PUCRS

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

GABRIEL DA SILVA SIMÕES

ABORDAGENS NEURAIS PARA CONTROLE DE CONTEÚDO PORNOGRÁFICO

Porto Alegre
2019

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

Abordagens Neurais para Controle de Conteúdo Pornográfico

Gabriel da Silva Simões

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

Orientador: Prof. Rodrigo Coelho Barros

Porto Alegre
2019

Ficha Catalográfica

S614a Simões, Gabriel da Silva

Abordagens neurais para controle de conteúdo pornográfico / Gabriel da Silva Simões . – 2019.

138 p.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rodrigo Coelho Barros.

1. Redes neurais. 2. Redes convolucionais. 3. Classificação. 4. Detecção. 5. Geração automática. I. Barros, Rodrigo Coelho. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Gabriel da Silva Simões

Abordagens Neurais para Controle de Conteúdo Pornográfico

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

Aprovado em 08 de novembro de 2019.

BANCA EXAMINADORA:

Prof. Dr. Eduardo Nunes Borges (PPGC/FURG)

Prof. Dr. Gustavo Pessin (PPGCC/UFOP)

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

Prof. Dr. Rodrigo Coelho Barros (PPGCC/PUCRS - Orientador)

DEDICATÓRIA

Inicialmente, dedico este Trabalho a todos aqueles que, de alguma forma, contribuíram comigo no sentido amplo da palavra. Nestes anos questionei-me, cresci intelectualmente, enfrentei desafios e reconstruí coisas na minha vida que, acredito, me fazem hoje uma pessoa melhor. Minha família tem papel chave nestas descobertas e reconstruções, por isso lhes dedico não só o Trabalho, mas tudo aquilo que pudemos viver e ser neste período. Obrigado!

"Therefore the problem is not so much, to see what nobody has yet seen, but rather to think concerning that which everybody sees, what nobody has yet thought."
(Schopenhauer)

AGRADECIMENTOS

Agradeço ao Professor Rodrigo Barros pela orientação, pela cobrança, pelas agruras transpostas, mas especialmente pelo apoio e pela amizade que se construiu no curso do Trabalho. Agradeço também ao Professor Duncan Ruiz, especialmente por me lembrar sempre do foco. Ao GPIN, nosso Grupo de Pesquisa, que não vou enumerar para não ser injusto. Agradeço ao Grupo como um todo, cada um deu sua parcela de apoio, ao seu jeito, e sou grato por isso.

Novamente agradeço minha família pelo apoio e pela paciência. Sei que estive distante e, muitas vezes, ausente neste período. Inclui-se nesta ausência os momentos em que estive por perto mas mentalmente distante, mergulhado nas metas, nos prazos, nos resultados, nas dificuldades. Obrigado por terem me aguentado, e por terem me ajudado a lembrar, pelo menos um pouco, de que mesmo doutorandos precisam viver e estarem felizes. Amo vocês!

Finalmente, agradeço a Motorola Mobility pelo apoio financeiro que custeou meus estudos e toda essa pesquisa. Não haveria Trabalho sem este apoio. Muito obrigado!

ABORDAGENS NEURAIS PARA CONTROLE DE CONTEÚDO PORNOGRÁFICO

RESUMO

O crescente volume de conteúdo adulto disponível na internet gera problemas de saúde e desordens comportamentais. O consumo de pornografia é favorecido pela facilidade de acesso, pelo baixo custo e pela anonimidade dos internautas. Quebrando um destes fatores, pode-se minimizar o consumo deste tipo de conteúdo, por outro lado, dado o volume, é necessário analisar o conteúdo automaticamente. Neste sentido, *Deep Learning* permite realizar tarefas complexas automaticamente. Esta tese ataca a facilidade de acesso à pornografia aplicando censuras automáticas através de 3 abordagens de *Deep Learning*: classificação, detecção de objetos e geração automática. Na abordagem de classificação, foram treinados e avaliados 8 modelos preditivos com diferentes arquiteturas de redes neurais, onde os resultados preditivos atingiram acurácias superiores a 99%, processando até 40 FPS. Observou-se que as regiões mais significativas para classificação de pornografia estão relacionadas especificamente às partes íntimas do corpo. A segunda abordagem censurou pornografia utilizando métodos de detecção de objetos. Foi construído um *dataset* para detecção de partes íntimas que permitiu o treinamento de modelos que atingiram resultados preditivos com mAP $= 0,6961$, censurando partes íntimas de corpo. Foi construída uma rede neural para detecção, chamada *CensorNet*, gerando resultados preditivos promissores. Foi construída também *CensorPlus*, uma rede composta por uma segunda saída para classificação, criando um método híbrido para detecção de objetos e classificação de imagem. Finalmente, a terceira abordagem desta tese apresenta *AttGAN*, um método baseado em tradução imagem-para-imagem que utiliza redes neurais para gerar censuras automáticas. O método utiliza máscaras de atenção geradas por *AttNET*, uma rede neural treinada para classificação, convertida para a geração de tais máscaras. Foram desenvolvidas 3 variações de *AttGAN*, comparadas por meio de uma avaliação *online* onde 21 participantes compararam os resultados. Os resultados evidenciaram vantagem para o método *AttGAN+*, escolhido como melhor método em 1.050 opiniões coletadas. O método *AttGAN+* foi incrementado, aplicando a mescla da imagem de entrada com a saída censurada, dando origem ao método *AttGAN++*, resultando em uma imagem censurada que preserva características periféricas da imagem original.

Palavras-Chave: redes neurais, redes convolucionais, classificação, detecção, geração automática, censura, pornografia.

NEURAL APPROACHES FOR PORNOGRAPHIC CONTENT CONTROL

ABSTRACT

The adult content available on the internet generates health problems and behavioral disorders. The consumption of pornography is favored by the ease of access, low cost and anonymity of Internet users. Breaking at last one of these factors can minimize the consumption of this content, however, given the volume, it is necessary to analyze the content automatically. In this sense, Deep Learning can perform complex tasks automatically. This thesis attacks the ease of access to pornography by applying automatic censorship through 3 Deep Learning approaches: classification, object detection and automatic generation. In the classification approach, 8 predictive models of different neural network architectures were trained and evaluated, where the predictive results reached accuracy above 99%, processing up to 40 FPS. It was observed that the most significant regions for pornography classification are related to the intimate body parts. The second approach censored pornography with object detection methods. An intimate body parts detection dataset was constructed which allowed the training of models for censoring intimate body parts that achieved mAP $= 0.6961$. A neural network for detection, called *CensorNet*, was built, generating promising predictive results. We build *CensorPlus*, a network composed by a second output for classification. This network creates a hybrid method for object detection and image classification. Finally, the third approach to this thesis presents *AttGAN*, a method based on image-to-image translation that uses neural networks to generate automatic censorship. The method utilizes attention masks generated by *AttNET*, a classification-trained neural network converted to generate such masks. Three *AttGAN* variations were developed, and we designed an online survey where 21 participants compared the results. The results indicate an advantage to the *AttGAN+* method, pointed as the best method in $1,050$ opinions collected. The *AttGAN+* method was incremented by merging the input image with the censored output, giving rise to the *AttGAN++* method, resulting in a censored image that preserves peripheral characteristics of the original image.

Keywords: neural networks, convolutional neural networks, classification, detection, pornographi, censorship.

LISTA DE FIGURAS

LISTA DE TABELAS

NOTAÇÕES

Funções

$boxDist$ – Função para calcular a distância entre um box e um centróide.

Inv – Inverte uma máscara de atenção.

IoU – Calcula *Intersection over Union* entre 2 objetos.

$Pinterp$ – Precisão Interpolada.

Termos

A – Domínio A

A_c – Acurácia

A_F – Altura do Filtro

A_I – Altura da Imagem

A_N – Rede de Atenção

AP – *Average Precision*

A_S – Altura da Saída

A_V – Altura do Volume

B – Domínio B

c – classe

C – Conjunto de Classes

$ctrd$ – Centroide

D – Discriminador

D_E – Dimensões de Entrada

D_F – Dimensões do Filtro

D_S – Dimensões de Saída

F_B – Imagem falsa do domínio B

F_N – Falsos Negativos

F_P – Falsos Positivos

G – Gerador

G_{AB} – Gerador que traduz do domínio A para o domínio B.

G_{BA} – Gerador que traduz do domínio B para o domínio A.

I – Imagem

L_C – Limiar de Confiança

L_F – Largura do Filtro

L_I – Largura da Imagem

L_{IoU} – Limiar de IoU

L_S – Largura da Saída

L_V – Largura do Volume

M_A – Mapa de Ativação

M_{AT} – Máscara de Atenção

mAP – *Mean Average Precision*

M_F – Medida F

N_C – Número de Classes

N_{PC} – Número de Predições por Célula

obj – objeto predito

O_P – Objetos Preditos

p – predição de um objeto

P_S – Profundidade do Cubo de Saída

P_V – Profundidade do Volume

P_r – Precisão

Q_{ctrd} – Quantidade de Centroides

Q_F – Quantidade de Filtros

Q_P – Quantidade de Parâmetros

Q_{PC} – Quantidade de Partes do Corpo

Q_M – Quantidade de Multiplicações

r – *Ground Truth* de detecção

S_e – Sensibilidade

T_B – Instâncias em um *Batch*

V_P – Verdadeiros Positivos

Z – Vetor de Espaço Latente

Z_P – *Zero Padding*

LISTA DE SIGLAS

API – *Application Programming Interface*

CGAN – *Conditional Generative Adversarial Networks*

CPU – *Central Processing Unit*

DPC – Dataset for Pornography Censorship

FPS – *Frames Per Second*

GAN – *Generative Adversarial Network*

GPU – *Graphical Processing Unit*

ILSVRC – *ImageNet Large Scale Visual Recognition Challenge*

IoU – *Intersection over Union*

LSTM – *Long Short-Term Memory network*

mAP – *Mean Average Precision*

MPAA – *Motion Picture Association of America's*

NMS – *Non-Maximal Suppression*

R-CNN – *Region-based Convolutional Network*

ReLU – *Rectified Linear Unit*

REST – *Representational State Transfer*

RPN – *Region Proposal Network*

SGD – *Stochastic Gradient Descent*

SSD – *Single Shot MultiBox Detector*

SVM – *Support Vector Machine*

YOLO – *You Only Look Once*

LISTA DE ABREVIATURAS

ACORDE. – *Adult Content Recognition with Deep Neural Networks*

AttGAN. – *Attention-based Generative Adversarial Networks*

AttNET. – *Attention Mask Generation Network*

Cocind. – Coordenação de Classificação Indicativa

ConvBlock. – Bloco de Convoluções

ConvNets. – Redes Neurais Convolucionais

CycleGAN. – *Cycle-Consistent Adversarial Networks*

DataSex. – *Dataset for Sexual Content Identification*

Grad-CAM. – *Gradient-weighted Class Activation Mapping*

InstaGAN. – *Instance-Aware Generative Adversarial Network*

SepBlock. – Bloco compostos por *Separable Convolutions*

SUMÁRIO

1. INTRODUÇÃO

O volume de conteúdo adulto disponível na internet cresce constantemente e, em paralelo, cresce também a quantidade de problemas que o uso indiscriminado deste pode causar. Casos de vício em sexo são frequentes [You08], e outras desordens como impotência e falta de apetite sexual começam a ser ralatadas[1][2]. Neste sentido, Cooper [Coo98] chamou de *Triple A Engine* a relação dos 3 fatores que motivam o consumo de pornografia na internet: i) *Accessibility*, em função da facilidade de acesso; ii) *Affordability*, referente ao baixo custo; iii) *Anonymity*, referente à anonimidade dos internautas.

Assumindo os 3 pilares do *Triple A Engine* propostos por Cooper [Coo98], entende-se que a quebra de um destes pilares pode contribuir para minimizar os impactos causados pelo consumo indiscriminado de conteúdo pornográfico, onde a identificação automática de imagens e/ou vídeos que contenham pornografia possibilitaria a aplicação de medidas restritivas, quebrando o pilar *Accessibility*. Desta maneira, tal solução passa por problemas relacionados à Visão Computacional, uma área interdisciplinar que estuda como computadores podem contribuir em tarefas de alto nível envolvendo imagens e vídeos, automatizando demandas executadas por humanos [BB82].

Métodos tradicionais para extração de características de imagens, referidos como *hand-crafted methods*, geram as chamadas *features* de baixo nível que, quando empregadas em tarefas de detecção de conteúdo pornográfico, alcançaram resultados promissores [WHH+09, WNHC12, HWG+14]. Por outro lado, a definição destas *features* de baixo nível é complexa, além de não serem suficientemente generalistas [NLW+16]. Neste cenário, assumindo que métodos de *Deep Learning* [LBH15] baseiam-se nos dados para naturalmente encontrar as *features* que melhor representam um conteúdo, abordagens baseadas em Redes Neurais Artificiais podem ser utilizadas na construção de mecanismos efetivos para contornar o problema.

Deep Learning [LBH15], uma iniciativa de Aprendizado de Máquina que permite aos sistemas computadorizados melhorar seus resultados por meio de dados e da experiência que esses dados representam, torna possível a construção de sistemas que possam desempenhar tarefas complexas do mundo real. *Deep Learning* é flexível e aprende a representar o mundo como uma hierarquia, onde conceitos de alto nível são definidos com base em conceitos mais abstratos [GBC16]. Dentre as estratégias de *Deep Learning* destacam-se as Redes Neurais Convolucionais [LJB+89], que vem definindo o atual estado-da-arte para diversos problemas de Visão Computacional [LJB+89, TLWT15, ZBS+15, ZLLH16, Li17, BDM+16, GSW+18, LSD15, BHC15, HGDG17].

Redes Convolucionais (ConvNets) são Redes Neurais Artificiais compatíveis com o processamento de estruturas de dados com topologia em grade. Na prática, ConvNets são Redes Neurais que substituem a multiplicação direta de matrizes por operações de convolução em pelo menos uma de suas camadas [GBC16]. Sua aplicação em tarefas de reconhecimento de dígitos escritos a mão, utilizando os *datasets* MNIST [LBBH98] e CASIA [LYWW11], além da classificação de

[1] https://goo.gl/RgyHOq
[2] http://goo.gl/tjuVVt

objetos diversos, utilizando o *dataset* ImageNet [DDS+09], superaram os resultados obtidos por humanos [CWF+15] [HZRS15]. ConvNets foram utilizadas em tarefas de classificação de imagens e reconhecimento de objetos definindo o estado-da-arte nestas tarefas [KSH12, CWF+15]. São capazes de aprender representações a partir de dados brutos, tendo sido aplicadas com sucesso em diferentes tarefas como classificação de imagens e vídeos [SWBR16, WSR+16], detecção de objetos [RHGS15, RDGF16], compreensão de vídeo [WB17b, WB17a], recuperação de conteúdo [WMB18, WLMB18], classificação de texto [BWCB17, WBCB17] e predição de funções proteicas [WBDC17], por exemplo. Neste sentido, ConvNets podem ser treinadas para identificar a ocorrência de conteúdo pornográfico, havendo iniciativas semelhantes publicadas na literatura (e.g. [Mou15, NLW+16, ZZG+16]).

Ao utilizar *datasets* que disponham de anotações com as posições e dimensões onde ocorrem objetos de diferentes conceitos como carros, pessoas, casas, aviões ou quaisquer outros, ConvNets podem ser treinadas para detectar a ocorrência de elementos relacionados a estes conceitos em imagens digitais. Iniciativas disponíveis na literatura [RHGS15, Gir15, RF16, RDGF16, LAE+16] reestruturam arquiteturas de ConvNets, tornando possível ir além da classificação de imagens como um elemento único, passando a identificar as coordenadas de localização que apontam exatamente onde cada objeto acontece no espaço da imagem. Desta maneira, ConvNets contribuem com a representação de *features* de alto nível para solucionar problemas de detecção de objetos, uma tarefa de Visão Computacional que lida com a detecção de instâncias de objetos relacionados a uma classe presentes em imagens ou vídeos digitais.

ConvNets também podem ser estruturadas para geração e transformação de conteúdo. Goodfellow et al. [GPAM+14] criaram as *Generative Adversarial Networks* (GANs), um método que utiliza duas redes neurais, conhecidas como geradora e discriminadora, que competem entre si para serem mais acuradas em suas tarefas. A rede geradora tem como tarefa produzir saídas artificiais relacionadas a um determinado domínio, sendo estas saídas suficientemente realistas para que possam se passar por reais. A rede discriminadora, por outro lado, tem como tarefa avaliar as saídas produzidas pela rede geradora, buscando distinguir saídas reais de saídas artificais. Com o treinamento das redes, o gerador passa a gerar saídas mais realistas, ao mesmo tempo em que o discriminador melhora sua capacidade de identificar saídas artificiais. Ao estender o conceito das GANs, Zhu et al. [ZPIE17] apresentaram o método *Cycle-Consistent Adversarial Networks* (CycleGAN), que é capaz de traduzir imagens entre diferentes domínios, tarefa conhecida como tradução imagem-para-imagem. O método não necessita de *datasets* anotados ou alinhados, contornando problemas como a indisponibilidades de *datasets* para domínios específicos.

Em geral, o treinamento de ConvNets para tarefas de classificação e detecção de objetos é um processo de aprendizado supervisionado [FLGC11], dependente do volume e da generalidade do conjunto de treinamento. Para que os modelos atinjam resultados comparáveis aos gerados por humanos, os *datasets* devem idealmente conter milhares ou mesmo milhões de instâncias. Neste sentido, *datasets* que relacionam conceitos específicos como lugares [ZKL+16], modelos de carros [KSDFF13] e celebridades [ZZWS12] são conhecidos. Existem diversos *datasets* relacionados na

literatura, muitos deles disponíveis abertamente, especialmente para tarefas de classificação e detecção de objetos. ImageNet [DDS+09] é o *dataset* que define os padrões para treinamento de modelos para classificação, dada sua quantidade de classes e instâncias. Para a tarefa de detecção de objetos, tanto PASCAL VOC [EEVG+15] quanto MS COCO [LMB+14] e *Open Images* [KDA+] são largamente utilizados. Por outro lado, conjuntos focados em pornografia são pouco relatados na literatura. Avila et al. [ATC+13] apresentam um *dataset* composto por imagens obtidas por meio de cenas de filmes pornográficos, sendo este conjunto aplicado no treinamento de ConvNets em [Mou15], no entanto, o *dataset* não dispõe de anotações em nível de posicionamento e dimensões de objetos.

Os métodos de detecção de objetos apoiados por *fetures* de alto nível extraídas por ConvNets podem ser utilizados para aplicar coberturas em imagens contendo elementos específicos, como partes relacionadas à pornografia. Além disso, assumindo que vídeos são composições de imagens, pode-se utilizar estas mesmas técnicas para aplicar coberturas automáticas em vídeos.

1.1 Contribuições

Tomando por base a problemática relacionada ao conteúdo pornográfico disponível abertamente na internet e as desordens que o consumo indiscriminado deste pode causar em públicos vulneráveis, especialmente crianças e adolescentes, esta tese adotou diferentes abordagens baseadas em Redes Neurais para criar 3 experiências distintas de censura ao conteúdo pornográfico. Foi adotada uma estratégia incremental, onde buscou-se em cada etapa reduzir o impacto da aplicação das censuras sobre o conteúdo, diminuindo a interferência na experiência do usuário. As abordagens utilizadas partiram i) pelo treinamento e avaliação de ConvNets para classificação de imagens, abordando censurar de pornografia como um problema de classificação de imagens, seguindo pela ii) utilização de métodos de detecção de objetos apoiados por *features* extraídas de ConvNets, passando a tratar a censura de partes do corpo que remetem à pornografia como um problema de detecção de objetos e, finalmente, iii) o uso de ConvNets adversárias que permitiu tratar censura à pornografia como um problema de tradução imagem-para-imagem. Nesta sequência incremental, as censuras partiram da simples remoção completa das imagens classificadas como pornográficas, passando pela oclusão de objetos específicos relacionados com partes íntimas do corpo e, finalmente, chegando na geração automática de peças de vestuário, neste caso biquínis, sobre partes nuas do corpo.

A Figura 1.1 apresenta uma visão geral desta tese, ilustrando as 3 abordagens empregadas para atacar o problema de censura à pornografia. Conforme suas particularidades, cada abordagem abre espaço para a aplicação automática de censuras que vão desde as mais invasivas, como a remoção completa da imagem ou *frame*, passando pela aplicação de coberturas sobre regiões específicas de acordo com o conteúdo exibido e, finalmente, modificando sutilmente a imagem gerando peças de vestuário sobre as partes que precisam ser censuradas.

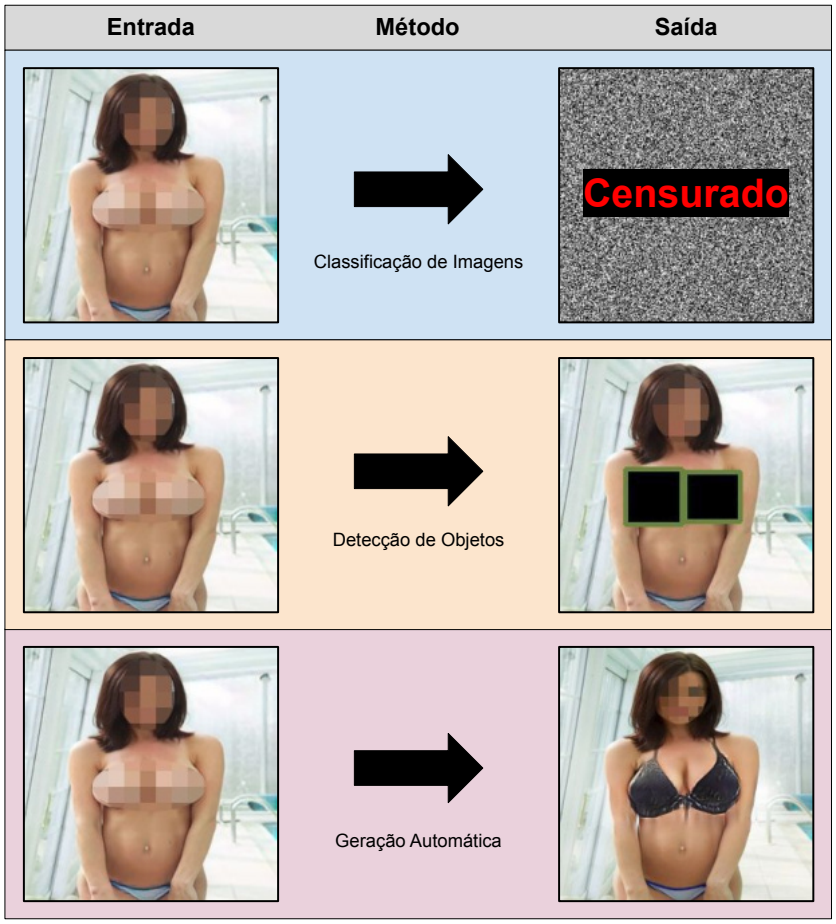


Figura 1.1: Visão incremental dos estágios para censura de pornografia trabalhados nesta tese.

Na abordagem de classificação de imagens pornográficas, esta tese construiu o maior *dataset* [SWP+16] para treinamento de modelos de classificação de imagens pornográficas até então relatado na literatura. Este *dataset* fui utilizado para treinar diferentes modelos de classificação. *DataSex* trata-se de um *dataset* binário para treinamento de modelos preditivos composto por 286.920 imagens, definindo as classes *free* e *porn*. Foram treinados e avaliados 8 modelos de classificação com diferentes arquiteturas de ConvNets onde os resultados preditivos, observados a partir do conjunto de teste de *DataSex*, atingiram acurácias superiores a 99%, permitindo o processamento de até 40 FPS. Além do desempenho preditivo foi possível observar que, para esses classificadores baseados em ConvNets, as regiões mais significativas para classificação de imagens pornográficas estão relacionadas a exposição de partes íntimas, e não somente a exposição de pele. Esta característica denota importante capacidade de generalização e robustez a variações de translação, escala e possíveis oclusões, importantes para a efetividade da censura automática.

Em termos de aplicações, classificadores de imagens pornográficas podem ser utilizados em mecanismos de controle parental, gerando predições para imagens avulsas ou, no caso de vídeos, para sequências de *frames*. Serão apresentados dois testes práticos, onde mídias contendo pornografia (tanto imagens quanto vídeo), identificados como tal por modelos treinados com *DataSex*, foram publicados sem qualquer restrição em redes sociais, neste caso Tumblr e YouTube, que disponibilizaram o conteúdo sem qualquer censura.

Na segunda abordagem, que tratou censura à pornografia como um problema de detecção de objetos, são avaliados os desempenhos dos métodos YOLO [RDGF16], *Faster* R-CNN [RHGS15] e SSD [LAE+16], treinando e avaliando modelos com um *dataset* construído especificamente para detecção de partes íntimas do corpo relacionadas à pornografia. *Dataset for Pornography Censorship* (*DPC*) é um *dataset* composto por 6.541 objetos anotados, distribuídos em 3.000 imagens, sendo o único *dataset* publicamente conhecido para detecção de partes íntimas do corpo. Os objetos foram rigorosamente anotados e revisados, seguindo um protocolo de validação cruzada. Experimentos apontaram resultados preditivos com mAP $= 0,6961$ no conjunto de teste de *DPC*.

As experiências com os diferentes métodos de detecção permitiram a criação de uma nova arquitetura totalmente convolucional para a tarefa de detecção, chamada *CensorNet*. Os resultados observados apontam que *CensorNet* é mais leve em termos de quantidade de parâmetros que YOLO *Tiny* [RDGF16], e atinge resultados preditivos melhores, tanto em tempo quanto em mAP. *CensorNet* permite a geração de modelos mais leves em termos de processamento e uso de memória, facilitando sua aplicação em dispositivos com recursos restritos. Foi construída também uma ConvNet composta por uma segunda saída específica para a tarefa de classificação, criando um método híbrido para classificação e detecção de objetos. Esta variação, chamada *CensorPlus*, resultou em um método constituído por uma única rede que executa uma única passada do volume de entrada, gerando predições de detecção de objetos e classificação da imagem como um todo. *CensorPlus* apresentou resultados preditivos, em termos de mAP e tempo, semelhantes aos observados nos métodos relacionados somente para detecção de objetos, além de ter sido capaz de classificar o conjunto de teste de *DataSex*, formado por 64.096 imagens, atingindo acurácia $= 0,9640$.

Finalmente, na terceira abordagem desta tese é apresentado *AttGAN*, um novo método baseado em tradução imagem-para-imagem utilizando redes geradoras para censura automática de conteúdo relacionado à pornografia. O método adiciona informação à entrada dos geradores utilizando máscaras de atenção oriundas de uma ConvNet chamada *AttNET*, treinada para classificação e convertida para a geração de tais máscaras. A intuição para criação do método parte do trabalho de More et al. [MSWB18], uma evolução de CycleGAN [ZPIE17] com enfoque na cobertura de nudez ao desenhar automaticamente biquínis sobre partes nuas do corpo em uma abordagem não-supervisionada, independente de *datasets* alinhados.

Foram desenvolvidas e apresentadas 3 variações do método *AttGAN*. As variações foram comparadas com o método de More et al. [MSWB18] por meio de um formulário *online* composto por 50 conjuntos de imagens censuradas pelos diferentes métodos, avaliadas por 21 participantes. Os resultados colhidos pelo formulário apontam vantagem para o método *AttGAN+*, indicado como o melhor método dentre 1.050 opiniões coletadas. A partir desta constatação, o método *AttGAN+* foi incrementado aplicando a mescla guiada da imagem de entrada com a saída censurada, chamado de método *AttGAN++*, resultando em uma imagem censurada que preserva características periféricas da imagem original.

1.2 Estruturação do Trabalho

Esta tese está estruturada de seguinte maneira: o Capítulo 2 aborda o problema da censura ao conteúdo pornográfico como uma tarefa de classificação de imagens. São apresentados conceitos fundamentais de Redes Neurais Convolucionais, indicando seus principais componentes e combinações, seguindo com a apresentação de um *dataset* específico para o treinamento de modelos para classificação de imagens pornográficas. São descritos os treinamentos de 8 modelos de classificação com diferentes arquiteturas de ConvNets, comparando seus resultados.

O Capítulo 3 segue a temática abordando censura à pornografia como um problema de detecção de objetos. São apresentados os métodos baseados em ConvNets para detecção de objetos e, em seguida, um *dataset* construído especificamente para o treinamento de modelos de detecção de partes íntimas do corpo relacionadas à pornografia. Este *dataset* foi utilizado para treinar modelos de detecção, além de propor duas novas arquiteturas, a primeira delas voltada para desempenho em função do tempo e a segunda, um método híbrido para classificação de imagens e detecção de objetos relacionados a partes íntimas do corpo.

O Capítulo 4 aborda censura de conteúdo pornográfico como um problema de tradução imagem-para-imagem. É proposto um novo método que utiliza redes geradoras para traduzir automaticamente imagens de um domínio relacionado a nudez para um segundo domínio, representando pessoas em trajes de banho. Este novo método adiciona informação à entrada dos geradores utilizando máscaras de atenção geradas por uma ConvNet treinada para classificação, convertida para geração de máscaras de atenção.

Finalmente, o Capítulo 5 conclui a tese com um resumo do que foi realizado, bem como com as limitações do trabalho e direções para trabalhos futuros.

Nota: dada a natureza desta tese, fica registrado o alerta de que o conteúdo gráfico ilustrativo, mesmo cuidadosamente coberto e/ou descaracterizado, pode gerar indisposições ou constrangimentos.

2. CLASSIFICAÇÃO DE IMAGENS PORNOGRÁFICAS

A publicação e o consumo de conteúdos na internet modifica hábitos da sociedade. Mudanças positivas, como a facilidade de comunicação, construção e disseminação de conhecimento são conhecidos. Por outro lado, a facilidade de acesso e o anonimato na rede expõe públicos sensíveis, como crianças e adolescentes, a conteúdos inapropriados [Coo98]. Inserido neste contexto está o conteúdo pornográfico, que quando consumido por públicos sensíveis pode resultar em transtornos ou patologias [You08].

A identificação prévia da presença de pornografia, especialmente em conteúdo público, é uma forma de minimizar o acesso inadvertido a este tipo de material. Por outro lado, dado o volume de conteúdo publicado diariamente, tal tarefa precisa acontecer automaticamente, seja no momento da publicação ou do consumo deste conteúdo. Inicialmente esta tese aborda a identificação de pornografia como um problema de classificação de imagens, apresentando métodos, arquiteturas e bases de imagens anotadas (*datasets*) para indução de modelos preditivos, discutindo seus resultados e possíveis aplicações.

Stockman et al. [SS01] definem uma classe como um conjunto de objetos que apresentam propriedades significativas comuns, sendo a classe de um determinado objeto representada por um rótulo c. Ainda de acordo com Stockman et al. [SS01], classificação é o processo de atribuir um rótulo c a um objeto a partir das propriedades do mesmo. Um classificador é um dispositivo que, dada a inserção de um conjunto de propriedades representativas de um objeto, aponta automaticamente a mais provável classe (c) em que se enquadra este objeto. O erro de classificação acontece quando o classificador aponta a classe c_i para um dado objeto que sabidamente pertence à classe c_j.

Esta tese estudou métodos baseados em ConvNets onde buscou-se atestar a viabilidade de utiliza-las para censurar conteúdo pornográfico automaticamente. Neste contexto, a classificação de imagens pornográficas permite aplicar contramedidas como a exclusão ou descaracterização da imagem, prevenindo exposições inapropriadas. O presente capítulo aborda a censura de conteúdo pornográfico como um problema de classificação de imagens.

Muitos métodos para detecção de pornografia utilizam a exposição de pele como parâmetro para classificação [ZGZL04, SBCC03]. Os métodos baseados em exposição de pele apresentam vantagens com relação à velocidade, às questões de invariância especialmente de rotação, além de serem pouco afetados pelas imagens de fundo. Por outro lado, a exposição de pele gera um viés para falsos positivos, fazendo com que parte das imagens que expõe pele sejam classificadas como pornográficas, mesmo tratando-se da prática de esportes ou exposição de partes do corpo na praia, por exemplo.

Classificadores de imagens pornográficas podem ser construídos partindo-se de um conjunto de regras, um processo intuitivo para extração de *features* que, como relatado por Yin et al. [YXY11], define regras para identificar os pixeis que representam pele. As imagens são classificadas como pornográficas caso a quantidade de pele seja compatível com um determinado limiar. Este método

apresenta restrições, já que é impossível definir um limiar ajustado para quantidade de pele em imagens genéricas. Imagens em um contexto de praia, por exemplo, apresentarão maior área de pele do que imagens que representem um contexto de sala de aula.

A tarefa de classificação também pode ser abordada como um problema de Recuperação de Informações. Estes métodos utilizam consultas sobre *handcrafted features* (como SIFT [Low04], SURF [BTVG06] ou HOG [DT05], por exemplo) extraídas de bases de imagens previamente categorizadas. As imagens resultantes das consultas são quantificadas por categoria, sendo estas quantidades comparadas com um limiar para definir a classe da imagem avaliada. Por outro lado, a abordagem de classificação por recuperação de informações apresenta fragilidades como a dificuldade de selecionar os métodos para extração de *features*.

Atualmente, métodos baseados em aprendizado vem apresentando resultados superiores aos gerados por humanos em tarefas de classificação [RDS+15]. Métodos baseados em aprendizado podem contornar as fragilidades observadas nos métodos construídos sobre regras ou recuperação de informação que utilizam *features* de baixo nível. Redes Neurais são treinadas utilizando *datasets* para tarefas específicas, induzindo modelos especializados em extrair as *features* que melhor descrevem os elementos vinculados a esta tarefa. *Features* extraídas por Redes Neurais são conhecidas como *features* de alto nível [LRM+12] e, neste contexto, Redes Neurais Convolucionais (ConvNets) tornam-se uma alternativa natural para extrair *features* que permitam classificar imagens. Esta tese estuda a efetividade da classificação de imagens pornográficas utilizando ConvNets.

Uma ConvNet é uma técnica de *Deep Learning* que combina três ideias para atingir um grau de variação de translação, escala e distorção sendo elas: i) campos receptivos locais (conhecidos como filtros), ii) parâmetros compartilhados e iii) *pooling* espacial. A operação de convolução substitui a multiplicação de matrizes totalmente conectadas, presentes em redes neurais convencionais. Esta operação garante as duas primeiras ideias listadas, além de reduzir a quantidade de parâmetros. Filtros convolucionais, também conhecidos como *kernels*, são otimizados em um processo de treinamento que utiliza o algoritmo *backpropagation* [RHW88].

$$v_{ij}^{xy} = relu\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right) \tag{2.1}$$

A Equação 2.1 define uma convolução onde (x, y) é a posição no j-ésimo mapa de *features* da i-ésima camada, m indexa o conjunto de mapas de *features*, b_{ij} representa o valor do *bias* correspondente, w_{ijm}^{pq} representa os valores dos pesos na posição (p, q), e P_i e Q_i representam largura e altura do filtro, respectivamente. Nesta equação, a função de ativação ReLU [KSH12] é aplicada como fonte de não-linearidade. Essencialmente, ReLU limita minimamente o valor de saída de convoluções a zero (i.e., $relu(v) = max(0, v)$).

Esta tese aborda a identificação de conteúdo pornográfico como um problema de aprendizado supervisionado, definido por Acharya et al. [AR05] como um modelo treinado com um grande conjunto de amostras de padrões de treinamento previamente rotulados, a fim de estimar os parâmetros estatísticos de cada classe de padrões. Foram utilizadas Redes Neurais Convolucionais

para treinar modelos de classificação, sendo utilizado *DataSex* [SWP+16], um *dataset* de imagens pornográficas desenvolvido no contexto desta tese. Foram treinados 8 modelos de ConvNets, construídas com base em arquiteturas descritas na literatura. Os modelos foram comparados em função do desempenho preditivo e das necessidades com relação ao uso de memória e tempo de predição, bem como possíveis aplicações práticas.

2.1 Modelagem

Ao aplicar o método baseado em aprendizado, esta tese treinou 8 ConvNets construídas a partir de Micro-arquiteturas apresentadas na literatura. Micro-arquiteturas são blocos compostos por camadas fundamentais como convoluções e *poolings*. Ao relacionar uma coleção de micro-arquiteturas, chega-se a uma rede formada por uma construção em blocos. A seguir serão apresentados os blocos que motivaram a construção das redes utilizadas para classificação de imagens avaliadas nesta tese para a tarefa de censura automática de conteúdo pornográfico.

2.1.1 GoogleNet/Inception

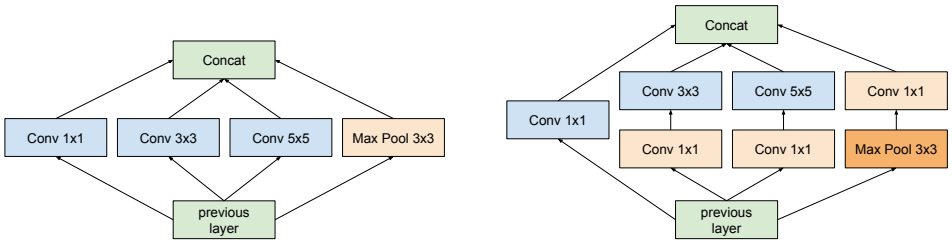
Desenvolvida por Szegedy et al. [SLJ+15], apresenta uma abordagem paralela formando blocos que exploram diferentes dimensões de filtros convolucionais (1×1, 3×3 e 5×5), além de uma camada de *pooling*. Szegedy et al. chamaram estas estruturas de blocos Inception. Em cada bloco, as saídas dos filtros paralelos são concatenadas, formando um novo volume sequencial que pode ser utilizado como entrada para um próximo bloco ou camada. Blocos Inception, quando observados isoladamente, definem uma arquitetura de rede, desta maneira, redes baseadas nestes blocos podem ser entendidas como redes formadas por composições de redes.

Inception foi um avanço importante no desenvolvimento de ConvNets para solução de problemas de classificação. Anteriormente, a principal estratégia de construção de ConvNets baseava-se fundamentalmente em pilhas cada vez maiores de camadas convolucionais, gerando redes cada vez mais profundas. Redes profundas sofrem com problemas de desaparecimento de gradientes [GBC16], resultando em camadas que não contribuem efetivamente com melhores resultados preditivos.

(a) grande (b) pequeno (c) vários.

Figura 2.1: Representação de diferentes dimensões do objeto mais significativo em uma imagem.

No bloco Inception, Szegedy et at. [SLJ+15] atacam o problema da profundidade das redes tornando-as, ao invés de mais profundas, mais largas. Um bloco relaciona camadas convolucionais em paralelo, resultando em mais camadas com menor profundidade, contribuindo para o fluxo dos gradientes e para o treinamento da rede. Soma-se também a aplicação de filtros com diferentes dimensões em cada convolução paralela, contribuindo para a indução de filtros especializados nas diferentes proporções de tamanho que o objeto mais significativo de uma imagem pode apresentar. Imagens referentes a classe gato, como as apresentadas pela Figura 2.1, podem apresentar gatos em diferentes tamanhos e quantidades, dificultando a definição adequada das dimensões de *kernel* para cada camada de convolução. Em ConvNets, *kernels* maiores são melhores com objetos maiores, enquanto *kernels* menores são melhores com objetos menores e possivelmente difusos.



(a) fundamento do bloco. (b) bloco Inception final.

Figura 2.2: Ilustração do bloco Inception, da intuição inicial ao bloco resultante. Baseando em [SLJ+15].

A Figura 2.2 ilustra os blocos Inception em duas leituras. A Figura 2.2a ilustra a visão conceitual do módulo, onde convoluções paralelas com diferentes dimensões de filtros processam o volume de entrada, resultando na concatenação de suas saídas e na geração de um novo volume. Na prática, a abordagem conceitual ilustrada pela Figura 2.2a gera custo computacional no tocante a parâmetros e operações matemáticas. Dado que a camada de *pooling* não reduz a profundidade do volume, ao concatenar as saídas das 3 convoluções (1×1, 3×3 e 5×5) com a saída da camada de *pooling*, obtém-se um volume de saída sempre crescente a cada módulo. Estas restrições foram solucionadas com a adição de camadas *bottleneck* (ou gargalo), como pode ser observado na Figura 2.2b.

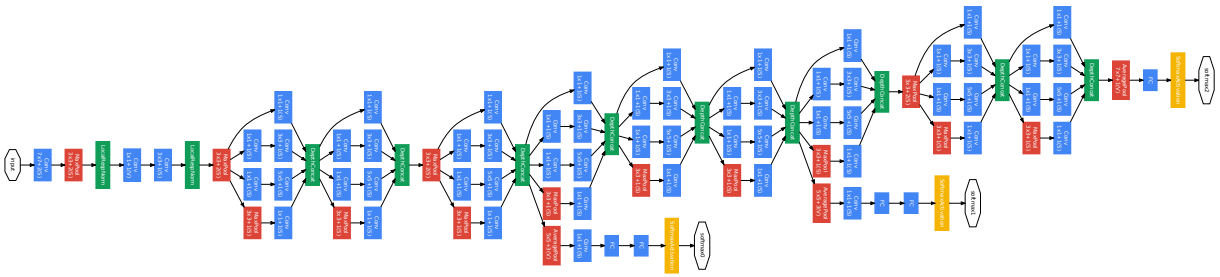


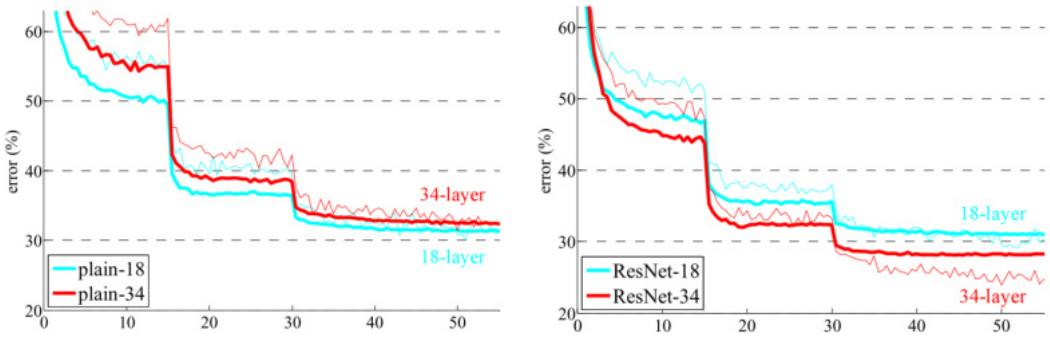
Figura 2.3: Arquitetura baseada em [SLJ+15], contendo 9 blocos Inception.

A Figura 2.3 mostra a primeira arquitetura de rede que utilizou blocos Inception. Esta arquitetura é conhecida como GoogleNet (ou Inception V1) e, em 2014, apresentou os melhores

resultados no *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) [RDS+15], uma competição onde um subconjunto do *dataset* ImageNet [DDS+09] é classificado em 1.000 diferentes classes. A arquitetura GoogleNet apresentou 6.7% de erro de classificação.

2.1.2 Resnet

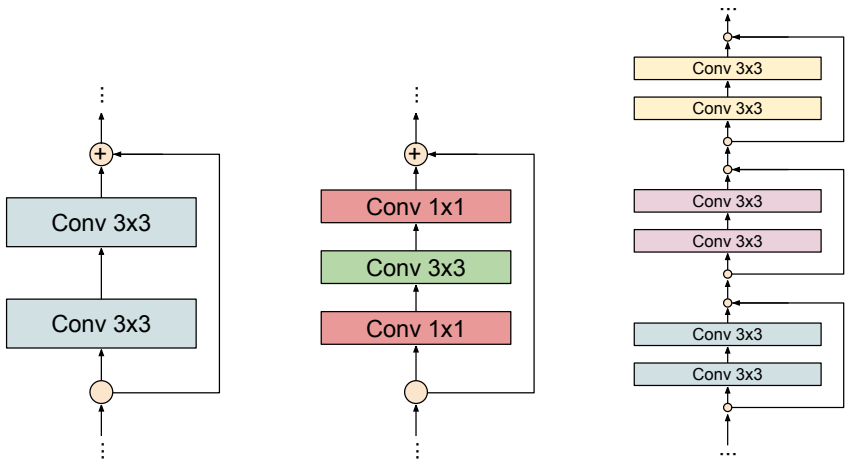
Sabendo que redes neurais muito profundas são difíceis de treinar, He et al. [HZRS16] atacaram o problema partindo de uma intuição diferente daquela observada nos blocos Inception [SLJ+15]. Ao comparar redes com diferentes profundidades (nesse caso, com 18 e 34 camadas), He [HZRS16] observou os erros de treinamento e validação ilustrados pelo gráfico da Figura 2.4a, onde é possível perceber que uma rede com 34 camadas apresenta maior erro de treino e validação do que outra rede de arquitetura semelhante, mas com 18 camadas. Em sua discussão, He at al. [HZRS16] evidenciaram um problema de otimização de modelo. Por outro lado, os resultados apresentados na figura 2.4b mostram arquiteturas equivalentes, também com 18 e 34 camadas, adicionando conexões residuais. As conexões residuais conectam diretamente camadas em diferentes níveis, geralmente somando os volumes no passo de propagação e, quando em treinamento, contribuindo com para a retro-propagação dos gradientes. Ao adicionar conexões residuais, foi observado que as arquiteturas maiores obtiveram resultados melhores, além de reduzir o tempo de treinamento. He at al. [HZRS16] experimentaram também arquiteturas com 50, 101 e 152 camadas, quando manteve-se a queda do erro de classificação.



(a) sem blocos residuais [HZRS16]. (b) com blocos residuais [HZRS16].

Figura 2.4: Comparação de desempenho entre ConvNets de 18 e 34 camadas, com e sem blocos residuais, apresentado em [HZRS16].

As conexões residuais formam blocos que somam volumes resultantes de diferentes camadas de uma ConvNet. A Figura 2.5 ilustra conexões residuais partindo de sua visão inicial (Figura 2.5a), onde o volume é transpassado de camadas rasas e somado a camadas mais profundas, passando para uma abordagem com *bottleneck*, reduzindo a quantidade de parâmetros na camada de dimensões 3×3 (Figura 2.5b) e, finalmente, a Figura 2.5c ilustra blocos Resnet inseridos em uma arquitetura de rede.



(a) bloco residual. (b) com *bottleneck*. (c) inseridos em uma arquitetura.

Figura 2.5: Ilustração dos blocos residuais inseridos em uma arquitetura de rede [HZRS16].

Em 2015, uma arquitetura com conexões residuais composta por 152 camadas obteve 3.6% de erro de classificação no ILSVRC [RDS+15], marcando o novo recorde até aquela data. Este foi o primeiro resultado que superou a classificação humana, conforme aponta Russakovsky et al. [RDS+15].

2.1.3 Darknet-19

Darknet-19 [RF16] é uma ConvNet composta por 19 camadas de convolução e 5 camadas de *pooling*. É uma arquitetura construída especialmente para tratar problemas de detecção de objetos, sendo base para o método YOLO [RF16]. Por outro lado, independente de sua aplicação principal, Darknet-19 apresenta resultados promissores para classificação de imagens, atingindo 8.8% de erro no ILSVRC [RDS+15].

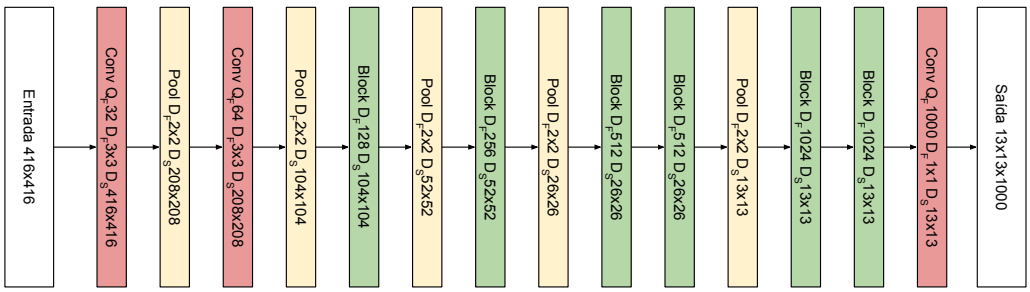


Figura 2.6: Arquitetura Darknet-19 com saída ajustada para um problema de classificação de 1000 classes, onde Q_F define a quantidade de filtros, D_F as dimensões destes filtros e D_S as dimensões de saída.

Darknet-19 é composta por blocos que intercalam convoluções 3×3 e 1×1, seguidos por uma operação de *pooling* que reduz as dimensões de saída dos blocos pela metade. Após

a redução de dimensionalidade, o bloco seguinte dobra a quantidade de filtros convolucionais. A Figura 2.6 ilustra Darknet-19, assumindo como entrada uma imagem de dimensões 416×416. Trata-se de uma arquitetura totalmente convolucional, onde cada camada mantem as proporções espaciais, característica importante para tarefas de detecção de objetos. Para tarefas de classificação, o número de filtros da última camada convolucional é definido de acordo com o número de classes, onde a média dos valores gerados por cada filtro é utilizada como entrada para a função de ativação SOFTMAX, resultando em probabilidades por classe.

2.1.4 Xception

Formada por 36 camadas convolucionais, Chollet [Cho16] define Xception como uma variação mais robusta de Inception [SLJ+15]. Xception desacopla a relação entre a profundidade de canais e o posicionamento espacial utilizando *Separable Convolutions* [SM14].

Uma *Separable Convolution* divide a operação de convolução convencional em dois estágios: i) *depthwise* e ii) *pointwise*. O estágio *depthwise* é definido por uma convolução espacial que aplica filtros independentes para cada canal do volume de entrada. Por serem independentes, os filtros da camada *depthwise* tem profundidade $= 1$, representando menos parâmetros que os filtros de camadas convolucionais convencionais, onde a profundidade de cada filtro é dependente da quantidade de canais do volume de entrada. O segundo estágio de uma *Separable Convolution*, conhecido como *pointwise*, é uma segunda convolução que aplica filtros 1×1 para recombinar o volume recebido da camada *depthwise*, sendo a profundidade destes filtros definida conforme a profundidade do volume recebido. A quantidade de filtros 1×1 da camada *pointwise* representa a profundidade do volume de saída de uma *Separable Convolution*.

Com relação às camadas de convolução convencionais, *Separable Convolutions* reduzem a quantidade de parâmetros e o número de operações de multiplicação computadas. Esta redução é atingida ao combinar as estratégias de filtros independentes de canais de entrada com os filtros 1×1. As Equações 2.2 e 2.3 formalizam, respectivamente, a quantidade de multiplicações computadas em convoluções convencionais e *Separable Convolutions*.

$$Q_M = Q_F \times L_F \times A_F \times P_V \times (L_V - L_F + 1) \times (A_V - A_F + 1) \tag{2.2}$$

$$Q_M = L_F \times A_F \times P_V \times (L_V - L_F + 1) \times (A_V - A_F + 1) + Q_F \times P_V \times L_V \times A_V \tag{2.3}$$

A Equação 2.2 representa a quantidade de multiplicações (Q_M) executadas em uma camada de convolução convencional, onde Q_F representa a quantidade de filtros, L_F e A_F as dimensões de largura e altura dos filtros, L_V e A_V representam largura e altura e P_V a profundidade do

volume de entrada. Em seguida, a Equação 2.3 formaliza a quantidade de multiplicações para uma *Separable Convolution*, mantendo os mesmos termos.

A título de comparação, assumindo 128 filtros de dimensões 3×3 para um volume de entrada com dimensões $7 \times 7 \times 3$, deslocando os filtros sobre o volume em saltos (*stride*) de tamanho $= 1$, uma convolução convencional resulta em 86.400 multiplicações. Mantendo as mesmas configurações para um volume de entrada com dimensões $32 \times 32 \times 3$, esta mesma convolução executa $3.110.400$ multiplicações. Por outro lado, utilizando uma *Separable Convolution*, os referidos exemplos resultarão em 19.491 e 417.516 multiplicações, respectivamente.

$$Q_P = Q_F \times L_F \times A_F \times P_V \tag{2.4}$$

$$Q_P = L_F \times A_F \times P_V + Q_F \times P_V \tag{2.5}$$

A Equação 2.4 define a quantidade de parâmetros Q_P para uma camada convolucional com Q_F filtros de dimensões $L_F \times A_F$ para um volume de entrada com profundidade P_V. Baseada nos mesmos termos, a Equação 2.5 representa quantidade de parâmetros para uma *Separable Convolution*.

Assumindo o mesmo exemplo que comparou quantidades de computações, utilizando uma camada convolucional com 128 filtros de dimensões 3×3 e um volume de entrada com 3 canais, uma convolução convencional será composta por 3.456 parâmetros. Estas mesmas configurações aplicadas a uma *Separable Convolution* representam 411 parâmetros.

Xception [Cho16] é composta por uma sequência de 14 módulos que utilizam conexões residuais e *Separable Convolutions*. Mesmo utilizando menos parâmetros e executando menos computações, Xception atinge um erro de classificação de 5.5% no ILSVRC [RDS+15].

2.1.5 Densenet

Dense Convolutional Network, chamada por Huang et al. [HLVDMW17] de DenseNet, é formada por blocos de convoluções onde a saída de cada convolução está conectada a todas as entradas das convoluções subsequentes. Esta abordagem é relacionada ao observado em Resnet [HZRS16], substituindo-se somas por concatenações, além de uma quantidade maior de conexões.

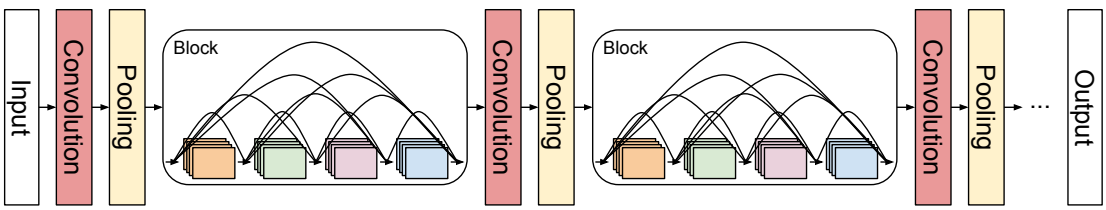


Figura 2.7: Rede formada por DenseBlocks seguidos por convolução/*pooling*.

Em um bloco DenseBlock, cada convolução recebe entradas residuais de todas as camadas anteriores, concatenando sua saída com todas as outras saídas das camadas subsequentes, até o final do bloco. Dadas as concatenações, os volumes de saída de cada bloco crescem em profundidade, que passa a ser limitada por uma camada convolucional que recebe o volume de saída do bloco. A Figura 2.7 ilustra blocos DenseBlock inseridos em uma arquitetura de rede, onde cada bloco ilustrado é seguido por uma convolução, limitando a profundidade do volume, e por um *pooling*, reduzindo as dimensões espaciais.

Como efeito prático, Huang et al. [HLVDMW17] destacaram a redução na perda dos gradientes, o reforço na propagação de *features* e o favorecimento do reuso, formando o que Huang et al. chamaram de consciência coletiva [HLVDMW17]. Ao utilizar *Separable Convolutions* [SM14], DenseBlocks também reduzem o número de parâmetros da rede. As conexões também contribuem para o treinamento, já que facilitam o fluxo dos gradientes das camadas mais profundas para as camadas mais rasas.

2.2 Trabalhos Relacionados

Redes neurais artificiais têm sido amplamente utilizadas em problemas de classificação de imagens [LJB+89, SLJ+15], segmentação semântica de imagens [LSD15, HGDG17], além de detecção de objetos [RHGS15, Gir15, RF16, RDGF16, LAE+16]. Iniciativas para detecção de conteúdo pornográfico baseadas em Aprendizado de Máquina, especialmente ConvNets, começam a ser relatadas. Por outro lado, até o momento estas iniciativas limitam-se à classificação, deixando em aberto tarefas como segmentação semântica e detecção de objetos.

O trabalho apresentado em [Mou15] foi o primeiro a utilizar ConvNets para a classificação de conteúdo pornográfico, neste caso em vídeos. O trabalho propõe um método que requer o treinamento de modelos utilizando duas diferentes arquiteturas, no caso *AlexNet* [KSH12] e *Inception* [SLJ+15], ambas pretreinadas com ImageNet [DDS+09]. Essas redes são então treinadas com o *dataset* proposto em Avila et al. [ATC+13], dividido em 5 *folds*[1]. O método querer o treinamento de 10 modelos distintos: um modelo para cada *fold* (5 modelos) para cada uma das arquiteturas ($\times 2$). Para evitar o *overfitting*, o autor reporta o uso de *dropout* e de aumento de dados, aplicando a seleção randômica de recortes (*crops*) e adição de ruído nas imagens.

Recentemente, Nian et al. [NLW+16] apresentaram um novo método para detecção de pornografia baseado em ConvNets. O treinamento dos modelos foi realizado com 13.300 imagens pornográficas obtidas a partir de consultas na internet. As imagens passaram por uma estratégia de aumento de dados que exigiu a rotulação manual de áreas pornográficas em cada imagem, permitindo a extração de 10 recortes compatíveis com pelo menos 90% da área pornográfica anotada em cada imagem. As imagens também foram rotacionadas e espelhadas, contando também com a inserção de ruído em seus canais de cores. O treinamento partiu de um modelo pretreinado com o

[1] Subconjuntos mutuamente exclusivos.

dataset ImageNet [DDS+09], utilizando arquitetura AlexNet [KSH12], onde foram mantidos os pesos das camadas de convolução. Durante o treinamento, todos os pesos foram ajustados, tanto nas camadas de convolução quanto nas camadas totalmente conectadas. Por outro lado, a estratégia de treinamento adotada conta com um conjunto de treino auxiliar, utilizado para compensar erros de validação. Na prática, para cada erro identificado no conjunto de validação, novas instâncias semelhantes ao erro são adicionadas ao conjunto de treino, favorecendo o aprendizado e a diminuição do erro. A estratégia produz resultados expressivos no conjunto de teste utilizado (98.6% de acurácia), no entanto, é necessário avaliar o método sobre diferentes bases de dados para descartar a ocorrência de *overfitting*.

Outra iniciativa recente para classificação de pornografia que utiliza uma abordagem híbrida é relatada em [ZZG+16]. O trabalho utiliza duas estratégias sendo elas: *coarse detection*, baseada em algoritmos para detecção de pele e face, e *fine detection*, baseada em uma ConvNet treinada a partir de um *dataset* descrito em [ZGZgL16], composto por 19.000 imagens sendo 8.000 pornográficas e 11.000 não-pornográficas. Os testes foram executados sobre um recorte de 8.000 imagens reservadas para teste, atingindo uma acurácia de 97.2%. Inicialmente o método utiliza *coarse detection* para avaliar se uma imagem apresenta grandes áreas de exposição de pele, encaminhando aquelas que ultrapassam um limiar para o módulo *fine detection*. O módulo *fine detection* utiliza uma ConvNet para classificar as imagens como pornográficas ou não-pornográficas. As imagens que não atingem o limiar definido no módulo *coarse detection* são classificadas como não-pornográficas.

Abordagens comerciais para detecção automática de conteúdo pornográfico são disponibilizadas por Google[2] e Microsoft[3]. A API *Google Cloud Vision* encapsula elementos de Aprendizado de Máquina, acessíveis a partir de interfaces de serviços REST, disponibilizando um conjunto de facilidades dentre as quais destaca-se Detecção de Conteúdo Explícito (*Explicit Content Detection*), que refere-se ao conteúdo adulto e cenas de violência. A Microsoft *Computer Vision API*, também disponibilizada por interfaces de serviços REST, classifica imagens de acordo com seus elementos e pode ser configurada para identificar a ocorrência de conteúdo adulto.

As iniciativas comerciais apontadas oferecem facilidades para desenvolvedores de software, especialmente por disponibilizarem interfaces de serviços que permitem o uso dos recursos na forma de caixa-preta. Por outro lado, por dependerem de comunicação em rede, impõe limitações que inviabilizam sua utilização em contextos que demandem pronta resposta (i.e. análise de vídeos).

2.3 *Datasets*

A literatura dispõe de poucas alternativas de *datasets* que abordam conteúdo essencialmente pornográficos. Iniciativas como ImageNet [DDS+09] possuem algum volume de imagens adultas, no entanto, em quantidade irrelevante frente ao *dataset* como um todo. Uma segunda alternativa, apresentada por Ávila et al. [ATC+13], disponibiliza um *dataset* formado por imagens

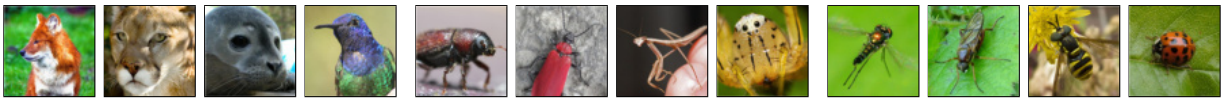
[2] goo.gl/MOXw3N
[3] goo.gl/KXcC2e

que representam filmes pornográficos e não-pornográficos, fazendo deste um *dataset* alinhado com os objetivos desta pesquisa. Por outro lado, dadas as características dos problemas para os quais estes *datasets* foram construídos, o volume de imagens especificamente pornográficas disponíveis torna-se insuficiente para o treinamento de modelos de ConvNets [LJB+89]. Desta maneira, surge a necessidade da construção de um novo *dataset* adaptado aos objetivos desta tese.

No contexto desta pesquisa foi desenvolvido *DataSex*, um *dataset* formado por duas classes, *free* e *porn*, compostas por imagens extraídas de duas fontes distintas: i) uma seleção de imagens não-pornográficas do ImageNet [DDS+09] e ii) consultas direcionadas coletadas por *web crawling* [KT00], resultando em um *dataset* significativamente maior (aproximadamente $17\times$) que o *dataset* apresentado por Ávila et al. [ATC+13]. *DataSex* não faz distinção entre as instâncias de suas classes, além de não manter qualquer relação temporal entre elas. Até o presente momento, pelo melhor que se conhece da literatura, nenhum outro *dataset* apresenta volume semelhante de instâncias, fazendo deste o maior conjunto rotulado de imagens pornográficas e não-pornográficas até então relatado. A seguir, serão apresentados os *datasets* relacionados ao tema encontrados na bibliografia.

2.3.1 *Dataset* ImageNet

ImageNet [DDS+09] é um *dataset* constituído por aproximadamente 14 milhões de imagens rotuladas. A estruturação de conteúdo segue o WordNet [KF99], um banco de dados léxico para língua inglesa que classifica seus conceitos na forma de conjuntos de sinônimos (*synonym set* ou *synset*). Atualmente o ImageNet possui 21.841 *synsets*, sendo uma de suas principais características a organização de suas classes que compõe uma hierarquia semântica. A Figura 2.8 apresenta um exemplo de hierarquia presente no ImageNet, onde o conceito ilustrado pela Figura 2.8c é uma especialização dos conceitos ilustrados pelas Figuras 2.8a e 2.8b.



(a) animais. (b) invertebrados. (c) artrópodes.

Figura 2.8: Exemplo de hierarquia entre classes no ImageNet [DDS+09].

O conteúdo do ImageNet [DDS+09] foi analisado manualmente em um processo de *crowdsourcing* [Bra08], sendo descrito por seus criadores como um *dataset* livre de ruído. Um dos objetivos do *dataset* é dispor de diversidade de instâncias para os conceitos por ele cobertos. Desta maneira, cada *synset* apresenta uma média aproximada de 650 imagens com diferentes composições de objetos, posições e planos de fundo, características que contribuem para a criação de modelos de classificação com alto poder de generalização.

Pornografia não é um conceito coberto pelo ImageNet [DDS+09], não existindo um *synset* dedicado para este tipo de conteúdo, o que impede o treinamento de modelos para classificação

de imagens pornográficas utilizando este *dataset* como única fonte de instâncias. Por outro lado, imagens contendo nudez são esparsamente encontradas em *synsets* relacionados à roupas íntimas, por exemplo. São encontrados *synsets* específicos para pessoas em trajes de banho, tanto masculinas como femininas, reforçando a exposição do corpo e da pele.

Por seu volume e diversidade, ImageNet é recorrentemente citado na literatura ([WSR+16, RF16, RHGS15, HGDG17]) como fonte de dados para iniciar o treinamento de ConvNets para os mais variados fins, em um processo conhecido como *transfer learning* [PY09]. A presença destes conteúdos amplia o potencial de utilizar ImageNet [DDS+09] como base para o pretreinamento de modelos, inclusive para classificação de imagens pornográficas.

2.3.2 *Dataset* apresentado por Ávila et al. [ATC+13]

O *dataset* apresentado por Ávila et al. [ATC+13] representa vídeos pornográficos e não-pornográficos a partir de imagens extraídas das cenas destes filmes. O *dataset* dispõe de duas classes: ponográfico e não-pornográfico. A classe não-pornográfico é dividida em duas subclasses, referidas como não-pornográfico fácil e difícil. Nessas sublasses, não-pornográfico fácil (Figura 2.9a) apresenta situações e/ou elementos genéricos do mundo real, tais como bicicletas, carros, aviões ou brinquedos, em ações como correr, caminhar ou dirigir, enquanto não-pornográfico difícil (Figura 2.9b) relaciona situações que destacam a exposição corporal, como fotos em trajes de banho, lutas ou amamentação de bebês. A Figura 2.9 apresenta exemplos com diferentes níveis de exposição de pele proporcionadas por situações cotidianas que não necessariamente significam pornografia.

(a) não-pornográfico fácil. (b) não-pornográfico difícil. (c) pornográfico

Figura 2.9: Exemplos de imagens presentes no *dataset* apresentado por Ávila et al. [ATC+13].

As instâncias das classes são imagens extraídas de aproximadamente 80 horas de vídeos pornográficos e não-pornográficos que foram preprocessados para extração dos *keyframes*[4] de todas as cenas. O resultado deste preprocessamento são 802 conjuntos contendo entre 1 e aproximadamente 320 *keyframes*, conforme a quantidade de cenas de cada vídeo. Ao todo foram identificadas 16.727 cenas, onde os vídeos pornográficos apresentam em média 15.6 cenas, os não-pornográficos fáceis 33.8 cenas e os não-pornográficos difíceis 17.5 cenas, totalizando 16.727 imagens. O conteúdo do *dataset* de Ávila et al. [ATC+13] apresenta diversidade étnica, já que participam das cenas asiáticos, negros, brancos e mestiços. Por outro lado, o *dataset* apresenta incompatibilidades com os objetivos desta pesquisa, sendo: i) a ocorrência de vídeos com uma única cena, além da ii) presença de desenhos, animações e das vinhetas de abertura dos filmes. Além disso, como iii) os *keyframes*

[4] *Keyframe* é o frame posicionado ao centro de uma dada cena.

de todas as cenas são admitidos como instâncias, cenas não-pornográficas como diálogos e tomadas externas resultam em imagens que não apresentam pornografia sendo rotuladas como pornográficas.

2.3.3 *DataSex*

Esta tese aborda métodos computacionais baseados em ConvNets para classificação, detecção e censura automática de conteúdo pornográfico. Desta maneira, a necessidade de induzir modelos para estes fins exige *datasets* compostos fundamentalmente por este tipo de conteúdo e, como relatado, as alternativas existentes não contemplam plenamente estas necessidades. Em função de seus objetivos, esta tese apresenta *DataSex*, um *dataset* formado pelas classes *free* e *porn*, compostas por imagens extraídas de duas fontes: i) uma seleção do ImageNet [DDS+09] contendo *synsets* com pessoas em atividades cotidianas e ii) consultas direcionadas, coletadas por *web crawling* [KT00], feitas sob fontes de conteúdo pornográfico. Somados, estes conjuntos resultam em um *dataset* significativamente maior (aproximadamente $17\times$) que o *dataset* apresentado por Ávila et al. [ATC+13]. *DataSex* não faz distinção entre as instâncias de suas classes, além de não manter qualquer relação temporal entre elas. Até o presente momento, pelo melhor que se conhece da literatura, nenhum outro *dataset* apresenta volume semelhante de instâncias, fazendo deste o maior conjunto rotulado de imagens pornográficas e não-pornográficas até então relatado.

2.3.3.1 Composição dos conjuntos

DataSex é dividido em subconjuntos de treino, teste e validação, compostos por imagens rotuladas distribuídas de maneira equilibrada entre as classes *free* e *porn*. *DataSex* distingue-se de Ávila et al. [ATC+13] fundamentalmente por i) tratar suas instâncias como imagens isoladas, ii) por restringir a classe *porn* em imagens contendo nudez e sexo explícito e, finalmente, iii) pelo volume de imagens aproximadamente $17\times$ maior (16.727 vs. 286.920). A distribuição das instâncias em *DataSex*, assim como seus respectivos volumes de dados, podem ser observados na Tabela 2.1. A Figura 2.10 exibe amostras para ambas às classes. Para exibição nesta tese, por motivos óbvios, imagens pertencentes à classe *porn* foram tarjadas ou descaracterizadas.

Tabela 2.1: Distribuição dos subconjuntos em *DataSex*.

Conjunto	*free*	*porn*	*free* + *porn*	Volume em Disco (≈)
Treino	95.388	95.388	190.776	18,0GB
Teste	32.048	32.048	64.096	5,9GB
Validação	16.024	16.024	32.048	3,0GB
Total	143.460	143.460	286.920	26,9GB

2.3.3.2 Extração e composição de dados

A composição de dados do *DataSex* foi realizada por estratégias distintas. A classe *free* foi criada a partir de um subconjunto de imagens extraídas do ImageNet [DDS+09], enquanto que a

(a) exemplos da classe *free*. (b) exemplos da classe *porn*.

Figura 2.10: Amostra dos conjuntos *free* e *porn* de *DataSex*.

classe *porn* utilizou imagens obtidas por meio de consultas guiadas na Web, onde foram predefinidas as fontes contendo este tipo de imagens.

A classe *free* foi construída com base em um sub-conjunto do ImageNet. O subconjunto é formado por 197 *synsets* (listados no Apêndice A.1) que possuem ao todo 93.116 imagens relacionando temas que exibem pessoas em contextos genéricos. Para equilibrar a quantidade de instâncias de ambas as classes, 50.344 imagens foram selecionadas randomicamente de outros 803 *synsets* do *dataset* ImageNet [DDS+09]. A estratégia de utilizar imagens não-pornográficas contendo pessoas em diferentes contextos do cotidiano teve por objetivo contrapor pessoas em cenários pornográficos, contribuindo para a indução de modelos generalistas, evitando relacionar pessoas diretamente à pornografia.

A classe *porn* de *DataSex* foi composta por 243.968 imagens de diferentes *web sites* de conteúdo adulto. As imagens foram obtidas por um *web crawler* configurado para realizar consultas com termos específicos (e.g. nudez, sexo, pornografia). Dentre o total de imagens, 138.475 são estáticas, enquanto que 105.493 são animações no formato GIF, das quais foram utilizados somente o primeiro *frame*. As imagens obtidas apresentam configurações de qualidade e dimensões diversas, além de diferentes formatos como GIF, JPG e PNG. Após a realização de preprocessamento, foram removidas as imagens duplicadas, corrompidas ou que não atingiram as dimensões mínimas (largura ou altura < 128), resultando em 143.460 imagens que compõe a classe *porn*.

2.3.3.3 Preprocessamento

A composição do *dataset DataSex* resultou em um número de imagens suficientemente grande que dificultou a análise manual de todas as instâncias. Assim, a remoção de duplicidades foi automatizada a partir de *features* extraídas de cada imagem utilizando uma ConvNet Inception [SLJ+15], pretreinada com o *dataset* ImageNet [DDS+09]. Por serem bastante representativas com relação ao conteúdo da imagem, *features* convolucionais podem ser aplicadas em tarefas como agrupamento, recuperação de conteúdo ou aproximação por similaridade [LB98].

O procedimento adotado, ilustrado pela Figura 2.11, extraiu para cada imagem 1024 *features* da última camada convolucional de uma ConvNet com arquitetura *Inception* [SLJ+15], treinada com o *dataset* ImageNet, gerando uma representação de cada imagem no formato de um vetor de 1024 valores. Os vetores foram utilizadas para construir uma estrutura de árvore do tipo KD-Tree [Ben75], que apresenta uma complexidade de tempo para operação de busca por vizinhos mais próximos de $O(\log n)$. Finalmente, sendo I o conjunto de todas as imagens e $Q(i)$ uma

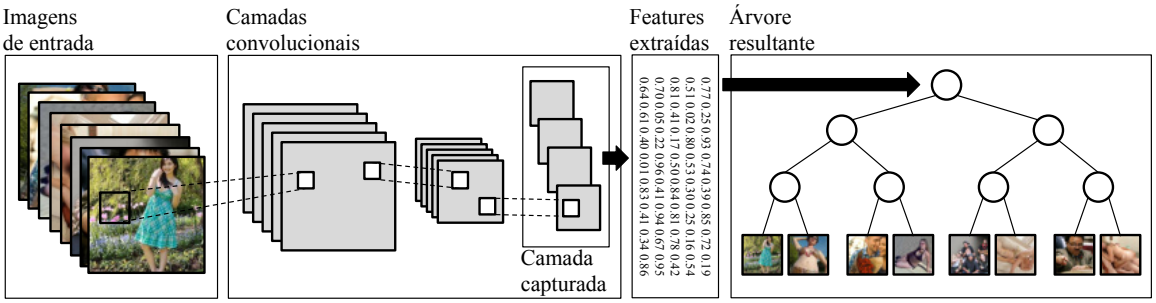


Figura 2.11: Geração de árvore para remoção de duplicatas.

operação de consulta que retorna $true$ para $q > 1$, onde q representa a quantidade de réplicas da imagem i no conjunto I, após a execução de $Q(i) \forall i \in I$ foram excluídas todas as ocorrências de i para valores de $Q(i) = true$. Observou-se que este procedimento resultou na identificação e remoção de 31.448 duplicatas presentes em *DataSex*.

2.4 Método

Para avaliar a efetividade do uso de *DataSex* para o treinamento de modelos de ConvNets para classificação de conteúdo pornográfico, foram avaliadas 8 arquiteturas construídas com base na literatura. A partir dos modelos obtidos, foram comparados os resultados observando quantidade de parâmetros, uso de memória, necessidade de armazenamento em disco e tempos de predição utilizando CPU e GPU. Foi comparado também o desempenho preditivo dos modelos observando precisão (Equação 2.6), sensibilidade (Equação 2.7), medida F (Equação 2.8) e acurácia (Equação 2.9).

Todos os modelos seguiram o mesmo protocolo de treinamento. Dada a quantidade de instâncias de *DataSex*, a convergência dos modelos aconteceu com poucas épocas, permitindo limitar o treinamento em 40 épocas, sendo cada modelo gerado avaliado utilizando o conjunto de validação. Os modelos iniciais foram pretreinados utilizando o *dataset* ImageNet [DDS+09]. Foi utilizado o otimizador Gradiente Descendente Estocástico (GSD), ajustado conforme mecanismo descrito em [Bot12], utilizando os seguintes hiperparâmetros: $\alpha = 1 \times 10^{-3}$, $\alpha(\gamma) = 4\%$, *momentum* $= 9 \times 10^{-1}$, *weight decay* $= 5 \times 10^{-4}$, sendo α a taxa de aprendizagem e $\alpha(\gamma)$ um fator para redução da taxa de aprendizagem aplicado a cada 2 épocas. Para inicialização dos pesos foi utilizado o método descrito em [HZRS15].

O treinamento utilizou aumento de dados aplicando espelhamento horizontal randômico, além de recortes randômicos que preservaram 90% da área da imagem. Não foi aplicado ruído e as proporções espaciais não foram preservadas, ocorrendo o achatamento para compatibilizar a imagem com as dimensões de entrada da rede. A Figura 2.12 exemplifica 4 diferentes configurações para uma mesma instância após o processo de aumento de dados.

Todos os treinamentos foram realizados com as mesmas configurações de *hardware*: GPU NVIDIA M40 com 12GB de memória, executando sobre um CPU Intel Xeon E5-2603 com 128GB

Figura 2.12: Exemplo de aumento de dados aplicado no treinamento de ConvNets para classificação.

de memória principal, com o *dataset* armazenado em disco de estado sólido. Quanto ao *software*, foi utilizado Keras [C+] + Tensorflow [AAB+] sobre sistema operacional Ubuntu 14.04 [5].

2.5 Validação Experimental

Para verificar a acurácia de modelos treinados com *DataSex*, assim como mensurar o tempo necessário para classificar conjuntos de imagens, foram executados experimentos que abordaram treinamento, validação e teste de modelos com arquiteturas baseadas em Darknet-19 [RF16], Densenet [HLVDMW17], Inception [SLJ+15], Inception Resnet [SIVA17], MobileNet [HZC+17], NasnetMobile [ZVSL18], ResNet [HZRS16] e Xception [Cho16]. Metodologia e resultados serão descritos a seguir.

2.5.1 Métricas de Avaliação

Os modelos de ConvNets para classificação treinados nesta tese foram avaliados utilizando as seguintes métricas: precisão, sensibilidade, medida F e acurácia. A precisão (Pr), representada pela Equação 2.6, é a habilidade que um classificador tem em não rotular como positiva uma instância que, de fato, é negativa. Neste caso, o melhor resultado para Pr é $1,0$ enquanto o pior é $0,0$. A sensibilidade (Se), representada pela Equação 2.7, é a habilidade de um classificador para encontrar todas as instâncias pertencentes a classe positiva. O melhor resultado esperado para sensibilidade é $Se = 1,0$, sendo $0,0$ o pior.

$$Pr = \frac{V_P}{V_P + F_P} \qquad (2.6) \qquad\qquad Se = \frac{V_P}{V_P + F_N} \qquad (2.7)$$

A medida F (M_F) é utilizada para avaliar a acurácia de teste a partir da média harmônica da precisão e da sensibilidade, provendo uma medida mais realista do que a observação isolada da precisão e/ou da sensibilidade. A medida F é apresentada pela Equação 2.8.

Nas Equações 2.6, 2.7, 2.8 e 2.9, V_P representa a quantidade de imagens da classe *porn* classificadas como *porn*, enquanto F_P representa imagens da classe *free* classificadas como *porn*. V_N são imagens da classe *free* classificadas como *free* e F_N representa imagens da classe

[5] releases.ubuntu.com/14.04/

porn classificadas como *free*. Pr refere-se a precisão, Se a sensibilidade, M_F a medida F e Ac a acurácia.

$$M_F = \frac{2 \cdot Pr \cdot Se}{Pr + Se} \tag{2.8}$$

$$Ac = \frac{V_P + V_N}{V_P + V_N + F_P + F_N} \tag{2.9}$$

2.5.2 Resultados Observados

A Tabela 2.2 apresenta os resultados obtidos pelos melhores modelos de cada ConvNet treinada. Analisando os resultados é possível observar que todos os modelos obtiveram acurácia preditiva superior a 96% no conjunto de teste. O conjunto de teste foi avaliado somente pelo melhor modelo gerado de cada arquitetura de rede, identificado a partir do conjunto de validação que foi avaliado em todos os modelos gerados para cada arquitetura ao fim de cada época.

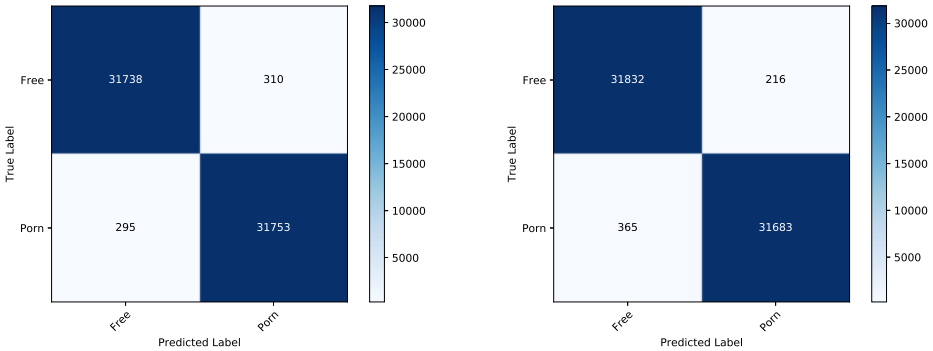
Tabela 2.2: Resultados observados por arquitetura.

Arquitetura	Parâmetros	Memória	Disco	CPU	GPU	Pr	Se	Fm	Ac
Darknet-19	38.716.386	342,11MB	296MB	551 ± 027ms	25 ± 01ms	0,9903	0,9908	0,9906	0,9906
Densenet	12.646.210	277,16MB	98MB	511 ± 004ms	45 ± 02ms	0,9896	0,9886	0,9891	0,9891
Inception	21.806.882	326,74MB	168MB	539 ± 007ms	40 ± 02ms	0,9638	**0,9941**	0,9787	0,9783
Inception Resnet	54.339.810	417MB	417MB	1.043 ± 065ms	70 ± 03ms	0,9932	0,9886	**0,9909**	**0,9909**
MobileNet	**3.230.914**	**84,31**MB	**25**MB	**130** ± 017ms	**09** ± 00ms	**0,9942**	0,9322	0,9622	0,9634
NasnetMobile	4.271.830	111,71MB	36MB	145 ± 007ms	45 ± 03ms	0,9855	0,9784	0,9819	0,9820
ResNet	23.591.810	563,56MB	181MB	1.262 ± 160ms	32 ± 01ms	0,9846	0,9852	0,9849	0,9849
Xception	20.865.578	539,16MB	160MB	1.231 ± 157ms	32 ± 02ms	0,9918	0,9860	0,9889	0,9890

A melhor acurácia observada foi de $99,09\%$, obtida com uma arquitetura baseada em Inception Resnet, errando 581 predições, conforme pode ser observado na matriz de confusão ilustrada pela Figura 2.13b. A segunda melhor acurácia observada atingiu $99,06\%$, com uma arquitetura baseada em Darknet-19, errando 24 predições a mais (Figura 2.13a). Por outro lado, analisando a Figura 2.13, que destaca as matrizes de confusão para os dois melhores modelos, pode-se observar que Darknet-19 apresenta maior tendência para a classe *porn*, o que em aplicações práticas pode minimizar a exposição ao conteúdo pornográfico. As Matrizes de Confusão referentes a todos os modelos de classificação treinados estão disponíveis no Apêndice B. Além da semelhança no desempenho preditivo observado entre as arquiteturas baseadas em Inception Resnet e Darknet-19, chama a atenção o *speedup* de Darknet-19 sobre Inception Resnet que, em GPU, atinge $2,8\times$.

A Figura 2.14 apresenta erros de classificação para as classes *free* e *porn*. Nesta ilustração, as imagens erroneamente classificadas como *porn* (2.14a) apresentam grande exposição de pele, mas não se tratam de nudez ou pornografia. Por outro lado, as imagens classificadas como *free* (2.14b), mesmo não representando sexo explícito, apresentam exposição de partes íntimas.

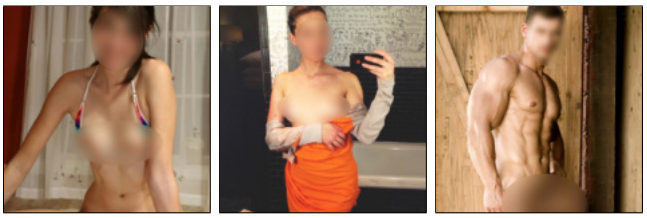
Para visualizar a distinção das *features* geradas por ConvNets treinadas com *DataSex*, foram analisadas as 242 *features* da última camada convolucional para todas as instâncias de validação, extraídas de uma ConvNet com arquitetura Inception. A Figura 2.15 apresenta gráficos com representações t-SNE [MH08], uma técnica que possibilita visualizar dados de alta dimensionalidade



(a) Darknet-19 (b) Inception Resnet

Figura 2.13: Matrizes de confusão dos dois melhores modelos.

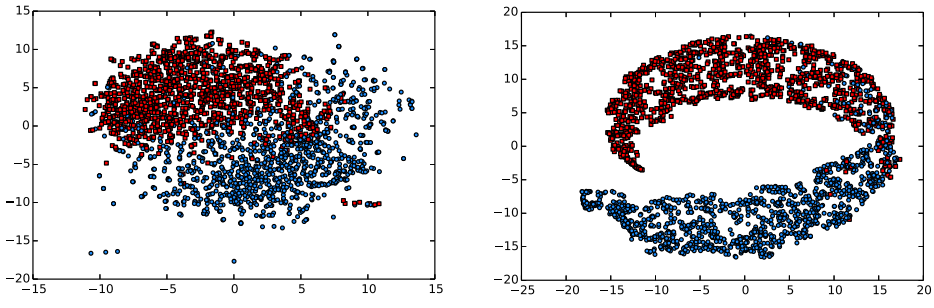
(a) erroneamente classificadas como *porn*.



(b) erroneamente classificadas como *free*.

Figura 2.14: Amostra de erros de classificação extraídos do conjunto de teste de *DataSex*.

posicionando cada ponto em um mapa bi ou tridimensional. Observando os gráficos é possível perceber significativa distinção dos conjuntos quando as *features* são extraídas do modelo treinado com *DataSex*(Figura 2.15b) em comparação com a separação gerada com *features* extraídas do modelo treinado com ImageNet [DDS+09] (Figura 2.15a), reforçando a eficácia de *DataSex* como fonte de dados para indução de modelos para detecção de conteúdo pornográfico.



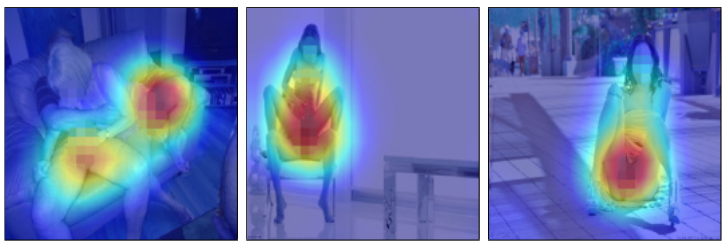
(a) modelo treinado com ImageNet. (b) modelo treinado com *DataSex*.

Figura 2.15: Representações t-SNE para instâncias de validação. Azuis =*free*, vermelhos =*porn*.

Conforme a literatura, a classificação de conteúdo pornográfico a partir de níveis de exposição de pele apresenta restrições, especialmente dada a necessidade de definição de limiares [ZGZL04, SBCC03]. Ao mesmo tempo, já que pele não necessariamente significa pornografia, a exposição de pele acaba por ser uma característica insuficiente para classificar esse tipo de conteúdo. A pornografia está relacionada não simplesmente a pele, mas a partes específicas do corpo e em como estas partes compõe a imagem. Desta maneira, é importante que um classificador de conteúdo pornográfico gere suas predições relacionando essas características.

Para identificar visualmente as partes das imagens que mais influenciaram os resultados gerados pelas ConvNets, foi utilizado *Gradient-weighted Class Activation Mapping* (Grad-CAM) [SCD+17], um método capaz de representar graficamente as decisões de uma ConvNet, tornando-as mais transparentes. Em problemas de classificação, Grad-CAM aplica uma imagem como entrada em uma ConvNet assumindo uma classe de interesse, calculando os *scores* de confiança para cada classe representada pela rede. Na camada de saída da rede, os gradientes são zerados para todas as classes, exceto para a classe de interesse. Os gradientes são retropropagados para as camadas anteriores e, quando combinados, permitem calcular um mapa de calor que representa as áreas da imagem que foram mais influentes no resultado predito pela rede [SCD+17].

(a) Grad-CAM de imagens da classe *free*.



(b) Grad-CAM de imagens da classe *porn*.

Figura 2.16: Grad-CAM para a classe *porn* gerados a partir de imagens das classes *free* e *porn*.

A Figura 2.16 ilustra um experimento onde foram analisadas imagens das classes *free* e *porn*, mantendo somente os gradientes da classe *porn*. Desta maneira, o método destaca visualmente as áreas com as maiores ativações para a classe *porn*, resultando em um mapa de calor onde as áreas vermelhas apontam maiores gradientes. Observando os mapas é possível notar que, nas imagens da classe *free* (Figura 2.16a), os gradientes mais intensos surgem nas pernas desnudas, parte do corpo bastante presente em imagens da classe *porn*. Por outro lado, ao observar os gradientes gerados pelas imagens da classe *porn* (Figura 2.16b), é possível perceber que os mapas indicam os maiores gradientes nas áreas relacionadas as genitálias, independente do sexo dos indivíduos e da interação entre as partes.

2.5.2.1 Aplicações

Paralelamente à massiva popularização do uso de dispositivos que permitem exibir e registrar imagens, surgem diversas possibilidade de aplicações para classificadores treinados com *datasets* de conteúdo adulto. Iniciativas como restrição de acesso a conteúdos inadequados em redes sociais ou controle parental são breves exemplos. Redes sociais recebem diariamente montantes significativos de conteúdo, sendo parte deste composto por imagens e vídeos. Dado o volume, a análise manual desse conteúdo torna-se impraticável. Por outro lado, um classificador automático pode contribuir nesta tarefa, promovendo o encaminhamento necessário aos casos onde conteúdo inapropriado seja identificado, seja bloqueando ou encaminhando o conteúdo para alguma moderação.

O consumo de séries, filmes e shows por *streaming* vem crescendo significativamente, promovendo mudanças profundas nas mídias tradicionais[6]. Estes conteúdos normalmente apresentam classificação indicativa previamente apontada por entidades reguladoras (i.e., Cocind[7], MPAA[8]). Mesmo nos casos onde o conteúdo é avaliado por entidade reguladora, a classificação indicativa representa a obra como um todo em uma análise holística, permitindo muitas vezes que partes específicas apresentem conteúdo inapropriado para públicos sensíveis. Desta maneira, a análise de vídeos em tempo de execução pode impedir automaticamente a visualização de segmentos inapropriados, conforme a avaliação automática de cada imagem.

Para avaliar a aplicação de ConvNets na censura automática de conteúdo pornográfico utilizando modelos treinados por esta tese, foram elaborados 2 testes práticos em tradicionais sites para publicação aberta de conteúdo: Tumblr[9] e YouTube[10]. Tumblr é uma rede social que conecta usuários em função de seus interesses sobre conteúdos diversos, como fotos, animações e vídeos. Conforme apontado em relatório mensal da consultoria SimilarWeb[11], Tumblr teve aproximadamente 371 milhões de visitas somente em maio de 2019. YouTube é uma rede social de compartilhamento de vídeos onde usuários disponibilizam e consomem conteúdo. Atualmente são assistidos em média 1 bilhão de horas de vídeo por dia no YouTube[12]. A título de comparação com Tumblr, SimilarWeb registrou, em maio de 2019, aproximadamente 25 bilhões de acessos ao YouTube.

Em 17 de dezembro de 2018, Tumblr implementou um mecanismo para banir todo o conteúdo adulto que seus usuários publicam em suas contas[13]. A partir desta data, todo conteúdo disponibilizado por meio do site é automaticamente classificado, tornando-se privado ao usuário sempre que o resultado da classificação aponta para conteúdo adulto. Todas as imagens que apresentem sexo ou nudez, especialmente exibindo genitálias, são classificadas como conteúdo adulto. As imagens automaticamente classificadas como conteúdo adulto não são excluídas, permanecendo privadas ao usuário e identificadas por uma tarja vermelha contendo a indicação "APPEL".

[6] forbes.com/sites/aalsin/2018/07/19/the-future-of-media-disruptions-revolutions-and-the-quest-for-distribution
[7] Coordenação de Classificação Indicativa
[8] Motion Picture Association of America's
[9] tumblr.com
[10] youtube.com
[11] similarweb.com
[12] youtube.com/yt/about/press/
[13] theguardian.com/technology/2018/dec/03/tumblr-to-ban-all-adult-content

Também em dezembro de 2018 o YouTube removeu 58 milhões de vídeos, dentre os quais muitos estavam relacionados a nudez e pornografia[14]. Em suas políticas de conteúdo, o YouTube proíbe claramente sexo explícito, exibição de genitálias e nudez em geral[15]. O YouTube não deixa claro o método que utiliza para classificar seu conteúdo, no entanto, é sabido que usuários podem denunciar conteúdos impróprios, inclusive atuando como moderadores, não evitando que muitos outros usuários tenham acesso a esses vídeos.

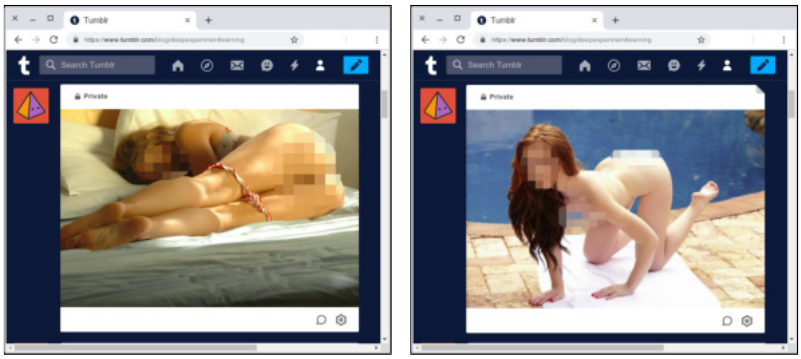


Figura 2.17: Imagens pornográficas não bloqueadas na rede social Tumblr.

Para avaliar a efetividade do método atualmente utilizado pelo site Tumblr, que não descreve publicamente como ou com que técnicas é construído, foram submetidas duas imagens classificadas como pornográficas por modelos de ConvNets treinados com *DataSex*. As imagens apresentam exposição de partes íntimas e não foram classificadas como conteúdo adulto pelo método de classificação utilizado pelo site Tumblr. A Figura 2.17 exibe as duas imagens utilizadas no teste listadas na conta de um usuário da rede Tumblr. Para exibição nesta tese, as partes íntimas e as faces exibidas na Figura 2.17 foram descaracterizadas. As imagens submetidas ao Tumblr foram mantidas originais.

Para o teste executado sobre o YouTube, foi utilizado um vídeo de 30 segundos composto por 13 cenas generalistas que reúnem situações cotidianas como discussões em grupo, passeios pela cidade, famílias na praia ou veículos em manobra. Além destas, existem 3 cenas $[3,8,12]$ de sexo explicito, extraídas de filmes pornográficos, distribuídas randomicamente dentre as outras 10 cenas. Inicialmente os frames desse vídeo foram classificados um-a-um por uma ConvNet treinada com *DataSex*, respeitando a ordem temporal. Foi utilizando o modelo baseado em Inception Resnet, que apresentou o melhor resultado de classificação nas avaliações realizadas com *DataSex*. A título de visualização, as probabilidades obtidas foram sincronizadas com as cenas, de maneira que seja possível avaliar a efetividade do modelo sobre este conteúdo.

A Figura 2.18 apresenta um gráfico com as probabilidades para cada um dos 800 *frames* pertencerem a classe *porn*. Ainda na Figura 2.18, o gráfico de probabilidades é sincronizado com o *keyframe* de cada cena que compõe o vídeo. Analisando o comportamento do gráfico é possível observar saltos nas probabilidades geradas pelos *frames* alinhados com as cenas 3, 8 e 12, onde os

[14] dailymail.co.uk/sciencetech/article-6493145/YouTube-pressure-problem-content-takes-58-mln-videos-quarter
[15] support.google.com/youtube/answer/2802002?hl=en-GB

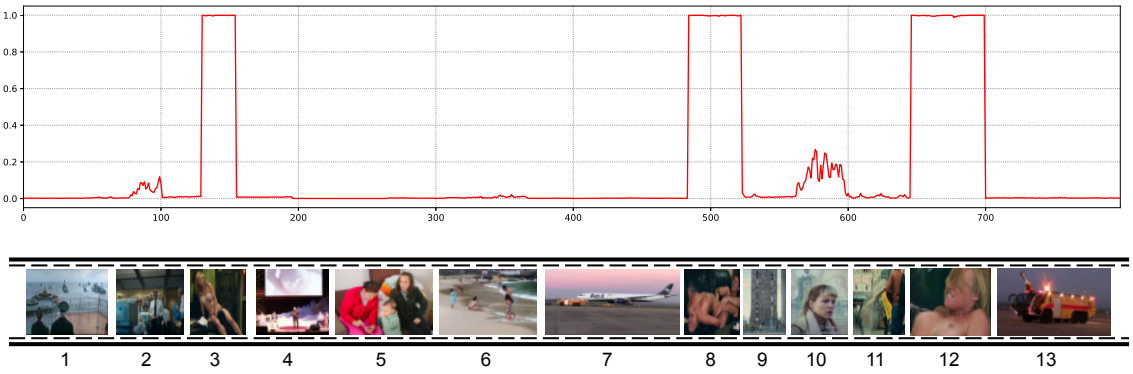


Figura 2.18: Linha do tempo exibindo os *frames* centrais de cada cena, alinhados com os *scores* de classificação para a classe *porn*.

valores oscilaram entre $\approx 0,97$ e $\simeq 1,00$. Nas cenas relacionadas à classe *free*, as probabilidades mantiveram-se próximas de zero, exceto na cena 2 e especialmente na cena 10, onde atingiram um pico de $\approx 0,27$.

No teste com a rede social YouTube, o vídeo de 13 cenas anteriormente avaliado *frame* a *frame* por um modelo treinando com *DataSex* foi submetido como conteúdo de acesso irrestrito. Conforme referido, dentre as 13 cenas do vídeo, 3 são efetivamente pornográficas. O resultado observado após a postagem foi a pronta disponibilidade do vídeo para acesso, sem qualquer tipo de restrição ou indicação. Certamente, após a visualização de alguns usuários, este vídeo poderia ser denunciado e, em função da denúncia, removido, o que não impediria o impacto para um conjunto de usuários.

A Figura 2.19 demonstra o vídeo disponibilizado na rede social YouTube, apontando duas das 3 cenas pornográficas que o compõe. Ao constatar a ausência de restrições ao conteúdo postado, foram feitos os registros visuais apresentados neste texto e o vídeo em questão foi removido.

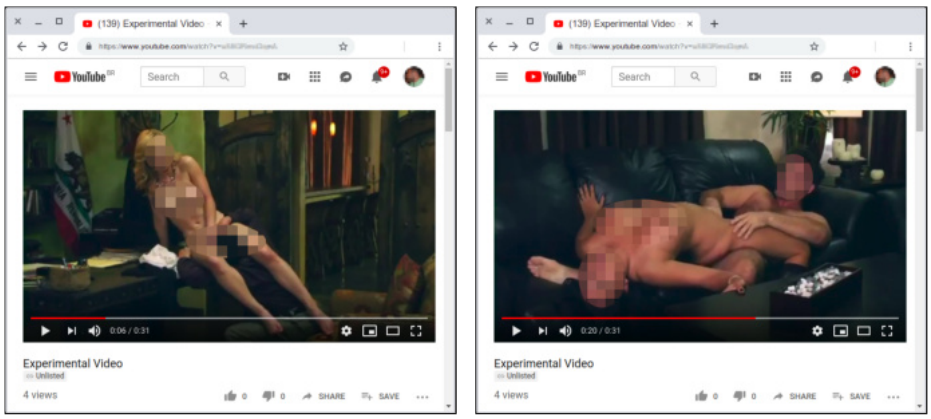


Figura 2.19: Vídeo contendo 3 cenas pornográficas não bloqueado pelo YouTube.

2.6 Considerações e Discussão

Este capítulo abordou o tema de censura automática de conteúdo pornográfico como um problema de classificação de imagens. Foi apresentado *DataSex*, um *dataset* binário para treinamento de modelos preditivos composto por 286.920 imagens, definindo as classes *free* e *porn*. Foram apresentados os resultados de 8 modelos de classificação com diferentes arquiteturas de ConvNets treinados com *DataSex*.

Os resultados preditivos, observados a partir do conjunto de teste de *DataSex*, atingiram acurácias superiores a 99%, permitindo o processamento de até 40 FPS utilizando GPU Tesla M40. Além do desempenho preditivo, foi possível observar que, para esses classificadores baseados em ConvNets, as regiões mais significativas para classificação de imagens pornográficas estão relacionadas a exposição de partes íntimas e não somente a exposição de pele. Esta característica denota importante capacidade de generalização, importantes para a efetividade da censura automática.

Em termos de aplicações, classificadores de imagens pornográficas podem ser utilizados em mecanismos de controle parental, gerando predições para imagens avulsas ou, no caso de vídeos, para sequências de *frames*. Foram apresentados dois testes práticos onde mídias contendo pornografia (tanto imagens quanto vídeo), identificados como tal por modelos treinados com *DataSex*, foram publicados sem qualquer restrição em redes sociais, neste caso Tumblr e YouTube. As redes disponibilizaram o conteúdo sem qualquer censura, mesmo contendo pornografia.

Dado que as predições geradas pelos classificadores apresentados resultam somente nos *scores* para as classes, não dispondo de qualquer referência espacial que permita direcionar uma ação para uma região específica, as soluções decorrentes de uma predição positiva ficam restritas a remoção ou a descaracterização completa da imagem ou *frame*. Neste caso, métodos que pudessem apontar regiões de interesse abririam espaço para a criação de mecanismos de censura automática mais elaborados e, especialmente, menos invasivos.

3. DETECÇÃO DE PARTES ÍNTIMAS

A censura automática de conteúdo pornográfico observada como um problema de classificação de imagens é factível. Resultados apontados na Seção 2.5.2 desta tese, listados na Tabela 2.2, demonstram que modelos de classificação treinados com o *dataset DataSex* atingem acurácia $\approx 0,99$, permitindo identificar e filtrar imagens ou *frames* de vídeos antes mesmo que estes sejam visualizados. Por outro lado, trata-se de uma abordagem intrusiva onde cada imagem terá de ser completamente removida, resultando em nítidas lacunas de conteúdo.

Dentre suas diferentes aplicações reportadas na literatura, Redes Neurais Convolucionais apresentam resultados promissores em tarefas de detecção de objetos [RHGS15, Gir15, RF16, RDGF16, LAE+16]. Avaliados sobre *datasets* generalistas como MS COCO [LMB+14], PASCAL VOC [EEVG+15] e *Open Images* [KDA+], os métodos baseados em ConvNets definem o atual estado-da-arte para esta tarefa. Além da precisão de suas predições, alguns destes métodos são capazes de processar imagens em tempo real, permitindo a análise de vídeos quadro-a-quadro. A partir destas características constitui-se a hipótese de que a detecção de objetos baseada em ConvNets pode constituir um avanço para a geração automática de censura à pornografia, permitindo cobrir ou descaracterizar partes íntimas do corpo e contornar as restrições inerentes à classificação de imagens, evitando a remoção completa de imagens classificadas como pornográficas.

Após analisar o problema utilizando métodos de classificação, seguindo seus objetivos, esta tese abordou a censura automática de conteúdo pornográfico como um problema de detecção de objetos. Dado que os métodos de detecção de objetos permitem identificar regiões específicas relacionadas a elementos de interesse, a censura baseada nestes métodos pode resultar em intervenções menos intrusivas, como a aplicação de descaracterizações ou tarjas sobre as partes sensíveis detectadas, evitando a remoção total de uma imagem ou *frame*.

Este capítulo estuda a geração de censuras automáticas de partes íntimas relacionadas à pornografia. Em função da necessidade de fontes de dados rotulados para o treinamento de modelos preditivos, inicialmente é apresentando *Dataset for Pornography Censorship* (*DPC*), o primeiro *dataset* de conteúdo pornográfico que rotula partes íntimas do corpo para treinamento de modelos de detecção de objetos. *DPC* é composto por 3.000 imagens que registram mais de 6.000 objetos anotados para 4 classes que relacionam partes íntimas do corpo: i) *butt*, ii) *breast*, iii) *frontalM* e iv) *frontalF*. *DPC* foi utilizado para treinar e comparar diferentes métodos de detecção de objetos na tarefa de detecção de partes pornográficas, onde o mesmo protocolo de treinamento e avaliação foi aplicado para todos os métodos avaliados. São apresentados experimentos com métodos baseados em ConvNets que definem o atual estado-da-arte para a tarefa de detecção de objetos, sendo eles: i) *Faster Region-based Convolutional Network* (*Faster* R-CNN) [RHGS15], ii) *Single Shot MultiBox Detector* (SSD) [LAE+16], e iii) *You Only Look Once* (YOLO) [RDGF16]. Também são descritas e apresentadas arquiteturas construídas especificamente para geração de predições em tempo real, compatíveis com o processamento de imagens e vídeos.

Mesmo os experimentos demonstrando que modelos baseados em *Faster* R-CNN atingem métricas de precisão superiores, constatou-se que estes modelos apresentam desempenho em função do tempo inferiores aos modelos baseados em métodos mais simples, como YOLO e SSD. A partir desta observação, esta tese baseou-se em YOLO para construir modelos de detecção com melhores resultados preditivos, explorando características do método que pudessem contribuir para melhores resultados, especificamente sobre conteúdo pornográfico. Para simplificar a elaboração dos experimentos, especialmente para a adição de rotinas de carga e aumento de dados, o método YOLO foi plenamente portado para a plataforma Keras/Tensorflow [C$^+$, AAB$^+$], utilizando linguagem Python.

3.1 Detecção de Objetos

Visão Computacional é uma área interdisciplinar que estuda como computadores podem contribuir em tarefas de alto nível envolvendo imagens e vídeos, automatizando demandas executadas por humanos [BB82]. Detecção de Objetos é uma tarefa de Visão Computacional que lida com a detecção de instâncias de objetos relacionados a uma classe em imagens ou vídeos digitais. Domínios bem pesquisados para tarefas de detecção de objetos incluem detecção de pedestres [TLWT15, ZBS$^+$15, ZLLH16], veículos [Li17, BDM$^+$16], sinais de trânsito [GSW$^+$18], dentre outros. Diferentes áreas de aplicação podem ser beneficiadas por métodos para detecção de objetos, como por exemplo: recuperação de imagens, sistemas de vigilância por vídeo-monitoramento, controle automático de tráfego e controle parental, neste último, podendo identificar a ocorrência de elementos relacionados à pornografia.

Diferentemente da tarefa de classificação, que resulta em probabilidades de uma imagem como um todo pertencer a uma determinada classe, os métodos de detecção de objetos resultam em n áreas de interesse, formadas por parâmetros de localização espacial (x, y) e dimensional (w, h) no espaço de uma imagem I, atribuindo para os n objetos uma probabilidade P_C onde C representa cada possível classe.

Anteriormente às ConvNets, a detecção de objetos era baseada em *features* extraídas a partir de regras implementadas por métodos como SIFT [Low04], SURF [BTVG06] e HOG [DT05]. Nos métodos baseados em ConvNets, as *features* são definidas com base em modelos induzidos a partir dos dados, tornando-as mais expressivas que aquelas obtidas pelos métodos *handcrafted*, contribuindo também para a tarefa de detecção de objetos. Os dados utilizados para indução de modelos de detecção de objetos são conjuntos de imagens onde todos os objetos de interesse devem ser previamente anotados, indicando posição, dimensão e classificação de cada objeto.

Algumas estratégias comuns aos métodos clássicos para detecção de objetos continuam contribuindo para melhorar os resultados, mesmo em técnicas baseadas em ConvNets. O aumento de dados cria instâncias sintéticas partindo de variações controladas de instâncias reais, gerando diversidade que contribui para modelos mais robustos a *overfitting*. O treinamento multi-escala gera modelos flexíveis, contribuindo para a detecção de objetos menores com relação aos demais, além

de constituir uma forma de aumento de dados. A predefinição de âncoras diminui instabilidades no treinamento de modelos ao evitar que valores lineares tenham de ser inferidos sem qualquer referência inicial. *Non-Maximal Suppression* (NMS) reduz a quantidade de predições conflitantes para um mesmo objeto. Finalmente, as medidas de avaliação para detecção de objetos, como *Mean Average Precision* (mAP) e *Intersection over Union* (IoU), continuam definindo os padrões para avaliação de desempenho preditivo em desafios de detecção de objetos [LMB+14, EEVG+15]. A seguir será formalizada a tarefa de detecção de objetos, passando ao detalhamento das estratégias aqui apontadas.

3.1.1 Definição do Problema

Pelo melhor que se sabe, esta é a primeira iniciativa que emprega detecção de objetos para censurar conteúdo pornográfico. Dada a detecção das áreas relacionadas à pornografia, é possível aplicar diferentes tipos de descaracterizações na imagem, como tarjas, desfoques ou borrões. Formalmente, o objetivo é aproximar uma função $f(I) = \{C, O_P\}$ onde I é uma imagem, $C = \{c_1, c_2, c_3, ..., c_{P_C}\}$ é um conjunto de classes para P_C partes do corpo detectadas que precisam ser censuradas, alinhadas com o vetor de objetos preditos $O_P = \{\mathbf{o_1}, \mathbf{o_2}, \mathbf{o_3}, ..., \mathbf{o_{P_C}}\}$. Cada predição representa coordenadas espaciais (x, y) e dimensionais (w,h). Para os casos onde a imagem I não apresenta qualquer objeto pornográfico, $f(I) = \{\emptyset, \emptyset\}$.

3.1.2 Aumento de Dados

Segundo Wong et al. [WGSM16], o aumento de dados contribui para o treinamento de modelos, atuando como regularizador na prevenção do *overfitting* em redes neurais, contribuindo para melhorar o desempenho em problemas onde os dados sejam escassos ou onde as classes não sejam equilibradas. Ainda de acordo com Wong et al. [WGSM16], diferentes métodos de classificação baseados em ConvNets [KSH12, SZ14, SLJ+15, HZRS16] apresentaram melhores resultados quando utilizaram aumento de dados.

Métodos específicos para detecção de objetos baseados em ConvNets utilizam estratégias de aumento de dados [LAE+16, RDGF16, RHGS15]. Para detecção de objetos, as estratégias de aumento de dados contribuem para a indução de modelos mais robustos a partir de variações sintéticas dos formatos, tamanhos, cores e proporções dos objetos reais presentes nos dados de treinamento. Durante o treinamento, essas variações são aplicadas nas imagens amostradas, ajustando arbitrariamente a intensidade de cada variação. Intuitivamente, toda variação espacial ou dimensional aplicada à imagem em tempo de execução precisa ser exatamente reproduzida sobre os rótulos que apontam os objetos presentes nestas imagens.

Dentre as variações mais utilizadas para aumento de dados em problemas de detecção de objetos destacam-se: i) a escala, ii) a instabilidade de proporções (conhecida como *jitter*), iii) as

(a) escala. (b) *jitter*. (c) iluminação e balanço de cores.

Figura 3.1: Exemplos de variações comuns utilizadas para aumento de dados em detecção de objetos.

variações de iluminação e balanço de cores, iv) os recortes arbitrários (também conhecidos como *crops*) e v) o espelhamento vertical/horizontal arbitrário (*flips*). A Figura 3.1 representa 3 amostras de aumento de dados variando intensidades onde: i) a escala assume valores $= \{.5, .75, 1.0\}$ (Figura 3.1a), variando dimensões de largura e altura proporcionalmente, ii) aplica-se *jitter* $= \{.25, .5, .75\}$ (Figura 3.1b), onde a variação da proporção entre largura e altura não é preservada e, finalmente iii) a iluminação e balanço de cores, onde a Figura 3.1c apresenta amostras que variam em $\{.75, .1, .25\}$ os percentuais de ajuste nestas características.

(a) imagem original. (b) amostras aumentadas.

Figura 3.2: Aumento de dados gerado sobre uma imagem anotada para detecção de objetos.

A Figura 3.2 ilustra 6 exemplos de aumento de dados (3.2b) sobre uma mesma imagem (3.2a), onde pode-se observar o efeito de todas as 5 varições de modificação relatadas. Variações de escala, quando resultam em uma imagem menor que o volume de entrada da rede, podem ser envelopadas em posições arbitrárias de uma imagem com coloração neutra ($RGB \approx [127, 127, 127]$), resultando nas porções em tons de cinza presentes em 5 dos 6 exemplos da Figura 3.2b. Já nos casos onde a variação de escala gerar imagens maiores que o volume de entrada, aplica-se o recorte em posição arbitrária de maneira que todo o volume de entrada seja coberto pelo recorte. Nestes casos, ilustrado em 1 dos 6 exemplos da Figura 3.2b, o recorte cobre todo o volume de entrada e não haverá porção cinza aparente.

Por se tratar da aplicação de aumento de dados em um problema de aprendizado supervisionado, toda a modificação espacial produzida em uma imagem de treinamento deve refletir precisamente sobre cada objeto anotado desta imagem. A Figura 3.2 ilustra a reprodução dos ajustes de aumento de dados nos objetos anotados por meio de caixas proporcionalmente desenhadas sobre cada objeto. Quando, em função do aumento de dados, um determinado objeto não permanece na imagem correspondente ao volume de entrada, este objeto deve ser removido do conjunto de ano-

tações. Comparando as Figuras 3.2a e 3.2b é possível perceber que as pessoas presentes na porção esquerda superior da imagem original (3.2a) são completamente excluídas em 2 dos exemplos após aumento de dados (3.2b).

3.1.3 Treinamento em múltiplas escalas

Arquiteturas totalmente convolucionais não restringem dimensões específicas ao volume de entrada. As camadas totalmente conectadas vinculam multiplicações de parâmetros com as dimensões espaciais do volume propagado na rede, característica não observada em camadas convolucionais. Camadas convolucionais compartilham um conjunto de parâmetros que é aplicado parte a parte por todo o espaço do volume. Arquiteturas totalmente convolucionais geram saídas que variam em função das dimensões da entrada, já que a saída de cada camada convolucional segue o padrão definido na Equação 3.1, onde D_E representa uma dimensão do volume de entrada, D_F uma dimensão de um filtro convolucional, Z_P uma borda de preenchimento formada por zeros (conhecida como *zero padding*), S o tamanho do salto de deslocamento do filtro sobre o volume de entrada (*stride*) e, finalmente, D_S representa a dimensão de saída da camada.

$$D_S = \left(\frac{D_E + 2Z_P - D_F}{S}\right) + 1 \tag{3.1}$$

A possibilidade de utilizar entradas com dimensões variáveis contribui para a geração de modelos invariantes espacial e dimensionalmente, características determinantes para detecção de objetos. O método YOLO [RDGF16], ao utilizar uma rede totalmente convolucional construída com base na arquitetura Darknet-19 [RF16], utiliza o treinamento multi-escala. Sua função de custo é agnóstica ao fator dimensional, comparando predições e anotações de objetos reais de maneira análoga, independentemente das dimensões utilizadas no volume de entrada.

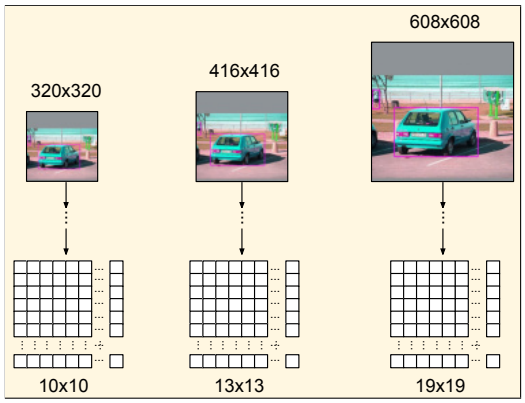


Figura 3.3: Impacto da variação de escala do volume de entrada na saída de uma rede totalmente convolucional baseada na arquitetura Darknet-19, utilizada para detecção de objetos.

A Figura 3.3 representa 3 exemplos de variação de dimensões de entrada para uma rede totalmente convolucional composta por 5 camadas, onde cada camada reduz as dimensões do volume

de entrada pela metade. Seguindo as dimensões definidas pelos exemplos, as saídas geradas por esta arquitetura seguirão sempre a proporção $D_E/2^5$.

3.1.4 Predefinição de proporções com Âncoras

Conforme relatado em [RF16], inicialmente o método YOLO era suscetível a instabilidade de gradientes que poderia inviabilizar modelos em treinamento. Esta instabilidade era potencializada pela estratégia adotada no método original [RDGF16], que assumia limites arbitrários para as dimensões dos objetos preditos, especialmente no início do treinamento. Em alguns casos, as predições iniciais poderiam ser muito distantes dos valores reais, impactando diretamente nos gradientes. Até chegar a estabilidade, a estratégia arbitrária fazia com que a rede oscilasse entre predições coerentes e predições muito distantes dos objetos reais. Muitos treinamentos resultavam em *overflow* de gradientes, forçando o início de um novo treinamento. Em outros casos a instabilidade persistia por algumas épocas, chegando ao equilíbrio e seguindo o treinamento até a convergência.

Objetos do mundo real não são arbitrários, sendo possível assumir padrões aproximados de suas dimensões e evitar os problemas de oscilação relatados em [RF16]. Assim, o método YOLO incorpora uma estratégia já utilizada por outros métodos de detecção, como [RHGS15], conhecida como âncoras de predefinição. As âncoras predefinem aproximações realistas dos limites que as predições de objetos devem assumir, sendo calculadas com base nos objetos reais anotados nos *datasets* utilizados para treinamento dos modelos.

Ao aplicar âncoras no treinamento de modelos, no lugar de gerar 5 predições de objetos com dimensões arbitrárias, o método toma as âncoras como base para que as predições passem a ser fatores de ajuste para proporções predefinidas. Neste caso, os valores preditos passam por ativação exponencial, limitados entre $(0, \infty)$, transformando-os em fatores de ajuste que agem sobre dimensões proporcionais aos objetos reais, diminuindo a arbitrariedade observada no método original.

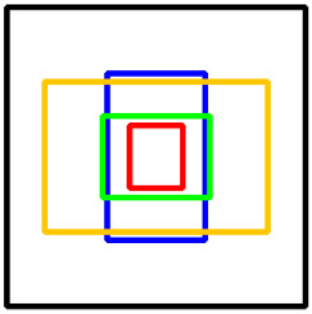


Figura 3.4: Âncoras geradas para um *dataset* de detecção de partes íntimas do corpo.

As âncoras devem representar o mais aproximadamente possível as dimensões e as proporções dos objetos reais presentes no *dataset* de treinamento. O método adotado para gerar essas âncoras é semelhante, tanto em [RDGF16] quanto em [RHGS15], baseando-se no agrupamento dos

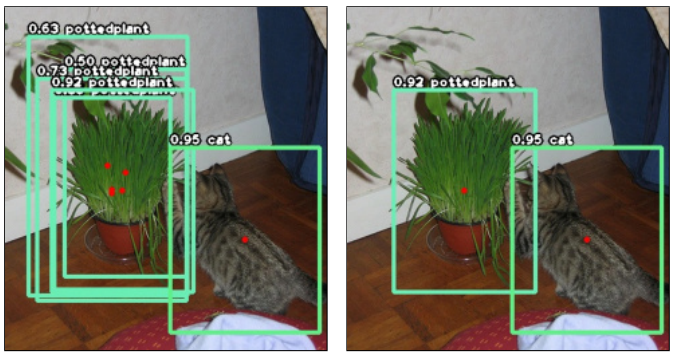
formatos dos objetos conhecidos. A Figura 3.4 demonstra as proporções para as 5 âncoras geradas a partir de *Dataset for Pornography Censorship* (*DPC*), um *dataset* para detecção de partes íntimas do corpo desenvolvido no contexto desta tese. As 5 âncoras, diferenciadas pelas cores vermelho, verde, azul, laranja e preto, estão centralizadas em um mesmo ponto e ilustram as proporções de largura e altura mais frequentes no *dataset*.

$$boxDist(obj, ctrd) = 1 - IoU(obj, ctrd) \tag{3.2}$$

O método de clusterização para geração das âncoras aplica K-means [Llo82], onde o valor de k é definido de acordo com o número de detectores desejado para o modelo. Para calcular os agrupamentos, todos os objetos são centralizados nas coordenadas $(0, 0)$ e k centroides são inicializados arbitrariamente. Cada objeto do *dataset* é associado a um dos Q_{ctrd} centroides, usando a Equação 3.2 como medida de distância, onde obj representa um objeto e $ctrd$ o centroide associado a este objeto, resultando em Q_{ctrd} agrupamentos de objetos. Os centroides são recomputados, buscando minimizar a medida de distância $boxDist$ com relação aos objetos já existentes no agrupamento. Após a convergência, os Q_{ctrd} centroides definem Q_{ctrd} âncoras que representam as proporções dimensionais dos objetos reais anotados no conjunto de treinamento do *dataset* utilizado, podendo contribuir para o treinamento dos modelos de detecção.

3.1.5 *Non-Maximal Suppression*

Apontado por Neubeck et al. [NVG06] como uma etapa de preprocessamento fundamental para diversas tarefas de processamento de imagens no contexto de detecção de objetos, *Non-Maximal Suppression* (NMS) representa a supressão das predições sobrepostas para um único objeto. Métodos de detecção de objetos baseados em ConvNets como YOLO [RDGF16], *Faster* R-CNN [RHGS15] e SSD [LAE+16] utilizam NMS para reduzir predições sobrepostas, sendo aplicado após a geração de todas as predições, somente sobre aquelas que forem compatíveis com o limiar de confiança adotado.



(a) $L_{IoU} = 0,8$ (b) $L_{IoU} = 0,3$

Figura 3.5: Exemplo de aplicação de NMS, removendo 4 predições sobrepostas.

A Figura 3.5 exemplifica a ocorrência de múltiplas predições sobrepostas sobre um mesmo objeto, apontando a mesma classe. No exemplo, existem 2 objetos que correspondem às classes *pottedplant* e *cat*, sendo ambos detectados. Por outro lado, como pode ser observado na Figura 3.5a, o detector gera 5 diferentes predições para o objeto da classe POTTEDPLANT com confiança superior ao limiar, neste caso $= 0,30$. Seguindo o exemplo, a aplicação de NMS relaciona todas as predições que apresentam IoU superior a um determinado limiar de sobreposição (L_{IoU}) e, ao mesmo tempo, apontem para a mesma classe. Todas as predições relacionadas são descartadas, exceto aquela que apresenta maior *score* de confiança, resultando na única predição mantida para o objeto. A Figura 3.5b exibe a plotagem das mesmas predições, na mesma imagem, após a aplicação de NMS, onde 3 das 4 predições para POTTEDPLANT com *scores* $= \{.5, .63, .68, .73\}$ foram suprimidas, restando apenas a predição com maior *score* de confiança ($= .92$). A predição para a classe *cat* não apresenta nenhuma sobreposição e manteve-se inalterada.

3.1.6 Medidas de Desempenho

Uma medida de desempenho precisa traduzir o quão adequada está sendo desempenhada uma determinada tarefa em um valor numérico. O problema de detecção de objetos apresenta características particulares, exigindo métricas que contemplem diversos elementos. Diferente de um problema de classificação de imagens, como relatado no Capítulo 2, onde assume-se que toda a imagem pertence a uma classe e que as medidas de desempenho, como a acurácia, comparam diretamente predito e real, na detecção de objetos esta relação 1×1 torna-se $n \times m$. Uma imagem pode conter um número indefinido de objetos, inclusive nenhum, e as predições que detectam estes possíveis objetos podem não estar perfeitamente alinhadas com os reais limites destes possíveis objetos. Além das particularidades de alinhamento espacial, detectores de objetos precisam apontar a qual classe um determinado objeto pertence, dada a quantidade de classes do *dataset* utilizado.

Desafios de detecção de objetos, como PASCAL VOC [EEVG+15] e MS COCO [LMB+14], consolidaram métricas comuns para esta tarefa. A seguir serão descritas as métricas *Intersection over Union* (IoU), *Average Precision* (AP) e *Mean Average Precision* (mAP), que constituem o conjunto de métricas fundamentais para avaliação desta tarefa.

3.1.6.1 Intersection over Union

Utilizada como medida de desempenho em praticamente todas as iniciativas para segmentação semântica de imagens [LSD15, HGDG17] e detecção de objetos [RDGF16, LAE+16, RHGS15], *Intersection over Union* (IoU) calcula a intersecção entre os limites de uma predição e os limites de um objeto real presente em uma imagem, dividindo a intersecção pela união dos mesmos limites. A Equação 3.3 formaliza a medida IoU, onde r representa a anotação real e p representa um objeto predito.

$$IoU(r,p) = \frac{area(r \cap p)}{area(r \cup p)} \tag{3.3}$$

O uso de IoU é comum tanto em abordagens de detecção de objetos, apontando predições que limitam a área de cada objeto, quanto para abordagens pixel-a-pixel utilizadas em segmentação semântica de objetos. Em ambas abordagens é utilizada a área de cada conjunto, independentemente de seu formato. A divisão pela união naturalmente penaliza as predições que apontam grandes áreas que extravasam os limites dos objetos reais como estratégia para obter bons resultados. A Figura 3.6 ilustra um objeto real (azul) e uma predição (vermelho), graficamente relacionados para cálculo da IoU, destacando a divisão de suas áreas de intersecção por união. Somente predições totalmente ajustadas ao objeto real terão IoU $= 1,0$, qualquer diferença de tamanho, proporção ou posicionamento impactará negativamente no valor, tendendo a zero.

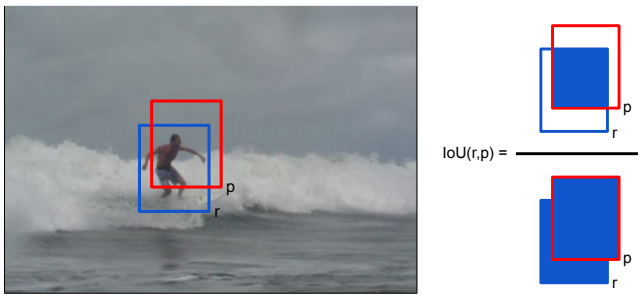
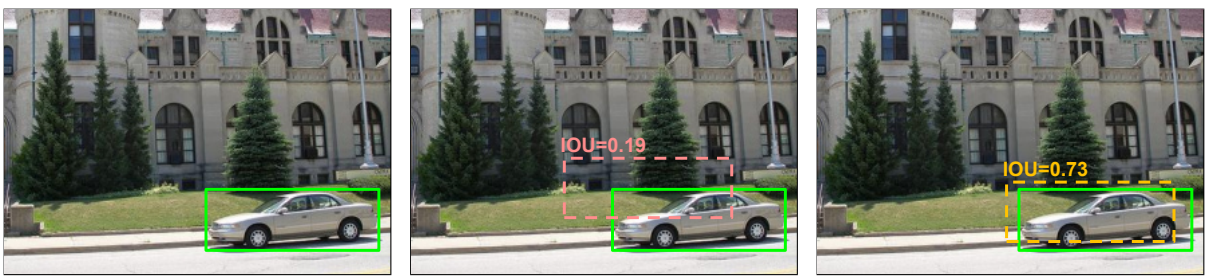


Figura 3.6: Relação entre objeto real e predição para cálculo de IoU.

3.1.6.2 Mean Average Precision

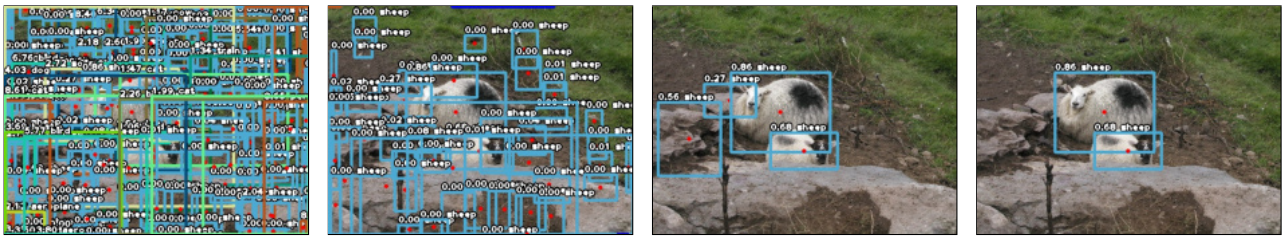
Mean Average Precision (mAP) é uma métrica utilizada para avaliação de modelos preditivos para detecção de objetos [EVGW+10, EEVG+15, LMB+14]. Dada a complexidade da tarefa, mAP relaciona diferentes métricas, sendo elas precisão, sensibilidade e IoU, resultando em uma *Average Precision* (AP) para cada classe de objetos presente no cenário avaliado, definido pelo número de classes N_C existentes no *dataset* utilizado. A média calculada entre as N_C APs representa o valor final de mAP.



(a) F_N. (b) F_P e F_N. (c) V_P.

Figura 3.7: Exemplos de predições (linha pontilhada) sobre um objeto real (linha contínua).

As medidas de precisão e sensibilidade, descritas pelas Equações 2.6 e 2.7, dependem das quantidades de verdadeiros positivos (V_P), falsos positivos (F_P) e falsos negativos (F_N). A predição de um objeto é um V_P quando está alinhada com um objeto real, apresentando $\text{IoU} >= 0,5$, e classificação compatível com a classe do objeto real. Uma predição é um F_P quando $\text{IoU} < 0,5$, quando a classe não corresponde ao objeto real ou quando ocorrem predições duplicadas. F_N são os objetos reais que não são apontados por nenhuma predição. A Figura 3.7 apresenta exemplos para V_P, F_P e F_N, ilustrando também a variação de IoU para duas predições de um mesmo objeto, presentes nas Figuras 3.7b e 3.7c, onde é possível perceber o impacto que os desencontros entre predito e real geram na métrica.



(a) $L_C = 0,00001$. (b) $L_C = 0,001$. (c) $L_C = 0,1$. (d) $L_C = 0,6$.

Figura 3.8: Plotagem de predições variando crescentemente o limiar de confiança (L_C).

A composição da medida AP é dada pela média dos valores de precisão observados em pelos menos 11 pontos de sensibilidade, obtendo um AP relacionado a cada classe. Destes valores derivam curvas precisão/sensibilidade que são construídas por meio da interpolação dos diferentes pontos de sensibilidade, obtidos ao variar o limiar de confiança (L_C) que aponta se uma predição é válida, partindo de $0,0$ até $1,0$. Quando o limiar de confiança é definido como 0, o número de predições válidas será exatamente o número de predições geradas pelo modelo de detecção, caindo de acordo com o crescimento do limiar. Variações do limiar de confiança impactam diretamente sobre as contagens de V_P, F_P e F_N. A Figura 3.8 plota as predições geradas por um mesmo modelo de detecção, para a mesma imagem de entrada, utilizando 4 valores crescentes como limiares de confiança, partindo de ≈ 0. Conforme L_C cresce, o número de predições plotadas diminui, no entanto, diminui também o número de F_p. Por outro lado, seguindo o exemplo da Figura 3.8, quando $L_C > 0,68$ haverá queda em V_P e crescimento de F_N, já que um objeto real deixará de ser plotado.

Na forma tradicional, a definição do número de pontos de precisão/sensibilidade considerados para plotar a curva assumia 11 valores de sensibilidade, distribuídos igualitariamente no intervalo $[0,0; 1,0]$, resultando especificamente em $[0,0; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0]$. A partir do desafio PASCAL VOC [EEVG+15], a definição dos valores passou a compreender todas as variações de sensibilidade observadas na distribuição, calculadas a partir da variação do limiar de confiança (L_C) das predições. Esta modificação resultou em uma medida mais precisa, permitindo comparar modelos que geram resultados semelhantes.

A interpolação da curva precisão/sensibilidade adota uma estratégia para reduzir o impacto das oscilações geradas pela variação do limiar de confiança L_C. O limiar de confiança L_C

define as predições aceitas com base na confiança que o modelo aponta para que uma determinada predição realmente corresponda a um objeto. Predições com confiança $< L_C$ são descartadas, não impactando nos cálculos de Pr e Se. A estratégia consiste em assumir sempre a maior precisão medida a partir da maior sensibilidade da distribuição até então observada. A Equação 3.4 define a precisão interpolada ($Pinterp$) de um ponto na curva para um nível de sensibilidade Se como a maior precisão observada $\forall \tilde{Se} \geq Se$. A partir das precisões interpoladas, a Equação 3.5 representa o cálculo da *Average Precision* (AP), assumindo a média das $Pinterp$ observadas ao variar os $num(L_C)$ no modelo. Por outro lado, a abordagem atual para a medida AP, ilustrada pela Equação 3.6, assume a área sob a curva precisão/sensibilidade calculada com base na integração numérica de todos os $num(L_C)$ pontos de sensibilidade observados. A Figura 3.9 representa a curva precisão/sensibilidade interpolada sob predições de um dado modelo, registrando $AP = 0,6325$ para a classe TVMONITOR do *dataset* PASCAL VOC [EVGW+], aplicando o cálculo de AP com base na Equação 3.6. A área destacada em azul claro representa AP interpolada, enquanto a linha azul escura liga os pontos precisão/sensibilidade antes da interpolação. A leitura do gráfico, da esquerda para a direita, representa o decremento do limiar de confiança, partindo de $1,0$ até chegar em 0. Quanto maior o limiar L_C, maior a precisão e menor a sensibilidade das predições geradas. Na prática, predições com confiança $\approx 1,0$ tendem a estar corretas, diminuindo F_p. Por outro lado, quando L_C tende a $0,0$, todas as predições tornam-se válidas, diminuindo F_n.

$$Pinterp(Se) = \underset{\tilde{Se}:\geq Se}{max}\ p(\tilde{Se}) \tag{3.4}$$

$$\text{AP} = \frac{1}{num(L_C)} \sum_{Se \in L_C} Pinterp(Se) \tag{3.5}$$

$$\text{AP} = \sum_{i=1}^{num(L_C)-1} (Se_{i+1} - Se_i) \times Pinterp(Se_{i+1}) \tag{3.6}$$

$$\text{mAP} = \frac{1}{N_C} \sum_{i=1}^{N_C} \text{AP}_i \tag{3.7}$$

Finalmente, a medida mAP resulta em uma métrica geral de desempenho do detector, onde são considerados os valores de AP para todas as classes representadas pelo *dataset*. A Equação 3.7

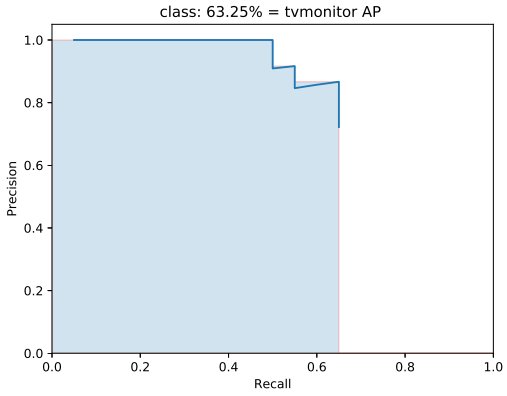


Figura 3.9: Curva Precisão/Sensibilidade interpolada.

descreve a medida para uma quantidade de classes N_C, onde AP_i representa o valor de AP para a i-ésima classe do *dataset*.

3.2 *Dataset for Pornography Censorship* (*DPC*)

Métodos de aprendizado supervisionado são fundamentalmente dependentes de dados previamente rotulados. Em se tratando de *datasets* anotados para tarefas de detecção de objetos contendo classes relacionadas à pornografia, a literatura não oferece alternativas. Para o problema de classificação de imagens pornográficas, por outro lado, esta tese criou *DataSex*, um *dataset* composto pelas classes *free* e *porn*, apresentado por Simões et al. [SWP+16]. Este *dataset*, já descrito na Seção 2.3.3, é o maior conjunto de imagens rotuladas para classificação binária de imagens pornográficas, contendo 286.920 imagens distribuídas igualitariamente entre as classes *free* e *porn*. Experimentos utilizando *DataSex* atingiram resultados de classificação com acurácia de teste $= 0,9906$ (reportados na Tabela 2.2) ao ajustar finamente uma ConvNet baseada em Darknet-19 [RF16], pretreinada com Imagenet [DDS+09]. Mesmo permitindo o treinamento de modelos de classificação acurados, *DataSex* não dispõe de anotação em nível de objetos, impedindo o treinamento de detectores de partes relacionadas à pornografia.

Para avaliar o tratamento do problema de geração automática de censuras de partes íntimas do corpo relacionadas à pornografia como uma tarefa de detecção de objetos, esta tese apresenta *Dataset for Pornography Censorship* (*DPC*). *DPC* é um *dataset* projetado para detecção de objetos composto por 3.000 imagens contendo exposição de partes íntimas. Cada imagem contém pelo menos 1 objeto relacionado a umas das seguintes classes: *butt*, *breast*, *frontalM* ou *frontalF*. *DPC* é dividido em conjuntos de treino (2.100 imagens), teste (600) e validação (300), contendo uma ampla variedade de elementos como:

- variação de escala/tamanho, com imagens variando entre 170 até 3.000 pixeis em largura e altura;
- variação de condições de iluminação;
- variação de composição de cenas; e
- variação de etnia dos participantes.

Para construir *DPC*, foi selecionada uma amostra do *dataset DataSex*. As imagens selecionadas foram anotadas para a tarefa de detecção de objetos passando por 2 passos: i) anotação e ii) revisão. No primeiro passo, as imagens foram divididas em 4 grupos, cada um deles atribuído a uma pessoa responsável por anotar cada objeto presente em cada imagem. No segundo passo, as anotações passaram por um processo de revisão cruzada, onde os 4 grupos de imagens foram intercambiados entre os anotadores. O processo completo de anotação levou aproximadamente 30 dias para ser concluído e exigiu a construção de uma ferramente de anotação.

Em função do assunto abordado por este *dataset*, partes íntimas do corpo, a área total dos objetos anotados é basante reduzida. A título de comparação, PASCAL VOC [EEVG+15] apresenta em média $20,8\%$ da área das imagens anotadas como objetos, enquanto que em *DPC* esta área corresponde a $\approx 11,8\%$. Ao todo, 6.500 objetos foram anotados manualmente, resultando em uma média de $3,4$ objetos por imagem (sendo pelo menos 1 e ao máximo 11 objetos por imagem). A Tabela 3.1 exibe a distribuição de objetos anotados e a quantidade de imagens relacionadas a cada classe presente no *dataset*;

Tabela 3.1: Distribuição das partes do corpo anotadas em *DPC*.

	butt	*breast*	*frontalM*	*frontalF*	total
objetos anotados	1.200	2.693	1.265	1.383	6.541
imagens	1.122	1.537	1.134	1.335	3.000

Tabela 3.2: Dimensões dos objetos anotados em *DPC*.

	butt	*breast*	*frontalM*	*frontalF*
relativas aos objetos	$16,96\%$	$3,35\%$	$7,15\%$	$4,66\%$
relativas às imagens onde ocorrem	$18,14\%$	$5,86\%$	$7,98\%$	$4,83\%$
relativas à área de todas as imagens	$6,79\%$	$3,00\%$	$3,02\%$	$2,15\%$

As dimensões médias das 3.000 imagens que compõe *DPC* apresentam largura $= 534$ e altura $= 525$, correspondendo a uma área média de 322.411 pixeis. A Tabela 3.2 apresenta percentuais que indicam a área referente aos objetos anotados para as 4 classes de *DPC* relativas i) aos objetos da própria classe, ii) à área de todas as imagens onde estes objetos acontecem e iii) à área de todas as imagens presentes no *dataset*.

DPC foi utilizado para treinar modelos de detecção de partes íntimas compatíveis com YOLO [RDGF16], SSD [LAE+16] e *Faster* R-CNN [RHGS15]. Foram calculadas âncoras de predefinição de proporções seguindo o método disposto em [RF16], ilustradas pela Figura 3.4. Os modelos treinados com *DPC* atingiram mAP $= 0,6961$, gerando censuras automáticas ajustadas às partes íntimas reais anotadas nas 900 imagens de validação e teste.

3.2.1 Ferramenta para anotação de objetos

A tarefa de anotação manual de imagens para detecção de objetos é lenta e detalhista. É preciso atenção e cuidado para gerar anotações ajustadas e padronizadas, já que métodos supervisionados são fundamentalmente dependentes destes dados para induzir modelos acurados. Para agilizar a anotação das 3.000 imagens de *DPC*, foi construída uma ferramenta gráfica baseada em teclas de atalho e poucos cliques de *mouse*, que contribuiu para a conclusão do trabalho em tempo razoável.

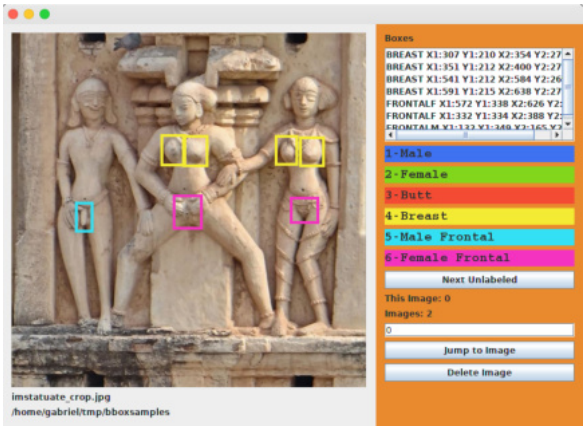


Figura 3.10: Ferramenta construída no contexto desta tese para anotação dos objetos presentes em *DPC*.

A ferramenta manteve as opções focadas somente nas classes relacionadas ao tema abordado pela tese, disponibilizando teclas de atalho objetivas para todas as tarefas. A definição da localização e área de cada objeto é feita com o uso do *mouse*, sendo possível desenhar retângulos em qualquer proporção e em qualquer quantidade. Caso uma anotação registrada tenha de ser removida, é possível selecionar cada uma por meio de uma listagem que relaciona todas as anotações da imagem corrente. O padrão de saída gerado pela ferramenta é compatível com o formato PASCAL VOC [EEVG+15].

A Figura 3.10 ilustra a ferramenta desenvolvida para anotação de objetos. No exemplo, uma imagem de uma escultura representando 3 pessoas tem anotações para 4 objetos da classe *breast*, 2 da classe *frontalF* e 1 da classe *frontalM*. A listagem dos objetos anotados está disposta no canto direito superior e as legendas das classes e seus respectivos botões numéricos de atalho aparecem abaixo da lista de anotações.

3.3 Arquiteturas para Censura de Pornografia

A censura de conteúdo pornográfico abordada como um problema de detecção de objetos requer modelos preditivos acurados e rápidos o suficiente para processar imagens em tempo real. Nesta tese, chama-se de 'tempo real' os métodos capazes de gerar predições para pelo menos 24 imagens por segundo, taxa tradicionalmente utilizada por películas de cinema. Mesmo relativamente baixa, poucos modelos de detecção são capazes de atingir estas taxas mantendo razoável desempenho preditivo. Nesta seção, será apresentado um comparativo entre diferentes arquiteturas de rede e métodos de detecção de objetos, onde modelos para detecção de partes íntimas relacionadas à pornografia foram treinados utilizando o *dataset DPC*. Os resultados gerados por este experimento balizaram a escolha dentre um dos métodos para maiores estudos.

Durante o comparativo, foram treinados modelos utilizando os métodos YOLO [RDGF16], SSD [LAE+16] e *Faster* R-CNN [RHGS15], seguindo um protocolo experimental equivalente, onde os parâmetros variaram em função de restrições dos métodos. Nesta etapa, também foi proposta

uma arquitetura alternativa de ConvNet para geração de predições rápidas e precisas para censurar conteúdo pornográfico, nomeada *CensorNet*. Os métodos de detecção de objetos avaliados no comparativo utilizam estratégias distintas, algumas delas complexas e lentas, impedindo sua aplicação em problemas de tempo real. YOLO, por outro lado, aplica uma abordagem simples para transformar a tarefa de detecção de objetos em um problema de regressão, baseado em uma única ConvNet. Com uma única propagação do volume através desta única ConvNet, YOLO gera predições acuradas em tempo real.

3.3.1 Arquiteturas Avaliadas

A arquitetura *CensorNet* foi construída com base em blocos que intercalam camadas de convolução, *pooling* e normalização, formando duas configurações de blocos referidas como *ConvBlock* e *SepBlock*. Os *ConvBlocks* adicionam convoluções com filtros de dimensões 1×1, reduzindo pela metade a quantidade de mapas de ativação propagados. Esta abordagem reduz o número de parâmetros e aumenta a não linearidade do modelo. Todas as outras camadas convolucionais empregam filtros de dimensões 3×3. Como notação, *Conv* representa uma única camada convolucional com filtros 3×3, seguida de *batch normalization* e ativação ReLU. Um *ConvBlock* é composto por uma convolução 3×3, seguida por normalização e ativação ReLU, com um total de Q_F filtros, seguido por uma convolução 1×1 com $Q_F/2$ filtros. Finalmente, um *SepBlock* compreende uma convolução *depth-wise* com filtros 3×3 sobre os canais de entrada, que é seguida por uma convolução *point-wise* com filtros 1×1. Observa-se que, ao usar essa abordagem baseada em *Separable Convolutions* [SM14], é possível reduzir pela metade o número de parâmetros e multiplicações, diminuindo a complexidade do modelo.

Uma camada de convolução comum, composta por 128 filtros 3×3, dado um volume de entrada de dimensões $7 \times 7 \times 3$, aplicando *stride* $= 1$, resulta em 86.400 multiplicações. As mesmas configurações, para uma entrada com dimensões $32 \times 32 \times 3$, resultam em $3.110.400$ multiplicações. Substituindo a convolução comum por uma *Separable Convolution*, os mesmos exemplos resultarão em respectivamente 19.491 e 417.516 operações.

A Tabela 3.3 compara estruturalmente *CensorNet* com duas variações de arquiteturas utilizadas pelo método YOLO. YOLO-Full é a arquitetura original baseada em Darknet-19 [RF16], modificada para atacar o problema de detecção de objetos. YOLO-Tiny é uma arquitetura reduzida que aplica menor número de camadas convolucionais com menor número de filtros, resultando em menos parâmetros e operações. Esta arquitetura apresenta melhor desempenho em função do tempo, por outro lado, perde acurácia preditiva. A arquitetura *CensorNet* é composta por blocos semelhantes aos utilizados pela arquitetura YOLO-Full, no entanto, ela substitui *ConvBlocks* com 1024 filtros por 2 *SepBlocks*, reduzindo o número de parâmetros na ordem de $\approx 9\times$.

Tabela 3.3: Estruturas e parâmetros das arquiteturas YOLO-Full, YOLO-Tiny e *CensorNet*.

YOLO-Full		YOLO-Tiny		*CensorNet*	
Conv	32	Conv	16	Conv	32
Pooling		Pooling		Pooling	
Conv	64	Conv	32	Conv	64
Pooling		Pooling		Pooling	
ConvBlock	128	Conv	64	Conv	128
Conv	128	Conv	64	Conv	128
Pooling		Pooling		Pooling	
ConvBlock	256	Conv	128	ConvBlock	256
Conv	256	Conv	128	Conv	256
Pooling		Pooling		Pooling	
ConvBlock	512	Conv	256	ConvBlock	512
ConvBlock	512			ConvBlock	512
Conv	512			Conv	512
Pooling		Pooling		Pooling	
ConvBlock	1024	Conv	512	SepBlock	512
ConvBlock	1024	MaxPool		SepBlock	512
Conv	1024	Conv	1024		
Conv	1024	Conv	512		
Conv	1024				
50,5Mp		15,7Mp		5,7Mp	

3.3.2 Configurações dos Experimentos

A seguir serão apresentados os métodos utilizados como *baseline* para comparação com *CensorNet*, indicando as configurações dos hiperparâmetros utilizados para cada método. Serão apontadas também as métricas utilizadas para comparação entre os modelos treinados.

3.3.2.1 Baselines

Faster R-CNN [RHGS15] tem origem no método *Region-based Convolutional Network* (R-CNN) [GDDM15], precursor na aplicação de ConvNets para tarefas de detecção de objetos. R-CNN é composto por 4 elementos sendo eles i) um módulo que propõe regiões de interesse, ii) uma ConvNet para extração de *features* que representam as regiões propostas em função das N_C classes relacionadas no problema, iii) N_C SVMs binários, um para cada classe e iv) um módulo de regressão para ajuste das posições das propostas que forem assumidas como objetos. R-CNN utiliza o algoritmo *Selective Search* [UvdSGS13] para propor regiões de interesse dado um número predefinido de propostas, em geral 2.000. O método exige treinamento parte-a-parte e leva até 50 segundos para gerar predições de objetos para uma única imagem. *Faster* R-CNN [RHGS15] é uma

evolução de R-CNN [GDDM15] e de *Fast* R-CNN [Gir15]. *Faster* R-CNN é composta por 3 redes interconectadas sendo elas: i) uma ConvNet que extrai *features* da imagem de entrada, ii) uma RPN, que é uma ConvNet para geração de propostas de predições que substitui *Selective Search* e iii) uma rede totalmente conectada para classificação e regressão. As propostas geradas pela RPN são unidas às *features* da imagem de entrada para serem finalmente avaliadas pelo classificador. O método *Faster* R-CNN é mais eficiente e preciso que seus precursores R-CNN e *Fast* R-CNN.

Para treinar o modelo de detecção de partes pornográficas baseado em *Faster* R-CNN, foi utilizada uma arquitetura *Inception Resnet* [SIVA17] que chegou a $59,4$ milhões de parâmetros. O modelo treinado foi ajustado finamente, partindo de um modelo de classificação pretreinado com Imagenet [DDS$^+$09]. Dadas as características do método, o tamanho do *batch* foi fixado em 1, com *momentum* igual a $0,9$ e taxa de aprendizado 4×10^{-4}, mantida por todo o treinamento até a convergência.

Single Shot MultiBox Detector (SSD) [LAE$^+$16] é um método de detecção de objetos que compartilha similaridades com o método YOLO. Assim como YOLO, SSD também utiliza uma única ConvNet, gerando predições com base em uma única passada do volume pela rede. As predições resultantes são formadas partindo de conjuntos de predições de diferentes escalas e proporções. SSD combina predições de vários mapas de ativação com diferentes resoluções, o que supostamente melhora os resultados para detecção de objetos com diferentes tamanhos. A abordagem de passada única contribui para que SSD possa processar 59 FPS em GPUs NVIDIA® M40.

A arquitetura de rede utilizada para treinamento do modelo SSD [LAE$^+$16] foi a *Inception Resnet*, pretreinada para classificação com Imagenet [DDS$^+$09], chegando a $54,3$ milhões de parâmetros. O tamanho do *batch* foi fixado em 24 imagens com *momentum* igual a $0,9$ e *weight decay* igual a $0,9$. A taxa de aprendizado foi fixada em 4×10^{-3}, mantida fixa até o final do treinamento.

Para treinar os modelos baseados em YOLO, foi utilizado *batch* de 24 imagens, *momentum* igual a $0,9$ e *weight decay* igual a 5×10^{-4}. A taxa de aprendizado seguiu um agendamento, iniciando em 10^{-3}, passando para 10^{-4} após 40 épocas e 10^{-5} depois de 60 épocas. O modelo YOLO-Full atingiu $50,5$ milhões de parâmetro enquanto que YOLO-Tiny limitou-se a $15,7$ milhões de parâmetros, $\approx 3\times$ menos parâmetros que YOLO-Full.

3.3.2.2 Métricas Observadas

Os experimentos executados compararam os modelos utilizando 4 métricas, sendo elas: i) quantidade de parâmetros, que impacta na quantidade de memória necessária para carga do modelo e no tempo de predição, ii) AP por classe, que permite avaliar o desempenho preditivo do modelo em função de cada classe, iii) mAP, que traduz o desempenho preditivo do modelo como um todo em um único valor numérico e, finalmente, iv) tempo de predição, que representa a possibilidade de utilizar os modelos em problemas de tempo real. A seguir serão listados os resultados observados.

3.3.3 Análise Experimental

Modelos de detecção de objetos treinados com *DPC* podem ser utilizados para geração de censuras automáticas, aplicando coberturas ou descaracterizações sobre partes do corpo que remetem à pornografia. Se suficientemente velozes, estes modelos poderiam ser embarcados em aplicações que reproduzem imagens ou vídeo estáticos mas, especialmente, sob demanda. Para verificar a efetividade da aplicação destes modelos, especialmente em problemas de tempo real, foram realizados experimentos utilizando métodos de detecção de objetos reconhecidos por seus resultados em desafios como MS COCO [LMB+14], PASCAL VOC [EEVG+15] e *Open Images* [KDA+]. Foram comparados os métodos YOLO [RDGF16], SSD [LAE+16] e *Faster* R-CNN [RHGS15], observando também o comportamento de diferentes arquiteturas para o método YOLO. Os experimentos foram executados em ambiente comum, constituído por CPU Intel® Xeon® E5-2603 e GPU NVIDIA® M40.

Todos os modelos gerados, independente do método, partiram de modelos pretreinados para classificação com Imagenet [DDS+09], sendo finamente ajustados para detecção de objetos utilizando *DPC*. A utilização de modelos pretreinados contribui para a inicialização dos pesos das redes e também transfere conceitos aprendidos de outros *datasets*. Além de partir de modelos pretreinados, todos os métodos avaliados utilizaram as mesmas estratégias de aumento de dados, aplicando recortes randômicos, ajustes de escala, variações de proporções e espelhamentos horizontais. O treinamento de cada método respeitou a mesma janela temporal, executando por 96 horas (4 dias), prazo definido em função da disponibilidade de recursos disponíveis no laboratório. A avaliação dos resultados seguiu os critérios definidos pelo desafio PASCAL VOC [EEVG+15], assumindo como corretas as predições que atingiram IoU maior ou igual a $0,5$, apontando a classificação correta do objeto predito.

3.3.3.1 Quantidade de Parâmetros vs. mAP

CensorNet foi criada com o objetivo de chegar-se a um detector leve e preciso, que pudesse ser utilizado em aplicações para detecção de conteúdo pornográfico em tempo real. Para tal, foi desenvolvida uma arquitetura de rede baseada em *Separable Convolutions* [SM14] que reduziu significativamente a quantidade de parâmetros do modelo. *CensorNet* é composta por $5.669.682$ parâmetros, aproximadamente $1/3$ dos parâmetros de uma arquitetura YOLO-Tiny. Após o treinamento da rede utilizando *DPC*, foi observado o melhor resultado preditivo após 82 épocas, atingindo mAP$=50$ no conjunto de validação, chegando a mAP$=51$ no conjunto de teste. Na Figura 3.11, *CensorNet* representa somente o $2º$ melhor mAP, superior apenas ao observado para YOLO-Tiny e próximo de SSD. Por outro lado, *CensorNet* tem a menor quantidade de parâmetros dentre todos os modelos avaliados. Mesmo com acurácia preditiva próxima de SSD, *CensorNet* exige $\approx 9,6\times$ menos parâmetros.

Faster R-CNN é um método reconhecido por definir o estado-da-arte em detecção de objetos nos desafios MS COCO [LMB+14] e PASCAL VOC [EEVG+15]. O melhor modelo *Faster*

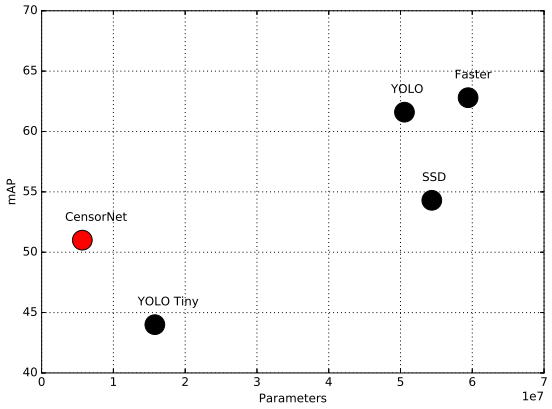


Figura 3.11: Comparação entre quantidade de parâmetros e performance preditiva (mAP).

R-CNN foi gerado após 36 épocas, atingindo mAP$=63,3$ no conjunto de validação e mAP$=62,8$ no conjunto de teste de *DPC*. Mesmo apresentando um resultado preditivo expressivo em termos de mAP, o melhor dentre todos os modelos ilustrados pela Figura 3.11, *Faster* R-CNN é o método que apresenta maior quantidade de parâmetros, $\approx 10,4\times$ maior que *CensorNet*.

O método SSD utiliza a estratégia de passada única do volume pela rede. O melhor modelo SSD observado no período de treinamento atingiu mAP$=60,1$ no conjunto de validação após 83 épocas. Este mesmo modelo apresentou resultados piores no conjunto de teste, registrando mAP$=54,3$. O resultado no conjunto de teste foi semelhante ao obtido por *CensorNet*, porém SSD utilizou um modelo significativamente maior em termos de parâmetros.

Para o método YOLO com sua arquitetura de rede original, o melhor modelo foi gerado após 97 épocas, atingindo mAP$=63,2$ no conjunto de validação. No conjunto de teste, este mesmo modelo obteve mAP$=61,6$. Os resultados foram semelhantes aos observados utilizando *Faster* R-CNN e superiores aos obtidos com SSD, mesmo com menos parâmetros. O desempenho preditivo de YOLO também foi superior ao observado com *CensorNet*. Já a variação mais leve do método YOLO, conhecido como YOLO-Tiny, gerou seu melhor resultado de validação após 88 épocas, atingindo mAP$=47$. O mesmo modelo resultou em mAP$=44$ no conjunto de teste de *DPC*, marcando os piores resultados de mAP dentre todos os modelos comparados, mesmo contando com mais parâmetros que *CensorNet*.

3.3.3.2 Análise por classe

Os resultados apresentados na Tabela 3.4 demonstram que *Faster* R-CNN apresenta os melhores desempenhos preditivos para mAP e, especificamente, AP para as classes *butt* e *frontalM*. *CensorNet* tem resultados interessantes para as classes de objetos maiores, como *butt* e *breast*. Pode-se observar que *CensorNet* é superior a YOLO *Tiny* em todas as classes, atingindo resultados próximos dos observados com SSD, sendo inclusive superior para as classes *butt* e *frontalF*, mesmo utilizando um modelo significativamente menor em termos de parâmetros. O método YOLO, com sua arquitetura original, atingiu resultados próximos aos observados com *Faster* R-CNN, inclusive

sendo superior para as classes *breast* e *frontalF*, que representam os menores objetos do *dataset* em termos de dimensões. Por ser um método de passada única, baseado em uma única ConvNet, os resultados obtidos por YOLO chamam a atenção, especialmente pela semelhança com *Faster* R-CNN.

Tabela 3.4: Resultados de teste observados para os melhores modelos treinados com *DPC*.

Método	*butt*	*breast*	*frontalM*	*frontalF*	mAP
CensorNet	$56,7$	$66,6$	$38,9$	$39,6$	$51,0$
Faster R-CNN	**66,3**	$71,0$	**64,8**	$49,3$	**62,8**
SSD	$53,5$	$68,8$	$56,0$	$38,8$	$54,3$
YOLO	$64,4$	**72,8**	$58,8$	**50,2**	$61,6$
YOLO-Tiny	$48,9$	$60,2$	$34,4$	$32,2$	$44,0$

3.3.3.3 Desempenho por Tempo

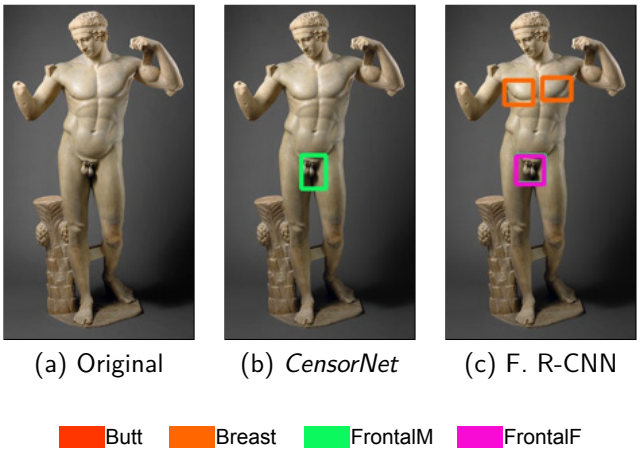
Com relação ao desempenho preditivo em função do tempo, a Tabela 3.5 reforça a intuição de que, ao se tratar de um modelo enxuto com relação ao número de parâmetros, YOLO-Tiny necessita de menos tempo para gerar predições. Mesmo sendo mais lento que YOLO-Tiny, ao considerar a relação mAP $\times$ tempo, *CensorNet* torna-se competitivo. A performance de tempo demonstrada por SSD foi superior a apresentada por YOLO, mas sua acurácia preditiva é inferior, tanto com relação a YOLO quanto a *Faster* R-CNN. Mesmo sendo o método que apresentou melhor acurácia preditiva, *Faster* R-CNN foi claramente o pior em termos de predições em função do tempo, não se enquadrando como um método de tempo real dada a incapacidade de gerar predições para 24 imagens por segundo. *CensorNet* apresenta um *speedup* de $175\times$ quando comparado ao tempo observado para *Faster* R-CNN e $1,5\times$ quando comparado com YOLO. Em uma visão geral, YOLO também demonstra resultados importantes ao atingir FPS$=45$, o que permite aplica-lo em tarefas de processamento de vídeos, mesmo em abordagens quadro-a-quadro. Considerando seu desempenho preditivo, apresentando mAP semelhante ao observado com *Faster* R-CNN, tratou-se do método que melhor combinou desempenho preditivo com desempenho em função do tempo.

Tabela 3.5: Tempos de predição por imagem (em milissegundos).

Método	Tempo	FPS
CensorNet	15 ± 0	66
Faster R-CNN	2700 ± 330	$0,37$
SSD	20 ± 4	50
YOLO	22 ± 0	45
YOLO-Tiny	$\mathbf{10} \pm 0$	**100**

3.3.3.4 Capacidade de generalização

Para verificar a capacidade de generalização dos métodos utilizados, foi utilizada uma imagem de uma estátua de um homem nu como entrada para todos os modelos treinados com *DPC*. Assumindo-se como hipótese que um modelo com grande capacidade de generalização deve identificar objetos em diferentes contextos, esperava-se que os modelos fossem capazes de detectar a genitália presente na estátua. A Figura 3.12 ilustra os resultados obtidos para esse experimento para *CensorNet* e para *Faster* R-CNN, ambos capazes de detectar partes íntimas na imagem. Por outro lado, *Faster* R-CNN gerou 3 erros: (i) *breast* esquerdo, (ii) *breast* direito, e uma genitália feminina. Reforçando, *DPC* anota objetos da classe *breast* somente para mulheres. Estes erros não são observados para *CensorNet*.



(a) Original (b) *CensorNet* (c) F. R-CNN

Figura 3.12: Capacidade de generalização sobre estátua masculina.

3.4 Método YOLO

Redmon et al. [RDGF16] afirmam que o método YOLO reformula a detecção de objetos ao transforma-la em um problema de regressão, partindo unicamente dos pixeis de uma imagem, resultando nas probabilidades por classe, nas coordenadas e nas dimensões das predições que delimitam os objetos. Além de simples, já que utiliza uma abordagem fim-a-fim com uma única ConvNet, também apresenta desempenho competitivo em função do tempo. Diferente de outros métodos de detecção de objetos baseados em ConvNets [GDDM15, Gir15], YOLO necessita apenas de uma propagação da entrada pela rede para gerar predições, apresentando desempenho compatível com métodos de detecção de objetos em tempo real [EVGW$^+$], mantendo a acurácia preditiva.

Com base nos resultados observados nos experimentos relatados, quando arquiteturas enxutas com relação à parâmeros e operações matemáticas foram aplicadas ao método YOLO e comparadas com outros métodos de detecção de objetos, somado à simplicidade de implementação e modificação de seus componentes, esta tese optou por pesquisar abordagens de detecção de partes

íntimas relacionadas à pornografia utilizando variações deste método. A seguir serão descritos o método YOLO e as variações desenvolvidas e avaliadas em um contexto de objetos relacionados à pornografia. Implementado sob um *framework* para *Deep Learning* chamado Darknet [Red], desenvolvido em linguagens C e CUDA pelo próprio autor, o método YOLO foi portado para plataforma Keras [C+] + Tensorflow [AAB+] que, dado seu alto nível de abstração, simplifica a criação de extensões ou mesmo modificações em estruturas fundamentais como carga e manipulação dos dados de entrada, arquiteturas de rede ou funções de custo. O código portado está disponível para acesso na plataforma *GitHub* por meio do link https://github.com/Pezaun/golo.git.

3.4.1 Arquitetura e representação das predições

O método YOLO [RDGF16] ataca o problema de detecção de objetos utilizando uma abordagem simples, baseado em uma única ConvNet, propagando o volume de entrada pela rede uma única vez para gerar todas as possíveis predições. O método modela o problema representando as predições como um cubo de dimensões variáveis em função da largura (L_I) e altura (A_I) da imagem de entrada, onde a largura (L_S) e a altura (A_S) do cubo correspondem as dimensões do volume de saída da última camada da rede, resultando em $L_I/32 \times A_I/32$. A profundidade do cubo representa os valores que compõe cada predição, sendo $5 + N_C$, onde N_C representa o número de classes. A profundidade do cubo está vinculada também a quantidade de detectores (N_D), sendo o valor mínimo de $N_D = 1$. Cada célula do cubo é responsável por N_D predições, determinadas pelo centro de cada objeto. Desta maneira, cada predição é formada por um conjunto de valores de acordo com o seguinte detalhamento: 1 *score* de confiança c, que indica se o centro de um objeto realmente está sobre a célula, 4 valores referentes ao posicionamento espacial (x, y) e dimensional (w, h), seguidos de N_C valores, correspondentes ao número de classes que permitem classificar cada predição. A profundidade mínima do cubo é dada por $1 + 4 + N_C$. O método assume N_D detectores por célula, de maneira que cada célula detecte de 1 até N_D possíveis objetos, refletindo na profundidade do cubo que passa a ser definida por $(1 + 4 + N_C) \times N_D$. A título de exemplo, para uma entrada com dimensões $448 \times 448 \times 3$, assumindo 5 detectores em um problema de 4 classes, obtém-se um volume de saída com dimensões $14 \times 14 \times (1 + 4 + 4) \times 5$, totalizando 8.820 valores que representam 980 possíveis objetos. A mesma predição para um volume de entrada com dimensões $512 \times 512 \times 3$, também com 5 detectores e 4 classes, gera um volume de saída com dimensões $16 \times 16 \times (1 + 4 + 4) \times 5$, resultando em 11.520 valores que representam 1.280 possíveis objetos.

A Figura 3.13 representa o fluxo completo do método YOLO [RDGF16]. É possível observar a arquitetura da ConvNet, neste caso Darknet-19 [RF16], indicando as dimensões da imagem de entrada e a progressiva redução das dimensões do volume propagado pelos blocos da rede. Quando aplicada no método de detecção YOLO, Darknet-19 recebe uma convolução paralela que leva *features* de resolução mais alta diretamente para a dimensão de saída, reorganizando o volume para torna-lo compatível com as dimensões de entrada do último bloco, permitindo a concatenação. A camada

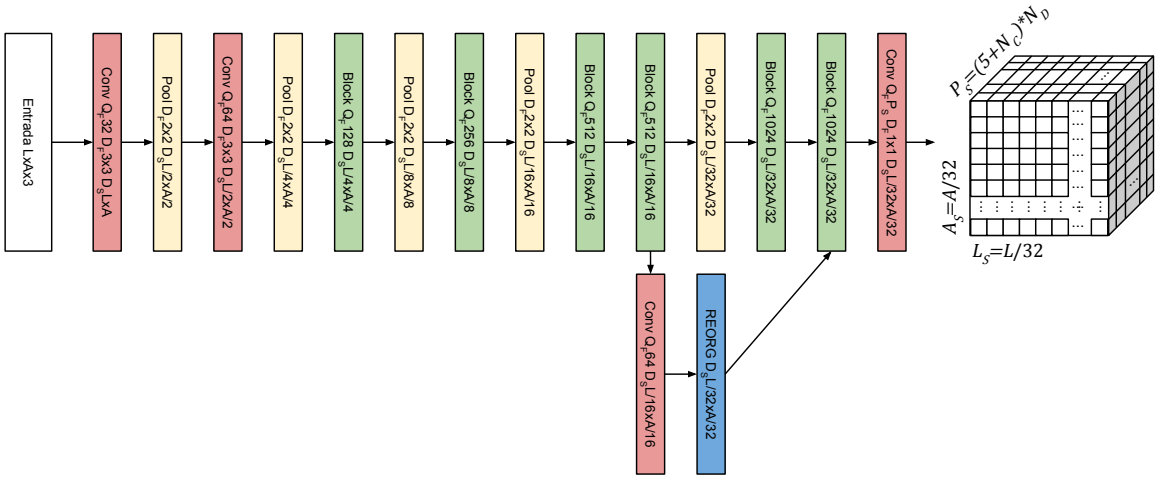


Figura 3.13: Representação da ConvNet e do cubo de saída que compõe o método YOLO.

REORG transforma um volume com dimensões $64 \times (L_I/16) \times (A_I/16)$ em $256 \times (L_I/32) \times (A_I/32)$, distribuindo *features* espaciais (de largura e altura) na profundidade do volume resultante, possibilitando a concatenação das saídas de 2 blocos com dimensões diferentes. A última camada da rede é formada por uma convolução com D filtros de dimensão 1×1 sem ativação, gerando como saída o cubo que representa $(L_I/32) \times (A_I/32) \times D$ valores correspondentes às possíveis predições. A dimensão de profundidade (P_S) do cubo de predições, resultante da quantidade de filtros da última camada convolucional, é dada por $P_S = (1 + 4 + N_C) \times N_D$.

A conversão dos valores extraídos do cubo de predições para a representação espacial 2D dos objetos preditos na imagem, associadas a uma das N_C possíveis classes, é calculada da seguinte maneira: a ativação SIGMOIDE do valor c representa a confiança de um detector conter um objeto, dado por $\sigma(c)$. A localização espacial é dada pela ativação SIGMOIDE dos valores (x, y), dado por $\sigma(x)$ e $\sigma(y)$, resultando em valores entre $(0, 1)$, que representam a proporção de largura e altura em que o centro do objeto predito ocorre na célula. As dimensões do objeto são resultado da ativação exponencial dos valores (w, h), dado por $exp(w)$ e $exp(h)$, limitando inferiormente os valores resultantes em $\approx 0,0$ cada. Os resultados de $exp(w)$ e $exp(h)$ são multiplicados pelas âncoras de predefinição de proporções, que são relacionadas aos seus respectivos detectores. O número de âncoras é dado pela quantidade de detectores N_D, predefinidos para cada *dataset* antes do treinamento. A definição de classificação de cada objeto predito é dada pela ativação SOFTMAX, aplicada aos N_C valores correspondentes ao número de classes definido pelo *dataset* utilizado.

A Figura 3.14 demonstra como exemplo uma imagem contendo 2 objetos da classe SHEEP, onde é projetada uma grade que representa um único detector ($N_D = 1$) em um problema de 4 classes. Originalmente, a imagem tem dimensões iguais a 500×375, sendo reescalada para 320×320 para entrada na ConvNet, gerando como saída um cubo de dimensões $10 \times 10 \times D$, sendo $D = (1 + 4 + N_C) \times N_D$, resultando em 900 valores $(10 \times 10 \times 9)$ que representam 100 possíveis objetos. Ocorrem duas predições para a classe SHEEP, localizadas respectivamente nas posições $(7, 2)$ e $(8, 5)$, observando os centros das predições marcados em vermelho. Dado que os centros são obtidos pela ativação SIGMOIDE, eles representam um percentual das dimensões da célula. No

Figura 3.14: Exemplo de predições para 2 objetos ocorrendo nas células $(7, 2)$ e $(8, 5)$ com $N_D = 1$ em um problema de 4 classes.

exemplo, os objetos preditos estão localizados a 70% da largura e 31% da altura da posição $(7, 2)$ e 40% da largura e 12% da altura da posição $(8, 5)$ na grade. As dimensões de largura e altura das predições consideram as âncoras de pré-definição como base, sendo ajustadas por um valor predito passando por ativação exponencial ($\exp$). Assumindo o número de detectores utilizado no exemplo ($N_D = 1$), tem-se apenas uma âncora onde, neste caso, as dimensões de largura e altura representam $3,4 \times 3,8$ células da grade. Assim, os valores de ativação para as predições serão de $1,17$ para a largura e $1,23$ para a altura da predição que ocorre na posição $(7, 2)$ e de $0,91$ para a largura e $0,87$ para a altura da predição que ocorre na posição $(8, 5)$, dado que as dimensões dos objetos são respectivamente iguais a (128×150) e (100×106), após serem escaladas de acordo com as dimensões de entrada (320×320). No exemplo, ocorre a sobreposição de predições com a mesma classe. Mesmo aplicando NMS, as predições são mantidas já que a IoU observada ($\approx 0,1$) é inferior ao limiar de corte utilizado $(0,25)$, resultando em predições coerentes com a realidade.

3.4.2 Função de Custo

Dentre os diferentes métodos de detecção de objetos baseados em ConvNets pesquisados nesta tese ([RDGF16, Gir15, RHGS15, LAE+16]), YOLO destaca-se pelo desempenho preditivo e pela velocidade, mas também por sua simplicidade. Trata-se de uma única rede convolucional composta por camadas convencionais (convoluções, *poolings* e *batch normalization*), estruturadas em blocos, modelando todas as possíveis predições em um tensor de 3 dimensões. O treinamento é feito com *batches* de imagens e aplica o método de otimização SGD. Por outro lado, a simplicidade da arquitetura e do processo de treinamento contrasta com uma engenhosa função de custo que pondera 5 diferentes partes do problema para cada possível objeto predito: i) posicionamento espacial, ii) dimensões, iii) confiança de ser um objeto, iv) confiança de não ser um objeto e, finalmente, v) classificação. Somados, os custos referentes aos 5 problemas compõe a função a ser otimizada.

$$
\begin{aligned}
&\lambda_{locdim} \sum_{i=0}^{L_S \times A_S} \sum_{j=0}^{N_D} \mathbb{1}_{ij}^{obj} \left[\left(\sigma(x_i) - \hat{X}_i \right)^2 + \left(\sigma(y_i) - \hat{Y}_i \right)^2 \right] \\
&+ \lambda_{locdim} \sum_{i=0}^{L_S \times A_S} \sum_{j=0}^{N_D} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{(exp(w_i) * L_{Ai}} - \sqrt{\hat{W}_i} \right)^2 + \left(\sqrt{(exp(h_i) * A_{Ai}} - \sqrt{\hat{H}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{L_S \times A_S} \sum_{j=0}^{N_D} \mathbb{1}_{ij}^{obj} \left(\sigma(c_i) - \hat{C}_i \right)^2 \\
&+ \lambda_{noobj} \sum_{i=0}^{L_S \times A_S} \sum_{j=0}^{N_D} \mathbb{1}_{ij}^{noobj} \left(\sigma(c_i) - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{L_S \times A_S} \mathbb{1}_{i}^{obj} \sum_{cls \in classes} \left(p_i(cls) - \hat{p}_i(cls) \right)^2 \quad (3.8)
\end{aligned}
$$

Conforme descrito por Redmon et al. [RDGF16], o custo referente a cada uma das 5 tarefas é representado pela soma de erros quadráticos. Segundo o autor, esta abordagem é simples de otimizar, mesmo que não garanta a melhor performance preditiva. Por outro lado é possível, ao particionar as tarefas, ponderar os erros para cada parte da função, ajustando cada tarefa por meio de parâmetros (λ) definidos de acordo com cada contexto de treinamento, com cada *dataset*. Na Equação 3.8, o parâmetro λ_{locdim} pondera as tarefas de localização e definição de dimensões, enquanto que λ_{noobj} pondera o custo de confiança quando os detectores não possuem objetos. Os termos $\mathbb{1}_{ij}^{obj}$ e $\mathbb{1}_{ij}^{noobj}$ são comutadores binários baseados nos dados reais de treinamento, indicando a existência ou não de um objeto no j-ésimo detector da i-ésima célula. Especificamente, seguindo as experiências relatadas em [RDGF16], esta tese manteve $\lambda_{obj} = 5$ e $\lambda_{noobj} = 1$, dada melhor performance relatada. Ainda assim, a soma dos erros quadráticos pondera igualitariamente qualquer predição válida, independente de suas dimensões. Idealmente, pequenos desvios em predições grandes não deveriam ser tão significativos quanto os observados em predições pequenas. Esta ponderação é feita pela função ao assumir a raiz quadrada das dimensões de largura e altura, tendendo a gerar custos maiores para predições de dimensões menores. O custo de classificação para cada detector é condicionado ao comutador $\mathbb{1}_{ij}^{obj}$, onde $p_i(cls)$ representa o vetor de probabilidades por classe predito, enquanto que $\hat{p}_{ij}(cls)$ representa o vetor de classificação real para o j-ésimo detector da i-ésima célula predita. Nos casos onde um detector estiver responsável por gerar predições para um objeto real ($\mathbb{1}_{ij}^{obj} = 1$), os custos de classificação, localização, dimensões e de confiança em conter um objeto serão todos computados e somados. Nos casos inversos, em que um detector não estiver vinculado a nenhum objeto real ($\mathbb{1}_{ij}^{noobj} = 1$), será computado somente o custo de confiança de existência de objeto, ponderado pelo termo λ_{noobj}, levando o valor desta confiança ao mais próximo possível de 0.

3.4.3 Resultados Observados

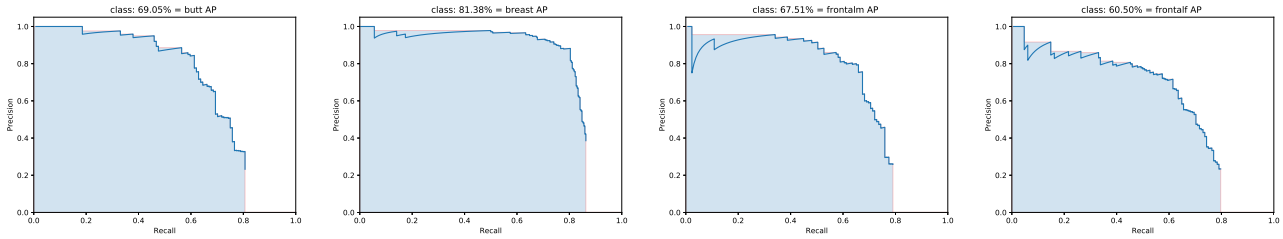
Durante o treinamento multi-escala observado em YOLO, as dimensões do volume de entrada variaram arbitrariamente de acordo com limites predeterminados pela arquitetura utilizada. Segundo Redmon et al. [RF16], dimensões maiores produzem resultados preditivos melhores, especialmente para objetos pequenos, enquanto dimensões menores melhoram o desempenho em função do tempo. Em ConvNets, a propagação de volumes menores resulta em menos operações matemáticas quando comparadas a volumes maiores, seguindo o exposto pela Equação 2.2.

Sabendo da influência das dimensões do volume de entrada no desempenho das camadas convolucionais, aproveitando as características da arquitetura utilizada, totalmente convolucional, onde os modelos foram treinados com múltiplas escalas, foram feitos experimentos para avaliar o impacto das dimensões de entrada na ConvNet, em tempo de predição. Foram observados os resultados preditivos, comparando AP por classe e mAP, além das médias dos tempos de predição para diferentes dimensões de entrada.

Tabela 3.6: Resultados observados ao variar as dimensões do volume de entrada utilizando o conjunto de teste.

Entrada (w,h)	*butt*	*breast*	*frontalM*	*frontalF*	mAP	tempo(ms)
384×384	$0,7032$	$0,7503$	$0,5672$	$0,5187$	$0,6348$	**27,00** $\pm 0,70$
416×416	**0,7330**	$0,7569$	$0,5932$	$0,5514$	$0,6586$	$29,40 \pm 0,66$
448×448	$0,6876$	$0,7648$	$0,5854$	$0,5975$	$0,6588$	$32,35 \pm 0,48$
480×480	$0,6944$	$0,8152$	$0,5191$	$0,5669$	$0,6489$	$33,30 \pm 0,64$
512×512	$0,6939$	$0,8119$	$0,6308$	$0,5940$	$0,6826$	$35,15 \pm 0,57$
544×544	$0,6893$	$0,8268$	$0,5890$	$0,5140$	$0,6548$	$40,30 \pm 1,05$
576×576	$0,6905$	$0,8138$	**0,6751**	**0,6050**	**0,6961**	$41,15 \pm 0,90$
608×608	$0,7259$	**0,8296**	$0,5861$	$0,5379$	$0,6699$	$45,10 \pm 0,89$
640×640	$0,6438$	$0,8009$	$0,5801$	$0,5651$	$0,6475$	$50,00 \pm 1,00$

A Tabela 3.6 apresenta mAP e AP por classe, comparando o desempenho preditivo de um mesmo modelo, variando as dimensões de entrada das imagens. Durante o treinamento deste modelo, a cada 10 *batches*, as dimensões das imagens de treinamento variaram arbitrariamente em múltiplos de 32, presentes no intervalo $[384, 416, ..., 576, 608]$, produzindo saídas com dimensões correspondentes ao intervalo $[12, 13, ..., 18, 19]$. O último registro da Tabela 3.6 apresenta os resultados observados para entradas de teste com dimensões iguai a (640×640), gerando uma saída (20×20). Mesmo esta configuração não constando no intervalo de dimensões de treinamento, pode-se observar que a capacidade de generalização do modelo ainda gera resultados coerentes, inclusive melhores que aqueles observados para dimensões de entrada (384×384), presente no intervalo de treinamento. Com relação ao desempenho em função do tempo, como esperado, o tempo cresce proporcionalmente de acordo com o aumento nas dimensões da imagem de entrada, impactando nas dimensões do volume propagado por toda a rede.



(a) *butt* (b) *breast* (c) *frontalM* (d) *frontalF*

Figura 3.15: Curvas Precisão/Revocação para detecção com entradas 576×576.

A Figura 3.15 apresenta as curvas precisão/sensibilidade para as 4 classes representando partes íntimas do corpo, reproduzindo os valores de AP apresentados na Tabela 3.15. O maior AP foi observado na classe *breast* (3.15b), onde pode-se perceber um desempenho preditivo estável, menos suscetível às variações do limiar de confiança. As classes *frontalM* e *frontalF* (3.15c e 3.15d) apresentam padrões semelhantes, exibindo impactos de precisão quando o limiar de confiança começa a cair. Por outro lado, mesmo que *frontalF* atinja maior sensibilidade, a robustez ao limiar de confiança faz com que *frontalM* apresente melhor AP. A classe *butt* (3.15a), ao mesmo tempo o maior e menos presente objeto do *dataset*, apresenta a segunda melhor sensibilidade, que cresce rapidamente a partir dos primeiros limiares de confiança plotados, coerente com a premissa de que objetos maiores tendem a ser facilmente detectados.

Como constatado na Seção 2.5.2, Darknet-19 [RF16] apresenta acurácia preditiva $= 0,9906$ para classificação no conjunto de teste do *dataset DataSex*, um *dataset* para classificação de imagens pornográficas apresentado na Seção 4.4.1.1. Esta mesma rede atingiu mAP $= 0,6961$ no conjunto de teste de *DPC* para o problema de detecção de objetos, mais complexo que o problema de classificação. O problema de classificação exposto utilizou redes com 2 valores de saída (representando cada classe), enquanto que a detecção de objetos para imagens com as mesmas dimensões (448×448) representa 8.820 valores de saída, exigindo mais do modelo. A partir desta percepção, foi construída uma variação da arquitetura de rede, adicionando uma segunda saída específica para a tarefa de classificação. Esta variação, chamada de *CensorPlus*, resultou em um método constituído por uma única rede, que executa uma única passada do volume de entrada, gerando predições de detecção de objetos e classificação da imagem como um todo.

De acordo com as características do problema, o modelo *CensorPlus* foi treinado utilizando um *dataset* que juntou *DPC* com um recorte do conjunto de treinamento de *DataSex*, estruturado para manter a compatibilidade com as duas tarefas. O conjunto total de imagens da junção dos *datasets* passou das 3.000 imagens originais de *DPC* para 6.000 imagens. Do todo, 3.000 imagens representaram a classe *porn* (para tarefa de classificação) e todas continham a anotação de pelo menos 1 objeto (os mesmos encontrados em *DPC* para a tarefa de detecção). Outras 3.000 imagens, um subconjunto de treinamento de *DataSex*, todas contendo pessoas em situações cotidianas e sem a anotação de qualquer objeto, representaram a classe *free*.

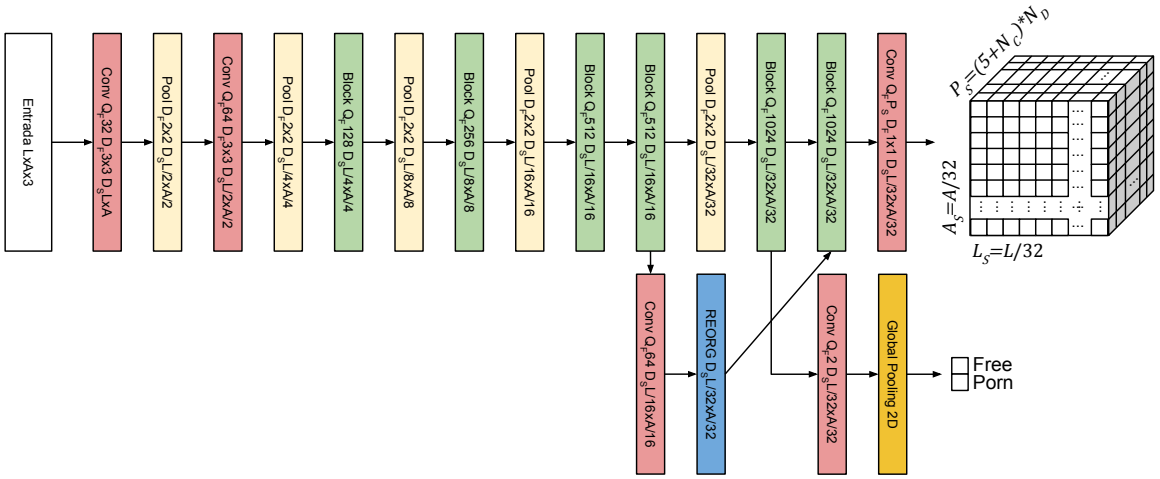


Figura 3.16: Adaptação do método para classificação de imagem e detecção de objetos.

A arquitetura desenvolvida, ilustrada pela Figura 3.16, partiu de Darknet-19 [RF16], adicionando um segundo braço de saída na rede onde 2 valores ativados com SOFTMAX representam probabilidades para as classes *porn* e *free*. O braço de saída parte do penúltimo bloco convolucional, aproveitando a porção da rede onde o volume representa menor dimensionalidade espacial e evitando a adição de profundidade resultante da concatenação com a porção paralela. O treinamento foi executado por 500 épocas, variando dimensões de entrada a cada 10 *batches*. A taxa de aprendizado variou de acordo com agendamento proposto por Redmon et al. [RF16].

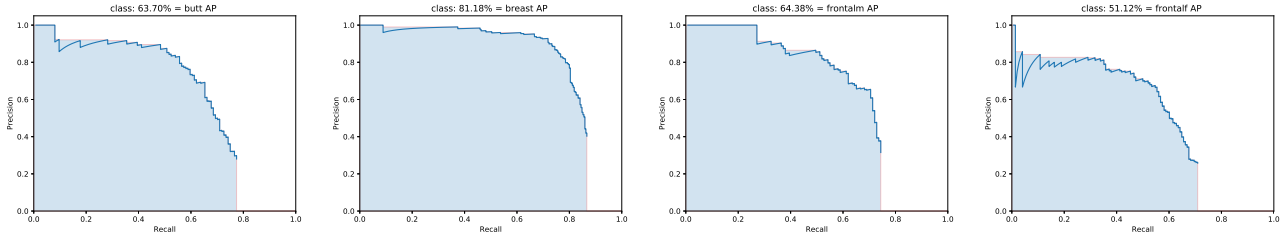
No treinamento do método para classificação de imagens e detecção de objetos foi utilizado o otimizador *Stochastic Gradient Descent* (SGD), aplicando a função de custo do método YOLO [RDGF16], definida pela Equação 3.8, para a saída de detecção de objetos. Para a saída de classificação, foi utilizada a entropia cruzada categórica sobre os 2 valores correspondentes às classes *free* e *porn*. O custo da rede foi dado pela soma dos custos das duas saídas.

Tabela 3.7: Resultados observados ao variar as dimensões do volume de entrada para tarefas de detecção de objetos e classificação de imagens utilizando o conjunto de teste.

Entrada (w,h)	*butt*	*breast*	*frontalM*	*frontalF*	mAP	ACC	tempo(ms)
384×384	$0,6552$	$0,6733$	$0,5153$	$0,4518$	$0,5739$	$0,9899$	**27,05** $\pm 1,28$
416×416	$0,6447$	$0,7367$	$0,5463$	$0,4718$	$0,5999$	$0,9899$	$29,80 \pm 1,78$
448×448	$0,6614$	$0,7386$	$0,5665$	$0,4317$	$0,5996$	$0,9915$	$30,70 \pm 1,27$
480×480	$0,6262$	$0,7706$	$0,5300$	**0,5194**	$0,6115$	**0,9917**	$32,90 \pm 0,54$
512×512	$0,6333$	$0,7947$	$0,6189$	$0,4664$	$0,6283$	$0,9899$	$35,60 \pm 1,50$
544×544	$0,6575$	$0,8125$	$0,6244$	$0,4504$	$0,6362$	$0,9849$	$39,70 \pm 1,90$
576×576	$0,5949$	**0,8358**	**0,6647**	$0,4801$	$0,6439$	$0,9849$	$40,50 \pm 1,56$
608×608	$0,6370$	$0,8118$	$0,6438$	$0,5112$	**0,6510**	$0,9849$	$44,00 \pm 1,34$
640×640	**0,6647**	$0,7904$	$0,5725$	$0,5059$	$0,6334$	$0,9849$	$48,20 \pm 1,54$

A Tabela 3.7 apresenta acurácia de classificação, mAP e AP por classe, comparando o desempenho preditivo de um mesmo modelo para as tarefas de classificação e detecção de objetos. Os resultados foram obtidos ao variar as dimensões de entrada das imagens, permitindo comparar o

impacto desta variação para as duas tarefas. Da mesma maneira que os resultados apresentados na Tabela 3.6, o modelo avaliado foi treinado com variação de dimensões de entrada, alternando arbitrariamente cada 10 *batches*, respeitando os mesmos intervalos de entrada ($[384, 416, ..., 576, 608]$) e saída ($[12, 13, ..., 18, 19]$). Analogamente, o último registro utiliza dimensões iguais a 640×640, gerando uma saída de dimensões 20×20, que não constam no intervalo de treinamento. Mesmo fora do intervalo, o modelo ainda mantém resultados coerentes para ambas as tarefas, demonstrando maior impacto negativo para a tarefa de detecção de objetos, onde a medida mAP apresenta queda em relação a observada com dimensões 608×608. A tarefa de detecção de objetos mantém-se estável com todas as dimensões analisadas.



(a) *butt* (b) *breast* (c) *frontalM* (d) *frontalF*

Figura 3.17: Curvas Precisão/Revocação para *CensorPlus* com entradas 608×608.

Semelhante ao exposto para o método unicamente de detecção, a Figura 3.17 apresenta as curvas precisão/sensibilidade para as 4 classes, reproduzindo os valores de AP apresentados na Tabela 3.7, resultantes do método *CensorPlus*. Da mesma maneira, o maior AP foi observado na classe *breast* (3.17b), seguindo padrão semelhante, onde percebe-se desempenho preditivo robusto ao limiar de confiança até atingir sensibilidade $\approx 0,65$, quando começa a perder precisão. As classes *butt* e *frontalM* (3.17a e 3.17c) geram AP aproximados e apresentam padrões semelhantes, atingindo sensibilidades também semelhantes. A classe *frontalM* mantém a precisão máxima para sensibilidade $\approx 0,25$, enquanto *butt* perde precisão já com sensibilidade $\approx 0,3$, resultando em AP maior para *frontalM*. A classe *frontalF* (3.17d), menor AP observado, apresenta queda de precisão já nos primeiros pontos de sensibilidade, mantendo-se $\approx 0,82$ até atingir sensibilidade $\approx 0,38$, passando a cair rapidamente. Observa-se que a classe *frontalF*, que representa os menores objetos anotados no *dataset DPC*, manteve maior dificuldade para detecção, assim como observado nos experimentos reportados na Tabela 3.6, avaliando somente detecção de objetos.

Ao verificar que *CensorPlus* apresentou resultados expressivos para a tarefa de classificação ($0,9917$ para dimensões de entrada 480×480), foi verificada a acurácia preditiva deste mesmo modelo com estas mesmas dimensões de entrada para todo o conjunto de teste de *DataSex*. O conjunto de teste de *DataSex* é composto por 64.096 imagens, sendo 32.048 *porn* e outras 32.048 *free*. *CensorPlus* atingiu $0,9640$ de acurácia no conjunto de testes de *DataSex*, errando $2,66\%$ a mais que o modelo derivado de Darknet-19 [RF16] (que atingiu $0,9906$, relatado na Tabela 2.2), treinado para classificação com todo o conjunto de treinamento de *DataSex*.

A Figura 3.18 apresenta amostras qualitativas de censuras geradas pelo melhor modelo de detecção de objetos treinado nos experimentos. Foram amostradas 13 imagens da classe *porn*,

extraídas do conjunto de teste de *DPC*, aplicando limiar de confiança $L_C = 0,30$ para plotar somente predições com confianças maiores. Nota-se que os exemplos apresentam coberturas ajustadas às partes íntimas diretamente expostas. As dimensões de entrada foram fixadas em 576×576, em consonância com o exposto na Tabela 3.6, que aponta o melhor mAP observado para este modelo utilizando estas dimensões.

A Figura 3.19 repete as 13 amostras apresentadas na Figura 3.18, aplicando agora o método *CensorPlus*, mantendo o limiar de confiança em $L_C = 0,30$. Somente predições com confianças maiores são plotadas. As dimensões de entrada foram fixadas em 608×608, em conformidade com os valores de mAP listados na Tabela 3.7. Neste exemplo, além das censuras geradas pelos objetos preditos, é registrada a classificação da imagem como um todo entre as classes *free* e *porn*. Os *scores* de confiança para ambas as classes são demonstrados no canto esquerdo superior de cada imagem. Nota-se que os exemplos mantiveram as coberturas ajustadas às partes íntimas presentes nas imagens, apresentando pequenas variações de posicionamento e confiança com relação ao exposto na Figura 3.18, que aplicou método exclusivamente para detecção de objetos. Os

Figura 3.18: Exemplos de imagens pornográficas expondo partes íntimas detectadas e oclusas automaticamente por método de detecção de objetos ajustado para entradas com dimensão 576×576.

resultados de classificação para cada imagem são coerentes com o conteúdo apresentado, mantendo-se todos $\approx 1,0$ para a classe *porn*.

Para contrastar com as imagens do conjunto de teste de *DPC*, foi selecionada uma amostra que representa pessoas em trajes de banho extraído do *dataset* Imagenet [DDS+09]. Esta amostra for processada pelo melhor modelo exclusivo para detecção de objetos e também por *CensorPlus*, utilizando dimensões de entrada fixadas em 576×576 e 608×608, em conformidade com o melhor desempenho de cada modelo. Os resultados qualitativos para ambos os métodos são ilustrados pela Figura 3.20. Foi mantido o limiar de confiança $L_C = 0,30$, o mesmo aplicado nos exemplos das Figuras 3.18 e 3.19. O exemplo não apresenta qualquer censura para ambos os métodos, mesmo se tratando de pessoas em trajes de banho com significativa exposição de pele. Com relação aos *scores* de classificação para a imagem como um todo, ilustrado na Figura 3.20b, a confiança para a classe *free* apontada pela porção de classificação do modelo foi $= 1,0$, consonante com o conteúdo representado pela imagem.

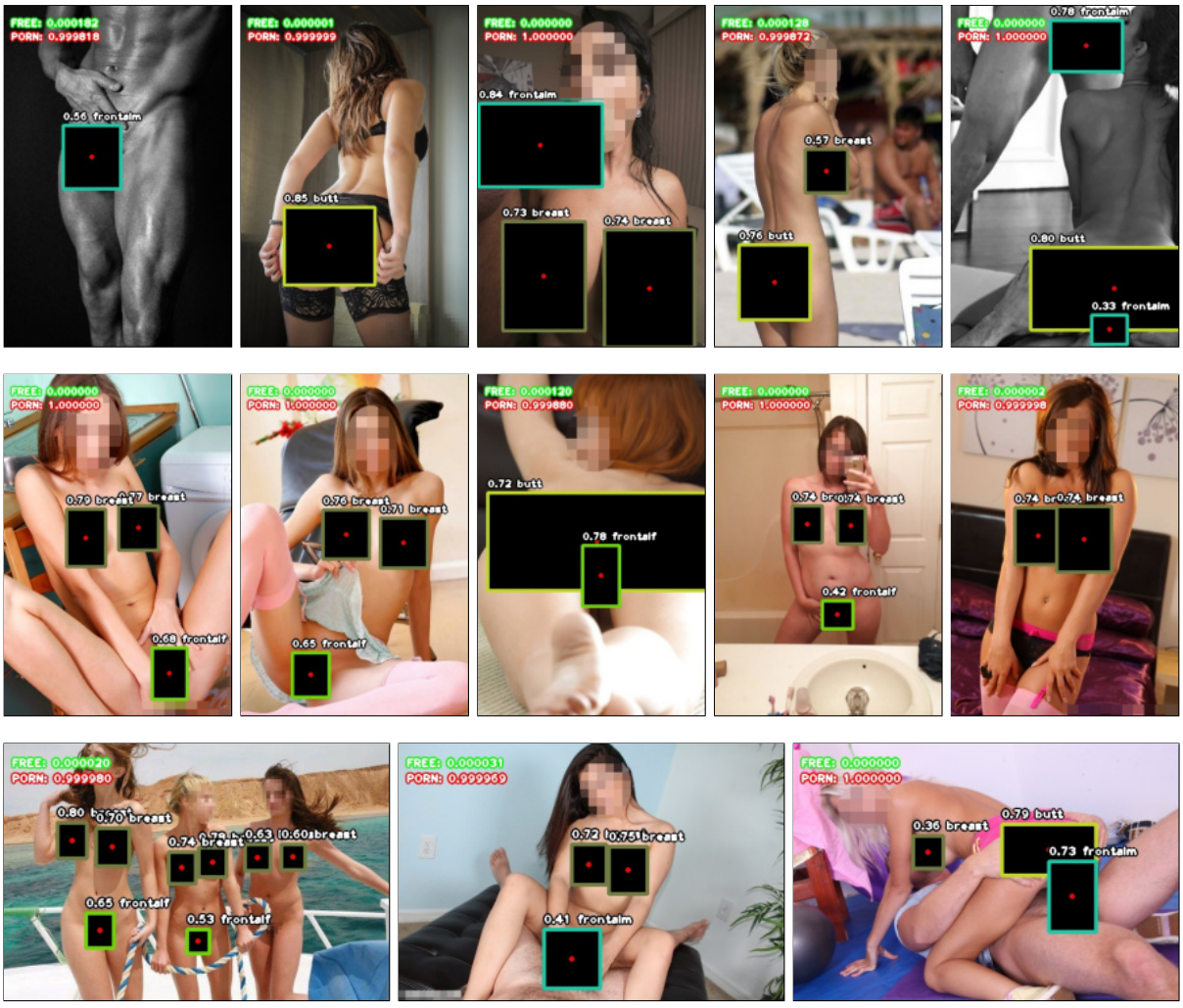


Figura 3.19: Reprodução dos 13 exemplos utilizados na Figura 3.18, aplicando o método híbrido *CensorPlus* com dimensões de entrada fixadas em 608×608.

(a) após detecção de objetos. (b) após *CensorPlus*.

Figura 3.20: Amostra de imagem da classe *free* processada pelo método de detecção de objetos e pelo método *CensorPlus*.

3.5 Considerações e Discussão

Esta etapa da tese atacou o problema da censura automática de conteúdo pornográfico como uma tarefa de detecção de objetos. Foram avaliados os métodos *Faster* R-CNN [RHGS15], SSD [LAE+16] e YOLO [RDGF16], quando foram treinados modelos utilizando *Dataset for Pornography Censorship* (*DPC*), um *dataset* para detecção de objetos que representam partes íntimas do corpo relacionadas à pornografia. *DPC* foi construído no contexto desta tese e representa, pelo melhor que se conhece da literatura, o único *dataset* destinado especificamente para detecção de partes íntimas do corpo relacionadas à pornografia. *DPC* é composto por 6.541 objetos anotados, distribuídos em 3.000 imagens. Os objetos foram rigorosamente anotados e revisados, seguindo um protocolo de validação cruzada.

As experiências com os diferentes métodos de detecção permitiram a criação de uma nova arquitetura totalmente convolucional para a tarefa de detecção, chamada *CensorNet*. Os resultados observados apontam que *CensorNet* é mais leve, em termos de quantidade de parâmetros, que YOLO-Tiny, e atinge resultados preditivos melhores, tanto em tempo quanto mAP. *CensorNet* permite a geração de modelos mais leves em termos de processamento e uso de memória, facilitando sua aplicação em dispositivos com recursos restritos.

Ao identificar que o método YOLO, mesmo mais simples que os outros métodos avaliados, atinge desempenho preditivo semelhante aos melhores resultados observados e apresenta melhor desempenho em função do tempo, a tese seguiu avaliando a tarefa de censura ao conteúdo pornográfico especificamente com este método. O *framework* original foi portado para plataformas abertas, flexibilizando a criação de novas estratégias e o uso de diferentes arquiteturas, disponibilizando as implementações para o uso da comunidade. Foi analisado o comportamento de modelos para detecção de objetos treinados com intenso aumento de dados e aplicação de múltiplas escalas, atingindo resultados superiores aos observados nos experimentos que utilizaram o método original. Observou-se também que a variação das dimensões do volume de entrada em fase de predição impacta os resultados, especialmente para os objetos das classes *frontalM* e *frontalF*.

Após contrastar os resultados obtidos para a tarefa de detecção de objetos utilizando arquitetura Darknet-19 [RF16] com os resultados de classificação de imagens utilizando a mesma arquitetura, apresentados e discutidos na Seção 2.5.2, foi construída uma nova arquitetura de rede composta por 2 saída: i) classificação de imagens e ii) detecção de objetos. Esta abordagem foi chamada de *CensorPlus* e combina a solução para os 2 problemas em um único fluxo. Mesmo demonstrando degradação de resultados preditivos em relação aos métodos de classificação e detecção isolados, *CensorPlus* atingiu resultados superiores aos observados no experimento inicial que treinou modelos com os formatos originais de *Faster* R-CNN [RHGS15], SSD [LAE+16] e YOLO [RDGF16].

Como contribuições específicas, esta etapa da tese deixa *Dataset for Pornography Censorship* (*DPC*), um *dataset* para detecção de objetos que representam partes íntimas do corpo relacionadas à pornografia, além de uma variação deste *dataset* para treinamento de modelos híbridos para classificação de imagens e detecção de objetos; a portagem do método YOLO para *frameworks* de *Deep Learning* difundidos entre a comunidade (Keras/Tensorflow [C+, AAB+]); *CensorNet* um método baseado em YOLO que utiliza uma arquitetura composta por blocos formados por *Separable Convolutions*, sendo mais leve que todos os outros métodos analisados; *CensorPlus*, um método híbrido para classificação de imagens e detecção de objetos relacionados à pornografia; ferramentas para anotação de imagens para tarefas de detecção de objetos compatível com o formato PASCAL VOC [EEVG+15], além de diversos modelos de detecção de partes íntimas com mAP que atinge até $0,7$ identificando as classes *butt*, *breast*, *frontalM* e *frontalF*.

4. GERAÇÃO AUTOMÁTICA PARA CENSURA DE PARTES ÍNTIMAS

Esta tese tem por objetivo desenvolver métodos baseados em ConvNets que possibilitem a censura automática de conteúdo relacionado à pornografia. No presente capítulo, a tarefa será abordada como um problema de geração automática de vestuário, neste caso o desenho de biquínis falsos para cobrir nudez. Conforme abordado nos Capítulos 2 e 3, até então esta tese havia atacado o problema de censura automática de pornografia utilizando estratégias baseadas em classificação de imagens e em detecção de objetos em imagens. Mesmo efetivas na identificação da ocorrência dos conteúdos, a classificação e a detecção de objetos não oferecem uma maneira pouco invasiva para censurar as imagens, o que se busca agora encontrar com métodos de geração automática.

O primeiro trabalho que possibilita a censura de conteúdo relacionado à pornografia baseado em ConvNets, apresentado em [Mou15], faz classificação de imagens, tanto isoladas quanto para *frames* de vídeos. Esta abordagem permite, ao identificar a ocorrência de pornografia, remover completamente o *frame*, sendo efetivo no sentido de censura, mas impactante com relação ao conteúdo, que passará a apresentará lacunas. Para minimizar estes problemas, abordagens baseadas em detecção de objetos podem ser menos invasivas já que, ao identificar as áreas especificamente relacionadas aos elementos pornográficos, permitiriam a aplicação de censuras mais discretas, como tarjas ou borrões direcionados nas áreas identificadas. Por outro lado, a aplicação de censuras direcionadas não permite descaracterizar o apelo inapropriado da imagem, já que a própria presença de tarjas denuncia as características da mesma. Uma alternativa às abordagens baseadas em classificação ou detecção de objetos são as gerações automáticas produzidas por *Generative Adversarial Network* (GANs).

Na tentativa de criar uma abordagem não-intrusiva para censurar pornografia, More et al. [MSWB18] trataram o problema como uma tarefa de tradução imagem-para-imagem, onde imagens pertencentes a um domínio A (mulheres nuas) são convertidas para outro domínio B (mulheres vestindo biquíni). Estes métodos tem a vantagem de traduzir imagens sem a necessidade de supervisão explícita, eliminando a necessidade de *datasets* com anotações de localização ou segmentação de objetos, ou mesmo de alinhamento de instâncias, onde uma mesma pessoa em uma mesma pose deverá estar nua e, em seguida, vestindo biquíni. Estes métodos contornam a ausência de supervisão utilizando 2 conjuntos de domínios, cada um deles representando um estado específico A ou B. Desta maneira, haverá um gerador treinado para mapear $G : A \rightarrow B$, que será capaz de traduzir imagens com mulheres nuas para sua contrapartida, mulheres vestindo biquínis. Outra contribuição apresentada em [MSWB18] é a construção de um novo *dataset* não-alinhado contendo tanto imagens de mulheres nuas quando mulheres vestindo biquíni.

Para desempenhar a tarefa de censura automática de conteúdo adulto, esta tese evoluiu o método CycleGAN [ZPIE17], utilizado em [MSWB18], tornando-o mais efetivo na tradução de elementos específicos e detalhistas presentes em uma imagem, como peças de roupa, por exemplo. O método CycleGAN [ZPIE17] é capaz de traduzir imagens entre diferentes domínios, tarefa conhecida

como tradução imagem-para-imagem. O método não necessita de *datasets* anotados ou alinhados, contornando problemas como a indisponibilidades de *datasets* para domínios específicos, que dado o tema, é uma problemática presente nesta tese.

O motivação do trabalho de More et al. [MSWB18] busca evitar indispor a experiência de usuários que consomem conteúdo que ocasionalmente pode conter nudez. O fluxo da solução proposta pelos autores é baseado no método CycleGAN [ZPIE17], que é adequado para o problema específico. More et al. [MSWB18] incrementam o método CycleGAN com a remoção do plano de fundo das imagens de entrada durante o treinamento, permitindo que o gerador foque no assunto principal (neste caso, o corpo das mulheres), gerando melhores resultados. Por outro lado, essa estratégia tem a desvantagem de perder completamente o plano de fundo das imagens, causando nítida interferência na experiência do usuário.

Recentemente, Mo et al. [MCS19] propuseram um método que incorpora as informações de múltiplos objetos alvo no fluxo de uma GAN, criando o que chamaram de *Instance-Aware* GAN (InstaGAN), que traduz tanto a imagem quanto seu correspondente conjunto de atributos de instância, ao mesmo tempo em que mantém as propriedades de invariância. O método utiliza máscaras de segmentação de objetos para informar instâncias já que, ao representarem as posições e os contornos, independentemente de cores e plano de fundo, são uma boa representação para os formatos dos objetos. Por outro lado, este método depende de dados rotulados para a tarefa de segmentação semântica (conhecido como anotação pixel-a-pixel) para o treinamento dos modelos, dificultando sua aplicação em problemas genéricos que não dispõe de *datasets* rotulados para esta tarefa.

Esta etapa da presente tese apresenta um método que busca contornar as restrições para tratar o problema da geração automática de censura, ao mesmo tempo em que mantém a qualidade das imagens geradas, minimizando os problemas relatados em More et al. [MSWB18] e Mo et al. [MCS19]. Os resultados gerados apresentam coberturas coerentes com biquínis reais, além de preservar características periféricas da imagem, especialmente o plano de fundo. O foco da transformação gerada na imagem de entrada é guiado para a parte do corpo que deve ser coberta. A solução é coposta por uma ConvNet multi-rótulo treinada para identificar 5 classes relacionadas às partes do corpo: i) *butt*, ii) *breast*, iii) *frontalM*, iv) *frontalF*, e v) *free*, onde *free* representa a ausência de partes sensíveis. Esta rede implementa uma arquitetura estruturada para a geração de máscaras de atenção escaladas [VSP+17], que focam no assunto relacionado às partes sensíveis presentes na imagem de entrada como um todo. Para incrementar o método original, as máscaras de atenção são incorporadas ao volume de entrada do gerador de uma CycleGAN [ZPIE17]. Seguindo a intuição, a máscara de atenção será capaz de destacar as áreas alvo da imagem de entrada, contribuindo para conduzir o gerador de maneira que as modificação produzidas sejam direcionadas às áreas sensíveis. Finalmente, para melhorar a qualidade das áreas periféricas da imagem resultante, as máscaras de atenção são utilizadas como guias para mesclar a imagem gerada com a imagem de entrada original.

Os resultados gerados pelo método apresentado por esta tese foram avaliados em uma consulta *online* que coletou mais de 1000 opiniões de 21 pessoas diferentes. Os resultados demonstraram

que, para a tarefa de censura de nudez, a qualidade das imagens geradas pelo método desenvolvido nesta tese são superiores quando comparadas às imagens geradas pelos métodos antecessores.

4.1 Intuição para elaboração do método

Após os resultados reportados em More et al. [MSWB18], foram observados experimentos feitos com CycleGAN [ZPIE17], onde algumas imagens foram geradas utilizando os modelos pretreinados, disponibilizados pelos autores. O modelo utilizado assume cavalos no domínio A e zebras no domínio B, traduzindo imagens entre estes domínios. O exemplo ilustrado pela Figura 4.1 mostra a tradução de uma imagem composta por dois animais, onde o maior trata-se nitidamente de um cavalo. O segundo animal, menor, parece um cavalo, mas não é possível precisar. No exemplo, ambos animais foram traduzidos para zebras, coerente com a suposição de que ambos são cavalos. Por outro lado, caso o animal menor não se trate mesmo de um cavalo, seria apropriado dispor de algum mecanismo que permitisse apontar ao gerador o que e quando traduzir.

Figura 4.1: Tradução de cavalos para zebras utilizando CycleGAN [ZPIE17] com modelo pré-treinado pelos autores.

Buscando desenvolver um mecanismo que permitisse induzir o gerador, foi utilizado o método descrito por He et al. [HGDG17] para segmentar todos os cavalos presentes no *dataset* cavalos-para-zebras, disponibilizado em [ZPIE17]. Foram geradas máscaras de segmentação que abrangem todos os pixeis de cada ocorrência de cavalo no *dataset*. A partir disso, foi treinado um modelo CycleGAN [ZPIE17] em que a entrada foi modificada, passando a contar com um quarto canal que representou a máscara de segmentação, além dos 3 canais de cores da própria imagem. Após o treinamento, a rede geradora seguiu traduzindo cavalos para zebras como em sua verão original, no entanto, quando a máscara de segmentação é removida do animal menor, a rede geradora o mantém inalterado. Além do incremento de informação na entrada, nenhuma outra característica da CycleGAN [ZPIE17] foi modificada.

A Figura 4.2 ilustra a experiência onde a máscara de segmentação compreende todos os animais presentes na imagem, resultando na tradução destes para zebras. Por outro lado, a Figura 4.3 apresenta a experiência em que a segmentação do animal menor foi suprimida. Neste caso, é possível perceber que somente o cavalo maior é traduzido para zebra, enquanto o animal menor tem suas características preservadas.

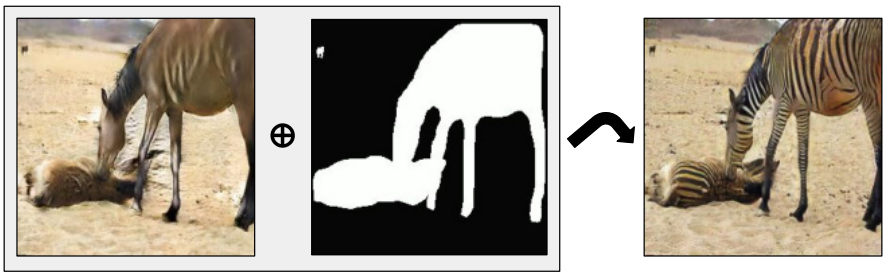


Figura 4.2: Tradução apoiada por máscaras de segmentação, onde todos os animais presentes na imagem foram segmentados.

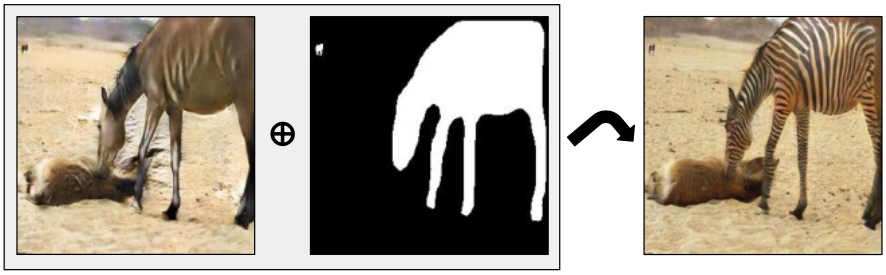


Figura 4.3: Tradução apoiada por máscaras de segmentação, onde a segmentação do animal menor foi suprimida.

Ao constatar que as máscaras de segmentação influenciaram a geração, foi definida a hipótese de que esta mesma abordagem poderia influenciar a rede para modificar somente partes específicas de uma imagem. A partir desta hipótese, o trabalho seminal de More et al. [MSWB18] foi revisado, de maneira que os planos de fundo das imagens não foram mais removidos mas, por outro lado, as máscaras de segmentação foram utilizadas como informação adicional. A Figura 4.4 representa esta hipótese, onde a máscara de segmentação reforça somente a região à ser coberta.

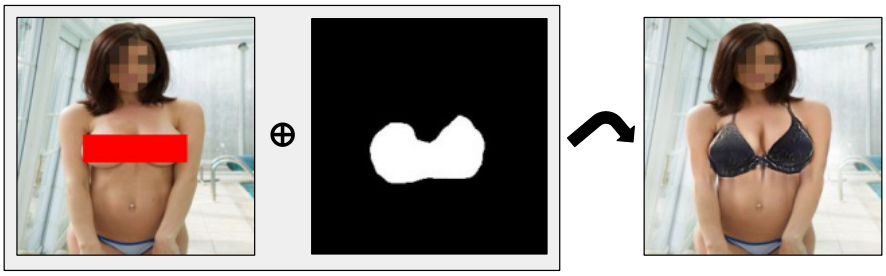


Figura 4.4: Tradução de nudez para biquíni apoiada por máscara de segmentação.

Por outro lado, a adição de máscaras de segmentação, no caso específico de tradução de nudez para biquíni, oferece um novo problema: não são disponíveis modelos previamente treinados para segmentação semântica das partes íntimas do corpo que correspondem à biquínis, como seios por exemplo. Não são disponíveis também *datasets* que permitam treinar modelos para esta tarefa. Para resolver este problema, as máscaras de segmentação foram substituídas por máscaras de atenção.

As máscaras de atenção foram geradas por uma ConvNet treinada para classificação multi-rótulo, onde foi utilizado um *dataset* anotado para detecção de objetos, que foi convertido para a tarefa de classificação. Originalmente este *dataset* dispõe de objetos anotados para 4 classes (*butt*, *breast*, *frontalM* e *frontalF*), onde 3 delas estão diretamente relacionadas às áreas que precisam ser cobertas pelos biquínis. A representação deste *dataset* foi ajustada para permitir o treinamento de uma ConvNet para classificação de 4 classes. Após o treinamento, esta rede foi ajustada para deixar de ser um classificador, passando a ser uma geradora de máscaras de atenção. As máscaras de atenção representam um mapa de calor, indicando áreas de interesse em uma determinada imagem. Nesse caso, as áreas de interesse estão alinhadas com as partes íntimas do corpo, que devem ser cobertas pelos biquínis desenhados pela rede geradora.

4.2 Trabalhos Relacionados

A seguir serão discutidos trabalhos que abordam os dois principais conceitos relacionados ao método proposto: *Generative Adversarial Networks* (GANs), no contexto da tradução imagem-para-imagem, e estudos que apresentam *datasets* e métodos para identificação/classificação de conteúdo adulto em imagens e vídeos.

4.2.1 Tradução Imagem-para-Imagem

Generative Adversarial Networks (GANs) [GPAM+14] podem ser compreendidas como um *framework* para o treinamento simultâneo de duas redes em um jogo de soma zero. Durante o treinamento, a rede geradora G produz imagens sintéticas, ao mesmo tempo em que o discriminador D aprende a identificar quando a entrada analisada é de fato real ou se é produzida artificialmente. Com o avançar do treinamento, a rede geradora G aprende a produzir imagens realistas que devem enganar a rede discriminadora D. O jogo é definido de acordo com a Equação 4.1, onde $\mathbf{z}$ é um vetor desenhado a partir de uma distribuição conhecida, normalmente gaussiana, representando o chamado espaço latente. O vetor $\mathbf{z}$ é utilizado como entrada para o gerador G, gerando uma imagem sintética que é avaliada pelo discriminador D. Finalmente, o vetor $\mathbf{I}$ representa imagens gerais amostradas do conjunto de treinamento. Nas formas iniciais de GANs, o gerador G pode gerar múltiplas imagens, variando a amostra do vetor de espaço latente $\mathbf{z}$.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{I}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] \ + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{4.1}$$

Nas GANs tradicionais, as imagens geradas são incondicionais. Por outro lado, *Conditional Generative Adversarial Networks* (CGANs) podem gerar imagens baseadas em uma certa entrada [MO14]. Este tipo de *framework* abre caminho para tarefas que focam na mudança de

características específicas de uma imagem, como ajustes em sua resolução [LTH+16] e a aplicação automática de retoques [PKD+16]. Genericamente falando, CGANs são o caminho para trabalhar com tarefas de tradução imagem-para-imagem, quando imagens de um certo domínio A devem ser mapeadas para sua imagem correspondente no domínio B.

A menos que os experimentos desejados tratem de domínios amplos e generalistas como dia e noite e inverno e verão, construir um *dataset* de imagens pareadas entre ambos os domínios será uma tarefa custosa. Para contornar esta limitação, Zhu et al. [ZPIE17] propuseram CycleGAN [ZPIE17], um método para tradução imagem-para-imagem que não depende de imagens pareadas. O método foi experimentado em diferentes contextos e apresentou resultados promissores, especialmente em tarefas onde características como cores ou texturas de grande objetos precisavam ser trocadas.

Motivados pelas potencialidades do método CycleGAN [ZPIE17], More et al. [MSWB18] estenderam o método para aplicação no problema de censura de partes íntimas. A estratégia adotada para gerar melhores resultados buscou destacar o corpo da pessoa presente na imagem. Para chegar ao destaque, o *dataset* utilizado no treinamento dos modelos foi preprocessado, quando todas as imagens foram submetidas ao método de segmentação semântica apresentado por He at al. [HGDG17], segmentado os formatos de cada corpo presente em cada imagem. A partir das segmentações obtidas, todos os pixeis que não estivessem inseridos em algum espaço segmentado foram transformados em branco. Esta solução contribuiu para melhores resultados, onde os biquínis desenhados tornaram-se mais realistas. Por outro lado, esta modificação gerou um efeito colateral indesejado, quando toda a informação periférica à segmentação dos corpos presentes nas imagens foi perdida. More et al. [MSWB18] também trazem como contribuição um *dataset* de imagens não-alinhadas para ambos domínios: A, mulheres nuas e B, mulheres vestindo biquínis.

4.3 Método

Esta tese apresenta *Attention-based Generative Adversarial Networks* (*AttGAN*), um novo método para censura de nudez em imagens que utiliza treinamento adversário em uma estratégia de tradução imagem-para-imagem para desenhar biquínis sobre o corpo de mulheres nuas. Além de gerar biquínis coerentes com os reais, o método preserva as partes periféricas da imagem, como o plano de fundo e os rostos das pessoas. O método embarca uma ConvNet de atenção, treinada para identificar partes íntimas do corpo, criando mapas de atenção que serão usados para guiar um gerador G em um fluxo de tradução imagem-para-imagem. As máscaras obtidas a partir da rede de atenção são utilizadas para adicionar informação ao volume de entrada A do gerador G, reforçando as áreas íntimas que devem ser cobertas na imagem de saída gerada B.

A Figura 4.5 ilustra a intuição inicial do método *AttGAN*, apresentando uma representação do fluxo onde consta a imagem de entrada, passando pela geração das máscaras de atenção,

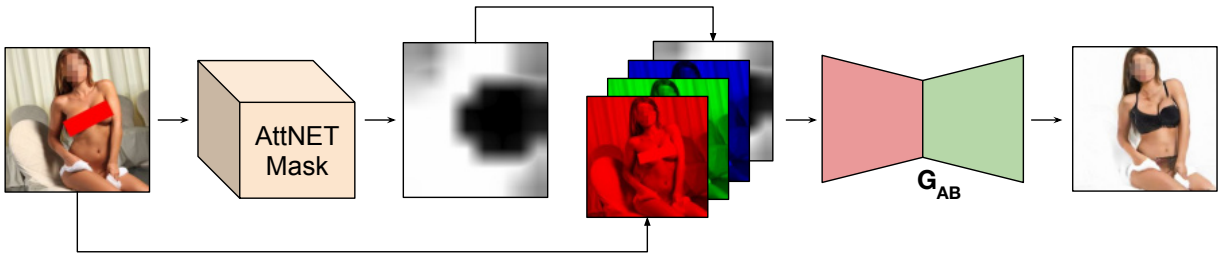


Figura 4.5: Visão geral do fluxo do método *AttGAN*.

seguindo pela composição da entrada para o gerador G_{AB} e, finalmente, resultando em uma saída automaticamente censurada.

4.3.1 Rede de Atenção - *AttNET*

A principal contribuição desta etapa da tese é a extensão do método CycleGAN [ZPIE17], onde uma rede de atenção responsável por apontar áreas de interesse em imagens que contém nudez foi incorporada ao fluxo do método original. Esta rede de atenção, nomeada *Attention Mask Generation Network* (*AttNET*), induziu a rede geradora G de maneira que esta focasse em regiões íntimas do corpo para produzir censuras automáticas coerentes com biquínis reais. *AttNET* é uma ConvNet construída com base na arquitetura ResNet-152 [HZRS16], tendo sua saída reestruturada para gerar mapas de atenção.

Para treinar a rede de atenção *AttNET*, idealmente seria necessário dispor de um *dataset* anotado para identificar a presença de determinadas partes do corpo (como seios, por exemplo), no entanto, nenhum dos *datasets* relatados dispunha deste tipo de característica. Para contornar a falta de dados de treinamento, foi adotada uma estratégia que converteu *Dataset for Pornography Censorship* (*DPC*), o *dataset* anotado para detecção de partes íntimas do corpo descrito na Seção 3.2 desta tese, transformando-o em um *dataset* para treinamento de modelos de classificação multi-rótulo.

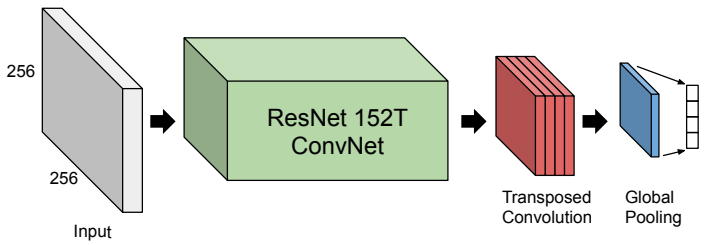


Figura 4.6: *AttNET* estruturada para treinamento de classificação com 4 classes.

Para aprender a gera os mapas de atenção, foi utilizado uma ResNet-152 [HZRS16] pre-treinada com ImageNet [DDS+09], onde foram removidas as duas últimas camadas: uma camada totalmente conectada e um *pooling* global. Após remover essas duas camadas, foi adicionada uma convolução com N_C filtros, resultando em um volume de saída com dimensões $L_S \times A_S \times N_C$,

onde L_S representa a largura e A_S é a altura dos mapas de ativação. A saída da rede gera um mapa de ativação distinto para cada classe, onde a camada convolucional adicionada tem N_C filtros, sendo N_C o número de classes que representam as partes íntimas buscadas. Para gerar os *scores* de classificação, foi aplicada uma camada de *pooling* global, de maneira que o as dimensões espaciais dos N_C mapas de ativação gerados pela camada convolucional anteriormente adicionada sejam sumarizadas, passando a representar N_C probabilidades independentes.

A Figura 4.6 retrata *AttNET*, uma ConvNet baseada em ResNet-152 [HZRS16], modificada para permitir o treinamento como um classificador de imagens dentre as 4 classes representadas pela versão de *DPC* adaptada para classificação. Foram adicionadas duas camadas na saída da rede, sendo uma convolução com 4 filtros, podendo esta ser substituída por uma convolução transposta, e finalmente uma camada de *pooling* global, responsável por sumarizar a saída em 4 valores independentes. O vetor de saída N_C-dimensional foi ativado com função $\text{SIGMOIDE}\ \sigma$, transformando valores lineares em predições independentes descritas por $\hat{\mathbf{Y}}$. Ao sumarizar o espaço dimensional diretamente para o espaço de classes, as áreas da imagem relacionadas a cada classe foram reforçadas, gerando *scores* grandes o suficiente para ultrapassarem aqueles gerados para às classes não presentes em uma determinada imagem.

A rede de atenção foi treinada para classificação multi-rótulo, já que uma imagem pode exibir diferentes partes do corpo ao mesmo tempo. Assim, similarmente ao reportado por [WBDC17, WCB18, WB17a], foi otimizada a entropia binária cruzada, definida na Equação 4.2, onde T_B representa o número de instâncias em um mini-batch, N_C é o número de classes, $\mathbf{Y}_{ij}$ é o j-ésimo rótulo real para a i-ésima instância e $\hat{\mathbf{Y}}$ representa as predições geradas por *AttNET*.

$$-\frac{1}{T_B}\sum_{i=1}^{T_B}\sum_{j=1}^{N_C}[\mathbf{Y}_{ij}\times log(\hat{\mathbf{Y}}_{ij})+(1-\mathbf{Y}_{ij})\times log(1-\hat{\mathbf{Y}}_{ij})] \tag{4.2}$$

Após o treinamento utilizando *DPC*, a rede de atenção foi modificada quando a última camada (o *pooling* global) foi removida, restando agora somente os mapas de ativação gerados pela última convolução. As posições espaciais resultantes destes mapas foram normalizadas no intervalo $[0,1]$ aplicando a ativação SOFTMAX na profundidade de cada um dos N_C mapas de ativação, gerando $L_S \times A_S \times N_C$ probabilidades relacionados às classes. Finalmente, todos os mapas representando cada parte do corpo foram agregados a partir de seu maior valor, utilizando a operação MAX-POOLING. Essa agregação resultou em um volume de atenção de tamanho $L_S \times A_S \times 1$, representando todas as regiões responsáveis por exibir partes íntimas do corpo. Para compatibilizar a máscara de atenção com a imagem de entrada foi aplicada a interpolação bilinear, escalando o volume de atenção de acordo com as dimensões da entrada, resultado na máscara de atenção.

A Figura 4.7 representa a rede *AttNET* após os ajustes para geração de máscaras de atenção. Para uma determinada imagem de entrada é gerado um mapa de atenção que representa as áreas de interesse que contribuirão para o treinamento guiado da rede geradora G. No exemplo, as partes escuras representam as áreas relacionadas à exibição de partes íntimas do corpo.

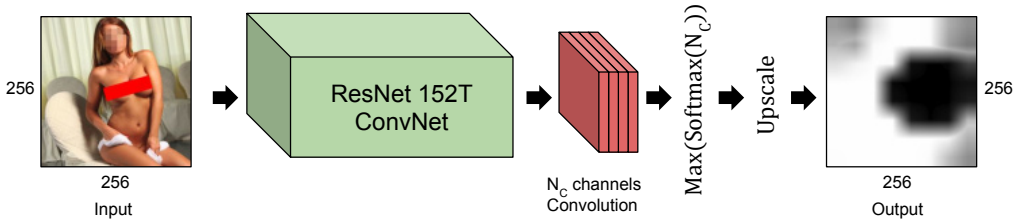


Figura 4.7: Visão da estrutura e do fluxo para geração de máscara de atenção da rede *AttNET*.

A introdução do mapa de atenção no gerador para tradução imagem-para-imagem deveria fazer com que este traduzisse somente áreas da imagem que contivesse conteúdo sensível, mantendo inalteradas as áreas relacionadas ao conteúdo livre. A passada na rede de atenção é utilizada para gerar os mapas de atenção denotados por $AN(I) = M_A$, onde M_A é a máscara de atenção de dimensões $L_S \times A_S$ gerada a partir da imagem de entrada I.

4.3.2 *AttGAN*

A primeira abordagem desenvolvida para combinar máscaras de atenção com a consistência de ciclo da tradução imagem-para-imagem foi chamada de *AttGAN*. No método *AttGAN*, as máscaras de atenção geradas por AN são utilizadas como informação adicional no fluxo de uma CycleGAN, onde as máscaras são concatenadas na entrada como novos canais de informação, aplicadas tanto no gerador G_{AB} quanto no gerador G_{BA}. A Figura 4.8 ilustra o método *AttGAN*, encadeando suas 4 etapas.

A solução é inspirada em [ZPIE17], onde foram preservados os geradores G e o discriminador D_B. As arquiteturas, tanto dos geradores quanto do discriminador, foram baseadas em More et al. [MSWB18], que utilizou a arquitetura ResNet [HZRS16] com 9 blocos como gerador, funcionando como um *autoencoder* que entrepõe conexões residuais e camadas *bottleneck*, inspiradas em [JAFF16].

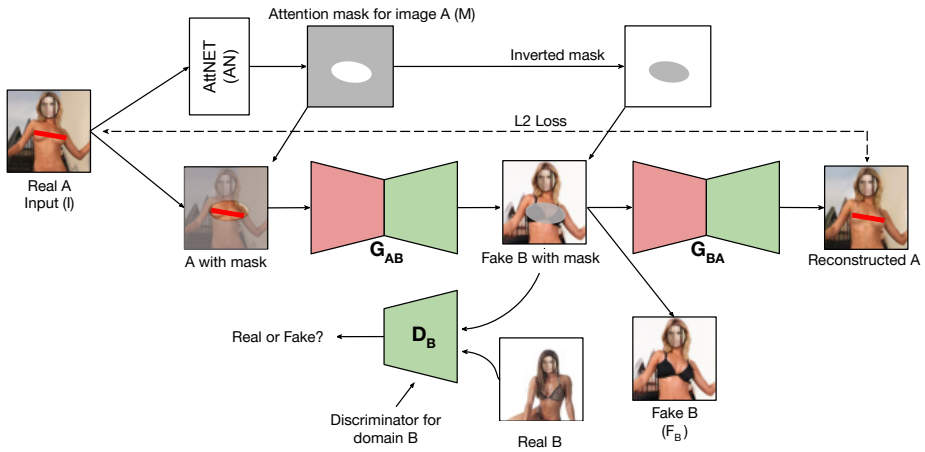


Figura 4.8: Encadeamento das 4 etapas de *AttGAN* inseridos no fluxo da CycleGAN [ZPIE17].

A partir da adição da rede *AttNET* para geração de máscaras de atenção, representada por AN, o método *AttGAN* implementa as 4 etapas seguintes: i) geração da máscara de atenção, quando a imagem de entrada A passa pela ConvNet de atenção, resultando em um mapa escalado de acordo com as dimensões da imagem de entrada, ii) a máscara de atenção é concatenada ao volume como uma dimensão adicional, criando um novo canal de informação na imagem de entrada do gerador G_{AB}, iii) o inverso da máscara de atenção é concatenado à imagem gerada (Fake B), resultando no volume que será passado para reconstrução no gerador G_{BA} e, finalmente, iv) a imagem reconstruída é comparada com a imagem original I. O resultado esperado para solucionar o problema de censura de partes íntimas é dado por *Fake B*.

4.3.3 *AttGAN+*

O método *AttGAN+* é uma evolução de *AttGAN* que adiciona novos passos em diferentes ordens para melhorar a geração de censuras automáticas sobre nudez. Nesta segunda abordagem, além da máscara de atenção gerada por AN ser mantida como informação adicional no fluxo de entrada dos geradores G_{AB} e G_{BA}, a máscara também é somada em todos os canais de saída da primeira camada convolucional dos geradores G_{AB} e G_{BA}.

A Figura 4.9 representa graficamente os 6 passos que implementam o método, sendo eles: i) geração das máscaras de atenção, quando a imagem de entrada A passa pela ConvNet de atenção, resultando em um mapa escalado de acordo com as dimensões da imagem de entrada, ii) a máscara de atenção é concatenada ao volume como uma dimensão adicional, criando um novo canal de informação na imagem de entrada do gerador G_{AB}, iii) a máscara de atenção é somada a todos os canais de saída da primeira convolução do gerador G_{AB}, iv) o inverso da máscara de atenção é concatenado à imagem gerada (Fake B), resultando no volume que será passado para reconstrução no gerador G_{BA}, v) o inverso da máscara de atenção é somado a todos os canais de saída da primeira convolução do gerador G_{BA} e, finalmente, vi) a imagem reconstruída é comparada com a imagem original I.

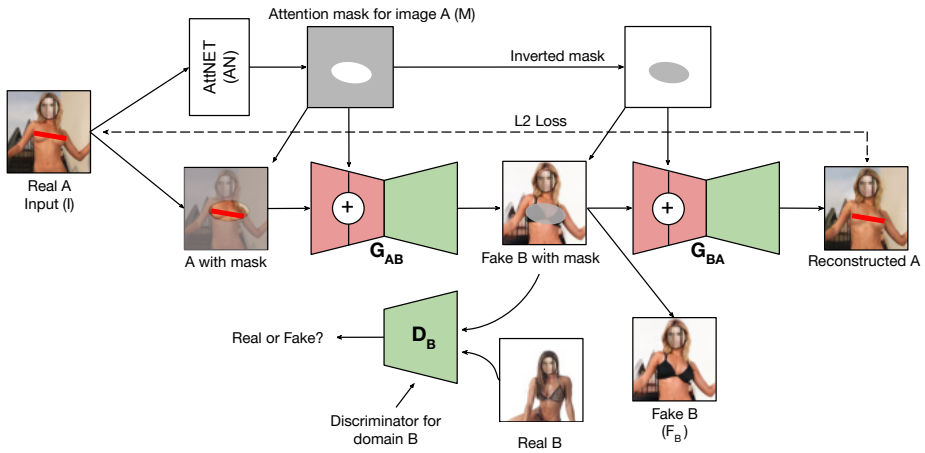


Figura 4.9: Encadeamento das 6 etapas de *AttGAN+* inseridos no fluxo da CycleGAN [ZPIE17].

4.3.4 *AttGAN++*

A terceira variação do método, chamada *AttGAN++*, incrementa *AttGAN+* ao mesclar a saída do gerador G_{AB} com a imagem de entrada original I. A mescla das imagens é guiada pelo mapa de atenção gerado por AN, de maneira que somente as áreas de interesse definidas na máscara de atenção são preenchidas pelo biquíni desenhado pelo gerador G_{AB}. As áreas não relacionadas pela máscara de atenção M_A são preenchidas pela própria imagem original, contribuindo para a manutenção de partes periféricas da imagem, como plano de fundo e rostos.

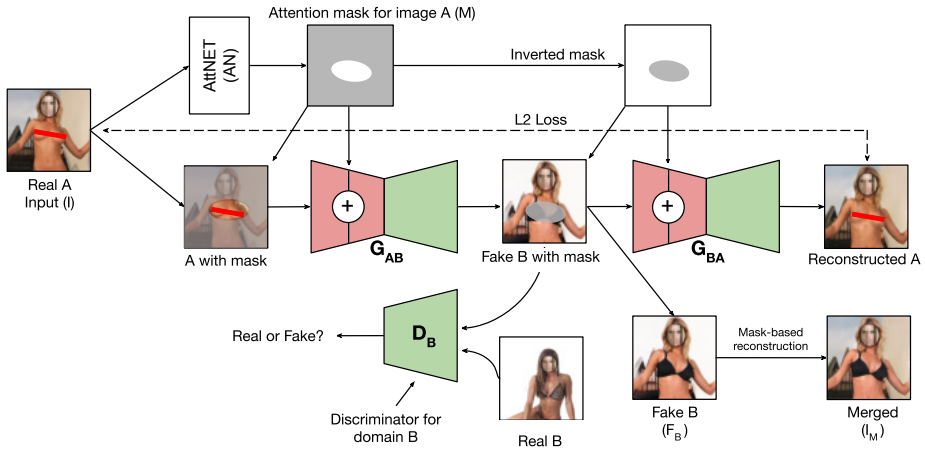


Figura 4.10: Encadeamento das 7 etapas de *AttGAN++* inseridos no fluxo da CycleGAN [ZPIE17].

A Figura 4.10 ilustra os 7 passos do método *AttGAN++*, onde os primeiros 6 passos são idênticos aos descritos em *AttGAN+*, adicionando um novo passo ao final da sequência, sendo ele: vii) o casamento da imagem de saída do gerador G_{AB} com a imagem de entrada original I, guiada pela máscara de atenção M_A. A imagem obtida após a mescla I_M é nitidamente mais rica em detalhes periféricos, como plano-de-fundo e rostos das pessoas, ao mesmo tempo em que preserva as peças do biquíni desenhado pelo gerador G_{AB}. A Equação 4.3 formaliza a mescla I_M da imagem original I com a imagem traduzida para o domínio B, representada por F_B, guiada pela máscara de atenção M_A. A função Inv retorna o inverso da máscara de atenção, passando a destacar aquilo que não está relacionado com partes íntimas do corpo.

$$I_M = (I \times Inv(M_A)) + (F_B \times M_A) \tag{4.3}$$

Todos os métodos resultantes desta etapa da tese, seja *AttGAN*, *AttGAN+* ou *AttGAN++*, foram construídos com foco na geração de coberturas automáticas para censura de conteúdo impróprio. Desta maneira, a etapa de reconstrução, onde a imagem traduzida para o domínio B retorna ao domínio A, foi mantida unicamente por ser parte da consistência de ciclo do método CycleGAN [ZPIE17], não havendo qualquer tipo de interesse ou avaliação sobre este conteúdo.

4.4 Configuração dos Experimentos

Nesta seção serão apontados os *datasets* utilizados para o treinamento da ConvNet de atenção *AttNET*, e para o treinamento dos modelos do método *AttGAN* e suas variações. Também será descrito o processo de avaliação pelo qual passaram as imagens geradas pelos modelos obtidos.

4.4.1 *Datasets*

A rede de atenção *AttNET* foi treinada inicialmente para classificação multi-rótulo, gerando predições independentes para 4 classes: *butt*, *breast*, *frontalM* e *frontalF*. Neste contexto, as classes representam a ocorrência de pelo menos um dos elementos que devem ser cobertos com biquínis. A rede *AttNET* foi treinada com *DPC*, um *dataset* originalmente construído para tratar o problema de censura à pornografia como uma tarefa de detecção de objetos. Para treinar os modelos das variações de *AttGAN*, foi utilizado o *dataset* não-alinhado de mulheres nuas e mulheres vestidas com biquínis apresentado em More et al. [MSWB18].

4.4.1.1 DPC: Dataset for Pornography Censorship

O *dataset DPC*, descrito na seção 3.2, foi projetado para detecção de partes íntimas do corpo que são relacionadas à pornografia. *DPC* é composto por 3.000 imagens contendo exposição de partes íntimas do corpo. Cada imagem contém pelo menos 1 objeto relacionado a umas das seguintes classes: *butt*, *breast*, *frontalM* e *frontalF*. *DPC* é dividido em conjuntos de treino (2.100 imagens), teste (600) e validação (300).

Para treinar *AttNET*, *DPC* foi adaptado para a tarefa de classificação multi-rótulo com base na simples ocorrência de elementos relacionados a qualquer uma das 4 classes do *dataset*. Desta maneira, a tarefa deixa de ser a geração de predições que apontem as posições espaciais (x, y) e largura e altura (w, h), relacionados a ocorrência de um objeto de uma das classes, passando a ser a simples ocorrência ou não de determinado tipo de objeto.

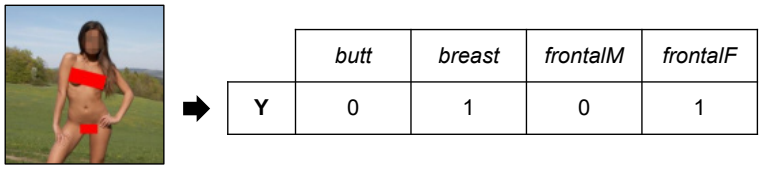


Figura 4.11: Conversão de detecção de objetos para classificação multi-rótulo.

A Figura 4.11 ilustra a conversão de uma imagem anotada, contendo 2 objetos da classe *breast* e 1 objeto da classe *frontalF*, para um vetor binário Y que representa a ocorrência de pelo menos 1 elemento para cada uma das classes.

4.4.1.2 Dataset dos Biquínis

O *dataset* dos biquínis que foi utilizado para treinar, validar e experimentar *AttGAN* e suas variações é o mesmo apresentado em More et al. [MSWB18]. Para construir esse *dataset*, os autores buscaram na Internet imagens de mulheres nuas, representando o domínio A, e de mulheres vestindo biquínis, represetando o domínio B. As imagens não são alinhadas, onde as poses e os posicionamentos das diferentes imagens são totalmente independentes. No entanto, foram utilizadas somente imagens contendo uma única pessoa para ambos os domínios. O *dataset* foi dividido em conjuntos disjuntos de treino e teste, contendo 1.965 imagens para treino e outras 220 imagens para teste. A classe que relaciona mulheres nuas (domínio A) dispõe de 921 imagens para treino e 103 para teste, enquanto que na classe que relaciona mulheres vestindo biquínis (domínio B), são 1.044 imagens para treino e 117 para teste.

4.4.2 Hiperparâmetros

O treinamento dos modelos foi iniciado do zero, não utilizando modelos pré-treinados, sendo a mesma estratégia utilizada por Zhu et al. [ZPIE17]. O treinamento foi executado por 500 épocas, com taxa de aprendizado inicial de 10^{-4}, mantida fixa por 100 épocas, passando a decair linearmente até zero por 400 épocas. Demais hiperparâmetros foram mantidos os mesmos utilizados por More et. al. [MSWB18]. Tanto geradores G_{AB} e G_{BA} quanto o discriminador D_B otimizaram as mesmas funções de custo descritas em [ZPIE17]. Foram feitos experimentos para avaliar duas variações do método: *AttGAN* e *AttGAN+*. O método *AttGAN++* foi construído após as avaliações, mantendo a porção gerada da imagem exatamente igual aquela gerada pelo método *AttGAN+*, passando a mesclar a saída F_B do gerador G_{AB} com a entrada original I, usando a máscara de segmentação M_A como guia.

4.4.3 Avaliação

Para avaliar qualitativamente os resultados gerados, foi criado um formulário composto de 50 grupos de imagens do conjunto de teste do *dataset* apresentado por More et al. [MSWB18]. O formulário foi respondido por 21 avaliadores que compararam as mesmas imagens censuradas automaticamente pele método descrito em More et al. [MSWB18], considerado o *baseline*, com os métodos *AttGAN* e *AttGAN+*. O formulário questionou os avaliadores quanto sua percepção sobre a qualidade dos biquínis desenhados nas imagens apresentadas. O formulário ofereceu também uma opção neutra, que deveria ser marcada quando o avaliador não encontrasse nenhum método com resultado suficientemente coerente com a tarefa de desenhar biquínis sobre as partes nuas do corpo. As opções de resposta foram randomizadas para evitar a geração de viés sob um método em específico. O método *AttGAN++* não foi incluído nesta avaliação, dado que foi construído após a

realização dessa avaliação. O Apêndice C apresenta 3 recortes ilustrativos (Figuras C.1, C.2 e C.3) do formulário aplicado para avaliação dos métodos, exemplificando 3 grupos de imagens avaliadas.

4.5 Experimentos

A seguir serão expostos os experimentos que apontam a performance para as 3 variações apresentadas do método: *AttGAN*, *AttGAN+*, e *AttGAN++*. Todas as variações mantiveram hiperparâmetros e funções de custo equivalentes as utilizadas em [MSWB18]. As dimensões de entrada das imagens foram mantidas em 256×256 para a rede de atenção e para os geradores G_{AB} e G_{BA}. A saída da rede de atenção AN tem dimensões de 8×8, sendo essa saída escalada de acordo com as dimensões da imagem de entrada. A saída foi escalada utilizando interpolação bilinear, tornando a máscara compatível com as dimensões da entrada dos geradores, ao mesmo tempo em que suaviza os contornos das áreas de interesse. A Figura 4.12 ilustra a melhora na qualidade das coberturas geradas em 5 diferentes épocas do treinamento, todas geradas pelo método *AttGAN++*, utilizando a mesma amostra do conjunto de teste. Observando as imagens, torna-se nítida a melhora na geração dos biquínis com o passar das épocas.

(a) 100 épocas (b) 200 épocas (c) 300 épocas (d) 400 épocas (e) 500 épocas

Figura 4.12: Amostra do domínio A traduzida para o domínio B pelo método *AttGAN++* com modelos colhidos em 5 diferentes épocas de treinamento. Por motivos de privacidade, os rostos presentes foram descaracterizados.

4.5.1 Resultados da Rede de Atenção

A rede *AttNET* foi treinada para classificação de 4 classes relacionadas a partes íntimas do corpo, sendo posteriormente convertida para a geração de mapas de atenção. Mesmo não sendo este o objetivo da criação e do treinamento dos modelos, o desempenho preditivo para a tarefa de classificação está relacionado a qualidade das áreas de interesse representadas pelas máscaras de atenção. Assim, todos os modelos de classificação *AttNET*, construídos a partir de variações de ResNet [HZRS16], foram avaliados observando a acurácia de classificação. Foram utilizados o conjunto de validação de *DPC*, o *dataset* utilizado para treinamento dos modelos *AttNET*, e o conjunto de treinamento do *dataset* de biquínis [MSWB18], utilizado para treinar os modelos de geração das variações do método *AttGAN*.

Tabela 4.1: Acurácia de classificação observada para diferentes variações de *AttNET*.

Arquitetura Base	Validação (*DPC*)	Treinamento (Biquínis)
ResNet-34	97,83%	98,42%
ResNet-34T	96,17%	96,92%
ResNet-50	97,33%	96,17%
ResNet-50T	97,33%	97,67%
ResNet-101	97,33%	97,82%
ResNet-101T	97,17%	98,57%
ResNet-152	97,67%	97,90%
ResNet-152T	**97,90%**	**98,65%**

A Tabela 4.1 exibe os resultados das 4 versões da rede de atenção AN, ResNet-[34, 50, 101, 152], mantendo a arquitetura compatível com classificação. Foram comparados os desempenhos das redes utilizando a última camada convolucional convencional com variações que utilizaram convolução transposta (representadas por *T*). Todos os modelos foram treinados e avaliados utilizando o conjunto de validação do *dataset DPC*. Adicionalmente foi verificada a acurácia para detecção da presença de nudez no conjunto de treinamento do *dataset* de Biquínis, construído por More et al. [MSWB18]. Os resultados demonstram que, para redes mais profundas, o uso de convoluções transpostas contribui para atingir melhor performance preditiva nos diferentes *datasets*. Pode-se perceber que o modelo com melhor performance utiliza arquitetura ResNet-152T, superior a todos os outros em ambos os conjuntos de avaliação. A partir dessa constatação, este modelo foi utilizado para geração de máscaras de atenção, integrado ao fluxo da CycleGAN e aplicado em todas as variações da *AttGAN*.

4.5.2 Resultados de Geração

Para comparar a capacidade de geração de coberturas automáticas entre o método *baseline* e as 3 variações do método *AttGAN*, foram treinados modelos por 500 épocas para cada uma das variações de *AttGAN*. Os 3 treinamentos utilizaram máscaras de atenção produzidas por uma *AttNET* com arquitetura baseada em ResNet-152T, a arquitetura que apresentou os melhores resultados nas avaliações de classificação no *dataset DPC* e no *dataset* apresentado por More et al. [MSWB18]. Os modelos foram utilizados para censurar amostras do conjunto de teste que foram analisadas por avaliadores humanos, quando estes puderam apontar o método que gerou as melhores censuras. O mesmo modelo de atenção *AttNET* utilizado para treinar as variações de *AttGAN* foi utilizado para gerar as censuras avaliadas.

A Figura 4.13 exibe 8 amostras do conjunto de teste do *dataset*, representadas pela primeira coluna, indicada por 4.13a. Estas amostras foram traduzidas do domínio A (mulheres nuas) para o domínio B (mulheres vestindo biquínis) pelo método *baseline* [MSWB18] e pelas 3 variações do método *AttGAN*. A coluna representada pela Figura 4.13b ilustra saídas geradas pelo

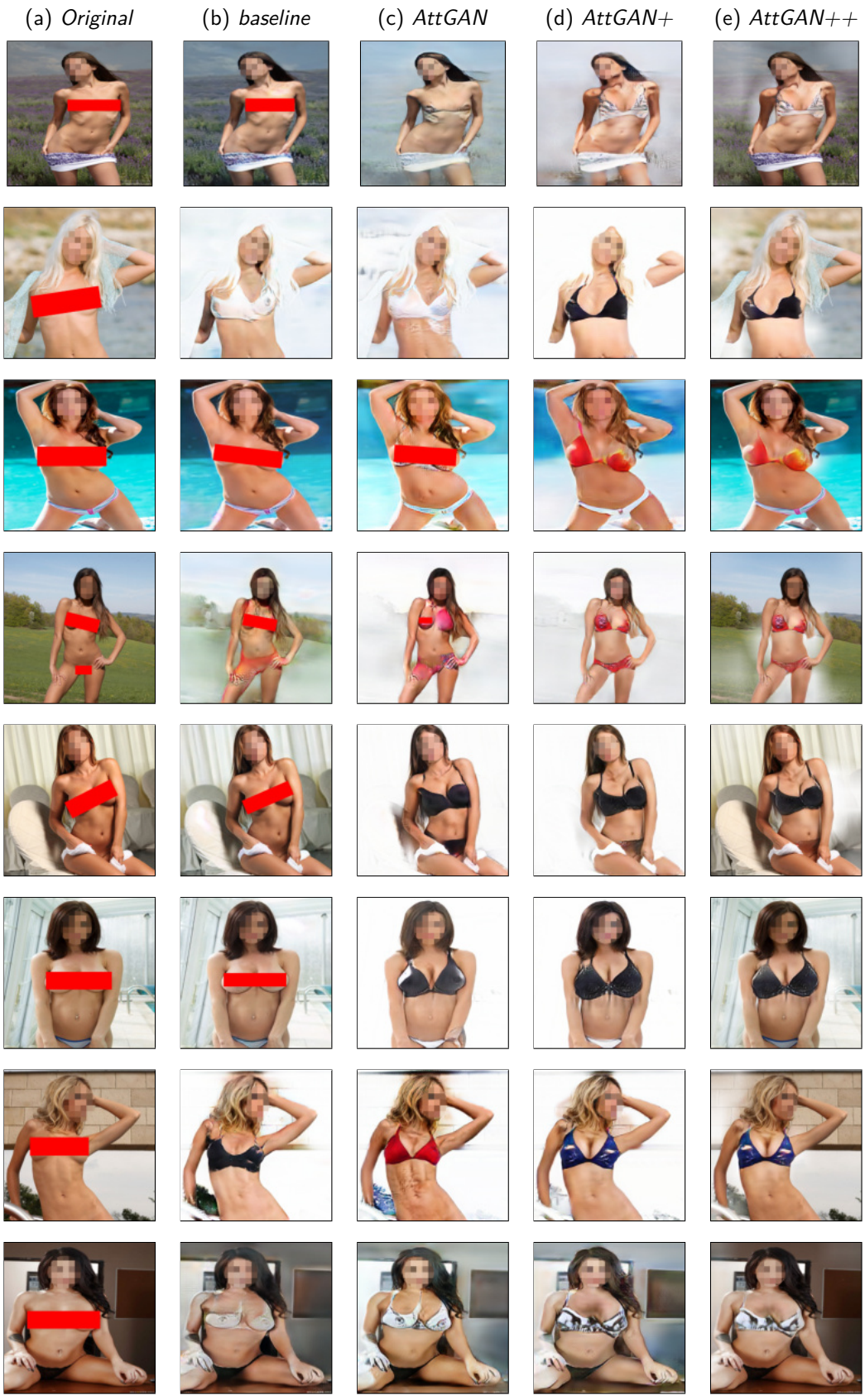
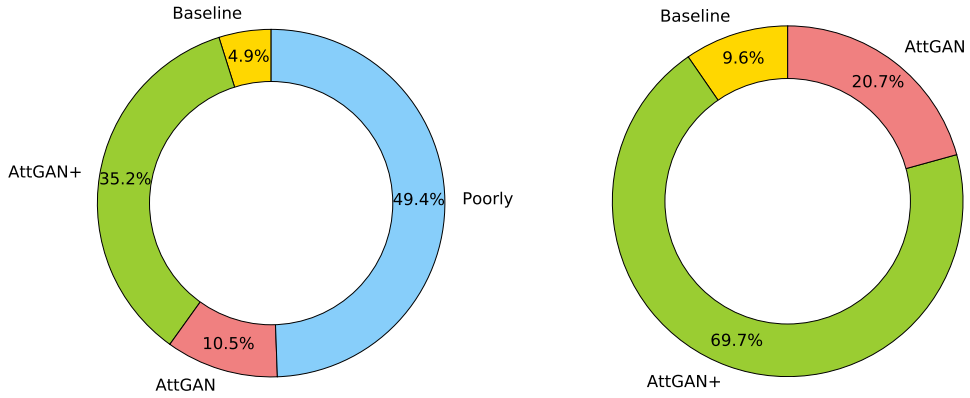


Figura 4.13: Comparação do método *baseline* [MSWB18] com *AttGAN*, *AttGAN+* e *AttGAN++*. Por questões de privacidade, os rostos foram descaracterizados e as partes íntimas expostas foram tarjadas em vermelho.

baseline utilizando um modelo treinando por 500 épocas, enquanto que as colunas representadas por 4.13c e 4.13d ilustram os exemplos gerados pelos métodos *AttGAN* e *AttGAN+*, também após 500 épocas de treinamento. Observando as imagens, pode-se identificar que o método *AttGAN+* gera imagens mais coerentes com as formas e com a cobertura de um biquíni real, quando comparadas às imagens geradas pelo método *AttGAN*. Na coluna representada pela Figura 4.13e são exibidas amostras para o método *AttGAN++*, que mescla a imagem de entrada original com a imagem gerada. A mescla é guiada pelas áreas determinadas na máscara de atenção, onde as áreas não relacionadas com partes íntimas do corpo são mantidas conforme a imagem original. O método *AttGAN++* apresenta resultados mais coerentes, já que aproveita a capacidade de geração dos biquínis do método *AttGAN+* e mantém os elementos periféricos, especialmente faces e plano de fundo.



(a) caso i (b) caso ii

Figura 4.14: Resultados colhidos após aplicação de formulário *online* para os casos i e ii.

Os métodos foram avaliados em um formulário *online*, onde diferentes participantes compararam os resultados gerados pelo *baseline* e pelos métodos *AttGAN* e *AttGAN+*. A Tabela 4.2, assim como a Figura 4.14, apresentam os resultados colhidos para os 2 casos observados: i) considerando a opção D (nenhum dos métodos é suficientemente bom), e ii) considerando somente as respostas correspondentes às opções A (*baseline*), B (*AttGAN*) e C (*AttGAN+*). A pesquisa foi respondida por 21 participantes, resultando em um total de 1050 respostas.

Para o primeiro cenário, ilustrado pela Figura 4.14a, observa-se que $49,4\%$ das imagens geradas foram avaliadas como insuficientes, enquanto $35,2\%$ escolheram o método *AttGAN+*, $10,5\%$ escolheram o método *AttGAN* e somente $4,9\%$ preferiram o método proposto por More et al. [MSWB18], utilizado como *baseline*.

Tabela 4.2: Resultados colhidos pelo formulário do avaliação.

	baseline (A)	*AttGAN* (B)	*AttGAN+* (C)	insuficiente (D)
Caso i	$4,9\%$	$10,5\%$	$35,2\%$	$49,4\%$
Caso ii	$9,6\%$	$20,7\%$	$69,7\%$	-

O segundo cenário, ilustrado pela Figura 4.14b, ignorou a opção D (insuficiente) e manteve as avaliações que optaram por um dos métodos apresentados. Neste cenário, o método *AttGAN+* é superior quando comparado ao método *AttGAN* e ao *baseline*, confirmando a hipótese de que a adição de informação proveniente da máscara de atenção contribui para que o gerador G_{AB} gere coberturas automáticas mais coerentes com biquínis reais.

4.6 Considerações e Discussão

A intuição para construir o método *AttGAN* tem origem na possibilidade de construir ConvNets capazes de gerar mapas que destacam áreas de interesse relacionadas a elementos específicos presentes em uma imagem. Assim, assumiu-se a hipótese de que a adição destas máscaras que destacam áreas de interesse poderiam guiar redes geradoras, contribuindo para seu treinamento e gerando melhores resultados. A hipótese foi validada ao observar os mapas de atenção gerados pela rede *AttNET* que, de fato, destacaram áreas onde ocorreram os objetos de interesse. Neste sentido, a Figura 4.15 ilustra mapas de atenção gerados pela camada de saída de uma *AttNET* para 3 imagens do conjunto de teste do *dataset* apresentado por More et al. [MSWB18]. Por motivos de privacidade, os rostos presentes nas imagens foram descaracterizados.

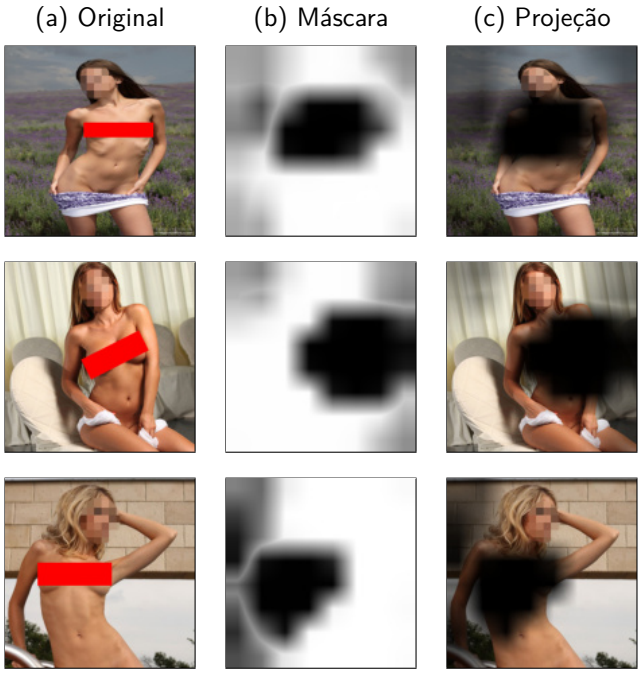
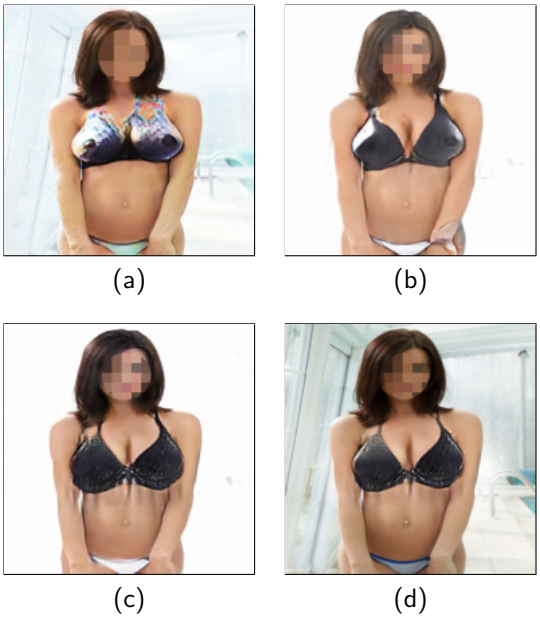


Figura 4.15: Máscaras de atenção geradas para 3 imagens do domínio A.

As regiões íntimas presentes nas imagens, ilustradas pela coluna referente à Figura 4.15a, são destacadas nos mapas de atenção representados pela coluna indicada em 4.15b. Os mapas de atenção, quando projetados sobre as imagens de entrada, são coerentemente alinhados com a imagem original. A coluna representada pela Figura 4.15c confirma a hipótese quando a máscara de atenção, ao ser projetada sobre a imagem de entrada, destaca as áreas relacionadas às partes íntimas,

constituindo a informação adicional que contribuirá para o treinamento dos geradores, especialmente G_{AB}, reforçando a tradução do domínio nudez (A) para vestida com biquíni (B).

Para identificar o impacto da adição de informação sobre as redes geradoras, foram investigados os efeitos dos componentes presentes nas variações de *AttGAN*. A Figura 4.16 ilustra 3 resultados para variações dos métodos que foram gerados progressivamente. É possível observar que, ao adicionar as máscaras de atenção como informação concatenada na entrada, os resultados para o método *AttGAN* seguem melhorando com o treinamento, onde a Figura 4.16a corresponde a 100 épocas e a Figura 4.16b corresponde a 500 épocas. Por outro lado, ao adicionar a soma da máscara de atenção na saída da primeira camada convolucional dos geradores G_{AB} e G_{BA} (método *AttGAN+*), o resultado obtido para 500 épocas (ilustrado pela Figura 4.16c) melhora. Finalmente, ao incluir a mescla guiada da imagem gerada por *AttGAN+* com a entrada original, o que foi chamado de *AttGAN++*, obtém-se o exemplo ilustrado pela Figura 4.16d, nitidamente melhor nos aspectos periféricos da imagem.



(a) (b) (c) (d)

Figura 4.16: Exemplos de 3 comparações. a) *AttGAN* após 200 épocas; b) *AttGAN* após 500 épocas; c) *AttGAN+* treinado por 500 épocas; d) *AttGAN++*. Por motivos de privacidade, as faces foram manualmente descaracterizadas.

Esta etapa da tese apresentou um novo método baseado em tradução imagem-para-imagem, utilizando redes geradoras para censura automática de conteúdo relacionado à pornografia. O método adiciona informação à entrada por meio de máscaras de atenção produzidas por uma ConvNet treinada para classificação, convertida para a geração de tais máscaras. A intuição para criação do método parte do trabalho de More et al. [MSWB18], uma evolução de CycleGAN [ZPIE17], com enfoque na cobertura de nudez numa abordagem não-supervisionada, independente de *datasets* alinhados.

Foram desenvolvidas e apresentadas 3 variações do método *AttGAN*. As variações foram comparadas com o método de More et al. [MSWB18] por meio de um formulário *online*, composto

por 50 conjuntos de imagens censuradas pelos diferentes métodos. O formulário foi respondido por 21 avaliadores, onde os resultados colhidos apontam vantagem para o método *AttGAN+*, escolhido como o melhor método dentre as 1.050 opiniões coletadas. A partir desta constatação, o método *AttGAN+* foi incrementado com a mescla guiada da imagem de entrada com a saída censurada, chamado de método *AttGAN++*, resultando em uma imagem censurada que preserva características periféricas da imagem original.

Esta etapa da tese deixa como contribuições o método *AttGAN* e suas variações *AttGAN+* e *AttGAN++*. Deixa também a rede de atenção para áreas relacionadas à pornografia *AttNET*, além da portagem do *dataset DPC* que permitiu o treinamento de modelos para classificação multi-rótulo, aplicada em *AttNET*.

5. CONCLUSÕES

A crescente quantidade de conteúdo pornográfico disponível abertamente na internet causa problemas como vício em sexo [You08], além de relatos que apontam a perda de interesse e a própria impotência sexual [1][2]. Neste sentido, Cooper [Coo98] apontou 3 pilares que motivam o consumo de ponografia na internet, sendo o primeiro deles *Accessibility*, que representa a facilidade de acesso ao conteúdo. Sabendo que a quantidade de ponografia disponível na internet cresce diariamente, esta tese assumiu que a quebra do pilar *Accessibility* pode minimizar o acesso indiscriminado a este tipo de conteúdo. Foram desenvolvidas 3 abordagens baseadas em Redes Neurais Artificiais para censurar automaticamente conteúdo pornográfico, onde cada abordagem buscou diminuir a exposição do internauta ao conteúdo pornográfico adotando estratégias cada vez menos intrusivas.

Inicialmente, a censura automática de conteúdo pornográfico foi atacada como um problema de classificação de imagens. Neste sentido foi apresentado *DataSex*, um *dataset* binário para treinamento de modelos preditivos composto por 286.920 imagens, definindo as classes *free* e *porn*. Este *dataset* permitiu o treinamento de 8 modelos de classificação com diferentes arquiteturas de ConvNets. Os resultados preditivos, observados a partir do conjunto de teste de *DataSex*, atingiram acurácias superiores a 99%, permitindo o processamento de até 40 FPS quando executados sobre GPU Tesla M40. Além do desempenho preditivo, foi possível observar que, para esses classificadores baseados em ConvNets, as regiões mais significativas para classificação de imagens pornográficas estão relacionadas a exposição de partes íntimas, e não somente a exposição de pele. Esta característica denota importante capacidade de generalização e robustez a invariância, importantes para a efetividade da censura automática.

Em termos de aplicações, classificadores de imagens pornográficas podem ser utilizados em mecanismos de controle parental, gerando predições para imagens avulsas ou, no caso de vídeos, para sequências de *frames*. Foram apresentados dois testes práticos, onde mídias contendo pornografia (tanto imagens quanto vídeo), identificados como tal por modelos treinados com *DataSex*, foram publicados sem qualquer restrição em redes sociais, neste caso Tumblr e YouTube, que disponibilizaram o conteúdo sem qualquer censura. Por outro lado, dado que as predições geradas pelos classificadores apresentados resultam somente nos *scores* para as classes, não dispondo de qualquer referência espacial que permita direcionar uma ação para uma região específica, as soluções decorrentes de uma predição positiva ficaram restritas à remoção ou à descaracterização completa da imagem ou *frame*.

Para aplicar censuras menos intrusivas, como oclusões ou descaracterizações de partes específicas das imagens, a tese atacou o problema da censura automática de conteúdo pornográfico como uma tarefa de detecção de objetos. Foram avaliados os métodos *Faster* R-CNN [RHGS15], SSD [LAE+16] e YOLO [RDGF16], utilizando *Dataset for Pornography Censorship* (*DPC*), um *dataset* para detecção de objetos que representam partes íntimas do corpo relacionadas à pornografia.

[1] https://goo.gl/RgyHOq
[2] http://goo.gl/tjuVVt

DPC foi construído no contexto desta tese, sendo o único *dataset* para detecção de partes íntimas do corpo relacionadas à pornografia já relatado. *DPC* é composto por 6.541 objetos rigorosamente anotados e revisados seguindo um protocolo de validação cruzada, distribuídos por 3.000 imagens contendo cenas pornográficas.

As experiências com os diferentes métodos de detecção avaliados na tese permitiram a criação de uma nova arquitetura totalmente convolucional para a tarefa de detecção, chamada *CensorNet*. Os resultados observados apontam que *CensorNet* é composta por menos parâmetros que YOLO-Tiny, atingindo resultados preditivos melhores tanto com relação ao tempo quanto mAP. *CensorNet* permite a geração de modelos mais leves, facilitando sua aplicação em contextos que disponham de recursos restritos.

Ao identificar que o método YOLO é mais simples e tão acurado quanto os métodos que apresentaram os melhores resultados, sendo ainda mais rápido que os outros métodos avaliados, a tese seguiu avaliando a tarefa de censura ao conteúdo pornográfico especificamente com este método. Para tal, o *framework* original do método YOLO foi portado para plataformas difundidas na comunidade de *Deep Learning*, flexibilizando a criação de novas estratégias e o uso de diferentes arquiteturas. Foi analisado o comportamento de modelos para detecção de objetos treinados com intenso aumento de dados e com a aplicação de múltiplas escalas, atingindo resultados superiores aos observados nos experimentos que utilizaram o método original. Observou-se também que a variação das dimensões do volume de entrada na fase de predição impacta nos resultados, em especial para os objetos das classes *frontalM* e *frontalF*.

Após contrastar os resultados obtidos para a tarefa de detecção de objetos utilizando arquitetura Darknet-19 [RF16] com os resultados de classificação de imagens utilizando a mesma rede, foi construída uma nova arquitetura composta por 2 saídas que permitiram a classificação de imagens e a detecção de objetos. Esta abordagem foi chamada de *CensorPlus* e combina a solução para os 2 problemas em um único fluxo. Mesmo demostrando alguma degradação de resultados preditivos quando comparada aos métodos de classificação e detecção de maneira isolada, *CensorPlus* atingiu resultados superiores aos observados no experimento que treinou modelos com os formatos originais dos métodos *Faster* R-CNN [RHGS15], SSD [LAE+16] e YOLO [RDGF16].

Como contribuições específicas para a tarefa de detecção de objetos relacionados à pornografia, a tese deixa *Dataset for Pornography Censorship* (*DPC*), um *dataset* para detecção de objetos que representam partes íntimas do corpo relacionadas à pornografia, além de uma variação deste *dataset* para treinamento de modelos híbridos para classificação de imagens e detecção de objetos. Também deixa a portagem do método YOLO para *frameworks* de *Deep Learning* difundidos entre a comunidade (Keras/Tensorflow [C+, AAB+]). Outra contribuição é *CensorNet*, um método baseado em YOLO que utiliza uma arquitetura composta por blocos formados por *Separable Convolutions*, sendo mais leve que todos os outros métodos analisados. *CensorPlus* é mais uma contribuição, sendo um método híbrido para classificação de imagens e detecção de objetos relacionados à pornografia. Finalmente, esta tese contribui com ferramentas para anotação de imagens em tarefas de detecção de objetos compatíveis com o formato PASCAL VOC [EEVG+15], além

de diversos modelos de detecção de partes íntimas que atingem até $0,7$ de mAP, identificando as classes *butt*, *breast*, *frontalM* e *frontalF*.

Buscando tornar as censuras ainda menos intrusivas que a aplicação de oclusões sobre partes íntimas, a tese apresentou *AttGAN*, um novo método baseado em tradução imagem-para-imagem que utilizou redes geradoras para desenhar censuras automaticamente sobre partes relacionadas à pornografia. O método parte de CycleGAN [ZPIE17], adicionando informação à entrada dos geradores utilizando máscaras de atenção produzidas por uma ConvNet treinada para classificação, convertida para a geração das máscaras. A intuição para criação do método partiu do trabalho de More et al. [MSWB18], que utilizaram CycleGAN [ZPIE17] em uma tarefa semelhante.

Foram desenvolvidas e apresentadas 3 variações do método *AttGAN*. As variações foram comparadas com o método de More et al. [MSWB18] por meio de um formulário *online* composto por 50 conjuntos de imagens censuradas pelos diferentes métodos, avaliadas por 21 participantes. Os resultados colhidos pelo formulário apontaram vantagem para o método *AttGAN+*. Sendo o melhor método avaliado no formulário, *AttGAN+* foi incrementado, adicionando a mescla guiada da imagem de entrada com a saída censurada produzida pelo gerador. A mescla utilizou a própria máscara de atenção para modificar somente as áreas da imagem relacionadas ao conteúdo pornográfico. O método incrementado foi chamado de *AttGAN++*, resultando em uma imagem censurada que preserva características periféricas da imagem original.

Como contribuições para a tarefa de censura ao conteúdo pornográfico abordada como um problema de tradução imagem-para-imagem, esta tese deixa um novo método baseado em GANs, chamado *AttGAN*, e suas variações *AttGAN+* e *AttGAN++*. Os métodos, especialmente *AttGAN++*, produzem censuras pouco intrusivas, que cobrem especialmente as partes íntimas do corpo ao desenhar peças falsas de roupa, no caso biquínis, sobre estas partes.

5.1 Limitações

Os modelos de classificação apresentam desempenho preditivo com acurácias próximas de 99% no conjunto de testes de *DataSex* e, conforme observado nas avaliações de aplicações relatadas na Seção 2.5.2.1, o modelo utilizado generalizou adequadamente, mesmo classificando imagens de fontes diferentes das usadas em treinamento. Por outro lado, ao classificar conteúdos de fontes distintas, foi possível identificar viés para a classe positiva. A Figura 2.18, especificamente no quadro 10, refere-se a uma cena fechada em um rosto feminino que resultou em picos nos *scores* para a classe *porn*, o que não é desejado. Também observa-se que o tempo de predição sobre CPU para o melhor modelo exige aproximadamente 1 segundo de processamento, inviabilizando sua utilização em problemas de tempo real. O modelo mais rápido gera predições sobre CPU em aproximadamente 130 milissegundos, o que ainda restringe a aplicação em problemas de tempo real, especialmente sob possíveis execuções em dispositivos com recursos de processamento e memória restritos como *smartphones* e *tablets*.

Assim como acontece com a abordagem de classificação, a censura automática utilizando métodos de detecção de objetos também apresenta restrições para aplicações de tempo real, especialmente sobre CPU. Utilizando imagens com dimensões de 576×576, o melhor modelo de detecção avaliado na Seção 3.4.3 necessitou de aproximadamente 2 segundos para gerar suas predições. Além das restrições relacionadas ao desempenho em função do tempo, nota-se também a presença de Falsos Negativos que resultam na exibição de nudez ou pornografia sem qualquer censura.

A abordagem para censura automática utilizando redes geradoras limita-se a cobrir nudez feminina com um único tipo de vestuário: biquínis. Além desta restrição, o tempo necessário para geração é de aproximadamente 4 segundos em CPU e próximo de 1 segundo em GPU, restringindo aplicações práticas.

5.2 Trabalhos Futuros

As abordagens para censura automática de nudez e pornografia apresentadas nesta tese demonstram resultados promissores e potencialidades para aplicações. Por outro lado, todas as abordagens deixam em aberto questões para trabalhos futuros.

Inicialmente, tanto classificação, detecção de objetos quanto geração automática devem ser otimizadas para reduzir o tempo de predição, permitindo aplicações em tempo real. Explorar arquiteturas de redes neurais mais enxutas pode reduzir o tempo preditivo, mantendo ou melhorando métricas de acurácia, precisão e IoU.

Especificamente sobre censura automática utilizando detecção de objetos, o modelo híbrido de classificação e detecção deve ser estudado buscando atingir melhores valores de mAP. A porção de classificação pode ser utilizada para ajustar os limiares de confiança de objetos, assumindo que imagens da classe *free* provavelmente não contém nudez ou pornografia, enquanto que imagens classificadas como *porn* certamente terão objetos detectados.

Finalmente, sobre o método de censura utilizando geração automática, deve-se avaliar o comportamento dos métodos *AttGAN* gerando peças de vestuário convencionais como calças e camisas. O método *AttGAN* e suas variações devem ser avaliados também em outros domínios conhecidos, especialmente aqueles já experimentados em outros métodos baseados em redes geradoras. Por outro lado, a dependência da rede de atenção *AttNET* requer o treinamento de novos modelos de atenção ajustados ao domínio avaliado, o que dependerá de dados rotulados para tal.

REFERÊNCIAS BIBLIOGRÁFICAS

[AAB+] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; ...; Zheng, X. "TensorFlow: Large-scale machine learning on heterogeneous systems". Capturado em: https://www.tensorflow.org/, Janeiro 2019.

[AR05] Acharya, T.; Ray, A. K. "Image processing: principles and applications". John Wiley and Sons, 2005, 455p.

[ATC+13] Avila, S.; Thome, N.; Cord, M.; Valle, E.; de A. Araújo, A. "Pooling in image representation: The visual codeword point of view", *Computer Vision and Image Understanding*, vol. 117, Mai 2013, pp. 453–465.

[BB82] Ballard, D. H.; Brown, C. M. "Computer Vision". Prentice Hall Professional Technical Reference, 1982, 523p.

[BDM+16] Bautista, C. M.; Dy, C. A.; Mañalac, M. I.; Orbe, R. A.; Cordel, M. "Convolutional neural network for vehicle detection in low resolution traffic videos". In: Proceedings of the IEEE Region 10 Symposium, 2016, pp. 277–281.

[Ben75] Bentley, J. L. "Multidimensional binary search trees used for associative searching", *Communications of the ACM*, vol. 18–9, Set 1975, pp. 509–517.

[BHC15] Badrinarayanan, V.; Handa, A.; Cipolla, R. "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling", *ArXiv E-prints*, vol. 1505.07293, Nov 2015, pp. 1–5.

[Bot12] Bottou, L. "Stochastic Gradient Descent Tricks". Springer Berlin Heidelberg, 2012, cap. 18, pp. 421–436.

[Bra08] Brabham, D. C. "Crowdsourcing as a model for problem solving an introduction and cases", *Convergence: The International Journal of Research Into New Media Technologies*, vol. 14, Fev 2008, pp. 75–90.

[BTVG06] Bay, H.; Tuytelaars, T.; Van Gool, L. "Surf: Speeded up robust features". In: Proceedings of the European Conference on Computer Vision, 2006, pp. 404–417.

[BWCB17] Becker, W.; Wehrmann, J.; Cagnini, H. E. L.; Barros, R. C. "An efficient deep neural architecture for multilingual sentiment analysis in twitter". In: Proceedings of the International Florida Artificial Intelligence Research Society Conference, 2017, pp. 246–251.

[C+] Chollet, F.; et al.. "Keras". Capturado em: https://keras.io, Janeiro 2019.

[Cho16] Chollet, F. "Xception: Deep Learning with Depthwise Separable Convolutions", *ArXiv E-prints*, vol. 1610.02357, Out 2016, pp. 1–8.

[Coo98] Cooper, A. "Sexuality and the internet: Surfing into the new millennium", *CyberPsychology & Behavior*, vol. 1, Jan 1998, pp. 187–193.

[CWF+15] Chen, L.; Wang, S.; Fan, W.; Sun, J.; Naoi, S. "Beyond human recognition: A CNN-based framework for handwritten character recognition". In: Proceedings of the Asian Conference on Pattern Recognition, 2015, pp. 695–699.

[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.

[DT05] Dalal, N.; Triggs, B. "Histograms of oriented gradients for human detection". In: Proceedings of the IEEE Computer Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[EEVG+15] Everingham, M.; Eslami, S. M. A.; Van Gool, L.; Williams, C. K. I.; Winn, J.; Zisserman, A. "The pascal visual object classes challenge: A retrospective", *International Journal of Computer Vision*, vol. 111, Jan 2015, pp. 98–136.

[EVGW+] Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; Zisserman, A. "The pascal visual object classes challenge 2007 (voc2007) results". Capturado em: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/, Janeiro 2019.

[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, Jan 2010, pp. 303–338.

[FLGC11] Faceli, K.; Lorena, A. C.; Gama, J.; Carvalho, A. "Inteligência Artificial: Uma abordagem de aprendizado de máquina". LTC, 2011, 394p.

[GBC16] Goodfellow, I.; Bengio, Y.; Courville, A. "Deep Learning". MIT Press, 2016, 800p.

[GDDM15] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. "Region-based convolutional networks for accurate object detection and segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, Mai 2015, pp. 142–158.

[Gir15] Girshick, R. B. "Fast R-CNN". In: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[GPAM+14] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. "Generative adversarial nets". In: Proceedings of the Conference on Neural Information Processing Systems, 2014, pp. 2672–2680.

[GSW+18] Ghilardi, M. C.; Simões, G.; Wehrmann, J.; Manssour, I. H.; Barros, R. C. "Real-time detection of pedestrian traffic lights for visually-impaired people". In: Proceedings of the IEEE International Joint Conference on Neural Networks, 2018, pp. 1–8.

[HGDG17] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. "Mask R-CNN". In: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.

[HLVDMW17] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. "Densely connected convolutional networks". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[HWG+14] Hong, R.; Wang, M.; Gao, Y.; Tao, D.; Li, X.; Wu, X. "Image annotation by multiple-instance learning with discriminative feature mapping and selection", *IEEE Transactions on Cybernetics*, vol. 44, Mai 2014, pp. 669–680.

[HZC+17] Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *ArXiv E-prints*, vol. 1704.04861, Abr 2017, pp. 1–9.

[HZRS15] He, K.; Zhang, X.; Ren, S.; Sun, J. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[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.

[JAFF16] Johnson, J.; Alahi, A.; Fei-Fei, L. "Perceptual losses for real-time style transfer and super-resolution". In: Proceedings of the European Conference on Computer Vision, 2016, pp. 694–711.

[KDA+] Krasin, I.; Duerig, T.; Alldrin, N.; Ferrari, V.; Abu-El-Haija, S.; Kuznetsova, A.; ...; Murphy, K. "Openimages: A public dataset for large-scale multi-label and multi-class image classification." Capturado em: https://storage.googleapis.com/openimages/web/index.html, Janeiro 2019.

[KF99] Kilgarriff, A.; Fellbaum, C. "Wordnet: An electronic lexical database", *The Library Quarterly*, vol. 69, Mai 1999, pp. 406–408.

[KSDFF13] Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. "3d object representations for fine-grained categorization". In: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 554–561.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[KT00] Kobayashi, M.; Takeda, K. "Information retrieval on the web", *ACM Computing Surveys*, vol. 32, Jun 2000, pp. 144–173.

[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 European Conference on Computer Vision, 2016, pp. 21–37.

[LB98] LeCun, Y.; Bengio, Y. "Convolutional networks for images, speech, and time series", *The Handbook of Brain Theory and Neural Networks*, vol. 3361, Out 1998, pp. 255–258.

[LBBH98] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, Nov 1998, pp. 2278–2324.

[LBH15] LeCun, Y.; Bengio, Y.; Hinton, G. "Deep learning". Nature Publishing Group, 2015, 801p.

[Li17] Li, B. "3d fully convolutional network for vehicle detection in point cloud". In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 2017, pp. 1513–1518.

[LJB+89] LeCun, Y.; Jackel, L.; Boser, B.; Denker, J.; Graf, H.; Guyon, I.; Henderson, D.; Howard, R.; Hubbard, W. "Handwritten digit recognition: Applications of neural network chips and automatic learning", *IEEE Communications Magazine*, vol. 27, Nov 1989, pp. 41–46.

[Llo82] Lloyd, S. "Least squares quantization in pcm", *IEEE Transactions on Information Theory*, vol. 28, Jan 1982, pp. 129–137.

[LMB+14] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. "Microsoft COCO: Common objects in context". In: Proceedings of the 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, Nov 2004, pp. 91–110.

[LRM+12] Le, Q. V.; Ranzato, M.; Monga, R.; Devin, M.; Chen, K.; Corrado, G. S.; Dean, J.; Ng, A. Y. "Building high-level features using large scale unsupervised learning". In: Proceedings of the International Conference on Machine Learning, 2012, pp. 507–514.

[LSD15] Long, J.; Shelhamer, E.; Darrell, T. "Fully convolutional networks for semantic segmentation". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[LTH+16] Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al.. "Photo-realistic single image super-resolution using a generative adversarial network", *ArXiv E-prints*, vol. 1609.04802, Set 2016, pp. 1–19.

[LYWW11] Liu, C.-L.; Yin, F.; Wang, D.-H.; Wang, Q.-F. "Casia online and offline chinese handwriting databases". In: Proceedings of the IEEE International Conference on Document Analysis and Recognition, 2011, pp. 37–41.

[MCS19] Mo, S.; Cho, M.; Shin, J. "Instagan: Instance-aware image-to-image translation". In: Proceedings of the International Conference on Learning Representations, 2019, pp. 1–26.

[MH08] Maaten, L. v. d.; Hinton, G. "Visualizing data using t-sne", *Journal of Machine Learning Research*, vol. 9, Nov 2008, pp. 2579–2605.

[MO14] Mirza, M.; Osindero, S. "Conditional generative adversarial nets", *ArXiv E-prints*, vol. 1411.1784, Nov 2014, pp. 1–7.

[Mou15] Moustafa, M. "Applying deep learning to classify pornographic images and videos", *ArXiv E-prints*, vol. 1511.08899, Nov 2015, pp. 1–9.

[MSWB18] More, M. D.; Souza, D. M.; Wehrmann, J.; Barros, R. C. "Seamless nudity censorship: an image-to-image translation approach based on adversarial training". In: Proceedings of the International Joint Conference on Neural Networks, 2018, pp. 1–8.

[NLW+16] Nian, F.; Li, T.; Wang, Y.; Xu, M.; Wu, J. "Pornographic image detection utilizing deep convolutional neural networks", *Neurocomputing*, vol. 210, Out 2016, pp. 283–293.

[NVG06] Neubeck, A.; Van Gool, L. "Efficient non-maximum suppression". In: Proceedings of the IEEE International Conference on Pattern Recognition, 2006, pp. 850–855.

[PKD+16] Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A. A. "Context encoders: Feature learning by inpainting". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.

[PY09] Pan, S. J.; Yang, Q. "A survey on transfer learning", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, Out 2009, pp. 1345–1359.

[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.

[RDS+15] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L. "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, vol. 115, Dez 2015, pp. 211–252.

[Red] Redmon, J. "Darknet: Open source neural networks in c". Capturado em: http://pjreddie.com/darknet/, Janeiro 2019.

[RF16] Redmon, J.; Farhadi, A. "YOLO9000: Better, Faster, Stronger", *ArXiv E-prints*, vol. 1612.08242, Dez 2016, pp. 1–9.

[RHGS15] Ren, S.; He, K.; Girshick, R. B.; Sun, J. "Faster R-CNN: towards real-time object detection with region proposal networks", *ArXiv E-prints*, vol. 1506.01497, Jan 2015, pp. 1–14.

[RHW88] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. "Learning Representations by Back-Propagating Errors". MIT Press, 1988, cap. 42, pp. 696–699.

[SBCC03] Schettini, R.; Brambilla, C.; Cusano, C.; Ciocca, G. "On the detection of pornographic digital images". In: Proceedings of the International Conference on Visual Communications and Image Processing, 2003, pp. 2105–2114.

[SCD+17] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[SIVA17] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. A. "Inception-v4, inception-resnet and the impact of residual connections on learning." In: Proceedings of the Association for the Advancement of Artificial Intelligence, 2017, pp. 4278–4284.

[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.

[SM14] Sifre, L.; Mallat, S. "Rigid-motion scattering for image classification", Tese de doutorado, Ecole Polytechnique, CMAP, 2014, 128p.

[SS01] Stockman, G.; Shapiro, L. G. "Computer Vision". Prentice Hall PTR, 2001, 650p.

[SWBR16] Simões, G. S.; Wehrmann, J.; Barros, R. C.; Ruiz, D. D. "Movie genre classification with convolutional neural networks". In: Proceedings of the International Joint Conference on Neural Networks, 2016, pp. 259–266.

[SWP+16] Simões, G.; Wehrmann, J.; Paula, T.; Monteiro, J.; Barros, R. C. "Datasex: um dataset para indução de modelos de classificação para conteúdo adulto". In: Proceedings of the Symposium on Knowledge Discovery, Mining and Learning, 2016, pp. 1–6.

[SZ14] Simonyan, K.; Zisserman, A. "Very deep convolutional networks for large-scale image recognition", *ArXiv E-prints*, vol. 1409.1556, Set 2014, pp. 1–14.

[TLWT15] Tian, Y.; Luo, P.; Wang, X.; Tang, X. "Pedestrian detection aided by deep learning semantic tasks". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5079–5087.

[UvdSGS13] Uijlings, J. R.; van de Sande, K. E.; Gevers, T.; Smeulders, A. W. "Selective search for object recognition", *International Journal of Computer Vision*, vol. 104, Abr 2013, pp. 154–171.

[VSP+17] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. "Attention is all you need". In: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[WB17a] Wehrmann, J.; Barros, R. C. "Convolutions through time for multi-label movie genre classification". In: Proceedings of the Symposium on Applied Computing, 2017, pp. 114–119.

[WB17b] Wehrmann, J.; Barros, R. C. "Movie genre classification: A multi-label approach based on convolutions through time", *Applied Soft Computing*, vol. 61, Dez 2017, pp. 973–982.

[WBCB17] Wehrmann, J.; Becker, W.; Cagnini, H. E.; Barros, R. C. "A character-based convolutional neural network for language-agnostic twitter sentiment analysis". In: Proceedings of the International Joint Conference on Neural Networks, 2017, pp. 2384–2391.

[WBDC17] Wehrmann, J.; Barros, R. C.; Dôres, S. N. d.; Cerri, R. "Hierarchical multi-label classification with chained neural networks". In: Proceedings of the Symposium on Applied Computing, 2017, pp. 790–795.

[WCB18] Wehrmann, J.; Cerri, R.; Barros, R. "Hierarchical multi-label classification networks". In: International Conference on Machine Learning, 2018, pp. 5225–5234.

[WGSM16] Wong, S. C.; Gatt, A.; Stamatescu, V.; McDonnell, M. D. "Understanding data augmentation for classification: when to warp?" In: Proceedings of the IEEE International Conference on Digital Image Computing, 2016, pp. 1–6.

[WHH+09] Wang, M.; Hua, X.-S.; Hong, R.; Tang, J.; Qi, G.-J.; Song, Y. "Unified video annotation via multigraph learning", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, Mar 2009, pp. 733–746.

[WLMB18] Wehrmann, J.; Lopes, M. A.; More, M. D.; Barros, R. C. "Fast self-attentive multimodal retrieval". In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2018, pp. 1871–1878.

[WMB18] Wehrmann, J.; Mattjie, A.; Barros, R. C. "Order embeddings and character-level convolutions for multimodal alignment", *Pattern Recognition Letters*, vol. 102, Jan 2018, pp. 15–22.

[WNHC12] Wang, M.; Ni, B.; Hua, X.-S.; Chua, T.-S. "Assistive tagging: A survey of multimedia tagging with human-computer joint exploration", *ACM Computing Surveys*, vol. 44, Ago 2012, pp. 1–25.

[WSR+16] Wehrmann, J.; Simões, G.; Rodrigo, B.; Paula, T.; Ruiz, D. "(deep) learning from frames". In: Anais da Conferência Brasileira de Sistemas Inteligentes, 2016, pp. 1–6.

[You08] Young, K. S. "Internet sex addiction: Risk factors, stages of development, and treatment", *American Behavioral Scientist*, vol. 52, Set 2008, pp. 21–37.

[YXY11] Yin, H.; Xu, X.; Ye, L. "Big skin regions detection for adult image identification". In: Proceedings of the IEEE Workshop on Digital Media and Digital Content Management, 2011, pp. 242–247.

[ZBS+15] Zhang, S.; Benenson, R.; Schiele, B.; et al.. "Filtered channel features for pedestrian detection." In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–4.

[ZGZgL16] Zhuo, L.; Geng, Z.; Zhang, J.; guang Li, X. "Orb feature based web pornographic image recognition", *Neurocomputing*, vol. 173, Jan 2016, pp. 511–517.

[ZGZL04] Zeng, W.; Gao, W.; Zhang, T.; Liu, Y. "Image guarder: An intelligent detector for adult images". In: Proceedings of the Asian Conference on Computer Vision, 2004, pp. 1080–1084.

[ZKL+16] Zhou, B.; Khosla, A.; Lapedriza, A.; Torralba, A.; Oliva, A. "Places: An Image Database for Deep Scene Understanding", *ArXiv E-prints*, vol. 1610.02055, Out 2016, pp. 1–12.

[ZLLH16] Zhang, L.; Lin, L.; Liang, X.; He, K. "Is faster r-cnn doing well for pedestrian detection?" In: Proceedings of the European Conference on Computer Vision, 2016, pp. 443–457.

[ZPIE17] Zhu, J.; Park, T.; Isola, P.; Efros, A. A. "Unpaired image-to-image translation using cycle-consistent adversarial networks", *ArXiv E-prints*, vol. 1703.10593, Mar 2017, pp. 1–20.

[ZVSL18] Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q. V. "Learning transferable architectures for scalable image recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8697–8710.

[ZZG+16] Zhou, K.; Zhuo, L.; Geng, Z.; Zhang, J.; Li, X. G. "Convolutional neural networks based pornographic image classification". In: Proceedings of the 2nd IEEE International Conference on Multimedia Big Data, 2016, pp. 206–209.

[ZZWS12] Zhang, X.; Zhang, L.; Wang, X.-J.; Shum, H.-Y. "Finding celebrities in billions of web images", *IEEE Transactions on Multimedia*, vol. 14, Jan 2012, pp. 995–1007.

APÊNDICE A – *SYNSETS* CONTENDO PESSOAS

A Tabela A.1 lista todos os *synsets* contendo pessoas utilizados na composição do *dataset DataSex*. Subconjuntos de *DataSex* foram utilizados para construir *Dataset for Pornography Censorship* (*DPC*) e suas variações.

Tabela A.1: *Synsets* contendo pessoas no ImageNet

ID	Descrição	ID	Descrição
n00007846	person, individual, someo	n10143725	granter
n02475669	Homo sapiens sapiens, mod	n10146927	greeter, saluter, welcome
n07942152	people	n10147262	grinner
n09606009	adventurer, venturer	n10149436	grunter
n09606527	anomaly, unusual person	n10150794	guesser
n09607630	appointee, appointment	n10159289	handyman, jack of all tra
n09610255	color-blind person	n10162194	hater
n09610405	commoner, common man, com	n10185148	hoper
n09613191	contestant	n10191001	hugger
n09616922	entertainer	n10212780	interpreter
n09617696	experimenter	n10213429	introvert
n09618957	face	n10219879	Jat
n09619168	female, female person	n10227266	junior
n09620078	inhabitant, habitant, dwe	n10236842	kink
n09620794	native, indigen, indigene	n10238272	kneeler
n09621232	native	n10240082	knower, apprehender
n09622049	juvenile, juvenile person	n10247358	lass, lassie, young girl,
n09622302	lover	n10247880	Latin
n09624168	male, male person	n10253122	left-hander, lefty, south
n09625401	national, subject	n10260706	life
n09626238	peer, equal, match, compe	n10266328	literate, literate person
n09629246	sensualist	n10282482	maid, maiden
n09629752	traveler, traveller	n10284965	malcontent
n09631129	unwelcome person, persona	n10289039	man
n09632274	unskilled person	n10289176	man
n09632518	worker	n10291110	manipulator
n09636339	Black, Black person, blac	n10299700	masturbator, onanist
n09638875	White, White person, Cauc	n10308066	nonmember
n09676884	Slav	n10314182	middlebrow
n09679170	gentile	n10323999	mixed-blood
n09752519	Gemini, Twin	n10328941	monolingual
n09753348	Sagittarius, Archer	n10341955	mutilator, maimer, mangle
n09753792	Pisces, Fish	n10344774	namer
n09756961	abomination	n10355142	neutral
n09763784	acquaintance, friend	n10361060	nondescript
n09764201	acquirer	n10361296	nonparticipant
n09764900	active	n10361525	nonperson, unperson
n09767197	actor, doer, worker	n10362003	nonresident

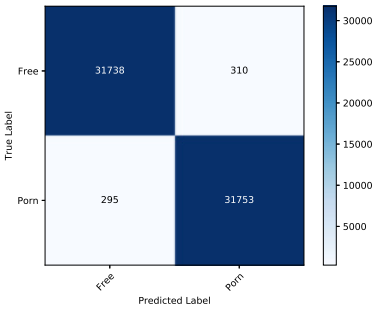
Tabela A.1: *Synsets* contendo pessoas no ImageNet

ID	Descrição	ID	Descrição
n09772330	adoptee	n10362319	nonsmoker
n09774167	advisee	n10370381	occultist
n09774783	advocate, advocator, prop	n10375314	old lady
n09779124	agnostic, doubter	n10375402	old man
n09796809	anti	n10380126	optimist
n09796974	anti-American	n10383094	organization man
n09801102	apprehender	n10384496	orphan
n09802445	appreciator	n10395390	pamperer, spoiler, coddle
n09804230	archaist	n10402824	party
n09824609	authority	n10417682	personage
n09827363	baby, babe, sister	n10418101	personification
n09828216	baby	n10418735	perspirer, sweater
n09828403	baby boomer, boomer	n10425946	philosopher
n09831962	bad person	n10431625	picker, chooser, selector
n09832633	bag lady	n10435716	pisser, urinator
n09833997	balker, baulker, noncompl	n10438172	planner, contriver, devis
n09836160	bullfighter, toreador	n10466918	preserver
n09843602	bather	n10495167	pursuer
n09845401	beard	n10498699	quarter
n09851165	best, topper	n10502046	quitter
n09856671	birth	n10513823	redhead, redheader, red-h
n09861287	bluecoat	n10518349	reliever, allayer, comfor
n09883452	buster	n10522759	rescuer, recoverer, saver
n09890749	candidate, prospect	n10525134	restrainer, controller
n09899289	cashier	n10526534	revenant
n09902954	celebrant, celebrator, ce	n10529231	rich person, wealthy pers
n09905530	chachka, tsatske, tshatsh	n10530959	right-hander, right hande
n09910374	charmer, beguiler	n10539160	rosebud
n09911226	charwoman, char, cleaning	n10540656	roundhead
n09918554	child, baby	n10560637	scientist
n09919451	chit	n10563711	scratcher
n09936825	colleen	n10569179	second-rater, mediocrity
n09947127	complexifier	n10575787	seeker, searcher, quester
n09950457	compulsive	n10576223	segregate
n09951274	computer user	n10585077	sex symbol
n09964411	copycat, imitator, emulat	n10592049	shop girl
n09970088	counterterrorist	n10616578	sneezer
n09976429	creature, wight	n10619492	socializer, socialiser
n09976728	creditor	n10656223	stifler, smotherer
n09990415	dancer, social dancer	n10659762	stooper
n09996481	deaf person	n10661216	stranger
n09997622	debtor, debitor	n10665302	struggler
n10024362	domestic partner, signifi	n10668666	subject, case, guinea pig
n10027246	double, image, look-alike	n10688811	tagger
n10033663	dribbler, driveller, slob	n10699752	tempter

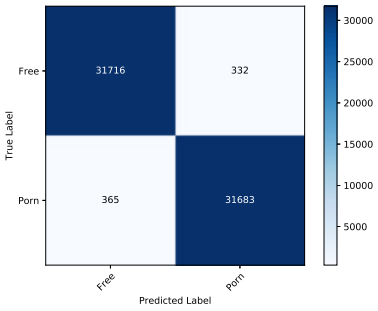
Tabela A.1: *Synsets* contendo pessoas no ImageNet

ID	Descrição	ID	Descrição
n10036266	drug user, substance abus	n10702167	termer
n10044682	ectomorph	n10702615	terror, scourge, threat
n10055730	endomorph	n10703336	testator, testate
n10056719	end man, corner man	n10709529	thrower
n10058411	enjoyer	n10715030	tomboy, romp, hoyden
n10074841	extrovert, extravert	n10724372	transfer, transferee
n10095420	flapper	n10728998	trier, attempter, essayer
n10099375	follower	n10745770	valley girl
n10109662	free agent, free spirit,	n10752480	victim, dupe
n10117739	gainer, weight gainer	n10753061	Victorian
n10117851	gal	n10756641	visionary
n10118844	gambler	n10781236	wiggler, wriggler, squirm
n10119609	gamine	n10783539	winker
n10129338	Gibson girl	n10791115	working girl
n10129825	girl, miss, missy, young	n10791890	worldling
n10138369	good guy	teste	teste

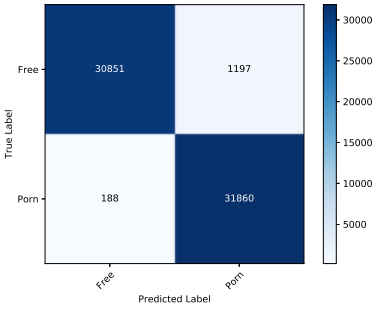
APÊNDICE B – MATRIZES DE CONFUSÃO PARA AVALIAÇÃO DE MODELOS DE CLASSIFICAÇÃO



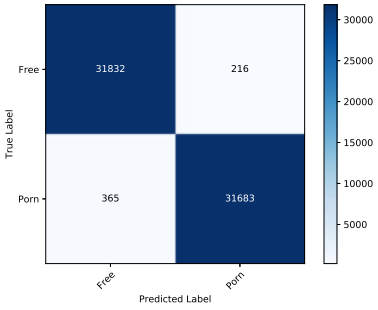
(a) Darknet19



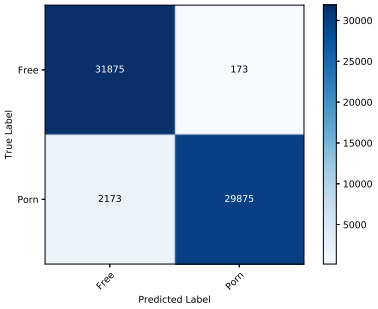
(b) Densenet



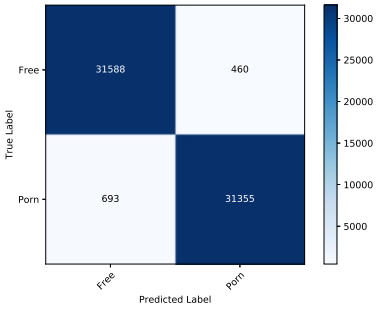
(c) Inception V3



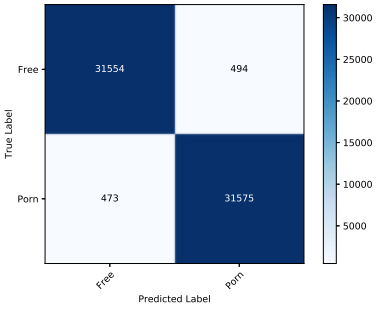
(d) Inception Resnet



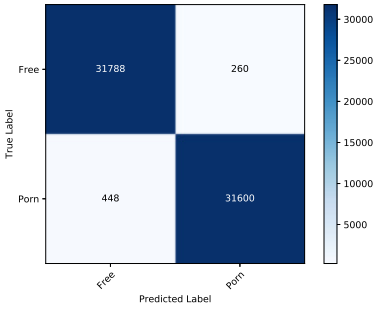
(e) Mobilenet



(f) Nasnet Mobile



(g) ResNet



(h) XCeption

Figura B.1: Matrizes de confusão obtidas ao classificar o conjunto de teste com os melhores modelos de cada arquitetura.

APÊNDICE C – AMOSTRA DO FORMULÁRIO DE AVALIAÇÃO APLICADO PARA OS MÉTODOS *ATTGAN* E *ATTGAN+*

As Figuras C.1, C.2 e C.3 apresentam recortes do formulário de avaliação que comparou os métodos *AttGAN* e *AttGAN+* com o *baseline*, neste caso, o método apresentado por More et al. [MSWB18].

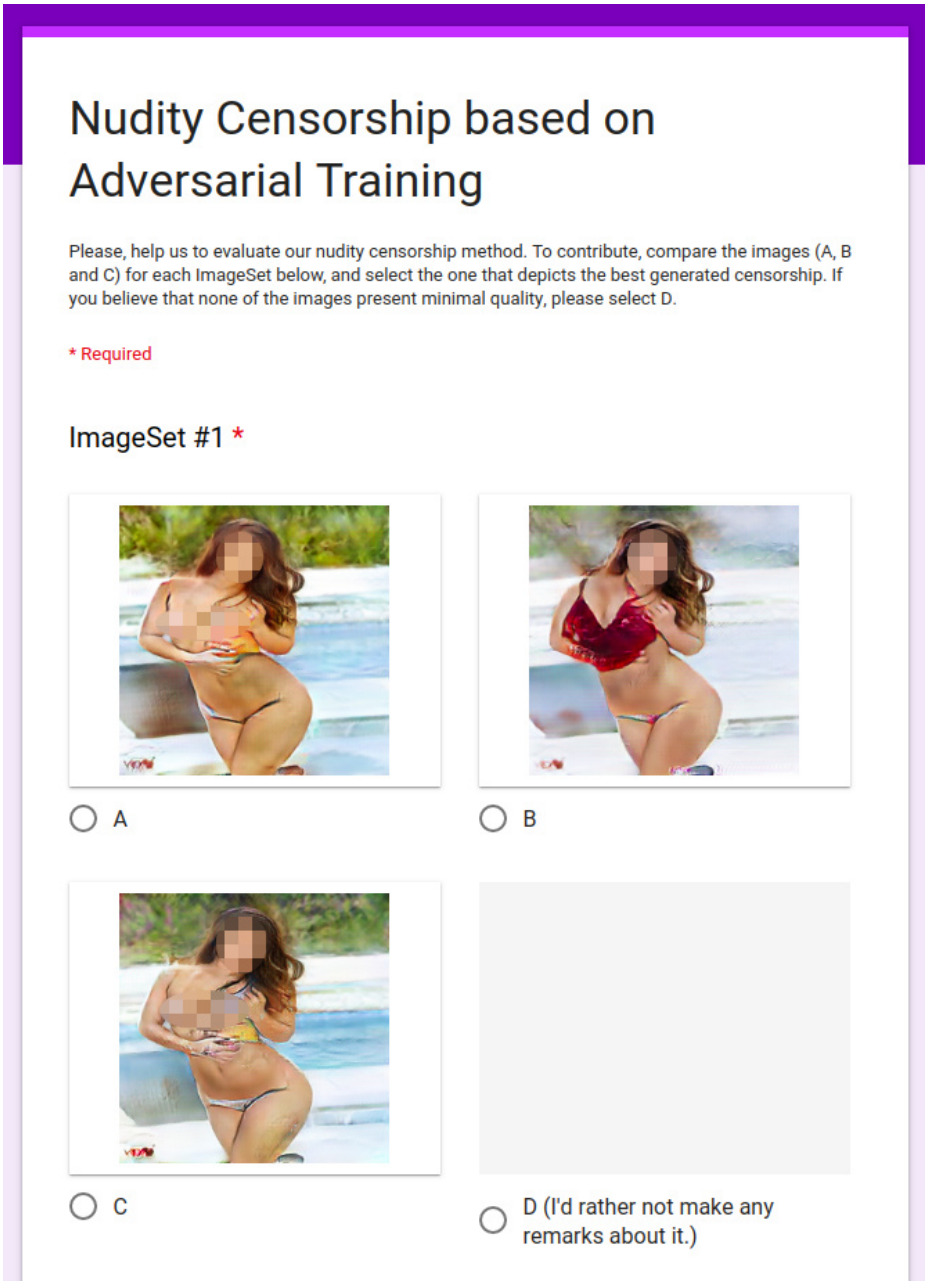


Figura C.1: Recorte do formulário de avaliação ilustrando o 1º conjunto de imagens apresentado ao avaliador.

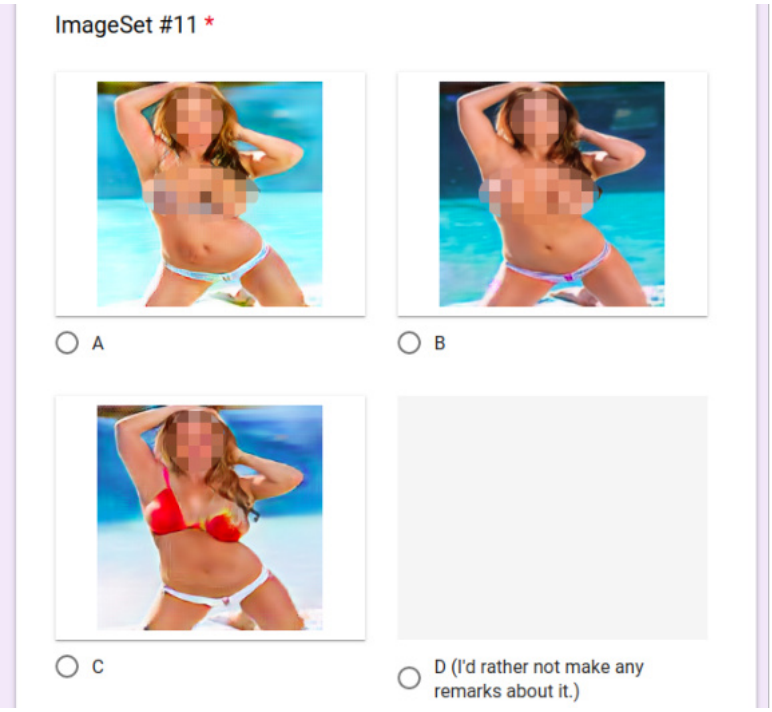


Figura C.2: Recorte do formulário de avaliação ilustrando o 11º conjunto de imagens apresentado ao avaliador.

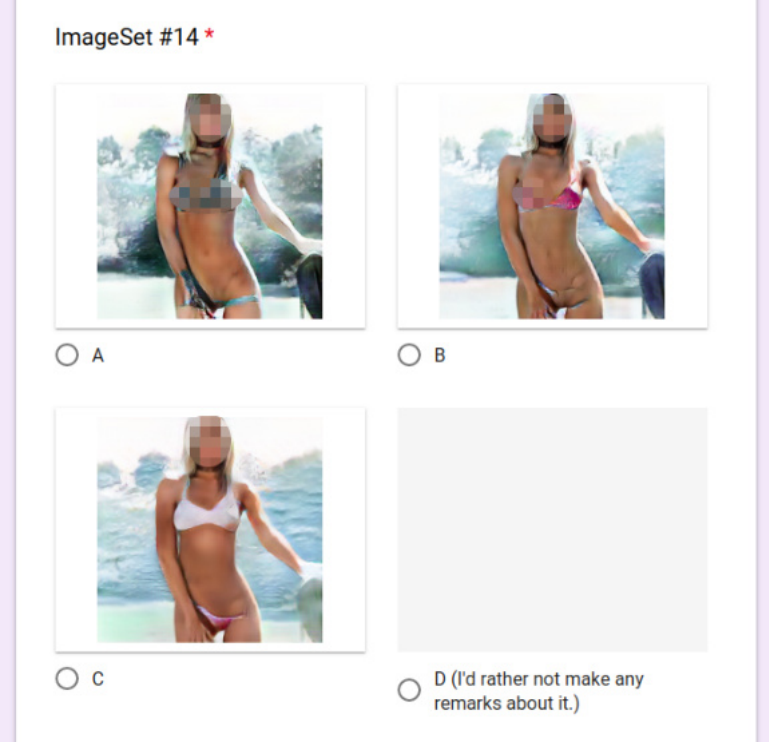


Figura C.3: Recorte do formulário de avaliação ilustrando o 14º conjunto de imagens apresentado ao avaliador.

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