Um sistema de prevenção de vazamento de dados de imagens baseado em aprendizado de máquina

Leandro Yukio Akune[1], Anderson Aparecido Alves da Silva[1234], Adilson Eduardo Guelfi[5], Marcelo Teixeira de Azevedo[2], José de Jesus Pérez Alcázar[2], Sergio Takeo Kofuji[2]

[1]Instituto de Pesquisas Tecnológicas (IPT)
[2]Universidade de São Paulo (USP)
[3]Universidade Paulista (UNIP)
[4]Centro Universitário SENAC
[5]Universidade do Oeste Paulista (UNOESTE)

leandro@akune.com.br, anderson@uol.com.br, guelfi@unoeste.br, marcelo.azevedo@pad.lsi.usp.br, {jperez,kofuji}@usp.br

Abstract. *Technological advancement contributes to the increased risk of loss of sensitive business and household data. Despite the evolution and wide availability of protection tools, in current data leak prevention systems such as Data Leak Prevent (DLP), lack of flexibility, clarity and functional limitations make it difficult to choose. There are several commercial solutions that often have a high cost of licensing, deployment, and limitations of prevention features. In this context, this work proposes the creation of a DLP proxy that prevents the unauthorized sending of sensitive data contained in images and videos captured by cameras. For the implementation of DLP, a neural network is used to detect objects in images.*

Resumo. *O avanço tecnológico colabora com o aumento do risco de perda de dados sensíveis de empresas e residências. Apesar da evolução e vasta disponibilidade de ferramentas de proteção, nos atuais sistemas de prevenção de vazamento de dados, como os Data Leak Prevention (DLP), a falta de flexibilidade, clareza e limitações funcionais dificultam a escolha. Existem diversas soluções comerciais que muitas vezes apresentam um alto custo de licenciamento, implantação, além de limitações de recursos de prevenção. Dentro deste contexto, este trabalho propõe a criação de um proxy DLP que impeça o envio não autorizado de dados sensíveis contidos em imagens e vídeos capturados por câmeras. Para a implementação do DLP é utilizada uma rede neural voltada para a detecção de objetos em imagens.*

1. Introdução

Os sistemas conhecidos como *Data Leak Prevention* (DLP) são usados para mitigar o risco de perda de informações confidenciais, tópico que é abordado na política de segurança da informação das corporações. Porém, o termo DLP ainda não é especificado em nenhum padrão ou regulamento oficial (HAUER, 2015).

Existem dezenas de DLPs comerciais, como Websense, McAffee, Symantec, Trend Micro, Check Point e Fidelis XPS, que podem ser integrados com sistemas de segurança existentes nas empresas. As soluções projetadas para detectar e prevenir vazamento de dados geralmente utilizam duas técnicas de análise: (1) a de conteúdo é a análise que verifica o conteúdo útil dos pacotes à procura de informações confidenciais; e (2) a de contexto busca, a partir de expressões regulares, a qual contexto uma informação compartilhada pertence (ALNEYADI; SITHIRASENAN; MUTHUKKUMARASAMY, 2016). As medidas preventivas e corretivas geralmente têm por base operações de registro, bloqueio, alerta, auditoria e quarentena. O que difere as soluções são os recursos opcionais, como por exemplo, a capacidade de detectar dados sensíveis em imagens, presente em algumas ferramentas de mercado (ALNEYADI; SITHIRASENAN; MUTHUKKUMARASAMY, 2016). As soluções comerciais protegem contra vazamento acidental. Porém, uma preocupação é que essas ferramentas comerciais não costumam oferecer proteção satisfatória contra ataques internos ou *malwares,* e nem sempre apresentam, com transparência, a forma de abordagem e as limitações (GUGELMANN, 2015). O receio da ocorrência de *malwares* se justifica porque este tipo de *script* malicioso pode ser usado para o roubo de informações confidenciais à partir da exploração de vulnerabilidades nos DLPs (KO; TAN; GAO, 2014).

O custo elevado com licenças e implantação também tornam inviável a utilização de ferramentas comerciais para a proteção de pequenas empresas e privacidade nas residências. Koutsourelis (2014) propõe uma arquitetura que combina as duas soluções gratuitas, OpenDLP (https://code.google.com/archive/p/opendlp/) e MyDLP (https://www.mydlp.com/), sendo que a primeira faz apenas uma varredura de dados armazenados em estações de trabalho e banco de dados, enquanto a segunda também previne vazamento de dados trafegados (em movimento) por meio de integração com o *Internet Content Adaption Protocol* (ICAP). Porém, a versão disponibilizada à comunidade possui limitações de recursos significativos, como ações de bloqueio e quarentena.

Além das limitações advindas da classificação, os trabalhos que implementam soluções contra vazamento de dados, como os de Alneyadi *et al.* (2016), Gugelmann *et al.* (2016), Vukovic *et al.* (2017), e Kouterelis (2014), em geral não são voltados para detecção e bloqueio de dados sensíveis em imagens e vídeos.

As redes neurais são modelos matemáticos, cuja estrutura tem por base o funcionamento cerebral e neurológico do ser humano. Trata-se de um ramo da Inteligência Artificial que consegue adaptar as suas próprias variáveis a partir da interação com o meio externo e melhorar o desempenho e os resultados (SANTOS, 2005). Há diversos tipos de redes neurais, sendo que algumas são apropriadas para aplicações específicas. Este é o caso das redes neurais usadas para identificação de objetos em imagens como o *You Only Look Once* (YOLO) (REDMON *et al.*, 2016), *Convolutional Neural Network* (CNN) (KARPATHY *et al.*, 2014) e *Regions with Convolutional Neural Network* (R-CNN) (GIRSHICK *et al.*, 2012). Estas redes neurais são importantes no contexto deste trabalho, já que no atual estado da arte, estas soluções não são integradas com sistemas de proteção de vazamento de dados.

O objetivo deste trabalho é propor uma arquitetura de *proxy* DLP capaz de impedir o envio não autorizado de dados sensíveis contidos no conteúdo de imagens, vídeos e

textos capturados por câmeras de vídeo. A validação é realizada com a aplicação de uma rede neural no reconhecimento de imagens de cartão de crédito e verificação de redução de Falso Positivos (FP) e Verdadeiro Negativos (VN), com treinamento de imagens de aparelhos celulares.

O restante do artigo está organizado conforme descrito a seguir. A Seção 2 apresenta os trabalhos relacionados. Já a Seção 3 expõe a proposta e a validação do método preventivo para tratar o vazamento de dados sensíveis. A Seção 4 aborda detalhes sobre o processo de treinamentos, enquanto a Seção 5 discute os resultados obtidos. Por fim, a Seção 6 apresenta a conclusão com as considerações finais sobre o trabalho.

2. Trabalhos Relacionados

Alneyad *et al.* (2016) divide um sistema DLP em três classes de atributos essenciais para proteger o vazamento de dados. Primeiro a capacidade de analisar conteúdo e contexto. Em segundo lugar, providenciar proteção de dados nos seus diferentes estados, como em trânsito, em uso e em repouso. Em terceiro lugar a capacidade de proteger dados por meio de diversas ações corretivas, como notificação, auditoria, bloqueio, criptografia e quarentena.

O foco principal do sistema proposto por Canabay *et al.* (2017) foi detectar ataques de modificação em palavras sensíveis em turco e prevenir vazamento de dados em movimento. Além de alteração, são tratados os ataques de adição e exclusão de caracteres e espaços em branco.

Kouterelis *et al.* (2014) propõem uma arquitetura automatizada que combina as ferramentas gratuitas OpenDLP e MyDLP de prevenção de perda de dados. O MyDLP se mostrou uma solução mais completa por atender os estados de dados em movimento, em repouso e em uso. No entanto, a versão disponibilizada à comunidade possui limitações de recursos significativos, como por exemplo, ausências de funcionalidades de varredura de dados em repouso, em estações que não utilizam sistema operacional *Windows*. Já o OpenDLP protege apenas contra vazamento de dados de estações de trabalho.

Girshick *et al.* (2012) propõem o R-CNN, que combina a rede neural convolucional – *Convolutional Neural Network* (CNN), algoritmos de propostas regionais e *Support Vector Machine* (SVM) para obter bons resultados de *mean Average Precision* (mAP[1]). O contexto do trabalho está voltado para a melhoria da precisão da detecção de imagens por redes neurais.

O processo de detecção consiste em descobrir um evento que está sendo executado em tempo real. Já uma classificação é o ato de categorizar objetos em grupos distintos a partir de um determinado critério. Sendo assim, a detecção é mais complexa que a classificação por exigir a localização precisa dos objetos dentro das imagens. No entanto, Girshick *et al.* (2012) mostram que além da CNN apresentar bons resultados na classificação de imagens, também é possível elevar o desempenho da detecção de objetos do *dataset* PASCAL VOC (EVERINGHAM *et al.*, 2010), comparando com sistemas que usam descritores baseados em histogramas como o HOG (GIRSHICK *et al.*, 2012).

[1] A média de precisão ou mean Average Precision (mAP) é um índice gerado pela rede neural que indica a precisão da identificação de um objeto no procedimento de validação do aprendizado.

Contrário ao método da janela deslizante e das técnicas baseadas em algoritmos de propostas regionais, o YOLO analisa toda a imagem durante o tempo de treinamento de informações contextuais sobre suas classes de objetos. De forma semelhante às abordagens R-CNN, também utiliza recursos convolucionais para marcar as caixas delimitadoras, ou seja, a borda em torno de um objeto no arquivo de imagem. Para evitar muitas detecções no mesmo objeto, o YOLO reduz para 98 caixas para delimitar um objeto por imagem, enquanto no método de pesquisa seletiva são usadas 2.000 marcações (REDMON *et al.*, 2016). Esta redução dificulta a classificação de vários objetos pequenos agrupados, como identificar um bando de pássaros em uma fotografia.

A rede neural escolhida neste trabalho foi o YOLO por apresentar resultados satisfatórios de velocidade e precisão nos trabalhos estudados de diversas áreas, como o próprio YOLO (REDMON *et al.*, 2016), criação de métodos para aprendizado de língua de sinais (KIM *et al.*, 2018) e monitoramento de cenas de trânsito para sistema de veículos autônomos (TAO *et al.*, 2017). O Fast R-CNN possui a vantagem em comparação com o YOLO por apresentar menos erros de localização. No entanto, para este trabalho, o importante é a identificação do objeto, independentemente da localização do mesmo na imagem.

Os dois principais *datasets* usados em conjunto com o YOLO são PASCAL VOC e COCO[2]. O PASCAL VOC foi selecionado neste trabalho por apresentar menor quantidade de classes de objetos pré-treinados, e quanto menor a quantidade de classes mais rápida fica a validação. O padrão de classes dos PASCAL VOC abrange vinte tipos de objetos enquanto o COCO trabalha com oitenta objetos.

3. Método Proposto

A Figura 1 apresenta o método proposto neste trabalho, com quatro módulos principais e os submódulos correspondentes. No primeiro módulo são definidos os requisitos da topologia e os critérios dos dados a serem protegidos. O segundo módulo aborda a criação de um banco de dados de *hashes* e o treinamento da rede neural. O terceiro módulo contém a forma de gestão do banco de dados e a classificação dos mesmos. O quarto e último módulo é responsável pelas ações que podem ocorrer de acordo com a classificação.

[2] O PASCAL VOC (http://host.robots.ox.ac.uk/pascal/VOC/) é um *dataset* público que possui uma grande quantidade de imagens e rótulos em 20 categorias principais coletadas entre 2007 a 2012. Já o COCO (https://cocodataset.org/#home) é um *dataset* público segmentado e rotulado que possui cerca de 1,5 milhão de instâncias de imagens categorizadas em diversas categorias.

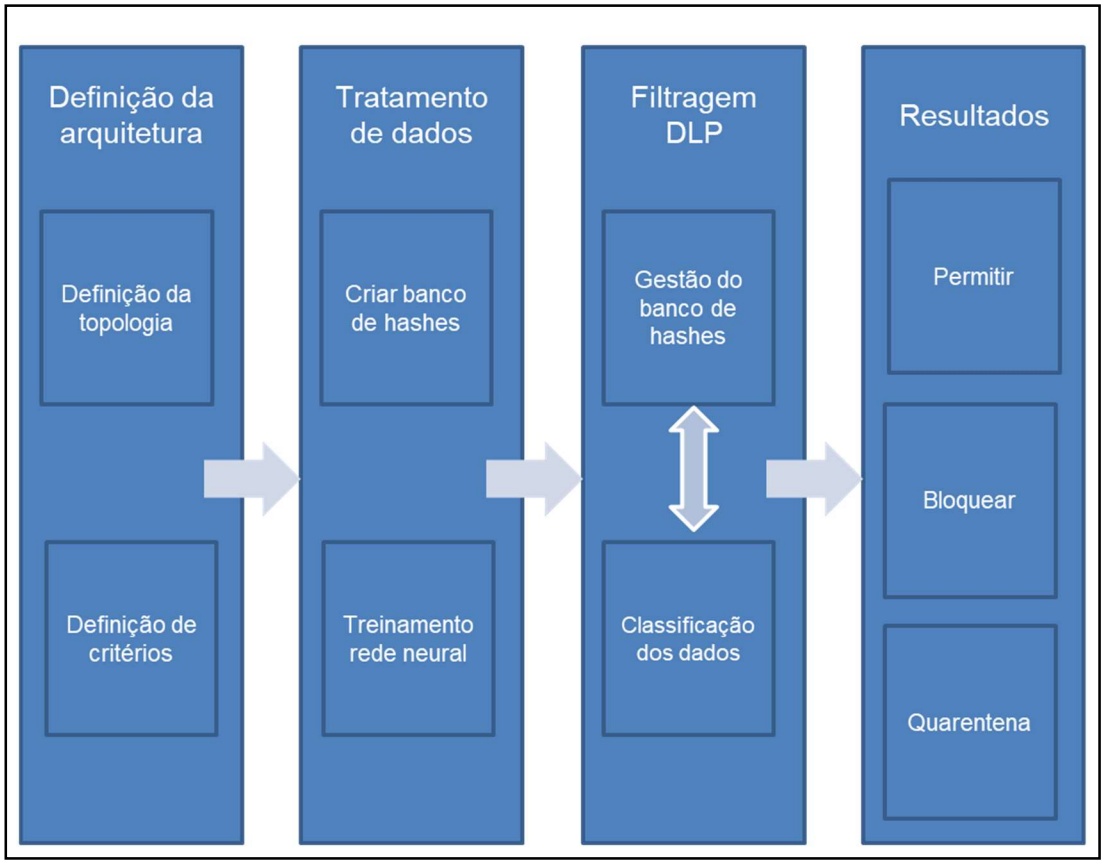

Figura 1. Proposta de método preventivo DLP

3.1 Módulo Definição da Arquitetura

A arquitetura envolve a *definição da topologia* (Figura 2) composta por uma *Local Area Network* (LAN), cujo acesso à Internet é realizado através de um *proxy* onde está implantado o DLP. Na Figura 2 também é possível ver o banco de dados de *hashes*.

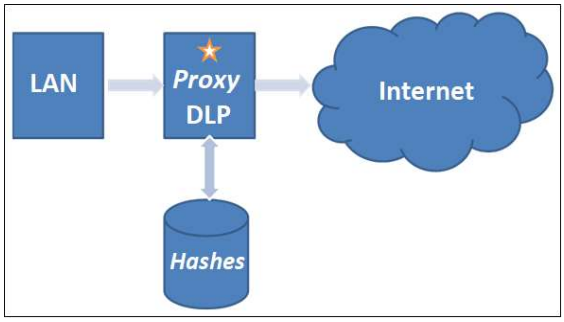

Figura 2. Topologia da rede proposta.

A *definição dos critérios* ocorre a partir de expressões regulares dos cartões das principais bandeiras, mostrados na Tabela 1).

Tabela 1 – Regras de distribuição de cartões por bandeira

Bandeira	Regra	Expressões Regulares
Visa	13 ou 16 dígitos, iniciando com 4	(4[0-9]{12}(?:[0-9]{3})?)
MasterCard	16 dígitos, com prefixo de 51 a 55	(5[1-5][0-9]{14})
Discover	16 dígitos, iniciando com prefixo 6011 ou 65	(6(?:011|5[0-9]{2})[0-9]{12})
American Express	15 dígitos, começando com 34 ou 37	(3[47][0-9]{13})
Diners Club	14 dígitos, começando com 300, 305, 36, ou 38	(3(?:0[0-5]|[68][0-9])[0-9]{11})
JCB	15 dígitos, começando com prefixo 2131 ou 1800, e 16 dígitos iniciando com prefixo 35	((?:2131|1800|35[0-9]{3})[0-9]{11})

Fonte: Adaptado de GOYVAERTS (2012)

3.2 Tratamento de Dados

O tratamento de dados envolve a *criação de hashes* dos dados classificados previamente como protegidos. Os *hashes* dos arquivos que passam pelo DLP são comparados com o conteúdo do banco de dados para evitar a saída de dados protegidos.

O processo de *treinamento da rede neural* também faz parte do tratamento dos dados. Há diversas redes neurais para tratamento de imagens: R-CNN (GIRSHICK *et al.*, 2012), Fast R-CNN, YOLO (REDMON *et al.*, 2016), O YOLO e R-FCN (TAO *et al.*, 2017). Também existem diversos *datasets* que podem ser utilizados para realizar um pré-treinamento da rede neural, como o *Digital National Security Archive* (KONGSGÅRD *et al.*, 2017), ILSVRC-2012 (DENG *et al*., 2012) e PASCAL VOC (EVERINGHAM *et al.*, 2010) (FELZENSZWALB *et al.*, 2009). A escolha do YOLO teve como base os bons resultados de rapidez e a precisão desta rede neural.

As fases de treinamento envolvem a definição da rede neural com base no desempenho e confiabilidade, um pré-treinamento com *dataset* público e um treinamento com imagens selecionadas para ensinar a rede a reconhecer objetos específicos.

3.3 Filtragem DLP

Este módulo se divide em duas partes.

A primeira, *gestão do banco de hashes,* consiste na alimentação do banco de dados.

A segunda, *classificação dos dados*, é executada pelo *proxy* e consiste na divisão do arquivo analisado pelo DLP em duas partes: (1) o conteúdo em formato de texto, enviado para classificação por expressões regulares; e (2) o conteúdo em formato de imagem, enviado para classificação pela rede neural.

3.4 Módulo Resultados

A partir do treinamento, neste módulo um arquivo que passa pelo DLP pode ser *liberado (permitir), bloqueado ou enviado para quarentena*. A Figura 3 apresenta o fluxo dos dados no DLP.

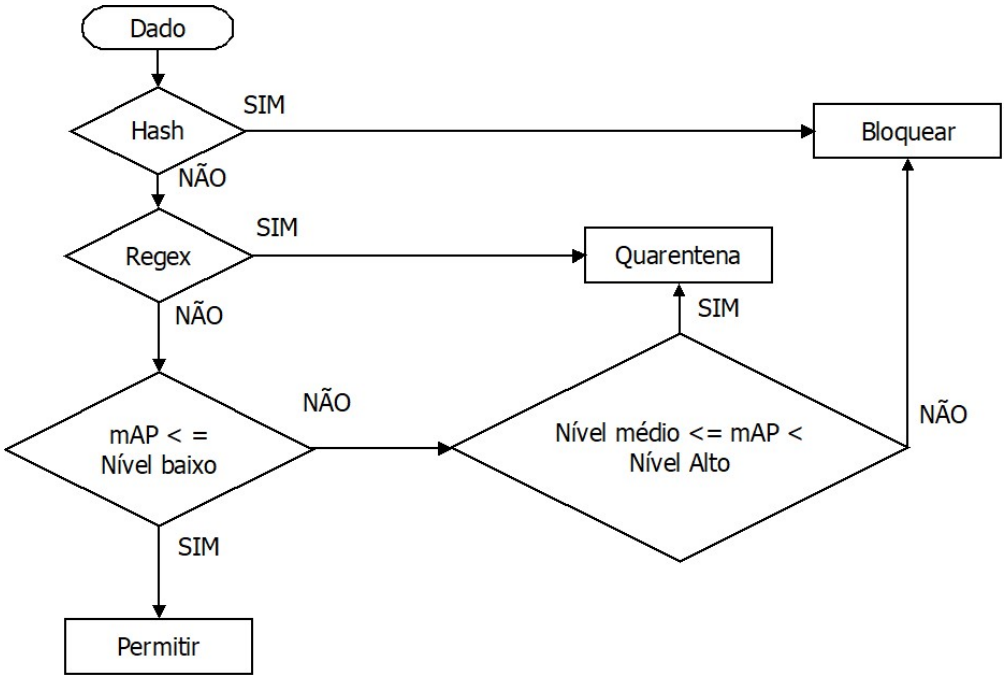

Figura 3. Fluxo de resultados dos dados

O ideal é que os valores de mAP sejam os mais altos possíveis. Para contextualização, neste artigo, definimos as seguintes faixas para os valores de mAP: (baixo) mAP < 60; (médio) 60 ≤ mAP < 80m; e (alto) mAP ≥ 80. Vale destacar que uma das vantagens do método proposto é que as três faixas de valores mAP (baixo, médio e alto) do fluxograma da Figura 3, podem ser definidas pelo administrador após os treinamentos, se adequando às necessidades de cada ambiente.

Como pode ser visto na Figura 3, o destino dos dados depende da classificação, da validação das expressões regulares *Regex* e do resultado mAP para os casos de imagens. A primeira etapa da validação consiste em consultar e verificar no banco de dados de *hashes* se o dado é classificado como protegido, caso positivo, ele é automaticamente bloqueado sem necessidade de verificação do conteúdo de texto ou imagem.

Conteúdos de textos que coincidem com as expressões regulares pré-determinadas são enviados para quarentena, para posterior análise mais apurada. Já conteúdos representados por imagens passam por avaliação de mAP.

Apenas valores mAP baixos são liberados (permitidos). Uma das vantagens do método proposto é que as três faixas de valores mAP (baixo, médio e alto) podem ser definidas pelo administrador após os treinamentos, se adequando às necessidades de cada ambiente.

3.5 Validação

Para validação do DLP é construída uma arquitetura com três instâncias em uma *cloud* pública. A primeira instância, ou máquina virtual, contém a instalação e configuração do *proxy* Squid integrado ao ICAP. A segunda é voltada para instalação, treinamento e teste da rede neural YOLO. Por fim a terceira instância tem a finalidade de funcionar como cliente de acesso à Internet. Todas as instâncias foram contratadas na modalidade de

Infraestrutura como Serviço (IaaS). A Figura 4 apresenta a topologia do ambiente proposto.

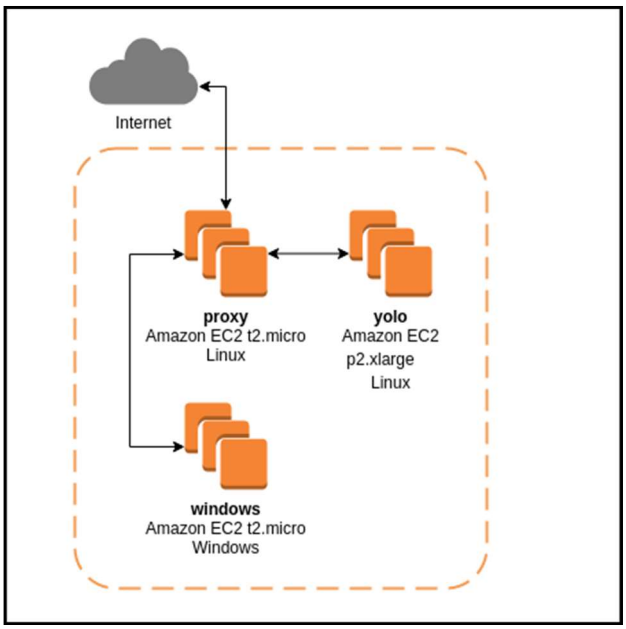

Figura 4. Topologia do ambiente DLP

Como visto na Figura 4, o acesso à Internet da máquina virtual Windows é controlado pela instância *proxy* que possui um banco de dados que também é alimentado pela instância YOLO. Nesta base de dados são registrados os *hashes* dos arquivos, a data de inserção e uma *flag* para indicar se o dado é definitivo ou se está em período de quarentena.

Na instância *proxy* o Squid recebe todas as requisições de acesso à Internet e consulta o protocolo ICAP para verificar se a saída deve ou não ser liberada. Para os arquivos de texto, a consulta verifica as expressões regulares mostradas na Tabela 1.

A instância yolo é responsável pelo treinamento e teste das imagens para identificar a existência de cartão de crédito. Para funcionamento do YOLO na instância *p2.xlarge* da Amazon Web Services (AWS), é necessário fazer um *download* ou um clone do projeto do YOLO de um repositório público[3]. O único pacote necessário para instalação é a ferramenta de gerenciamento da interface GPU *(nvidia-cuda-toolkit)*.

4. Considerações sobre o treinamento

O treinamento e o teste são realizados considerando os cenários de classes do PASCAL VOC de 2007 à 2012 disponíveis no próprio site do projeto do YOLO (REDMON et al., 2016). Para o treinamento da rede neural é necessário um conjunto de pesos convolucionais previamente treinados na base de dados Imagenet[4] (REDMON et al., 2016). O YOLO utiliza por padrão arquivos de nomes da base COCO.

---

[3] https://github.com/pjreddie/darknet.

[4] Imagenet é um conjunto de dados de imagens público e estruturado obtido em pjreddie.com/media/files/darknet19_448.conv.23

Detalhes sobre as imagens utilizadas, com ilustrações de cartão de crédito e com imagens de aparelho celular, podem ser vistos na Tabela 2.

Além disso, também são analisados conjuntos de pesos e percentuais de mAP diferentes para avaliar a melhor aplicação do YOLO no ambiente de DLP.

Tabela 2 – Divisão de imagens para treinamento e teste.

	Imagens de Cartão	**Imagens de Celular**
Total	350	350
Treinamento 80%	280	280
Teste 20%	70	70

Ajustes da variável *set batch* define o multiplicador da quantidade de imagens usadas no treinamento, como apresentado na Tabela 3.

Tabela 3 – Tempo de treinamento por quantidade de imagens

Sufixo arquivo Backup	**set batch =1**		**set batch =32**	
	Imagens (qtde)	**Tempo (hh:mm)**	**Imagens (qtde)**	**Tempo (hh:mm)**
_100.weights	100	0:01	3200	0:06
_200.weights	200	0:01	6400	0:12
_300.weights	300	0:01	9600	0:17
_400.weights	400	0:01	12800	0:23
_500.weights	500	0:02	16000	0:30
_600.weights	600	0:02	19200	0:37
_700.weights	700	0:02	22400	0:42
_800.weights	800	0:03	25600	0:49
_900.weights	900	0:03	28800	0:55
_10000.weights	10000	0:25	320000	10:27
_20000.weights	20000	0:49	640000	20:57
_30000.weights	30000	1:13	960000	31:23
_40000.weights	40000	1:37	1280000	41:44
_50000.weights	50000	2:01	1600000	52:08
_60000.weights	60000	2:25	1920000	62:32
_70000.weights	70000	2:49	2240000	73:04
_80000.weights	80000	3:13	2560000	83:18
_final.weights	80200	3:13	2566400	83:37

No caso do YOLOv2 com PASCAL VOC cada set batch corresponde a 80.200 imagens, um set batch = 32 corresponde a um treinamento com 2.566.400 imagens. Sendo assim, como mostra a Tabela 3, nota-se que a atribuição do valor do *set batch* implica nas quantidades de imagens do PASCAL VOC e consequentemente no tempo necessário para realização do treinamento.

A primeira coluna da Tabela 3 apresenta o sufixo aplicado nos arquivos de *backup*, ou seja, a fragmentação que o YOLO realiza durante os treinamentos. O valor numérico corresponde a quantidade de imagens usadas em cada etapa, o _final é o valor total e a base de multiplicação, que no caso do YOLOv2 com PASCAL VOC são 80.200 imagens. Quanto mais alto o sufixo, mais precisos são os resultados de detecção.

Dependendo do cenário, o simples uso de um sufixo substitui um treinamento mais extenso.

A quantidade de imagens usadas no treinamento implica diretamente a eficácia do YOLO. No entanto, por motivos de custo, este estudo se limitou a treinamentos com *set batch* com valor máximo = 32, o que implica em treinar 2.566.400 imagens em um tempo de mais de 83 horas por cenário. A Tabela 4 apresenta os custos para cada cenário de treinamento na AWS.

Tabela 4 – Custo na AWS para realização de treinamentos

Classes Imagens	Set batch =1 (hh:mm)	Set batch =32 (hh:mm)	Preço AWS/hora ($)	Tempo (horas)	Custo AWS ($)
VOC + Cartão	3:13	83:37	$0,90	86,83	$78,15
VOC + Cartão + Celular	3:13	83:37	$0,90	86,83	$78,15
				Total	$156,29

De acordo com a Tabela 4, para realizar apenas quatro treinamentos, com variações dos valores do multiplicador da quantidade de imagens *set bacth* e das classes do PASCAL VOC com imagens de cartão de crédito e em conjunto com aparelho celular, o custo fica em $156,29 dólares americanos. O problema deste custo por hora da AWS é que qualquer imprevisto ou erro na configuração do treinamento, além da perda de tempo, causa um alto prejuízo financeiro.

Além dos possíveis erros operacionais que podem ocorrer ao ajustar os arquivos de configurações do YOLO, observa-se que o clone disponível no projeto original do YOLO, gera algumas confusões nas marcações dos rótulos, pois mesmo realizando treinamento do PASCAL VOC, o YOLO utiliza por padrão arquivos de nomes da base COCO. No teste ocorrem as trocas dos nomes dos rótulos, por exemplo, classificando pessoa como pássaro e cartão de crédito como pessoa, conforme as ilustrações das Figura 5 e Figura 6.

Figura 5 – Exemplo de troca de rótulo de pessoa por pássaro

Como visto na Figura 5 o YOLO identificou corretamente as três pessoas, porém marcou de forma equivocada o rótulo como sendo pássaros. Este erro ocorreu porque o

YOLO utilizou as posições do arquivo de nomes de rótulos do *dataset* COCO, mesmo especificando para usar o PASCAL VOC.

Figura 6 – Exemplo de troca de rótulo de cartão de crédito por pessoa

Na Figura 6 observa-se que o YOLO identificou o cartão de crédito, mas o rotulou como pessoa.

5. Resultados e Discussão

Os conceitos de FP, VP e VN são adotados para análise dos resultados de acordo com as descrições da Tabela 5.

Tabela 5. Conceitos usados nas análises dos resultados

FP	Imagens sem ilustração, mas que são classificadas pelo YOLO
VN	Imagens com ilustração, mas que não são devidamente classificadas pelo YOLO
VP	Imagens com ilustração e são devidamente classificadas pelo YOLO

A Tabela 6 apresenta os resultados obtidos no teste de imagens com cartão de crédito e aparelho de celular, considerando o treinamento apenas com as imagens do PASCAL VOC e os nomes de rótulos do COCO.

Tabela 6. Treinamento com PASCAL VOC e rótulos COCO

Treinamento PASCAL VOC + COCO	FP	VN	VP	FN
mAP Final Peso > 50%	21,43%	18,57%	81,43%	78,57%

Na Tabela 6 é considerado apenas o melhor resultado, sendo o peso final com 2.566.400 imagens do PASCAL VOC usadas no treinamento e o mAP padrão acima de 50%. Além disso, é possível observar um alto índice de VP, pois a classe de aparelhos celulares não é nova nem desconhecida nos *datasets* do YOLO. Contudo, observa-se que o nível de FP se deve principalmente às imagens de cartão de crédito que foram reconhecidas como sendo do aparelho celular.

Nas validações com treinamento apenas com cartão de crédito e em conjunto com aparelhos celulares, foram avaliados os pesos 100, 10.000, 50.000 e final, estes sendo proporcionais às quantidades de imagens usadas no treino, sendo o peso final múltiplo de 80.200. Como o treinamento conjunto não apresentou resultado com o peso 10.000, e os resultados com pesos baixos como 100 geraram inúmeras marcações imprecisas foram considerados apenas os pesos 50.000 e final.

A Tabela 7 apresenta um resumo dos resultados do YOLO com treinamento das 2.566.400 imagens do PASCAL VOC com mais as 280 imagens de cartão de crédito, com validações de pesos 50.000 e final, e considerando mAP maior que 50%, maior que 60%, maior que 70%, maior que 80% e maior que 90%.

Tabela 7. Treinamento com PASCAL VOC e imagens de cartão de crédito

Treinamento cartão	mAP	FP	VN	VP	FN
Peso Final	>50%	**2,86%**	**74,29%**	25,71%	97,14%
	>60%	0,00%	77,14%	22,86%	100,00%
	>70%	0,00%	81,43%	18,57%	100,00%
	>80%	0,00%	82,86%	17,14%	100,00%
	>90%	0,00%	90,00%	10,00%	100,00%
Peso 50000	>50%	**7,14%**	**71,43%**	**28,57%**	92,86%
	>60%	5,71%	72,86%	27,14%	94,29%
	>70%	1,43%	75,71%	24,29%	98,57%
	>80%	1,43%	84,29%	15,71%	98,57%
	>90%	0,00%	88,57%	11,43%	100,00%

Como está destacado na Tabela 7, o melhor resultado, principalmente para um sistema de DLP, foi o que apresenta o menor índice possível de VN, pois a intenção é evitar ao máximo o vazamento de dados. Neste cenário, apesar dos altos valores devido ao treinamento com uma quantidade pequena de imagens de cartão, o teste que apresentou resultado com menor índice de VN tem peso 50.000, ou seja, com aproximadamente 1.600.000 de imagens do PASCAL VOC foi possível obter uma melhora no treinamento com um pequeno conjunto de dados próprio.

A Tabela 8 apresenta um resumo dos resultados do YOLO com treinamento das 2.566.400 imagens do PASCAL VOC com acréscimo de 280 imagens de cartão de crédito e inserção de 280 de aparelhos celulares, com validações de pesos 50.000 e final, e considerando mAP maior que 50%, maior que 60%, maior que 70%, maior que 80% e maior que 90%.

Tabela 8. Treinamento com PASCAL VOC, imagens de cartão e celular

Treinamento cartão e celular	mAP	FP	VN	VP	FN
Peso Final	>50%	10,00%	**78,57%**	21,43%	90,00%
	>60%	7,14%	84,29%	15,71%	92,86%
	>70%	1,43%	91,43%	8,57%	98,57%
	>80%	0,00%	95,71%	4,29%	100,00%
	>90%	0,00%	97,14%	2,86%	100,00%
Peso 50000	>50%	12,86%	82,86%	17,14%	87,14%
	>60%	10,00%	85,71%	14,29%	90,00%
	>70%	4,29%	87,14%	12,86%	95,71%
	>80%	1,43%	94,29%	5,71%	98,57%
	>90%	0,00%	95,71%	4,29%	100,00%

Como visto na Tabela 8, o menor índice de VN foi de 78,57%, considerando o peso final com mAP acima de 50%, porém o resultado de VN no treinamento apenas com as imagens de cartão de crédito apresentou melhores resultados, sendo 74,29% com peso final e 71,43% com peso 50000, como visto na Tabela 7.

Com base nos índices apresentados de VP e VN é possível medir a acurácia do YOLO para o cenário proposto, ou seja, a qualidade dos resultados ou o quão próximo este valor obtido se aproxima do valor correto. O cálculo da acurácia é realizado conforme descrito na Equação 1.

$$Acurácia = \frac{(VerdadeirosPositivos+VerdadeirosNegativos)}{(Positivos+Negativo\ )} \quad (1)$$

A Tabela 9 apresenta os valores calculados para os mesmos itens considerados nas medições, sendo pesos 50.000 e final, mAP maior que 50%, maior que 60%, maior que 70%, maior que 80% e maior que 90% e para os treinamentos com cartão de crédito e em conjunto com aparelhos celulares.

Tabela 9. Acurácia calculada para os treinamentos com cartão e em conjunto com celular

Acurácia	mAP	Cartão %	Cartão e Celular %
Peso Final	>50%	**61,4285714**	**55,7142857**
	>60%	**61,4285714**	54,2857143
	>70%	59,2857143	53,5714286
	>80%	58,5714286	52,1428571
	>90%	55,0000000	51,4285714
Peso 50000	>50%	60,7142857	52,1428571
	>60%	60,7142857	52,1428571
	>70%	**61,4285714**	**54,2857143**
	>80%	57,1428571	52,1428571
	>90%	55,7142857	52,1428571

Na Tabela 9 é possível observar que os melhores índices de exatidão ocorreram no treinamento apenas com imagens de cartão de crédito. Ao comparar os níveis de mAP podemos observar que quanto mais alto o mAP menor a acurácia. As únicas exceções, porém, são com peso 50.000 e mAP acima de 70%. Este caso ocorre, pois neste ponto a discrepância dos índices de VN são maiores tanto no treinamento com cartão de 4,28% quanto no treinamento com cartão e aparelho de celular de 5,71%.

Além de avaliar a acurácia, também é avaliada a eficácia, ou seja, das imagens classificadas como VP, quantas efetivamente foram classificadas corretamente, para este cálculo foi utilizada a Equação 2.

$$Eficácia = \frac{VerdadeirosPositivos}{(VerdadeirosPositivos+FalsosPositi\quad)} \quad (2)$$

A Tabela 10 apresenta os valores calculados para os mesmos itens medidos nos cálculos de acurácia, ou seja, com pesos 50.000 e final, mAP maior que 50%, maior que 60%, maior que 70%, maior que 80% e maior que 90% e para os treinamentos com cartão de crédito e em conjunto com aparelhos celulares.

Tabela 10. Eficácia calculada para os treinamentos com cartão e em conjunto com celular

Eficácia	mAP	Cartão %	Cartão e Celular %
Peso Final	>50%	90,0	68,1
	>60%	100,0	68,7
	>70%	100,0	85,7
	>80%	100,0	100,0
	>90%	100,0	100,0
Peso 50000	>50%	80,0	57,1
	>60%	82,6	58,8
	>70%	94,4	75,0
	>80%	91,6	80,0
	>90%	100,0	100,0

Conforme a Tabela 10, o peso 50.000 para ambos os casos apresentou eficácia inferior que o peso final, o que é esperado para uma rede neural, pois quanto mais se treina a rede maior também será a eficácia do sistema. Nota-se que no treinamento apenas com cartão de crédito e com peso final o sistema se mostra totalmente preciso com mAP a partir de 60%. Devido ao treinamento conjunto de imagens de cartão de crédito apresentar um aumento das ocorrências de FP, a eficácia deste cenário também mostrou resultados menos satisfatórios, atingindo uma eficácia absoluta apenas com mAP acima de 80% para o peso final e acima de 90% para o peso 50.000.

É importante notar que a acurácia representa a precisão do sistema (fórmula 1), enquanto a eficácia indica se o sistema se comporta bem em algumas categorias (fórmula 2). Apesar de o sistema apresentar uma acurácia baixa, obtendo um número alto de VN, a eficácia em relação às detecções VP foi alta, pois, as imagens classificadas dessa maneira tiveram um alto índice de acerto. Vale destacar também que os resultados não apresentam *overfitting*, pois em nenhum dos cenários testados foi possível reduzir significativamente o VN. Isso indica que o real motivo dos resultados é a necessidade de treinamentos com um volume maior de imagens, o que foi uma das limitações deste trabalho.

Por fim, a Tabela 11 apresenta a média de tempo gasto no teste do YOLO para os cenários de testes com pré-treinamento do PASCAL VOC, com treinamento com imagens de cartão de crédito e em conjunto com imagens de celular. Como os valores são próximos, independente da imagem receber ou não a classificação dos objetos, os valores apresentados são uma média aritmética de todos os valores de tempo coletados em cada cenário.

Tabela 11. Média de tempo no teste com cartão e em conjunto com celular por imagem

	Pascal VOC	Cartão	Cartão e Celular
Tempo médio (ms)	196,852181	164,9734	171,714773

Com base nestas médias de tempo foi possível observar que, com esta infraestrutura ou com os treinamentos realizados, a rede neural não está apta para o reconhecimento de cartão de crédito nos casos de vídeos transmitidos em tempo real, pois no período de 1 segundo, o YOLO foi capaz de avaliar em torno de 6 imagens apenas.

Porém, o DLP pode ser aplicado na verificação de imagens e se o *delay* não for levado em conta, vídeos também podem ser verificados.

6. Conclusão

Este trabalho propôs uma arquitetura que integra uma ferramenta de detecção de objeto com um *proxy* DLP, para impedir o envio não autorizado de dados sensíveis em arquivos de imagem e vídeo. Além de analisar de forma transparente o funcionamento, a eficácia e as limitações desta solução, foi realizada a avaliação do treinamento e teste da rede neural para reconhecimento de imagens de cartão de crédito e verificação de redução de Falso Positivos (FP) e Verdadeiro Negativos (VN), com treinamento de imagens de aparelhos celulares.

A análise dos resultados com o YOLO mostrou-se precisa para as imagens que foram classificadas, porém, realizando o treinamento com uma pequena quantidade de imagens, tanto nos casos com cartão de crédito, como naqueles em conjunto com aparelho celular, não foi possível obter um grande volume de VP.

A previsão de que o treinamento em conjunto das classes de cartão de crédito e de aparelho celular reduziria a taxa de FP não se confirmou, pelo contrário, ocorreu um aumento em todas as validações realizadas. Notamos que quanto mais imagens são usadas no treinamento, melhores são os resultados obtidos no teste. Porém, ao aumentar a quantidade de classes de objetos ocorreu uma redução significativa nos índices de VP, o que implica também em redução da acurácia. A razão principal para estes resultados está na necessidade de um treinamento mais amplo, com mais imagens, o que se tornou inviável devido aos custos do serviço de nuvem que hospedava as máquinas virtuais. Além disso, acertos nas configurações e categorias usadas pelo nas bases do PASCAL VOC e COCO também poderiam gerar melhores resultados, já que parte dos erros de acurácia se concentraram em problemas relacionados às categorias das imagens.

Com a estrutura construída e com os treinamentos realizados, o YOLO foi capaz de avaliar por volta de 6 imagens por segundo, o que torna este sistema não recomendado para reconhecimento de cartão de crédito em vídeos transmitidos em tempo real. Porém, mesmo que implique em um *delay*, é possível aplicar este DLP na verificação de *upload* tanto de arquivos de imagens como de vídeos.

O trabalho também contribuiu para avaliar a precisão de uma rede neural para detecção de objetos (YOLO/Darknet) em relação na avaliação das tecnologias de treinamento e

O investimento em segurança da informação representa um custo significativo e qualquer aplicação é inversamente proporcional a usabilidade, porém, o vazamento de dados críticos pode implicar em prejuízos incalculáveis. O uso de uma rede neural na proteção de vazamento de dados de cartão de crédito mostrou ser uma abordagem promissora, com a disponibilização de maiores recursos de processamento e com a validação de novos *datasets* para treinamento.

Em resumo, os resultados obtidos podem ser complementados com alguns trabalhos futuros:

a) Aumentar a duração e o volume de imagens do treinamento com o *dataset* PASCAL VOC usando valores mais altos para as variáveis *set batch e subdivision*;

b) Realizar treinamentos com gradativo aumento da quantidade de imagens de cartão de crédito, para verificar o crescimento da taxa de acurácia;
c) Validar outros conjuntos do *dataset* com o YOLO ou possivelmente outra rede neural com o PASCAL VOC;
d) Comparar os resultados alcançados no reconhecimento de objeto com sistemas que realizam reconhecimento de caracteres do tipo OCR e analisem expressões regulares;
e) Combinar melhores treinamentos com infraestrutura mais robusta para aumentar a quantidade de imagens validadas por segundo para permitir a aplicação de uma rede neural que reconheça imagens de cartão de crédito em vídeos transmitidos em tempo real.

Agradecimento: este trabalho teve suporte técnico da parceria Huawei-USP.

Referências

ALNEYADI, S.; SITHIRASENAN, E.; MUTHUKKUMARASAMY, V. (2016). A survey on data leakage prevention systems. Journal of Network and Computer Applications, v. 62, p. 137–152, 2016.

EVERINGHAM, M. *et al.* (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, v. 88, n. 2. 2010 p. 303–338.

GIRSHICK, R. *et al.*(2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014. p. 580-587.

GUGELMANN, D. *et al.*(2015). Can Content-Based Data Loss Prevention Solutions Prevent Data Leakage in Web Traffic? IEEE Security & Privacy, v. 13, n. 4, 2015. p. 52–59.

HAUER, B. (2015) Data and information leakage prevention within the scope of information security. IEEE Access, v. 3. 2015 p. 2554–2565.

KARPATHY, A. *et al.*(2014). Large-scale video classification with convolutional neural networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2014. p. 1725–1732, 2014.

KIM, S.; JI, Y.; LEE, K. (2018). An Effective Sign Language Learning with Object Detection Based ROI Segmentation. In: 2018 Second IEEE International Conference on Robotic Computing (IRC). IEEE, 2018. p 330–333.

KO, Ryan KL; TAN, Alan YS; GAO, Ting. A Mantrap-Inspired, User-Centric Data Leakage Prevention (DLP) Approach. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science. IEEE, 2014. p. 1033-1039.

KOUTSOURELIS, D.; KATSIKAS, S. K. (2014). Designing and developing a free Data Loss Prevention system. In: Proceedings of the 18th Panhellenic Conference on Informatics (PCI). New York, USA: ACM Press, 2014. p 1–5.

REDMON, J. *et al.*(2016). You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016 p. 779–788.

SANTOS, A. et al. Usando redes neurais artificiais e regressão logística na predição da hepatite A. Revista Brasileira de Epidemiologia, v. 8, p. 117-126, 2005.

TAO, J. *et al.* (2017). An object detection system based on YOLO in traffic scene. In: 6th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2017.

VUKOVIC, M. *et al.*(2017). Rule-based system for data leak threat estimation. In: Software, Telecommunications and Computer Networks (SoftCOM) 25th International Conference on IEEE, 2017 . p. 1–5.