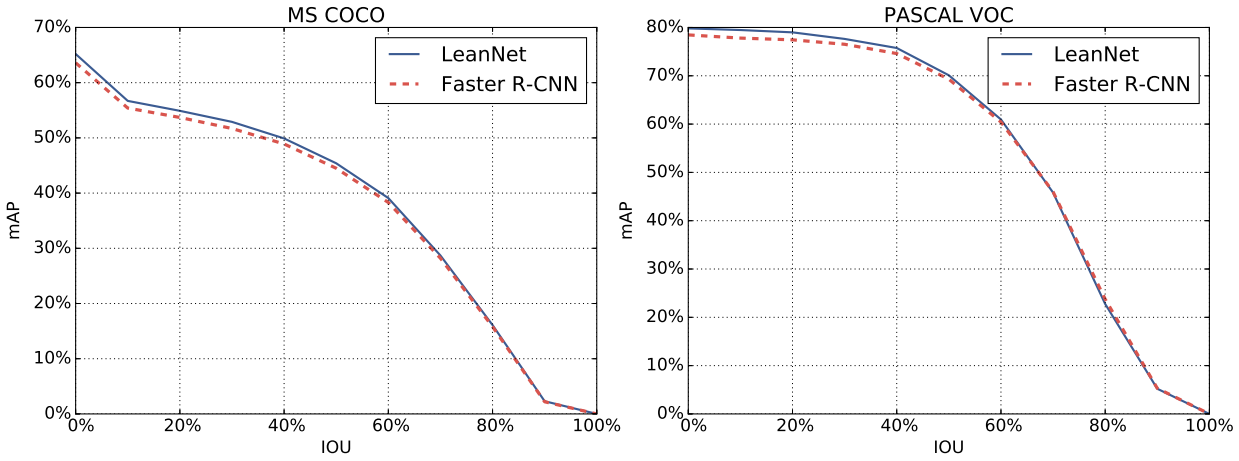


Figura 5.1 – mAP para os *datasets* MS COCO e PASCAL VOC 2007 variando os valores de *Intersection over Union* (IoU).

Embora os valores de mAP obtidos pela LeanNet sejam similares aos valores obtidos pela Faster R-CNN, é importante salientar que a LeanNet é desenvolvida com o intuito de classificar objetos que estão em um contexto, enquanto a Faster R-CNN não se preocupa em levar em conta o contexto da imagem. Assim, nossa abordagem pode aprender um determinado contexto para objetos que não dependem de contexto, *e.g.*, a classe pessoa que pode ocorrer em qualquer contexto, e em imagens de teste diminuir a probabilidade de classificação desses objetos caso eles apareçam fora do contexto aprendido. Por exemplo, nos dados de treino a classe pessoa normalmente aparece associada à cidade como contexto (prédios, carros, ruas, *etc.*.), porém nos dados de teste ela pode aparecer no campo ou na

praia, que não é o contexto aprendido nos dados de treino. Assim, a Faster R-CNN poderia classificar melhor uma pessoa, visto que ela usa apenas as características da pessoa para a classificação, enquanto nossa rede iria diminuir a probabilidade da classe pessoa por levar em conta o contexto. Uma análise deste tipo de problema é realizado nas seções seguintes.

Além da mAP, também analisamos a precisão média (*average precision* - AP) obtida em cada classe. Para tanto, fixamos o valor de IoU em 50% em ambos *datasets* PASCAL VOC 2007 [EVGW+10] e MS COCO [LMB+14]. Os resultados são apresentados nas Subseções 5.1.1 e 5.1.2.

5.1.1 Resultados para o *dataset* PASCAL VOC 2007

Conforme ilustrado na Figura 5.2, utilizando os dados do *dataset* PASCAL VOC 2007, os valores de precisão da LeanNet são superiores para quase todas as classes, perdendo somente para as classes *cow*, *cat*, *motorbike* e *aeroplane*, com diferenças de -1.16%, -1.06%, -0.38% e -0.05% respectivamente. A classe que obteve o maior ganho foi *horse* com 4.06%, seguido de *sheep* com 3.23% e *pottedplant* com 2.78% a mais que a Faster R-CNN. Observamos que o contexto não aumentou o escore de todas as classes porque alguns objetos preenchem toda a imagem, dificultando o reconhecimento do contexto.

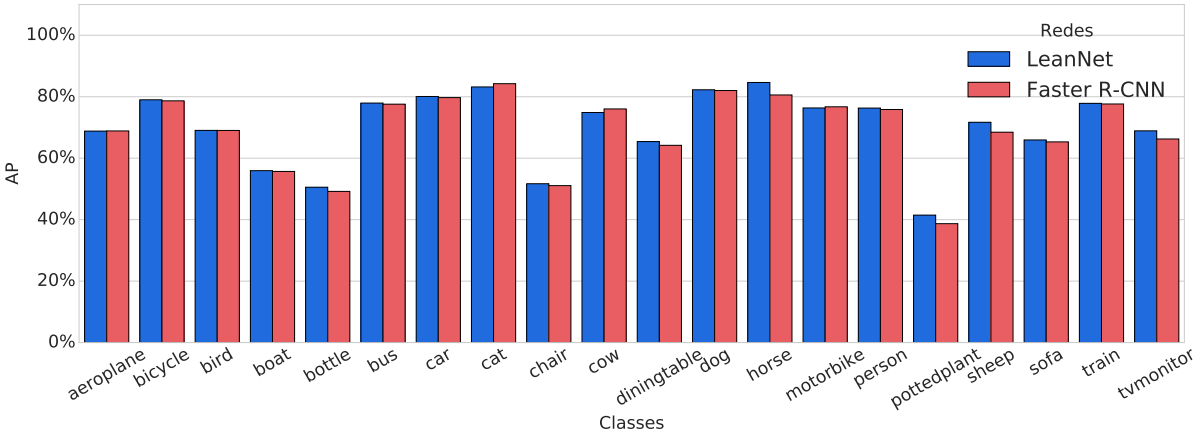


Figura 5.2 – Precisão média por classe usando IoU fixado em 50% no *dataset* PASCAL VOC 2007.

Analisando o resultado da relação das categorias de objetos com as categorias de ambientes, conforme mostrado na Tabela 5.1, a nossa abordagem perdeu apenas quando os objetos que pertencem a categoria de objetos internos (*indoor*) aparecem nas categorias de ambientes externos ("*outdoor, man-made*" e "*outdoor, natural*"). Acontece que os objetos da categoria de objetos internos que aparecem em ambientes externos estão fora de

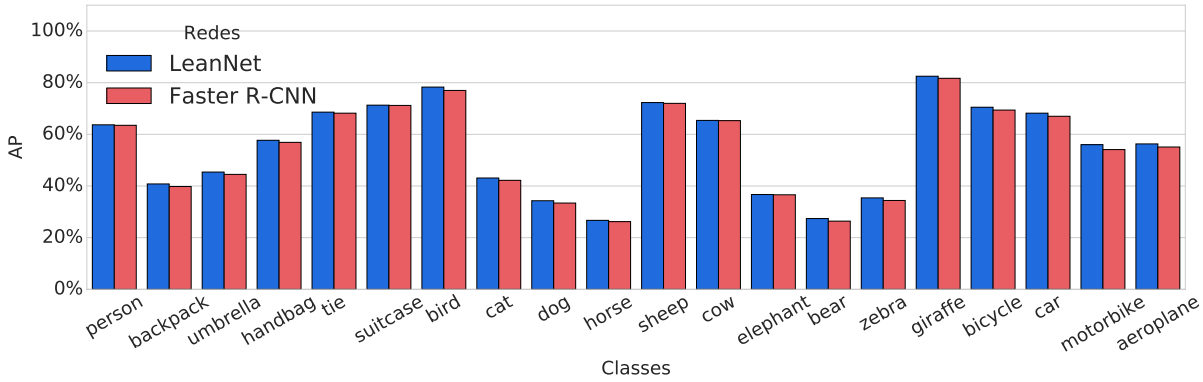
contexto, e por esse motivo a nossa abordagem não conseguiu melhorar o reconhecimento destes objetos. Por outro lado, os objetos que foram encontrados em ambientes internos tiveram uma melhora na classificação. As outras categorias de objetos tiveram ganho em todos os ambientes. Os dados da Tabela 5.1 podem ser melhor vistos na Figura A.1 do Apêndice A.

Tabela 5.1 – Quantidade de acertos, erros e diferenças entre Faster R-CNN e LeanNet, agrupado por categorias de objetos do PASCAL VOC 2007 e categorias de ambientes da Places365.

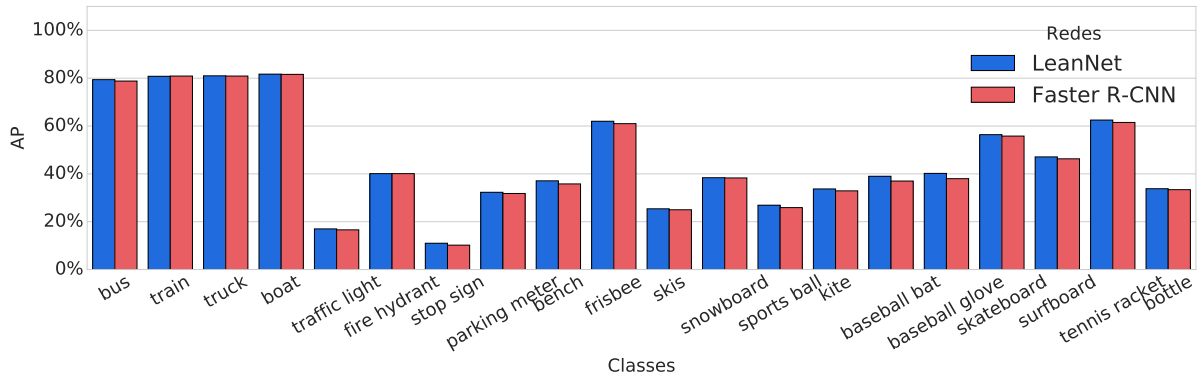
Categorias		Divergências pró		Total	
Objetos	Ambientes	LeanNet	Faster R-CNN	Acertos	Erros
Animal	*indoor*	**14.44%**	11.02%	48.20%	26.34%
	outdoor, man-made	**12.51%**	10.46%	51.61%	25.42%
	outdoor, natural	**14.10%**	10.75%	45.06%	30.09%
indoor	*indoor*	**12.30%**	10.00%	54.44%	23.26%
	outdoor, man-made	8.96%	**10.70%**	61.50%	18.84%
	outdoor, natural	9.75%	**14.98%**	51.99%	23.28%
person	*indoor*	**17.32%**	4.58%	31.41%	46.69%
	outdoor, man-made	**13.28%**	4.95%	41.17%	40.60%
	outdoor, natural	**14.46%**	6.24%	50.98%	28.32%
vehicles	*indoor*	**5.34%**	4.90%	81.47%	8.29%
	outdoor, man-made	**5.79%**	4.98%	79.06%	10.17%
	outdoor, natural	**6.67%**	4.97%	77.44%	10.92%

5.1.2 Resultados para o *dataset* MS COCO

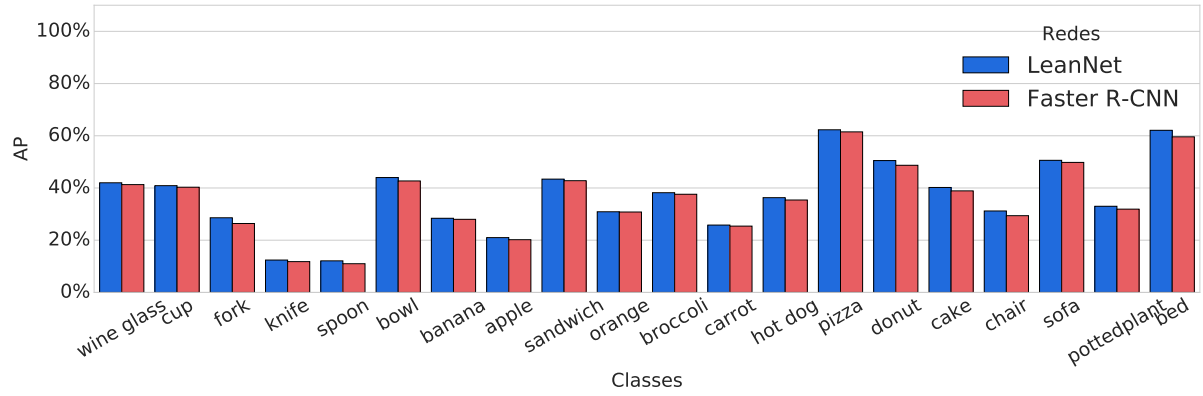
No *dataset* MS COCO, separamos as 80 classes em 4 gráficos contendo 20 classes cada, devido à limitação do espaço da página. Observando a precisão média no gráfico da Figura 5.3, percebemos que os valores de precisão da LeanNet também foram superiores para quase todas as classes, perdendo somente para a classe *train* com -0.1% de diferença. A classe com melhor resultado foi *remote* com 3.7%, seguido de *hair drier* com 2.6% e *bed* com 2.5%. Podemos notar que os dois primeiros objetos (*remote* e *hair drier*) são objetos pequenos, obtendo assim uma visualização melhor do ambiente, melhorando o reconhecimento e a inferência do contexto na classificação do objeto. Por outro lado, para a classe *train* por ser um o objeto grande, não teve melhora quando usamos o contexto. Para o contexto poder inferir sobre esses objetos maiores, os objetos devem estar um pouco distantes, permitindo assim capturar melhor o contexto da cena.



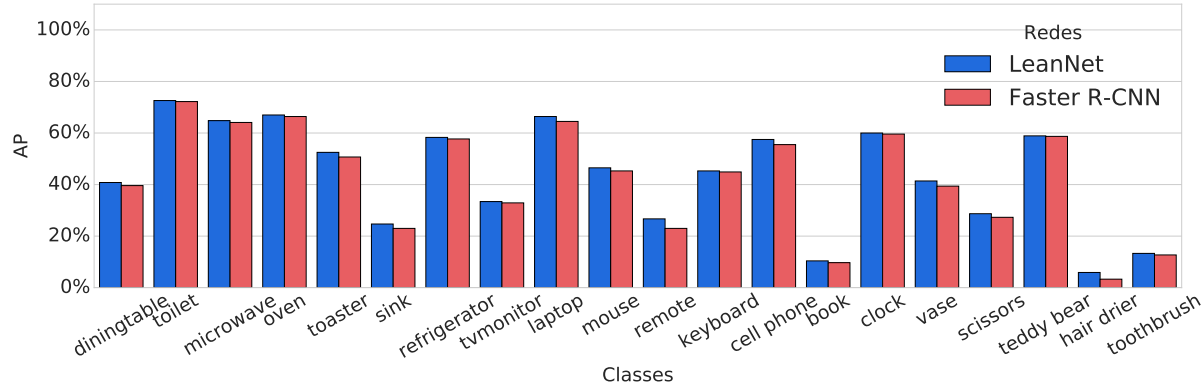
(a) MS COCO classes de 1 a 20



(b) MS COCO classes de 21 a 40



(c) MS COCO classes de 41 a 60



(d) MS COCO classes de 61 a 80

Figura 5.3 – Precisão média por classe usando IoU fixado em 50% no *dataset* MS COCO.

Na análise das relações entre as categorias de objetos do MS COCO e as categorias de ambientes da Places365-Standard, o mesmo comportamento obtido sobre o PASCAL VOC 2007 (*i.e.*, quando o objeto está fora do contexto o escore de classificação é reduzido) pode ser observado com as categorias de objetos *Furniture* e *Kitchenware*, como mostrado na Tabela 5.2. Na categoria Mobília (*Furniture*) a nossa abordagem não teve melhor resultado porque os objetos da categoria não aparecerem comumente em ambiente externos (*outdoor*), sendo assim a diferença nessas categorias de ambientes foi de -1.51% e -1.79%. Por outro lado, para ambientes internos (*indoor*), apesar de não conseguir um resultado melhor que a Faster R-CNN, a nossa abordagem conseguiu se aproximar, ficando com uma diferença de apenas -0.07%, visto que os objetos da categoria Mobília fazem parte de ambientes internos (*e.g.*, casa, quarto, escritório, *etc.*). Esse foi o único caso que uma categoria objetos em contexto não obteve um resultado superior ao da Faster R-CNN.

Na categoria de Utensílios de Cozinha (*Kitchenware*), tivemos um resultado inferior em comparação a Faster R-CNN para a categoria de ambientes externos na natureza (*outdoor, natural*), ficando com uma diferença de -2.40%. Esses resultados acontecem porque é incomum de ver utensílios de cozinha na natureza, e a nossa abordagem conseguiu aprender essa relação, reduzindo assim o escore quando encontrados fora de contexto. Nas categorias de ambientes internos (*Indoor*) e externos feito pelo homem (*outdoor, manmade*), nossa abordagem reconheceu melhor os objetos, com uma diferença de 0.60% e 1.22%, porque esses objetos aparecem com mais frequência nesses ambientes.

Para as categorias de objetos Eletrônicos (*Electronics*), Utensílios (*Appliance*) e Objetos Internos (*Indoor Objects*), nossa abordagem empatou com a Faster R-CNN quando estavam na natureza, porque essas categorias não se relacionam com natureza. Para os demais ambientes a nossa abordagem obteve um resultado superior. Nas demais categorias de objetos Animais (*Animal*), Alimentos (*food*), Objetos Externos (*Outdoor Objects*), Pessoa e Acessórios (*Person & Acessory*), Esportes (*sport*) e Veiculos (*vehicle*) nossa abordagem teve resultado superior em todas as categorias de ambientes. Os dados da Tabela 5.2 podem ser melhor vistos na Figura A.2 do Apêndice A.

Tabela 5.2 – Quantidade de acertos, erros e diferenças entre Faster R-CNN e LeanNet, agrupado por categorias de objetos do MS COCO e categorias de ambientes da Places365.

Categorias		Divergências pró		Total	
Objetos	Ambientes	LeanNet	Faster R-CNN	Acertos	Erros
Animal	*indoor*	**6.30%**	3.12%	55.74%	34.84%
	outdoor, man-made	**5.79%**	2.79%	62.93%	28.49%
	outdoor, natural	**4.96%**	3.01%	69.68%	22.35%
Appliance	*indoor*	**4.42%**	1.94%	63.06%	30.58%
	outdoor, man-made	**4.69%**	3.91%	41.40%	50.00%
	outdoor, natural	0.00%	0.00%	26.67%	73.33%
Electronics	*indoor*	**3.92%**	2.69%	58.44%	34.95%
	outdoor, man-made	**4.44%**	2.44%	36.29%	56.83%
	outdoor, natural	4.29%	4.29%	31.90%	59.52%
Food	*indoor*	**4.88%**	2.73%	58.79%	33.60%
	outdoor, man-made	**4.13%**	1.65%	64.38%	29.84%
	outdoor, natural	**3.86%**	1.75%	69.16%	25.23%
Furniture	*indoor*	3.95%	**4.02%**	61.55%	30.48%
	outdoor, man-made	3.01%	**4.52%**	50.29%	42.18%
	outdoor, natural	3.14%	**4.93%**	52.80%	39.13%
Indoor objects	*indoor*	**4.05%**	1.88%	63.85%	30.22%
	outdoor, man-made	**1.92%**	1.55%	65.05%	31.48%
	outdoor, natural	2.54%	2.54%	54.97%	39.95%
Kitchenware	*indoor*	**4.03%**	3.43%	50.36%	42.18%
	outdoor, man-made	**4.22%**	3.00%	39.59%	53.19%
	outdoor, natural	1.85%	**4.25%**	34.94%	58.96%
Outdoor Obj.	*indoor*	**5.10%**	1.80%	45.56%	47.54%
	outdoor, man-made	**2.98%**	1.63%	51.78%	43.61%
	outdoor, natural	**5.30%**	2.59%	55.83%	36.28%
Person & Acessory	*indoor*	**2.10%**	1.39%	79.82%	16.69%
	outdoor, man-made	**1.63%**	1.07%	81.13%	16.17%
	outdoor, natural	**1.69%**	1.43%	82.23%	14.65%
Sport	*indoor*	**3.24%**	2.46%	49.13%	45.17%
	outdoor, man-made	**3.62%**	1.54%	56.13%	38.71%
	outdoor, natural	**3.57%**	1.59%	63.90%	30.94%
Vehicle	*indoor*	**4.44%**	3.67%	63.14%	28.75%
	outdoor, man-made	**4.59%**	3.07%	64.90%	27.44%
	outdoor, natural	**4.35%**	3.81%	58.43%	33.41%

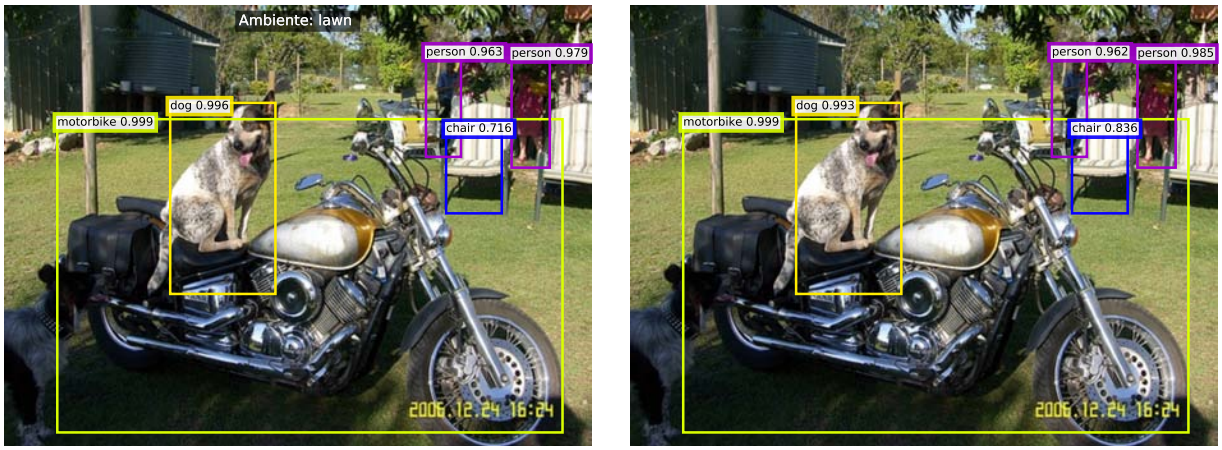
5.2 Análise das imagens

Observando as imagens dos *datasets* PASCAL VOC 2007 e MS COCO, selecionamos 6 exemplos nos casos que a LeanNet e a Faster R-CNN acertam e erram a detecção de objetos. No primeiro caso, mostrado na Figura 5.4, a LeanNet reconhece uma mesa de jantar (*diningtable*) com um escore 23.9% maior que a Faster R-CNN e algumas cadeiras (*chair*) relacionados ao contexto, classificado como sala de jantar (*dining room*). Na Faster R-CNN além de reconhecer a mesa de jantar com um escore baixo, também não reconhece todas as cadeiras ao fundo. Como a nossa abordagem utiliza o contexto da cena, e cadeiras normalmente aparecem em salas de jantar, a nossa abordagem conseguiu reconhecer e assim fazer a detecção.

(a) LeanNet (b) Faster R-CNN

Figura 5.4 – Detecção de objetos entre LeanNet (usando o contexto) e a Faster R-CNN (sem o uso de contexto) no *dataset* PASCAL VOC 2007, com todos objetos em contexto.

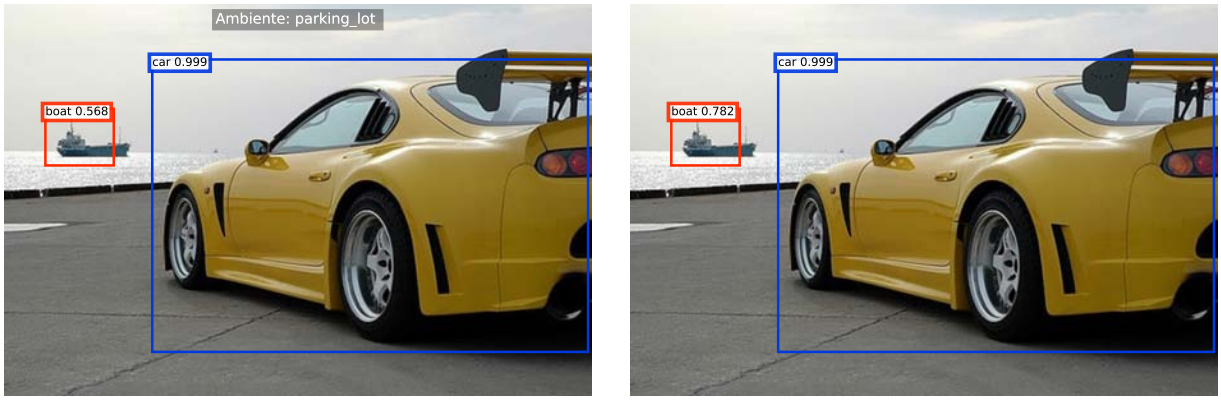
No segundo caso, ilustrado na Figura 5.5, a LeanNet diminui o escore de uma cadeira (*chair*) visto que a cadeira está associada com a categoria de objetos internos (*indoor*). Assim, embora o objeto seja uma cadeira, a nossa abordagem aprendeu que uma cadeira não tem uma forte relação com o ambiente classificado como gramado (*lawn*). Por esse motivo o escore da cadeira foi reduzido em -12% com relação à classificação realizada pela Faster R-CNN.



(a) LeanNet (b) Faster R-CNN

Figura 5.5 – Detecção de objetos entre LeanNet (usando o contexto) e a Faster R-CNN (sem o uso de contexto) no *dataset* PASCAL VOC 2007, com alguns objetos fora de contexto.

No terceiro caso o contexto é ambíguo, podendo ser classificado como porto (*harbor*) ou estacionamento (*parking lot*). Podemos ver na Figura 5.6 que o contexto infere na classificação do barco (*boat*) ao fundo da cena, reduzindo em -21.4% o escore com relação a Faster R-CNN ao classificar o ambiente como estacionamento. A rede aprendeu que um barco é difícil de aparecer em um estacionamento e por esse motivo ela reduziu o escore. Caso o ambiente houvesse sido classificado como porto, existem grandes chances do escore de classificação do barco aumentar. Por outro lado, com o ambiente sendo classificado como porto, o escore do carro seria reduzido.

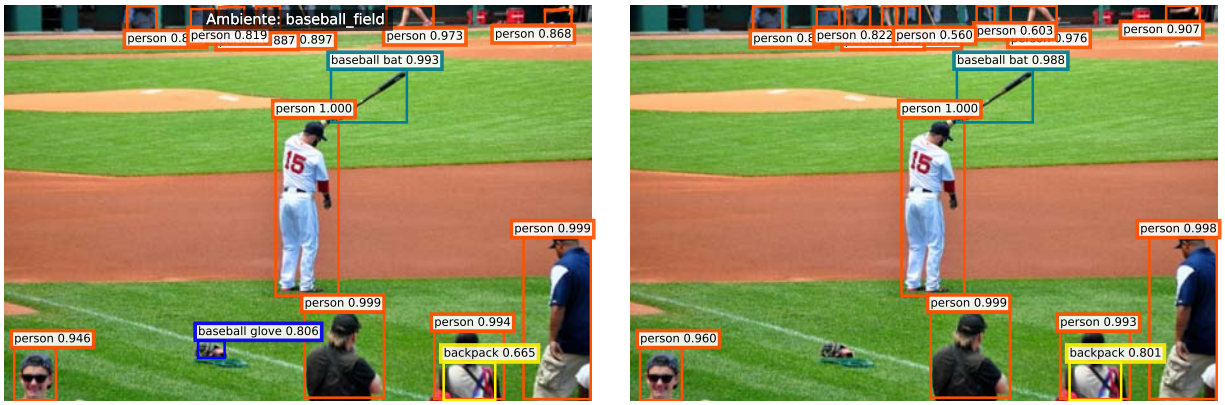


(a) LeanNet (b) Faster R-CNN

Figura 5.6 – Detecção de objetos entre LeanNet (usando o contexto) e a Faster R-CNN (sem o uso de contexto) para o *dataset* PASCAL VOC 2007, quando a classificação do contexto é ambígua.

No quarto caso, mostrado na Figura 5.7 e utilizando o *dataset MS COCO*, a LeanNet conseguiu detectar uma luva de beisebol (*baseboll glove*) com um escore $56,31\%$ maior que a Faster R-CNN para o ambiente classificado como campo de beisebol (*baseball*

field). A LeanNet conseguiu aumentar o escore da luva pois ela aprendeu através do treinamento que a luva de beisebol tem forte relação com o campo de beisebol. Por outro lado, a Faster R-CNN não fez a detecção da luva pois apenas objetos com probabilidade acima de 50% são classificados como objetos.



(a) LeanNet (b) Faster R-CNN

Figura 5.7 – Detecção de objetos entre LeanNet (usando o contexto) e a Faster R-CNN (sem o uso de contexto) no *dataset* MS COCO, com todos objetos em contexto.

O quinto caso, mostrado na Figura 5.8, a nossa abordagem não detectou uma televisão (*tv*) ao fundo da imagem, porque o ambiente foi classificado como cozinha (*kitchen*). O erro de detecção de uma televisão na cozinha se deve ao fato que nossa abordagem aprendeu que a televisão não aparece com muita frequência em cozinhas, assim, o escore da televisão foi reduzido em $-40,66\%$ com relação a Faster R-CNN.

(a) LeanNet (b) Faster R-CNN

Figura 5.8 – Detecção de objetos entre LeanNet (usando o contexto) e a Faster R-CNN (sem o uso de contexto) no *dataset* MS COCO, com alguns dos objetos fora de contexto.

Nesse último exemplo. ilustrado na Figura 5.9, o objeto cobre quase toda a imagem, obtendo assim pouca informação sobre o ambiente. Por esse motivo a rede acabou errando a classificação do contexto e classificando o ambiente como parque de diversões (*amusement park*). Devido ao erro do contexto, o objeto *snowboard* teve seu escore reduzido em -17.4% em relação a Faster R-CNN, pois a LeanNet aprendeu que *snowboard* não se relaciona com parque de diversões. Podemos ver também que existe um *bounding box* sobre uma parte do *snowboard* classificando como pipa (*kite*). Sabemos que essa detecção está errada, mas em um parque de diversões é mais comum aparecer uma pipa do que um *snowboard*, por esse motivo o *bounding box* classificado como pipa aumentou o score em 1.6%.

(a) LeanNet (b) Faster R-CNN

Figura 5.9 – Detecção de objetos entre LeanNet (usando o contexto) e a Faster R-CNN (sem o uso de contexto) no *dataset* MS COCO, com a classificação contexto incorreta.

5.3 Comparações com trabalhos relacionados

Nesta seção comparamos a nossa abordagem com alguns trabalhos recentes que utilizam os mesmos *datasets* desta dissertação. A Tabela 5.3 contém os resultados obtidos pela nossa abordagem e as seguintes abordagens:

Tree Context: Desenvolvida por Choi *et al.* [CTW12], utiliza uma estrutura da dados em forma de árvore para fazer as relações entre os objetos. A abordagem não faz uso de redes neurais para detectar os objetos.

Context-SVM: Chen *et al.* [CSD+15] constroem uma abordagem que usa um classificador de imagens como contexto do detector de objetos e o mesmo ocorre ao contrário. A abordagem utiliza o algoritmo de classificação *support vector machine* (SVM) para fazer a fusão das informações.

DeepID-Net: Ouyang *et al.* [OWZ+15] explora o contexto dos objetos usando uma rede pré-treinada no ImageNet e utiliza o algortimo de classificação SVM para fazer o treinamento do modelo.

Faster R-CNN: Esta rede foi desenvolvida por Ren *et al.* [RHGS15] e é utilizada em parte de nossa arquitetura. Uma descrição da mesma é apresentada no capítulo 2.5.

AC-CNN: Desenvolvida por Li *et al.* [LWL+17], esta abordagem utiliza redes neurais convolucionais para detectar objetos e extrair o contexto. O contexto é extraído usando uma camada *long short-term memory* LSTN e a detecção dos objetos usando múltiplas escalas. A fusão da rede é realzada usando uma camada totalmente conectada.

Deep feature contextual: Chu e Cai [CC17] desenvolveram uma abordagem para extrair o contexto utilizando uma rede neural convolucional. A abordagem modifica a Faster R-CNN adicionando uma *conditional random field* (CRF) depois de gerar as propostas de regiões.

ION: Bell *et al.* [BLZBG16] utilizam redes recorrentes em sua abordagem para extrair características do contexto e explora informações dentro e fora da região de interesse usando múltiplas escalas da região de interesse.

Avaliando nossa abordagem com relação aos trabalhos relacionados mostrados na Tabela 5.3, vemos que conseguimos resultado superior para algumas abordagens, ficando atrás da *Deep feature contextual* [CC17], AC-CNN [LWL+17] e da ION [BLZBG16]. Embora o resultado geral não tenha sido melhor entre todos os trabalhos relacionados, nossa abordagem consegue classificar melhor objetos quando estão em contexto e reduzir o escore quando estão fora de contexto.

Tabela 5.3 – Comparação de mAP da nossa abordagem com os trabalhos relacionados utilizando uma IoU de 50% nos *datasets* MS COCO e PASCAL VOC 2007.

Abordagens	MS COCO	PASCAL VOC 2007
ION [BLZBG16]	55.7%	75.6%
Deep feature contextual [CC17]	-	73.5%
AC-CNN [LWL+17]	-	72.4%
LeanNet	45.4%	70.1%
Faster R-CNN [RHGS15]	41.5%	69.9%
DeepID-Net [OWZ+15]	-	64.1%
Context-SVM [CSD+15]	-	37.7%
Tree Context [CTW12]	-	27.9%

5.4 Teste estatístico

Nesta seção realizamos um teste estatístico para responder a questão “É possível usar uma rede neural pré-treinada que classifica ambientes para extrair o contexto semântico e fazer a fusão com uma rede que detecta objetos, a fim de melhorar a classificação do objeto?”. Para tanto, utilizamos o teste de *Wilcoxon* por se tratar de um teste estatístico não paramétrico que avalia dois modelos. O teste é realizado sobre a diferença entre os scores dos objetos da Faster R-CNN e da LeanNet. Conforme apresentado na Tabela 5.4, obtivemos o valor de P para o *dataset* MS COCO de 9.86×10^{-145} e para o PASCAL VOC 2007 de 2.40×10^{-109}. Neste caso como o valor de P é menor que o nível de significância de 0.05, rejeitamos a hipótese nula e aceitamos a hipótese alternativa, afirmando que a fusão de duas redes neurais, uma para detectar objetos e outra para classificar ambientes, melhora a detecção de objetos.

Tabela 5.4 – Resultado do teste estatístico não paramétrico de *Wilcoxon* usando a diferença dos escores entre a Faster R-CNN e a LeanNet.

Dataset	*Bounding boxes*	Valor de Z	Valor de P
PASCAL VOC 2007	12.032	22.2161	2.40×10^{-109}
MS COCO	291.874	25.6170	9.86×10^{-145}

6. TRABALHOS RELACIONADOS

Nesse capitulo faremos um resumo dos trabalhos que utilizam o contexto para melhorar o reconhecimento de objetos em imagens. Destacando quais tipos de contextos foram utilizados e quais *datasets* usados para realização dos experimentos, mostrando os resultados encontrados.

6.1 Abordagens

Rabinovich *et al.* [RVG+07] propôs uma nova abordagem usando o contexto semântico do objeto (*i.e.*, relação entre os objetos e não entre os objetos e o ambiente), para reduzir a ambiguidade de alguns objetos. A abordagem processa informações do contexto após a detecção, facilitando a utilização com vários tipos de detector de objetos. As características dos objetos foram extraídas e categorizadas pelo *framework bag of features* (BoF) [NJT06], que após extrair as características da imagem, realiza um agrupamento das características similares, separando em categorias. A abordagem utiliza o *framework conditional random fiel* (CRF), para aprender sobre o contexto através do treinamento verificando a frequência que os objetos se relacionam. O método melhora a detecção do objeto de acordo com a relevância do contexto. Os experimentos foram realizados usando o PASCAL VOC [EVGW+10] e atingiu 74.2% e MSRC atingindo 68.4% de acurácia.

Galleguillos *et al.* [GRBD08] apresentam uma nova abordagem que utiliza o contexto semântico das relações entre os objetos e o contexto espacial analisando a localização e aparência dos objetos. Os dois tipos de contextos são aprendidos simultaneamente utilizando os dados de treino. As características dos objetos foram extraídas usando o *framework bag of features* (BoF), e para fazer a inferência do contexto sobre os objetos foi usado o *conditional random filed* (CRF). O contexto espacial foi descrito pela relação dos pares de objetos e divido em 4 categorias: *above*, *below*, *inside* e *around*. O modelo foi avaliado usando os *datasets* PASCAL VOC 2007 atingindo 36.7% e o MSRC alcançando 68.38% de acurácia.

Galleguillos *et al.* [GMB+10] desenvolveram uma nova abordagem para a localização de objetos explorando 3 tipos de contextos: informações do fundo da imagem, posição da região de um objeto em relação ao outro, e as relações entre os objetos. As características dos objetos foram extraídas da segmentação dos objetos rotulados usando 4 algoritmos: SIFT [Low04], SSIM [SI07], LAB *histogram* e PHOG. A distância entre os objetos foi calculada usando a distância de *Mahalanobis* e o algoritmo *Multiple Kernel LMNN* (MKLMNN). O treinamento foi realizado usando o algoritmo SVM para prever quando dois segmentos pertencem ao mesmo objeto. O saída do SVM é enviada para o *conditional*

random filed (CRF) para fazer classificação usando do contexto. Para avaliar o modelo foi usado os *datasets* PASCAL VOC 2007 e o MSRC atingindo 36% e 70% de acurácia.

Choi *et al.* [CTW12] desenvolveram uma abordagem que cria uma estrutura de árvore com relações entre os objetos (*Tree Context*), afim de explorar a informação contextual entre os objetos para melhorar a classificação. A arvore é criada a partir da escolha manual do nodo raiz, sendo esse nodo um objeto que aparece com bastante frequência em imagens (*e.g.*, céu). Os objetos com forte relação entre si, são interligados com uma aresta com um valor positivo, enquanto os objetos que não possuem relação entre si, são interligados com valores negativos. Quanto maior a relação entre os objetos, maior é o valor do peso da ligação. A abordagem funciona com vários tipos de detectores de objetos, pois o processo para utilizar informações do contexto é realizado depois da detecção. A abordagem funciona melhor para datasets com muitas categorias de objetos. Os experimentos realizados com PASCAL VOC 2007 atingiram 27.90% de mAP e com o SUN09 [JLTW10] alcançaram 26.08% de mAP.

Chen *et al.* [CSD+15] propôs uma nova abordagem chamada *Contextualized Support Vector Machine* (Context-SVM) que utiliza a saída de um classificador de objetos como contexto do detector de objetos, e o mesmo ocorre ao contrário, a saída do detector como contexto do classificador. O método não se baseia na frequência que um objeto se relaciona com outro, ao invés disso, envia para o *Context-SVM* o objeto com maior escore dado pelo classificador junto com os objetos do detector, e vise-versa, para assim realizar o treinamento do modelo. Os experimentos realizados usando o PASCAL VOC 2007 e 2010 atingiram 37.7% e 74.5% de mAP na detecção e 70.5% e 74.5% na classificação.

Ouyang *et al.* [OWZ+15] construíram uma nova abordagem que captura deformações do objeto, (*e.g.*, forma aredondada de um objeto), e informações do contexto usando a classificação da imagem inteira entre as 1000 classes do *dataset* ImageNet. A abordagem concatena as 1000 classes do vetor de características com os escores dos objetos usando o algoritmo de classificação *support vector machine* (SVM). Depois de treinado o SVM, os valores das 1000 classes são usados para fazer inferência em cada classe de objeto. Os experimentos foram realizados usando o ILSVRC2014 e o PASCAL VOC 2007 obtendo 50.7% e 64.1% de mAP.

Bell *et al.* [BLZBG16] exploram uma ideia contextual e multi-escalar, apresentando uma nova abordagem denominada Inside-Outside Net (ION), que é capaz de detectar objetos explorando informações internas e externas da região de interesse. Bell et al. usa um detector de propostas de objetos e pooling dinâmico para avaliar a diferença entre as regiões de interesse (RoI) candidatas em uma imagem. Uma rede neural recorrente espacial computa os recursos a partir da informação contextual fora da região de interesse. O objeto e as informações contextuais são transmitidos e passam por várias camadas totalmente conectadas para classificação. Os experimentos usando o PASCAL VOC 2007 e 2012 tiveram

75.6% e 77.9% de mAP e usando o MS COCO [LMB+14] dataset alcançam 55.7% de mAP com IoU de 50%.

Liu *et al.* [LGMZ16] propôs uma abordagem para resolver um problema de classificação usando uma CNN contextualizada de dois canais. Sua abordagem usa uma rede, denominada *content net*, para capturar características dos objetos e uma rede, denominada *context net*, para capturar características do fundo da imagem. Enquanto a *content net* é fornecida com imagens extraídas dos *bounding boxes* de objetos, a rede *context net* é alimentada com toda a imagem, tendo a região do *bounding box* preenchida com pixels da posição equivalente da média da imagem calculada através do conjunto de treinamento. As redes são unidas em uma camada de fusão que aprende automaticamente os valores da característica do objeto e do contexto e exibe a classificação final. Os experimentos foram realizados usando três datasets públicos: *Flower102*[NZ08], *CUB2010*[WBM+10] e *CUB2011*[WBW+11], e alcançaram 94.5%, 41.8% e 76.9% de acurácia respectivamente.

Li *et al.* [LWL+17] desenvolveram uma nova abordagem chamada *attention to context CNN* (AC-CNN), baseada no modelo de detecção de objetos. Sua abordagem encorpora informações contextuais globais e locais em uma CNN usando duas sub-redes: *global contextualized* (AGC) e *multi-scale local contextualized* (MLC). A sub-rede AGC é responsável por destacar locais contextuais globais úteis através das múltiplas camadas de *long short-term memory* (LSTM) [HS97]. Enquanto a (MLC) captura informação contextual interna e externa. O contexto global e local são unidos por camadas totalmente conectadas antes da classificação. Os experimentos foram realizados usando o PASCAL VOC 2007 e atingiram 72.4% of mAP e no PASCAL VOC 2012 alcançaram 70.6% of mAP.

Chu e Cai *et al.* [CC17] criaram uma nova abordagem que usa informações do contexto, extraindo características das relações entre os objetos e de toda a imagem através de uma rede neural convolucional. Utilizando a Faster R-CNN [RHGS15], adicionaram uma *fully connected conditional random field* (CRF) depois da rede gerar as propostas de regiões (do inglês *region proposals*). A CRF usa as propostas de regiões de acordo com a informação do contexto, e realiza a inferência. Os experimentos usando o PASCAL VOC 2007 alcançaram 73.5% de mAP.

6.2 Análise

As abordagens apresentadas mostram o quanto o uso do contexto para o reconhecimento de objetos é pouco explorado, e que a utilização dessa informação contribui para melhorar o reconhecimento dos objetos. A maioria dos trabalhos fazem uso do contexto semântico das relações entre os objetos e não com o ambiente, e quando os autores se referem ao contexto da cena, eles não estão se referindo as informações do fundo da imagem, e sim de todos os elementos que estão na imagem (*i.e.*, objetos e ambiente). Liu

et al. [LGMZ16] é o único trabalho apresentado que explora o fundo da imagem, retirando os objetos da cena. A abordagem porém aprende a classificar o ambiente com o nome de um dos objetos, assim como as demais abordagem que exploram a característica de toda a imagem. Diferente dos trabalhos apresentados, nossa abordagem utiliza uma rede neural já treinada para explorar as características do fundo da imagem para capturar informações do ambiente, e usamos essa informação para melhorar o reconhecimento dos objetos que dependem do contexto.

7. CONCLUSÃO

Nesse trabalho, desenvolvemos uma nova arquitetura chamada LeanNet para detecção de objetos baseada em redes neurais convolucionais (Faster R-CNN [RHGS15] e a Places365-CNN [ZLK+17]). A arquitetura inclui uma CNN com foco em detectar objetos e uma outra CNN para classificar o ambiente. Modelos pré-treinados das redes foram utilizados para melhorar a qualidade da detecção e reduzir o tempo de treinamento. Concatenamos as características dos objetos com as características do contexto e predizemos a classe de cada objeto.

Realizamos experimentos usando os *datasets* PASCAL VOC 2007 [EVGW+10] e MS COCO [LMB+14]. Os resultados dos experimentos mostram que nossa abordagem de reconhecimento melhora o desempenho geral quando comparado com a rede Faster R-CNN. Nossa abordagem funciona muito bem quando os objetos estão situado no seu devido contexto. A análise dos resultados da nossa arquitetura demonstra que a probabilidade de um objeto tende a aumentar quando o contexto da cena está relacionado com o objeto, e reduzir quando o objeto está fora do contexto. Os resultados mostram também que o uso de uma rede neural pré treinada para classificar ambientes para extrair o contexto semântico da cena, melhorou a classificação dos objetos.

Foi publicada uma versão inicial desse trabalho na conferência KDMILE 2017 [dSGMR17], no qual treinamos a LeanNet liberando mais camadas para o treinamento, porém como as camadas liberadas eram compartilhadas com a camada que fazia a predição dos *bounding boxes*, o contexto acabou inferindo na predição, resultando em uma redução do IoU dos *bounding boxes*. Apesar dessa versão inicial não superar a Faster R-CNN, foi observado nos objetos que estavam fora do contexto a redução do escore da classificação, e quando estavam no contexto apropriado foi observado um aumento. A principal diferença entre a versão publicada e a desta dissertação é que neste trabalho o treinamento ocorre somente na camada de classificação do objeto, não tendo o compartilhamento dos pesos com a camada de predição dos *bounding boxes*. O artigo foi selecionado como um dos 13 melhores e convidado para enviar uma versão estendida com as modificações e resultados atuais para o *Journal of Information and Data Management* (JIDM).

7.1 Contribuições

Propusemos uma arquitetura de uma rede neural profunda que combina resultados prévios de redes que fazem detecção de objetos e classificação de ambientes. A arquitetura proposta melhora da classificação de objetos no qual o contexto é relevante, como por exemplo um hidrante que sempre vai aparecer na calçada.

7.2 Limitações

Nossa abordagem tende a reconhecer melhor os objetos que não preenchem toda a imagem, pois assim facilita o reconhecimento do ambiente. Porém quando o objeto preenche toda a imagem, a rede tem maior dificuldade de fazer o reconhecimento do contexto, resultando em uma classificação errada do ambiente. Isso pode inferir nos objetos que dependem do contexto, reduzindo os escores da classificação. Nossa abordagem portanto depende da classificação correta do ambiente para auxiliar no reconhecimento dos objetos que dependem do contexto.

7.3 Trabalhos futuros

Para trabalhos futuros pretendemos criar um *dataset* com rótulos dos objetos e dos ambientes, porque dessa forma poderíamos avaliar também a acurácia da classificação do ambiente enquanto avalia a detecção. Pretendemos também explorar o uso do contexto de escala, usando a regressão para predizer o tamanho dos objetos e assim fazer a relação entre os tamanhos dos objetos. Uma estrutura de dados em forma de árvore pode ser usada para fazer as relações entre os tamanhos dos objetos, assim como a abordagem desenvolvida por Choi *et al.* [CTW12] que utiliza uma árvore para relacionar os objetos e o ambiente.

REFERÊNCIAS BIBLIOGRÁFICAS

[Bis06] Bishop, C. “Pattern recognition and machine learning”. Springer-Verlag, 2006, vol. 20, 738p.

[BLZBG16] Bell, S.; Lawrence Zitnick, C.; Bala, K.; Girshick, R. “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2874–2883.

[BMR82] Biederman, I.; Mezzanotte, R. J.; Rabinowitz, J. C. “Scene perception: Detecting and judging objects undergoing relational violations”, *Cognitive Psychology*, vol. 14–2, Abril 1982, pp. 143–177.

[CC17] Chu, W.; Cai, D. “Deep feature based contextual model for object detection”, *Neurocomputing*, vol. 275–31, Janeiro 2017, pp. 1035–1042.

[CKZ+16] Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. “Monocular 3d object detection for autonomous driving”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2147–2156.

[CSD+15] Chen, Q.; Song, Z.; Dong, J.; Huang, Z.; Hua, Y.; Yan, S. “Contextualizing object detection and classification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37–1, Janeiro 2015, pp. 13–27.

[CTW12] Choi, M. J.; Torralba, A.; Willsky, A. S. “A tree-based context model for object recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34–2, Fevereiro 2012, pp. 240–252.

[DDS+09] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. “Imagenet: A large-scale hierarchical image database”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[dSGMR17] da Silva, L. P.; Granada, R.; Monteiro, J.; Ruiz, D. D. “Using scene context to improve object recognition”. In: Proceedings of the 5th Symposium on Knowledge Discovery, Mining and Learning, 2017, pp. 105–112.

[EVGW+10] Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; Zisserman, A. “The pascal visual object classes (voc) challenge”, *International Journal of Computer Vision*, vol. 88–2, Junho 2010, pp. 303–338.

[FFFP04] Fei-Fei, L.; Fergus, R.; Perona, P. “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object

categories”. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, 2004, pp. 178–178.

[FLGC11] Faceli, K.; Lorena, A. C.; Gama, J.; Carvalho, A. “Inteligência Artificial: Uma abordagem de aprendizado de máquina”. Livros Técnicos e Ciêntificos Editora Ltda, 2011, vol. 1, 375p.

[GB10] Galleguillos, C.; Belongie, S. “Context based object categorization: A critical survey”, *Computer Vision and Image Understanding*, vol. 114–6, Junho 2010, pp. 712–722.

[GBC16] Goodfellow, I.; Bengio, Y.; Courville, A. “Deep Learning”. MIT Press, 2016, vol. 1, 775p.

[GHP07] Griffin, G.; Holub, A.; Perona, P. “Caltech-256 object category dataset”, Relatório Técnico CNS-TR-2007-001, California Institute of Technology, 2007, 20p.

[Gir15] Girshick, R. “Fast r-cnn”. In: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[GMB+10] Galleguillos, C.; Mcfee, B.; Belongie, S.; Lanckriet, G.; Science, C.; Diego, S. “Multi-Class Object Localization by Combining Local Contextual Interactions”. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 113–120.

[GRBD08] Galleguillos, C.; Rabinovich, A.; Belongie, S.; Diego, S. “Object Categorization using Co-Occurrence, Location and Appearance”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[HS97] Hochreiter, S.; Schmidhuber, J. “Long short-term memory”, *Neural Computation*, vol. 9–8, Novembro 1997, pp. 1735–1780.

[HZRS16] 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, 2016, pp. 770–778.

[JLTW10] Jin, C. M.; Lim, J. J.; Torralba, A.; Willsky, A. S. “Exploiting hierarchical context on a large database of object categories”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 129–136.

[KSH12] 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, 2012, pp. 1097–1105.

[KTS+14] Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. “Large-scale video classification with convolutional neural networks”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[LAE+16] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. “Ssd: Single shot multibox detector”. In: Proceedings of the 14th European Conference on Computer Vision, 2016, pp. 21–37.

[LBH15] LeCun, Y.; Bengio, Y.; Hinton, G. “Deep learning”, *Nature*, vol. 521–7553, Maio 2015, pp. 436–444.

[LFF07] Li, L.-J.; Fei-Fei, L. “What, where and who? classifying events by scene and object recognition”. In: Proceedings of the IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.

[LGMZ16] Liu, J.; Gao, C.; Meng, D.; Zuo, W. “Two-stream contextualized cnn for fine-grained image classification”. In: Proceedings of the 13th AAAI Conference on Artificial Intelligence, 2016, pp. 4232–4233.

[LH17] Li, Z.; Hoiem, D. “Learning without forgetting”, *Transactions on Pattern Analysis and Machine Intelligence*, vol. PP–99, Novembro 2017, pp. 1–13.

[LMB+14] 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: Proceedings of the 13th European Conference on Computer Vision, 2014, pp. 740–755.

[Low04] Lowe, D. G. “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, vol. 60–2, Novembro 2004, pp. 91–110.

[LSP06] Lazebnik, S.; Schmid, C.; Ponce, J. “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories”. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.

[LWL+17] Li, J.; Wei, Y.; Liang, X.; Dong, J.; Xu, T.; Feng, J.; Yan, S. “Attentive contexts for object detection”, *IEEE Transactions on Multimedia*, vol. 19–5, Maio 2017, pp. 944–954.

[NJT06] Nowak, E.; Jurie, F.; Triggs, B. “Sampling strategies for bag-of-features image classification”. In: Proceedings of the 9th European Conference on Computer Vision, 2006, pp. 490–503.

[NZ08] Nilsback, M.-E.; Zisserman, A. “Automated flower classification over a large number of classes”. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, 2008, pp. 722–729.

[OT07] Oliva, A.; Torralba, A. “The role of context in object recognition”, *Trends in Cognitive Sciences*, vol. 11–12, Dezembro 2007, pp. 520–527.

[OWZ+15] Ouyang, W.; Wang, X.; Zeng, X.; Qiu, S.; Luo, P.; Tian, Y.; Li, H.; Yang, S.; Wang, Z.; Loy, C.-c.; Tang, X. “DeepID-Net : Deformable Deep Convolutional Neural Networks for Object Detection”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2403–2412.

[PH12] Patterson, G.; Hays, J. “Sun attribute database: Discovering, annotating, and recognizing scene attributes”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2751–2758.

[PY10] Pan, S. J.; Yang, Q. “A survey on transfer learning”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22–10, Outubro 2010, pp. 1345–1359.

[QT09] Quattoni, A.; Torralba, A. “Recognizing indoor scenes”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 413–420.

[RDGF16] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. “You only look once: Unified, real-time object detection”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[RF17] Redmon, J.; Farhadi, A. “Yolo9000: Better, faster, stronger”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6517–6525.

[RHGS15] Ren, S.; He, K.; Girshick, R.; Sun, J. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015, pp. 91–99.

[RHW86] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. “Learning representations by back-propagating errors”, *Nature*, vol. 323–6088, Outubro 1986, pp. 533–536.

[RVG+07] Rabinovich, A.; Vedaldi, A.; Galleguillos, C.; Wiewiora, E.; Belongie, S. “Objects in context”. In: Proceedings of the IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.

[SHK+14] Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. “Dropout: A simple way to prevent neural networks from overfitting”, *Journal of Machine Learning Research*, vol. 15, Junho 2014, pp. 1929–1958.

[SI07] Shechtman, E.; Irani, M. “Matching local self-similarities across images and videos”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[SLJ+15] 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, 2015, pp. 1–9.

[SZ14] Simonyan, K.; Zisserman, A. “Two-stream convolutional networks for action recognition in videos”. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, pp. 568–576.

[SZ15] Simonyan, K.; Zisserman, A. “Very deep convolutional networks for large-scale image recognition”. In: Proceedings of the 3rd International Conference on Learning Representations, 2015, pp. 1–13.

[Uni17] University, S. “CS231n Convolutional Neural Networks for Visual Recognition”. Capturado em: http://cs231n.github.io/, Janeiro 2018.

[WBM+10] Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. “Caltech-UCSD Birds 200”, Relatório Técnico CNS-TR-2010-001, California Institute of Technology, 2010, 15p.

[WBW+11] Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. “The Caltech-UCSD Birds-200-2011 Dataset”, Relatório Técnico CNS-TR-2011-001, California Institute of Technology, 2011, 8p.

[XHE+10] Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; Torralba, A. “Sun database: Large-scale scene recognition from abbey to zoo”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.

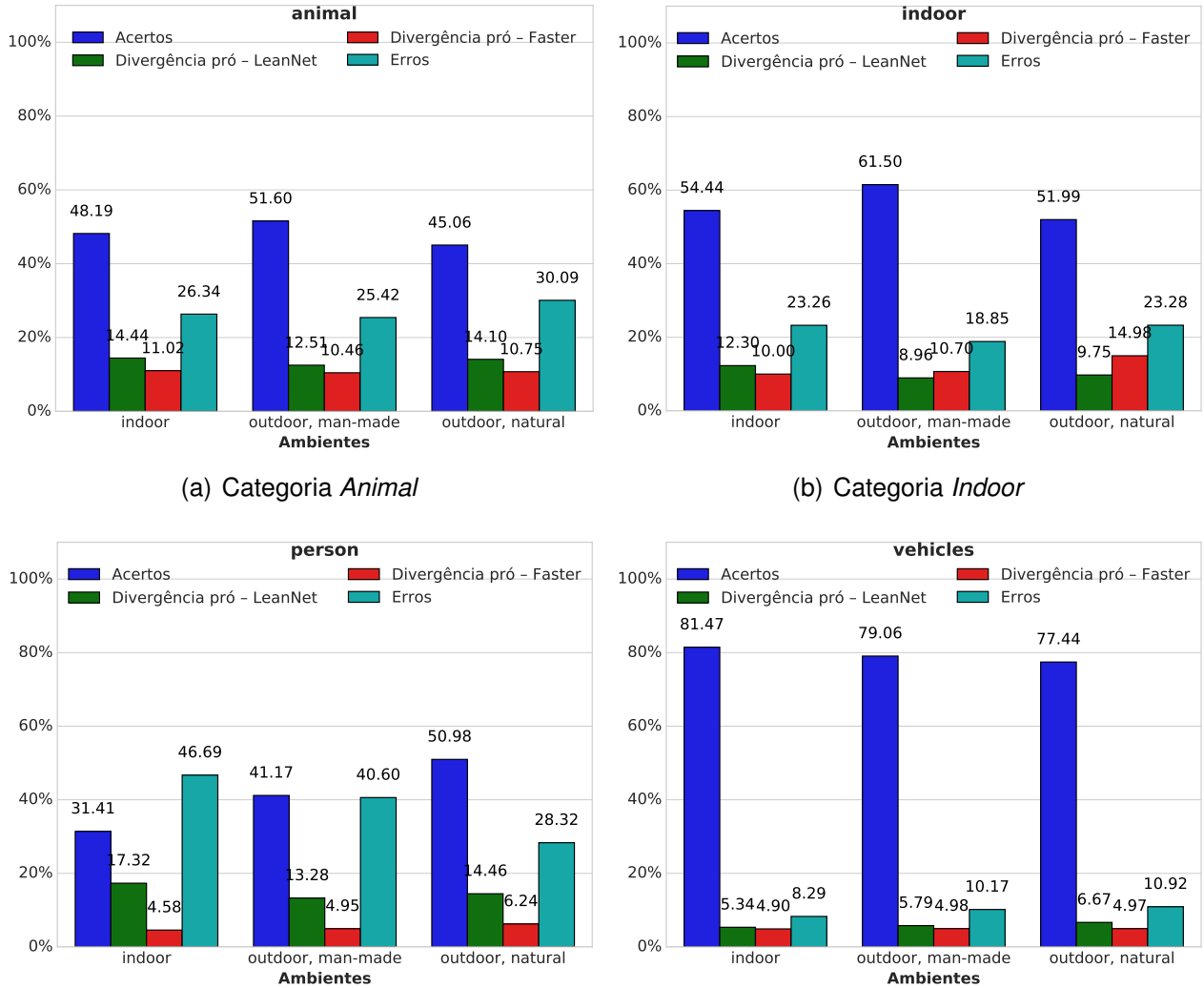
[YJK+11] Yao, B.; Jiang, X.; Khosla, A.; Lin, A. L.; Guibas, L.; Fei-Fei, L. “Human action recognition by learning bases of action attributes and parts”. In: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 1331–1338.

[YTA14] Yi, C.; Tian, Y.; Arditi, A. “Portable camera-based assistive text and product label reading from hand-held objects for blind persons”, *IEEE/ASME Transactions on Mechatronics*, vol. 19–3, Junho 2014, pp. 808–817.

[ZF14] Zeiler, M. D.; Fergus, R. “Visualizing and understanding convolutional networks”. In: Proceedings of the 13th European Conference on Computer Vision, 2014, pp. 818–833.

[ZLK+17] Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. “Places: A 10 million image database for scene recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP–99, Julho 2017, pp. 1–14.

APÊNDICE A – GRÁFICOS DAS CATEGORIAS DE OBJETOS E AMBIENTES



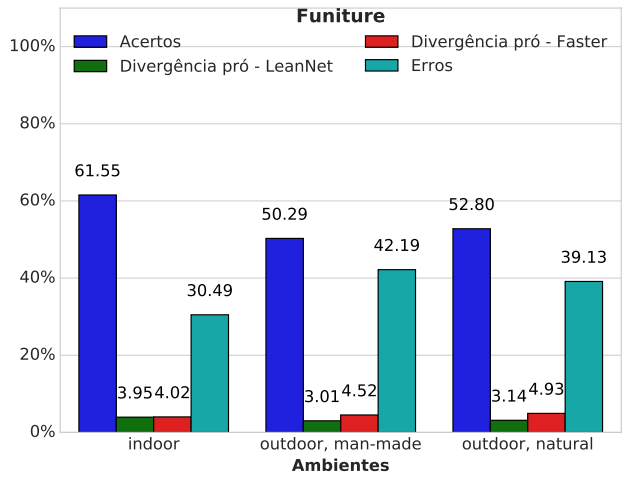
(a) Categoria *Animal*

(b) Categoria *Indoor*

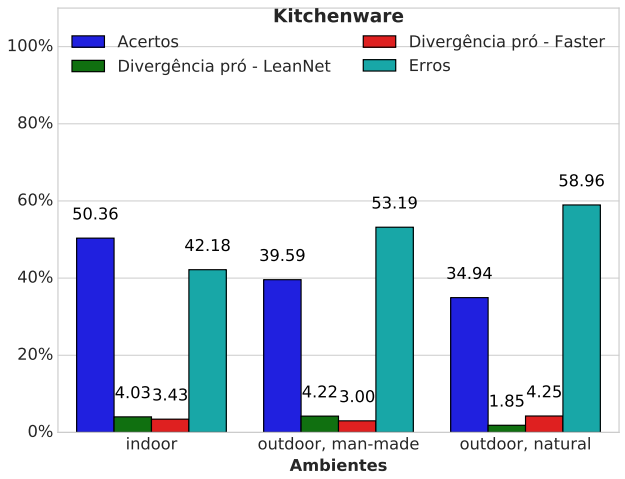
(c) Categoria *Person*

(d) Categoria *Vehicles*

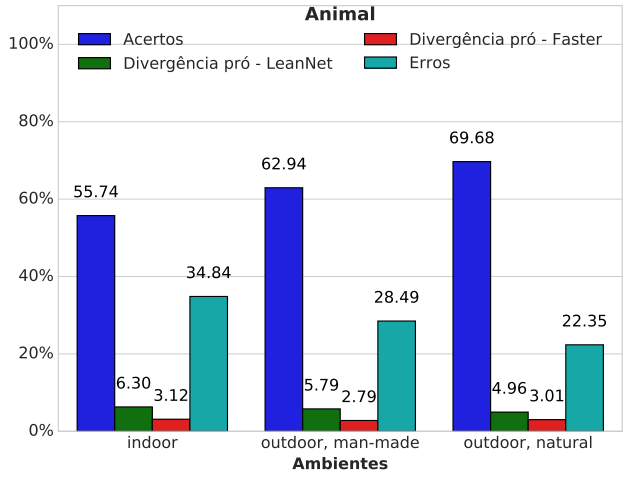
Figura A.1 – Quantidade de acerto, erros e diferenças entre a Faster R-CNN e a LeanNet, agrupado por categorias de objetos do PASCAL VOC 2007 e categorias de ambientes da Places365.



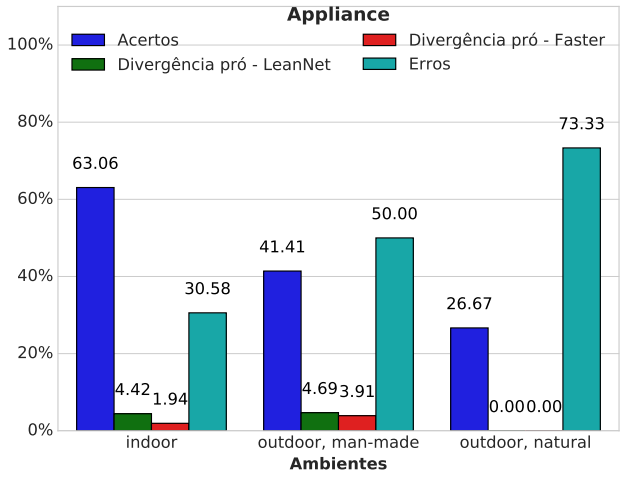
(a) Categoria *Funiture*



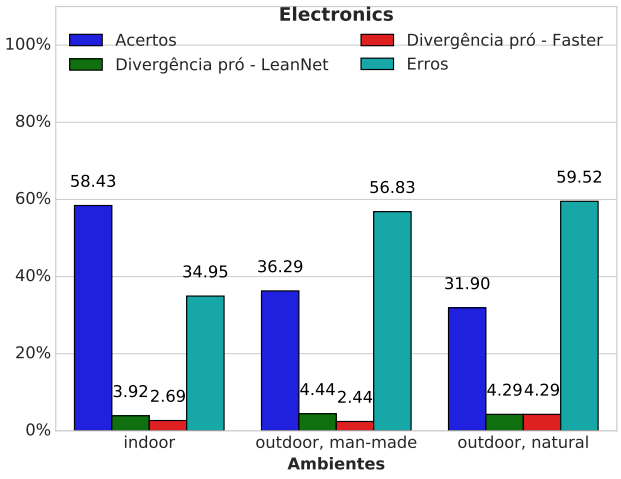
(b) Categoria *kitchenware*



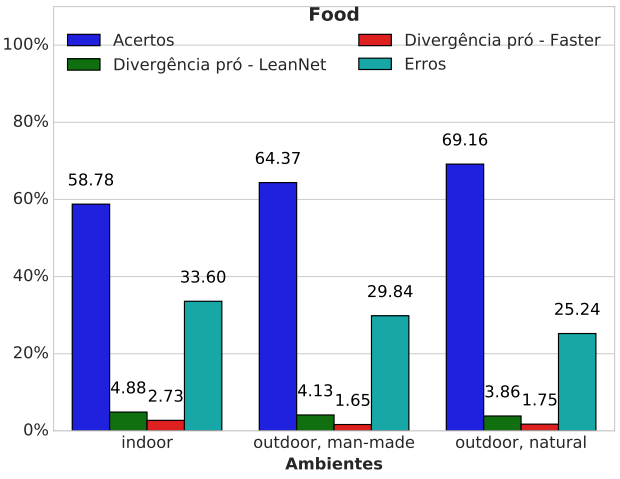
(c) Categoria *Animal*



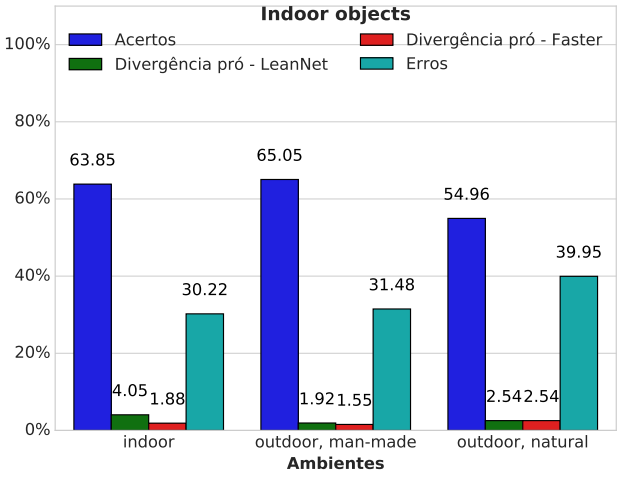
(d) Categoria *Applience*



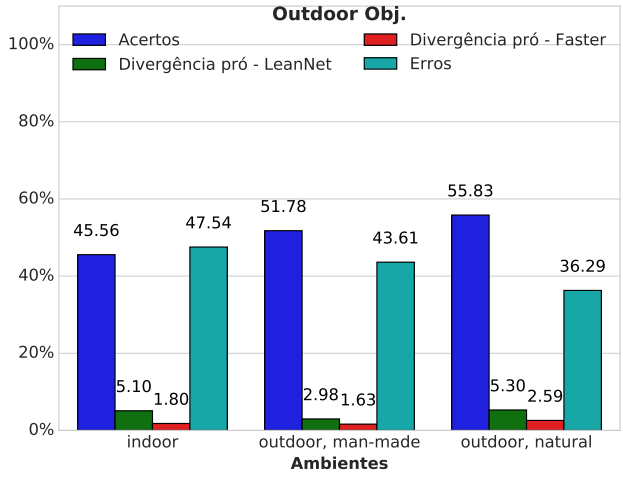
(e) Categoria *Electronics*



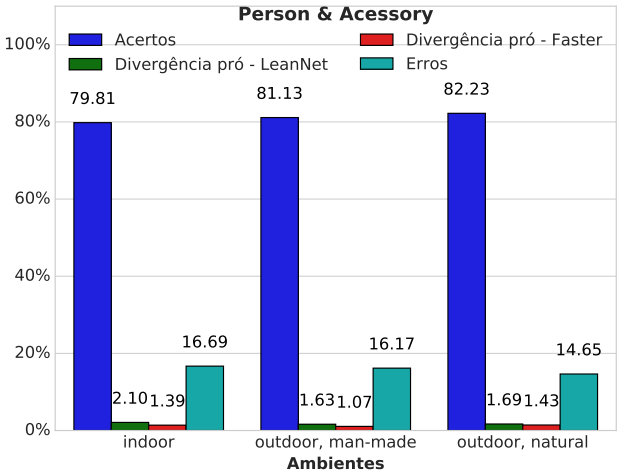
(f) Categoria *Food*



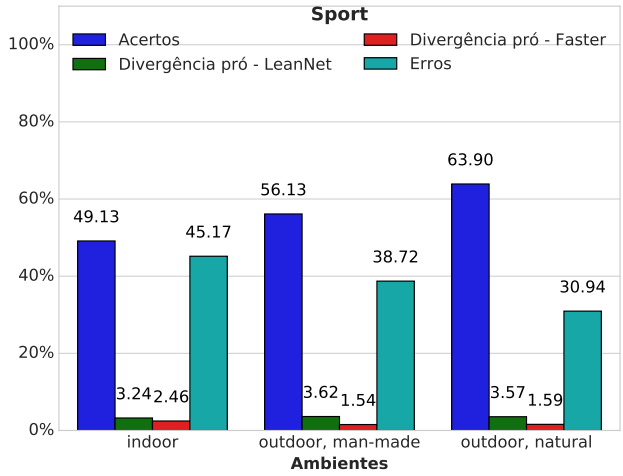
(g) Categoria *Inddor Objetcts*



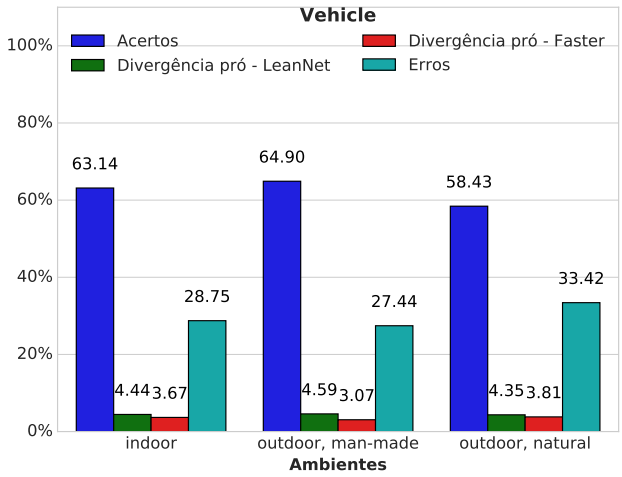
(h) Categoria *Outdoor Objects*



(i) Categoria *Person & Acessory*



(j) Categoria *Sport*



(k) Categoria *Vehicle*

Figura A.2 – Quantidade de acerto, erros e diferenças entre a Faster R-CNN e a LeanNet, agrupado por categorias de objetos do MS COCO e categorias de ambientes da Places365.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br