



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

MAINARA CAVALCANTI DE FARIAS

CLASSIFICAÇÃO DE IMAGENS USANDO WEAK SUPERVISION

CAMPINA GRANDE - PB

2019

MAINARA CAVALCANTI DE FARIAS

CLASSIFICAÇÃO DE IMAGENS USANDO WEAK SUPERVISION

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharela em Ciência da Computação.

Orientador: Leandro Balby Marinho

CAMPINA GRANDE - PB

2019

MAINARA CAVALCANTI DE FARIAS

CLASSIFICAÇÃO DE IMAGENS USANDO WEAK SUPERVISION

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Leandro Balby Marinho

Orientador – UASC/CEEI/UFCG

Hyggo Oliveira de Almeida

Examinador – UASC/CEEI/UFCG

Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 21 de novembro de 2019.

CAMPINA GRANDE - PB

RESUMO

Modelos de aprendizagem de máquina precisam de grandes conjuntos de dados para serem bem treinados. Porém, a criação desses dados rotulados ainda é um grande desafio para a aprendizagem de máquina supervisionada, já que muitas vezes é um processo manual bastante demorado e custoso. Por isso, usualmente são utilizadas bases de dados já existentes, no entanto, em muitos casos esses dados não refletem perfeitamente o contexto em que o modelo irá atuar. Como alternativa a essa abordagem tradicional, surgiu o paradigma data programming, utilizado para a criação programática de conjuntos de treinamento, nesse paradigma os usuários expressam estratégias de weak supervision para criar funções de rotulagem, as quais eventualmente são ruidosas. A fim de criar um modelo robusto aos ruídos produzidos por essas funções, pesquisadores da universidade de Stanford criaram o sistema Snorkel, o qual se beneficia do acordo entre essas funções para a criação do seu modelo. Nesta pesquisa, o Snorkel é utilizado com o objetivo de rotular pessoas em imagens, uma tarefa diferente da qual o Snorkel vem sendo utilizado, pois as aplicações que fazem seu uso são geralmente no contexto de processamento de linguagem natural, pois é mais simples criar heurísticas que atuam em textos. O ambiente em que as imagens foram extraídas foi um laboratório no bloco CN, localizado na UFCG. Com o propósito de comparar o desempenho do modelo treinado com os dados do ambiente específico (gerados pelo Snorkel) e uma base de dados já existente (genérica), um modelo mais sofisticado foi treinado com os diferentes conjuntos de dados. A acurácia final do modelo treinado com os dados gerados pelo Snorkel foi de 84,94%, enquanto que o treinado com imagens genéricas foi de apenas 30%, indicando que o desempenho de um algoritmo de aprendizagem de máquina especializado em um ambiente específico é bastante superior a um treinado com dados genéricos.

IMAGE CLASSIFICATION USING WEAK

ABSTRACT

Machine learning models need large datasets to be well trained. However, creating this labeled data is still a major challenge for supervised learning, as it is often a very time consuming and costly manual process. Therefore, existing databases are usually used, however, in many cases this data does not perfectly reflect the context in which the model will act. As an alternative to this traditional approach, the data programming paradigm arose, used for the programmatic creation of training sets, in this paradigm users express weak supervision strategies to create labeling functions, which eventually are noisy. In order to create a robust model to the noise produced by these functions, researchers at Stanford University created the Snorkel system, which benefits from the agreement of these functions to create their model. In this research, Snorkel is used for the purpose of labeling people in images, a different task from which Snorkel has been used, since the applications that make use of it are generally in the context of natural language processing, because it is simpler to create heuristics that act in texts. The environment in which the images were extracted was a laboratory in the CN block, located at UFCG. In order to compare the performance of the trained model with the specific environment data (generated by Snorkel) and an existing (generic) database, a more sophisticated model was trained with the different datasets. The final accuracy of the model trained with the data generated by Snorkel was 84.94%, while the one trained with generic images was only 30%, indicating that the performance of a specialized machine learning algorithm in a specific environment is far superior to one trained with generic data.

Classificação de imagens usando *weak supervision*

Trabalho de Conclusão de Curso

Mainara Cavalcanti de Farias
mainara.farias@ccc.ufcg.edu.br
Universidade Federal de Campina Grande - UFCG
Campina Grande, PB

Leandro Balby Marinho
lbmarinho@computacao.ufcg.edu.br
Universidade Federal de Campina Grande - UFCG
Campina Grande, PB

ABSTRACT

Machine learning models need large datasets to be well trained. However, creating this labeled data is still a major challenge for supervised learning, as it is often a very time consuming and costly manual process. Therefore, existing databases are usually used, however, in many cases this data does not perfectly reflect the context in which the model will act. As an alternative to this traditional approach, the data programming paradigm arose, used for the programmatic creation of training sets, in this paradigm users express weak supervision strategies to create labeling functions, which eventually are noisy. In order to create a robust model to the noise produced by these functions, researchers at Stanford University created the Snorkel system, which benefits from the agreement of these functions to create their model. In this research, Snorkel is used for the purpose of labeling people in images, a different task from which Snorkel has been used, since the applications that make use of it are generally in the context of natural language processing, because it is simpler to create heuristics that act in texts. The environment in which the images were extracted was a laboratory in the CN block, located at UFCG. In order to compare the performance of the trained model with the specific environment data (generated by Snorkel) and an existing (generic) database, a more sophisticated model was trained with the different datasets. The final accuracy of the model trained with the data generated by Snorkel was 84.94%, while the one trained with generic images was only 30%, indicating that the performance of a specialized machine learning algorithm in a specific environment is far superior to one trained with generic data.

RESUMO

Modelos de aprendizagem de máquina precisam de grandes conjuntos de dados para serem bem treinados. Porém, a criação desses dados rotulados ainda é um grande desafio para a aprendizagem de máquina supervisionada, já que muitas vezes é um processo manual bastante demorado e custoso. Por isso, usualmente são utilizadas bases de dados já existentes, no entanto, em muitos casos esses dados não refletem perfeitamente o contexto em que o modelo irá atuar. Como alternativa a essa abordagem tradicional, surgiu o paradigma *data programming* [1], utilizado para a criação programática de conjuntos de treinamento, nesse paradigma os usuários expressam estratégias de *weak supervision* [2] para criar funções de rotulagem,

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

as quais eventualmente são ruidosas. A fim de criar um modelo robusto aos ruídos produzidos por essas funções, pesquisadores da universidade de Stanford criaram o sistema Snorkel [3], o qual se beneficia do acordo entre essas funções para a criação do seu modelo. Nesta pesquisa, o Snorkel é utilizado com o objetivo de rotular pessoas em imagens, uma tarefa diferente da qual o Snorkel vem sendo utilizado, pois as aplicações que fazem seu uso são geralmente no contexto de processamento de linguagem natural, pois é mais simples criar heurísticas que atuam em textos. O ambiente em que as imagens foram extraídas foi um laboratório no bloco CN, localizado na UFCG. Com o propósito de comparar o desempenho do modelo treinado com os dados do ambiente específico (gerados pelo Snorkel) e uma base de dados já existente (genérica), um modelo mais sofisticado foi treinado com os diferentes conjuntos de dados. A acurácia final do modelo treinado com os dados gerados pelo Snorkel foi de 84,94%, enquanto que o treinado com imagens genéricas foi de apenas 30%, indicando que o desempenho de um algoritmo de aprendizagem de máquina especializado em um ambiente específico é bastante superior a um treinado com dados genéricos.

PALAVRAS-CHAVE

Aprendizagem de máquina, rotulagem de dados, Snorkel, weak supervision

1 INTRODUÇÃO

Os algoritmos de aprendizagem de máquina vêm ganhando espaço há algum tempo nas mais diversas áreas, isso porque eles podem ser usados para automatizar diferentes atividades que antes eram realizadas apenas por humanos, como: classificação de imagens e geração de textos. Além disso, no mundo dos negócios, as decisões baseadas em dados fazem cada vez mais a diferença entre acompanhar a concorrência ou ficar para trás, pois ao criar modelos precisos, uma organização tem mais chances de identificar oportunidades lucrativas, ou evitar riscos desconhecidos.

O aprendizado de máquina tem seu foco em desenvolver programas de computador que usam dados para aprender por si mesmos, sem a necessidade de serem explicitamente programados, ou seja, ao invés de implementar funções de forma manual, a máquina é treinada usando uma grande quantidade de dados. Esses algoritmos operam construindo modelos (i.e abstrações) a partir de entradas com o objetivo de encontrar padrões implícitos potencialmente úteis.

Existem diversas abordagens de aprendizagem de máquina, as duas mais utilizadas são: aprendizagem supervisionada [4] e não supervisionada [5]. Na aprendizagem supervisionada o algoritmo aprende com dados de treinamento rotulados, ou seja, cada exemplo

no conjunto de dados de treinamento é marcado com a resposta que o algoritmo deve aprender por si próprio. Em outras palavras, o algoritmo recebe variáveis de entrada e uma variável de saída e usa o algoritmo para aprender a função de mapeamento da entrada para a saída. O objetivo é aproximar tão bem a função de mapeamento que, quando novos dados forem passados como entrada, o algoritmo poderá prever as variáveis de saída para esses dados. Esses algoritmos são geralmente divididos em regressão [6] e classificação [7].

A aprendizagem não supervisionada, por outro lado, é uma técnica em que o modelo aprende regras sobre os dados por conta própria. O conjunto de dados de treinamento é uma coleção de exemplos sem rótulos, e assim o algoritmo tenta encontrar automaticamente a resposta correta, extraindo recursos úteis e analisando sua estrutura. As abordagens que utilizam dessa técnica são *clustering* [8] e *association* [9].

No entanto, um dos desafios mais difíceis presentes nas soluções que utilizam aprendizagem de máquina supervisionada é a criação de conjuntos de dados de treinamento com alta qualidade, pois a qualidade dos dados que são utilizados no treinamento refletem na performance do modelo. Em diversos projetos que envolvem aprendizado de máquina, o tempo gasto coletando e rotulando dados costuma ser maior do que criando os modelos de aprendizagem [10]. Além desse problema, os dados de treinamento rotulados de forma manual não são adaptáveis ou flexíveis e, portanto, são totalmente inadequados para tarefas de aprendizado que mudam com o tempo. Como resultado disso, houve um grande movimento em direção a geração de rótulos de treinamento de maneira programática e eficiente, mas eventualmente ruidosa, geralmente referida como *weak supervision*.

O Snorkel é um sistema utilizado para auxiliar na criação de dados de treinamento de forma mais rápida. Ele foi construído em torno do paradigma *data programming*, o qual é uma abordagem em que usuários especialistas no assunto codificam várias funções de rotulagem que expressam diferentes estratégias de *weak supervision*, a partir dos rótulos produzidos por essas funções um modelo generativo [11] criado através da biblioteca Snorkel minimiza os ruídos produzidos por essas funções e gera apenas um rótulo para cada dado, sem acessar o rótulo verdadeiro.

Sendo assim, neste trabalho foi analisada a utilização da biblioteca Snorkel para criar programaticamente conjuntos de dados de treinamento de forma rápida e sem a necessidade de rotulagem manual. O tipo de dado que rotulou-se foi imagens, classificando-as em duas categorias: com pessoa ou sem pessoa. Com o auxílio do Snorkel, foi criado um modelo baseado em heurísticas para detectar pessoas, essas heurísticas representam diferentes estratégias de *weak supervision*, a finalidade desse modelo foi utilizá-lo para criar um conjunto de dados de treinamento para um ambiente específico, o ambiente escolhido foi um laboratório no bloco CN, localizado na UFCG. O principal objetivo dessa pesquisa foi comparar o desempenho de um modelo complexo treinado com um conjunto de dados genérico e com o conjunto de dados gerado pelo Snorkel, sendo os dois modelos avaliados utilizando imagens do ambiente escolhido (o laboratório). A criação dos rótulos de forma programática além de ser mais rápida ainda é flexível a mudanças no contexto da aplicação, pois apenas realizando alguns ajustes nas funções de rotulagem é possível fazer as adaptações desejadas.

O grande desafio enfrentado nesse trabalho é a utilização do Snorkel em um contexto onde o dado a ser rotulado são imagens, pois criar funções para rotular imagens não é uma tarefa fácil, já que toda a informação que se tem são os pixels presentes na imagem, por esse motivo as aplicações que utilizam o Snorkel são geralmente relacionadas ao Processamento de Linguagem Natural.

Para realizar os experimentos foram gravados 3 vídeos com duração de 5 horas e 38 minutos, 12 minutos e 10 minutos, respectivamente, eles foram utilizados para executar o experimento e criar o conjunto de dados de treinamento. Como resultado, foram obtidas 1.967 imagens, sendo 760 rotuladas como "não pessoa" e 1207 como "pessoa". O modelo Xception [12] foi o escolhido para ser treinado com os dados gerados pelo Snorkel e com a base de dados genérica, pois é um modelo de classificação no estado da arte que atende as necessidades desse estudo. A base de dados genérica utilizada para fazer a comparação nesse experimento foi a INRIA Person [13], que possui imagens coletadas como parte de um trabalho de pesquisa sobre detecção de pessoas em imagens e vídeos, contendo no total 1.832 imagens, sendo 1218 exemplos negativos (sem pessoa) e 614 exemplos positivos (com pessoa), essa base de dados foi escolhida porque ela é uma das mais utilizadas no contexto de classificação de pessoas. A acurácia final do modelo treinado com os dados gerados pelo Snorkel foi de 84,94%, enquanto que o treinado com a base de dados INRIA Person foi de apenas 30%, a validação foi realizada em um conjunto de 352 imagens capturadas da sala em que o experimento foi realizado.

Esta pesquisa está organizada da seguinte maneira: na Seção 2, será discutida a fundamentação teórica e trabalhos relacionados, na Seção 3 a metodologia proposta e ao final do trabalho são apresentados os resultados e conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Snorkel e *Weak supervision*

Já foram estudadas e implementadas diversas aplicações que utilizam a biblioteca Snorkel para rotular dados automaticamente [14–16], entretanto essas aplicações foram criadas com o objetivo de rotular dados textuais, isso ocorre porque a dificuldade para criar funções de rotulagem que atuam em imagens é bem maior, visto que apenas os pixels presentes nas imagens não possuem muita informação. A abordagem *weak supervision* [2] também foi estudada, pois o Snorkel é baseado nela, no entanto ele utiliza um método baseado em uma matriz de acordos [17] para minimizar os ruídos produzidos por essa abordagem.

Weak supervision é um ramo do aprendizado de máquina em que fontes ruidosas, limitadas ou imprecisas são usadas para fornecer um sinal de supervisão com o objetivo de rotular grandes quantidades de dados de treinamento em um ambiente de aprendizado supervisionado [18]. Essa abordagem é muito importante pois conjuntos de treinamento rotulados à mão são caros e demorados para serem criados, especialmente quando é necessário conhecimento de domínio. Além disso, apesar dos rótulos serem ruidosos, eles ainda podem ser usados para a criação de modelos preditivos fortes. Exemplos que podem ser considerados *weak supervision* incluem: (1) heurísticas sobre o domínio, (2) uso de um banco de dados externo já existente que não são exatamente adequados para a tarefa

em questão, mas próximos o suficiente para serem úteis (tradicionalmente chamado de *distant supervision* [19]) e (3) *crowdsourcing* [20].

A finalidade do Snorkel é justamente usar um conjunto de funções de rotulagem que expressam diferentes estratégias de *weak supervision* para criar um modelo baseado na eficácia dessas diferentes estratégias. Ao observar a frequência em que as funções de rotulagem concordam ou discordam, as acurácias estimadas para cada fonte de supervisão é aprendida. Dessa forma, os votos de todas as funções de rotulagem são combinados, de maneira que um modelo generativo [11] atribui programaticamente rótulos probabilísticos a diversos dados. Por fim, o objetivo é treinar um modelo mais robusto que possa generalizar mais do que as funções de rotulagem individuais.

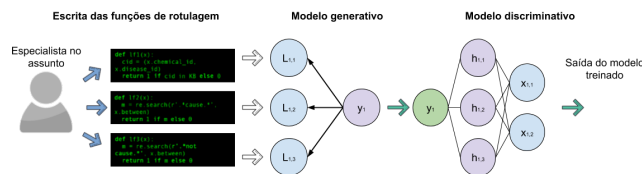


Figura 1: Arquitetura do Snorkel: (1) um especialista no assunto escreve funções de rotulagem, (2) cada função de rotulagem retorna um valor, o modelo generativo combina essas saídas e atribui apenas um rótulo a cada dado, (3) a partir dos rótulos gerados pelo modelo generativo, um modelo discriminativo (ex. uma rede neural) é treinado

2.2 Trabalhos relacionados

A biblioteca Snorkel já foi e ainda está sendo utilizada em diversos estudos, em um deles foi proposto um paradigma de classificação de textos clínicos [14], o objetivo do estudo foi similar ao que está sendo proposto neste trabalho, utilizar *weak supervision* para gerar automaticamente rótulos para dados de treinamento e, em seguida, treinar modelos de aprendizagem de máquina com os dados gerados. A validação da eficácia da abordagem utilizada foi feita usando dois estudos de caso institucionais: classificação do status de fumantes e fratura do fêmur proximal (quadril) e um estudo de caso usando um conjunto de dados público. No estudo foram testados quatro diferentes modelos: *Support Vector Machine* (SVM), *Random Forest* (RF), *Multilayer perceptron neural network* (MLPNN) e *Convolutional Neural Network* (CNN), os resultados de ambos os experimentos institucionais mostraram que a CNN foi, nesse caso, a melhor opção e que ela pode superar os algoritmos de Processamento de Linguagem Natural (PNL) baseados em regras.

No trabalho de Abraham Starosta [16] ele construiu um classificador para detectar *tweets* antisemitas combinando *weak supervision* e *transfer learning* [21], ele utilizou a seguinte metodologia: (1) coletou aproximadamente 25.000 *tweets* que possuíam a palavra "judeu" e rotulou 600 deles, (2) construiu um conjunto de treinamento com o Snorkel utilizando 24 funções de rotulagem e utilizou os 600 *tweets* rotulados para validar as funções criadas, por fim, (3) criou o modelo de classificação, para isso ele treinou o modelo ULMFiT [22]. Ao final do estudo o autor calculou a precisão e o *recall*, sendo

os resultados 95% e 39%, respectivamente, sendo assim, com o auxílio do Snorkel foi possível construir um bom classificador sem a utilização de um grande conjunto de dados previamente rotulado.

Em outro estudo [23], o Snorkel foi utilizado para rotular dados médicos relacionados à radiografia, tomografia computadorizada e eletroencefalografia, nesse caso o estudo conseguiu provar que é possível construir modelos de aprendizado de máquina em medicina de maneira mais rápida e flexível, pois ao contrário da abordagem tradicional, onde são passados meses rotulando dados, apenas são necessárias algumas horas para a implementação das funções de rotulagem. No entanto, um problema enfrentado por esse estudo foi a criação de funções para imagens que identificam precisamente a patologia, por isso, os autores optaram por utilizar dados de relatórios ou até metadados sobre as imagens para criar as funções.

A metodologia utilizada nessa pesquisa é semelhante aos trabalhos citados anteriormente, onde resumidamente ocorre as etapas de (1) coleta de dados, (2) implementação das funções de rotulagem, (3) criação do conjunto de treinamento e, finalmente, (4) o treinamento de um modelo de aprendizagem de máquina utilizando o conjunto de dados gerado.

Além de trabalhos em que o Snorkel é utilizado para a criação de dados de treinamento, existem também outros sistemas que foram criados sobre a estrutura dessa biblioteca. O Osprey [15] é um exemplo, sendo um sistema de *weak supervision* utilizado quando se tem dados altamente desbalanceados, o sistema foi aplicado em dados de mídia social altamente desbalanceados (0,05% positivo) e os resultados mostraram ganhos de 18,5 pontos de acurácia e 28,5 pontos de cobertura sobre abordagens anteriores tradicionalmente supervisionadas e baseadas em regras.

Os casos de estudo e aplicações mais famosos que utilizam o Snorkel em sua implementação estão citados no site oficial da biblioteca [24].

3 METODOLOGIA

Para realizar a tarefa escolhida nesse estudo (classificação de pessoas em imagens) foi necessário primeiramente escolher o ambiente em que as imagens foram extraídas. Em uma primeira tentativa, foram instaladas duas câmeras na copa do Laboratório de Sistemas Distribuídos localizado na UFCG, cada uma delas visando um ângulo diferente, mas ainda no mesmo ambiente. Porém, a diferença no contraste de luz ao passar do dia dificultou o desempenho das funções de rotulagem, então foi decidido utilizar um ambiente mais controlado, um laboratório no bloco CN, também na UFCG, pois assim foi possível controlar a luz do ambiente. Imagens das duas câmeras do primeiro cenário são mostradas nos apêndices desse trabalho na tentativa de evidenciar o porque da utilização desse ambiente ter sido abortada.

Para facilitar a coleta dos dados foram gravados 3 vídeos com duração de 5 horas e 38 minutos, 12 minutos e 10 minutos, respectivamente, esses vídeos foram capturados entre às 9 horas e 16 horas. Com a finalidade de salvar alguns *frames* do vídeo, foi criado um *script* em que a cada 330 *frames* um era salvo, gerando assim um conjunto de 1.967 imagens, esse foi o conjunto de imagens rotuladas ao final da pesquisa. Porém, além dessas imagens, foram coletadas mais 352 para auxiliar na validação das funções de rotulagem, essas imagens também foram obtidas a partir de 3

Tabela 1: Acurácia das funções de rotulagem em um subconjunto de imagens obtido de *datasets* genéricos

Função de rotulagem	Acurácia
HOG	68%
Haar Cascade (perfil)	92%
Haar Cascade (frontal)	88%

vídeos, com duração média de 25 minutos, cada vídeo foi gravado em um horário diferente, um no horário da manhã, outro no horário do almoço (entre 12 e 14 horas) e o último no horário da tarde. Essa validação foi realizada para verificar e garantir que as funções de rotulagem tivessem o desempenho desejado, portanto, essas 352 imagens foram rotuladas manualmente para calcular métricas de acurácia e *F1 score* [25] das funções.

Depois da coleta dos dados, foram estudadas heurísticas para detecção de pessoas, inicialmente, foram escolhidas o *Histogram of oriented gradients* (HOG) [26] e o Haar Cascade [27] (para detecção de face frontal e perfil). Essa escolha foi feita porque ao avaliar essas funções de rotulagem em 36 imagens aleatórias obtidas dos *datasets* INRIA Person e FEI Face Database [28] os resultados foram satisfatórios (os resultados estão expostos na Tabela 1).

Além das funções de rotulagem citadas no parágrafo anterior, foram acrescentadas uma *Convolutional Neural Network* (CNN) e um detector de pele, e as duas funções de rotulagem que detectavam faces frontais e de perfil passaram a ser apenas uma, totalizando até o momento 4 funções de rotulagem.

Até agora as funções de rotulagem não estavam sendo avaliadas no cenário proposto, e sim com um subconjunto de imagens obtido de *datasets* genéricos (INRIA Person e FEI Face Database), sendo assim, as 352 imagens obtidas de vídeos do ambiente foram rotuladas manualmente para realizar essa avaliação. Quando o cenário foi alterado as acurácias do HOG e da CNN diminuíram muito, 5% e 9%, respectivamente. Isso se deu ao fato de que esses modelos haviam sido pré-treinados com imagens em que as pessoas estavam em evidência e mostrando o corpo completo, enquanto que no cenário escolhido neste trabalho [Figura 7] as pessoas estão sentadas na maior parte do tempo, desta forma, foi decidido que essas funções de rotulagem deviam ser substituídas.

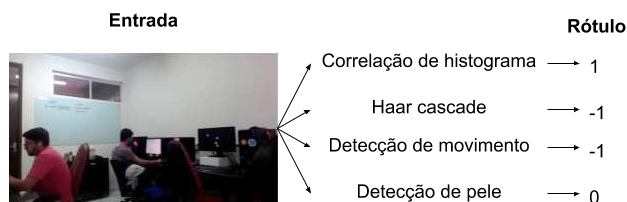


Figura 2: Exemplo dos diferentes rótulos produzidos pelas funções de rotulagem para uma mesma imagem

Com o objetivo de melhorar o resultado das funções de rotulagem, foram realizados alguns pré-processamentos nas imagens, como alterar o brilho através da correção *gamma*, que é uma operação não linear usada para codificar e decodificar valores de luminância

em sistemas de vídeo ou imagem fixa, além dessa correção, também foi feita a equalização do histograma da imagem. Porém, o melhor resultado foi recortando a imagem onde há objetos, por isso, foi decidido acrescentar uma função de rotulagem para detectar movimentos com o objetivo de criar os *bound boxes* ao redor dos objetos e passar a imagem recortada para as funções de rotulagem. No entanto, preferiu-se eliminar as funções de rotulagem que estavam com um desempenho abaixo do esperado e utilizar o detector de movimentos como uma outra função de rotulagem independente. Além dela foi adicionada outra para comparar histogramas entre duas imagens, substituindo o HOG e a CNN. A seguir serão explicadas quais as funções de rotulagem que finalmente foram utilizadas e seus funcionamentos:

- **Correlação de histograma:** foi selecionada uma imagem do *background* do ambiente para compará-la com as imagens a serem rotuladas. Essa comparação foi feita utilizando a função *compareHist* com o método distância *Bhattacharyya* da biblioteca OpenCV [29], essa função retorna um número que expressa o quanto dois histogramas se combinam. Portanto, depois de realizar alguns experimentos foi verificado que os limites inferiores e superiores deviam ser 0.07 e 0.1, respectivamente, ou seja, quando o valor é inferior a 0.07 a imagem pertence a classe "não pessoa", entre 0.07 e 0.1 a função se abstém de rotular, e acima de 0.1 é rotulada como "pessoa". Como o *background* do cenário é sempre o mesmo, essa função de rotulagem foi escolhida pois ela consegue identificar pessoas através de alterações no histograma de cores entre o *background* fixo e outra imagem qualquer do ambiente.
- **Haar cascade:** foram adicionadas mais funcionalidades a essa função de rotulagem, além de detectar faces frontais e perfis, adicionou-se também a detecção de corpo completo, parte do corpo superior e inferior. Essa função utiliza o classificador *CascadeClassifier* também do OpenCV, com pesos pré-treinados. Nesse caso, a função nunca se abstém de realizar a classificação, se nada for detectado então a função classifica como "não pessoa", caso contrário, como "pessoa". Essa função foi escolhida porque na maioria dos casos, no cenário proposto, apenas parte do corpo das pessoas está sendo exposta, já que usualmente as pessoas estão sentadas.
- **Detecção de movimento:** nessa função de rotulagem foi utilizada a técnica *Mixture of Gaussian* (MOG), disponibilizada através da função *createBackgroundSubtractorMOG* do OpenCV. A ideia do algoritmo é detectar movimentos a partir de um fundo estático, então sempre que existir algum movimento a função de rotulagem classifica o *frame* como "pessoa", caso não exista movimento é classificado como "não pessoa", essa função de rotulagem assim como a anterior nunca se abstém.
- **Detecção de pele:** essa função atua na tonalidade das imagens, inicialmente, é definido um intervalo que representa as cores consideradas pele, a partir desse intervalo, calcula-se o percentual de pele na imagem, caso ele seja maior do que um determinado limiar, considera-se que existe pessoa na imagem. O limiar inferior escolhido foi 0.09 e o superior 0.1.

Ao avaliar o desempenho dessas funções de rotulagem no subconjunto de 352 imagens, os resultados foram positivos [Tabela 2], por isso, o experimento continuou apenas com essas 4 funções de rotulagem.

Após realizar a avaliação das funções de rotulagem, foi realizado o experimento de criar os diferentes conjuntos de treinamento para cada uma delas. Os dados utilizados para criar os *datasets* foram os 3 vídeos com duração de 5 horas e 38 minutos, 12 minutos e 10 minutos, respectivamente. Como citado anteriormente, o *script* rotula um a cada 330 *frames*, o mesmo *frame* foi rotulado por todas as funções de rotulagem, como algumas se abstém de classificar então a quantidade de imagens final de cada conjunto de dados gerado não é necessariamente igual.

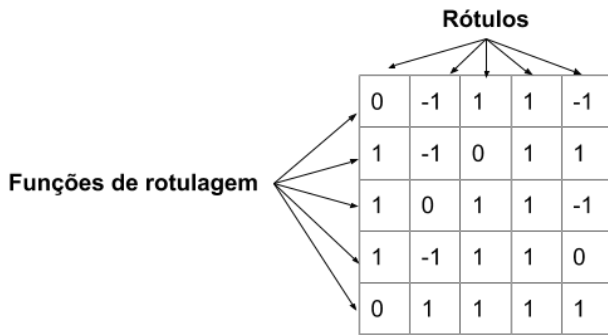


Figura 3: Exemplo de matriz que o modelo generativo do Snorkel recebe, nesse caso, 5 funções de rotulagem geraram rótulos para 5 imagens, cada coluna na matriz representa diferentes rótulos gerados para a mesma imagem

Depois da geração dos *datasets* para cada função de rotulagem, foi criado o modelo generativo a partir da biblioteca Snorkel, esse modelo recebe uma matriz [Figura 3] onde as linhas representam as funções de rotulagem e as colunas são os rótulos gerados por elas. No experimento realizado nesse trabalho, a matriz tem dimensão 4x1967, já que existem 4 funções de rotulagem e 1.967 imagens. O modelo generativo obtém vantagem dos conflitos entre as funções de rotulagem com a finalidade de estimar suas acurácias, ao observar a frequência com que as funções de rotulagem concordam ou discordam uma das outras, o modelo aprende acurácias estimadas para cada função de rotulagem (por exemplo, uma função de rotulagem em que as outras tendem a concordar terá uma alta acurácia, enquanto outra que parece discordar de todas as outras sempre que votar no mesmo exemplo terá uma baixa acurácia). Ao combinar os votos de todas as funções de rotulagem (ponderadas por suas acurácias estimadas), atribui-se um rótulo "sensível ao ruído" (entre 0 e 1, em vez de 0 ou 1) para cada exemplo. Portanto, ao rotular um novo dado, cada função de rotulagem dará um voto: positivo, negativo ou abstenção (caso a função tenha a opção de se abster), e então com base nesses votos e nas estimativas de acurácia das funções de rotulagem o modelo generativo atribui programaticamente rótulos probabilísticos aos dados. O Snorkel se beneficia das abstenções na criação do modelo generativo, o ideal é que as funções tenham uma média de abstenção, pois se houver muita abstenção o rótulo pode se desviar muito do voto da maioria, e caso nunca haja abstenção

Tabela 2: Desempenho das funções de rotulagem e do modelo generativo no subconjunto de 352 imagens

Função de rotulagem	Acurácia	F1 score
Correlação de histograma	69.60%	0.81
Haar Cascade	76.70%	0.79
Detecção de movimento	98.01%	0.97
Detecção de pele	69.60%	0.81
Modelo generativo (Snorkel)	84.94%	0.89

o voto da maioria acabará sendo ideal e a utilização do modelo generativo não será necessária. Nesse experimento o modelo foi treinado por 100 épocas [30].

Com o modelo generativo já treinado, foram passadas todas as imagens para que ele fizesse a classificação e gerasse o conjunto de treinamento final. Ao contrário de algumas funções de rotulagem, esse modelo não se abstém de rotular, então o conjunto de dados final gerado foi de 1.967 imagens.

Em seguida, foram realizados dois experimentos, o primeiro foi o treinamento do modelo Xception com os diferentes conjuntos de dados gerados por cada função de rotulagem e também pelo modelo generativo e o segundo foi o treinamento do mesmo modelo (Xception) com o conjunto de treinamento INRIA Person, esse conjunto de dados possui 1.832 imagens no total, ou seja, a quantidade de imagens é próxima ao do *dataset* gerado pelo modelo generativo.

Ao final desses experimentos, foram calculadas as métricas de acurácia e F1 score para o modelo com os diferentes tipos de treinamento. Os parâmetros utilizados nos treinamentos foram os seguintes: otimizador SGD, função de *loss categorical crossentropy* e as imagens foram redimensionadas para 299X299 pixels.

Nas seções seguintes são mostrados os resultados e conclusões obtidos a partir da realização dos experimentos e ao final do artigo estão anexadas outras informações julgadas importantes.

4 RESULTADOS E DISCUSSÕES

A partir dos experimentos mostrados anteriormente, obteve-se a coleta de algumas informações importantes. Antes de chegar aos resultados finais, serão apresentados alguns resultados parciais que ajudarão no entendimento dos experimentos como um todo.

Antes de rotular os dados do ambiente escolhido, foram selecionadas 36 imagens aleatórias obtidas dos *datasets* INRIA Person e FEI Face Database para avaliar o desempenho do modelo generativo na geração de seus rótulos, nesse experimento inicial foram utilizadas apenas 3 funções de rotulagem que foram anteriormente citadas, HOG, Haar cascade para detecção de faces frontais e perfis, o desempenho das funções já foi mostrado na Tabela 1, o modelo generativo teve nesse caso 88% de acurácia, o que foi um bom desempenho, mas ainda igual ao da função *Haar cascade* (frontal), porém, o conjunto de dados nesse caso foi muito pequeno, então não é possível tirar conclusões até o momento.

Ao avaliar o desempenho dessas funções de rotulagem no subconjunto de 352 imagens do nosso contexto, foram obtidos os resultados presentes na Tabela 2.

Os resultados do modelo generativo foram bons, só não foram superiores aos da função de rotulagem que utiliza a detecção de

movimento. Porém, existem algumas observações que devem ser levadas em consideração. Primeiro, o Snorkel deve atuar com rotuladores fracos, ou seja, com acurácia entre 55% e 80%, mas ao observar os valores obtidos pelo detector de movimento pode-se concluir que para esse contexto ele já é um rotulador forte, segundo, foi realizado um teste para verificar a correlação das funções de rotulagem utilizando o coeficiente *Cohen's Kappa* [31], o resultado está presente na Figura 4.

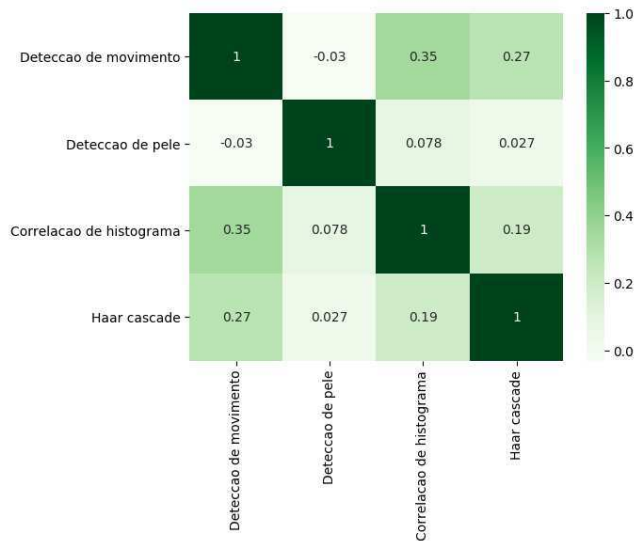


Figura 4

De acordo com os resultados mostrados na Figura 4, observa-se que não há correlação muito forte entre as funções de rotulagem, o que é bom, pois funções muito correlacionadas não agregam tanto valor ao modelo generativo construído pelo Snorkel.

Além disso, outro ponto importante é que algumas funções de rotulagem podem descartar dados ao gerar o modelo final, ao contrário do modelo generativo. No nosso caso, as funções de rotulagem de correlação de histograma e detecção de pele obtiveram um conjunto de dados de treinamento menor do que as outras, o que pode ter prejudicado seus desempenhos, ainda mais a de detecção de pele que gerou menos de 50% dos dados iniciais. Esses dados podem ser observados na Figura 5.

No entanto, o resultado mais importante para o nosso estudo foi a comparação de desempenho do modelo Xception treinado com os dados gerados pelo modelo generativo e o mesmo modelo treinado com a base de dados INRIA Person, lembrando que a validação foi realizada com os dados do ambiente escolhido (sala no bloco CN).

Esse resultado deixa claro o quanto os dados de treinamento podem influenciar no desempenho do modelo. Apesar de ambos os modelos terem sido treinados para classificar pessoas, o ambiente de classificação escolhido é bem diferente das imagens do INRIA Person, como mostrado na Figura 7, e encontrar *datasets* que sejam semelhantes ao contexto escolhido é em muitos casos uma tarefa difícil ou até mesmo impossível, por isso, a utilização de *weak supervision* e do modelo generativo do Snorkel é importante para

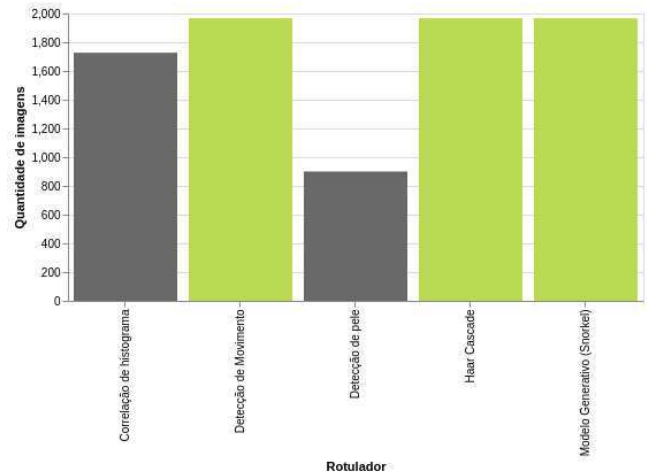


Figura 5: Quantidade de imagens geradas por cada rotulador

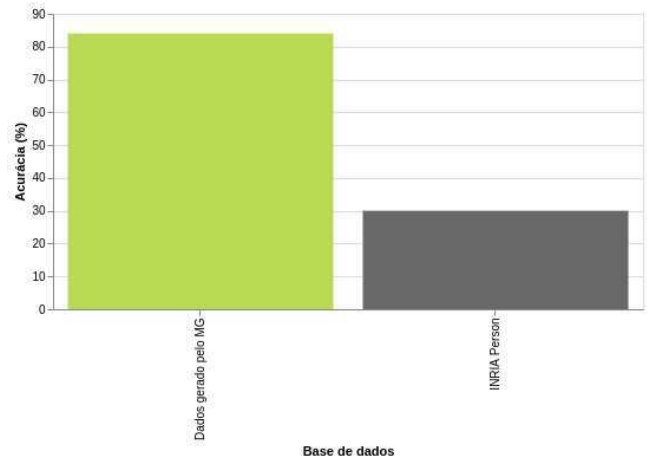


Figura 6: Acurácia do modelo Xception treinado em diferentes bases de dados

auxiliar na construção de conjuntos de treinamento para modelos de aprendizagem de máquina.

5 CONCLUSÃO

A *weak supervision* é uma abordagem que facilita a criação de conjuntos de treinamento para modelos de aprendizagem de máquina. O resultado obtido através do modelo generativo tem um grande impacto na geração de rótulos de forma programática. Uma vez que as funções de rotulagem já estejam implementadas, apenas é necessário alimentá-las para que novos dados rotulados sejam gerados. Neste trabalho foi possível observar que quando um modelo é treinado com dados do ambiente em que será utilizado, o seu desempenho é muito melhor do que se treinado com uma base de dados genérica já existente, além disso, se houver qualquer mudança no ambiente, apenas será preciso fazer alguns ajustes nas funções de rotulagem e elas ainda continuarão atendendo as necessidades desejadas, refletindo que utilizar *weak supervision* é algo



Figura 7: A imagem superior é um exemplo de uma imagem extraída do *dataset* INRIA Person, a outra foi extraída do conjunto de validação utilizado nesse estudo

que realmente pode auxiliar e facilitar a construção de grandes conjuntos de dados de maneira rápida. Ainda foi possível observar que o Snorkel encapsula essa técnica de forma inteligente, apesar de a função de rotulagem que utiliza detecção de movimento ter resultados superiores, talvez se tivessem sido criadas mais funções de rotulagem com acurácia entre 55% e 80% o resultado do modelo generativo fosse superior, porém, como já citado anteriormente, a criação de funções de rotulagem para atuar em imagens é bem mais difícil do que em textos, por exemplo.

Nesta perspectiva, a utilização da *weak supervision* permite que grandes conjuntos de dados sejam rotulados sem a necessidade de que pessoas conhecedoras do domínio passem meses rotulando dados, ao contrário disso, são necessárias apenas algumas horas para a criação das funções rotuladoras. Sendo assim, além de diminuir o tempo de criação de conjuntos de dados, o custo para a rotulagem também acaba diminuindo.

Como trabalhos futuros, pretende-se estudar a criação de novas funções de rotulagem e ajustar alguns limiares das funções já existentes, para que elas consigam refletir ainda mais o cenário desejado. Além disso, planeja-se também aplicar a metodologia utilizada nessa pesquisa para os mais diversos cenários, para assim aumentar a confiabilidade das abordagens utilizadas. Por fim, um comparativo entre várias aplicações que utilizaram essa abordagem e a abordagem tradicional pode fazer com que a relevância dessa técnica seja ainda mais reconhecida e utilizada.

6 APÊNDICES

Esse foi um cenário muito difícil de ser controlado [Figuras 8 e 9], pois a parede de vidro permitia que as mudanças fora do laboratório



Figura 8: Diferença no contraste que prejudicou a realização do experimento na copa do Laboratório de Sistemas Distribuídos. Ângulo 1.



Figura 9: Diferença no contraste que prejudicou a realização do experimento na copa do Laboratório de Sistemas Distribuídos. Ângulo 2

fizessem com que a luz oscilasse a quase todo momento, por isso,

foi escolhido um ambiente mais simples e fácil de controlar para a realização desse estudo.

REFERÊNCIAS

- [1] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly, 2016.
- [2] Braden Hancock Chris Ré e outros membros do Hazy Lab Alex Ratner, Paroma Varma. Weak supervision: A new programming paradigm for machine learning, March 2019. [Online; acessado em 16 de outubro de 2019].
- [3] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel. *Proceedings of the VLDB Endowment*, 11(3):269–282, Nov 2017.
- [4] Supervised learning, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [5] Unsupervised learning, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [6] Apoorva Dave. Regression in machine learning, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [7] Sidath Asiri. Machine learning classifiers, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [8] Surya Priy. Clustering in machine learning, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [9] Association rule learning, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [10] Astasia Myers. Data labeling: Creating ground truth, 2019. "[Online; acessado em 20 de novembro de 2019]".
- [11] Background: What is a generative model? [Online; acessado em 22 de outubro de 2019].
- [12] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2016.
- [13] Navneet Dalal. Inria person dataset, 2005. [Online; acessado em 1 de outubro de 2019].
- [14] Sijia Liu Feichen Shen Liwei Wang Elizabeth J. Atkinson Shreyasee Amin e Hongfang Liu Yanshan Wang, Sunghwan Sohn. A clinical text classification paradigm using weak supervision and deep representation. January 2019.
- [15] Eran Bringer, Abraham Israeli, Yoav Shoham, Alex Ratner, and Christopher Ré. Osprey: Weak supervision of imbalanced extraction problems without code. pages 1–11, 06 2019.
- [16] Abraham Starosta. Building nlp classifiers cheaply with transfer learning and weak supervision, February 1019. [Online; acessado em 16 de outubro de 2019].
- [17] Weak supervision, 2019. [Online; acessado em 21 de outubro de 2019].
- [18] Distant supervision, 2019. "[Online; acessado em 13 de novembro de 2019]".
- [19] B S Abhigna, Nitasha Soni, and Shilpa Dixit. Crowdsourcing – a step towards advanced machine learning. 2018.
- [20] Jason Brownlee. A gentle introduction to transfer learning for deep learning, December 2017. [Online; acessado em 24 de outubro de 2019].
- [21] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.
- [22] Jared Dunnmon, Alexander Ratner, Nishith Khandwala, Khaled Saab, Matthew Markert, Hersh Sagreiya, Roger Goldman, Christopher Lee-Messer, Matthew Lungren, Daniel Rubin, and Christopher Ré. Cross-modal data programming enables rapid medical machine learning, 2019.
- [23] Snorkel applications, 2019. "[Online; acessado em 26 de outubro de 2019]".
- [24] F1 score, 2019. "[Online; acessado em 14 de novembro de 2019]".
- [25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.
- [26] Viola-jones object detection framework, 2019. "[Online; acessado em 28 de outubro de 2019]".
- [27] Carlos Eduardo Thomaz. Fei face database. "[Online; acessado em 28 de outubro de 2019]".
- [28] Opencv, 2019. "[Online; acessado em 16 de novembro de 2019]".
- [29] Sagar Sharma. Epoch vs batch size vs iterations, 2017. "[Online; acessado em 21 de novembro de 2019]".
- [30] Cohen's kappa, 2019. "[Online; acessado em 01 de novembro de 2019]".
- [31] Cohen's kappa, 2019. "[Online; acessado em 01 de novembro de 2019]".