Evaluation of Change Detection Algorithms using Difficulty Maps

S. R. R. Sanches, C. G. Corrêa, B. R. Brum, P. H. Bugatti, P. T. M. Saito, C. M. Silva and E. Custódio Júnior

***Abstract*—The evaluation of a change detection algorithm should show its superiority over state-of-the-art algorithms' performances. Evaluating an algorithm involves executing it to segment a set of videos and comparing the results with the ground truth. Here, we used the difficulty level to classify each pixel of each frame of the videos of a dataset as an algorithm performance measure. A structure called "difficulty map" stores information about the difficulty of classifying each pixel in a frame. Based on these maps, we developed a metric that aims to evaluate the performance of algorithms on the difficulty map. The results showed that there are algorithms with the characteristic of classifying pixels that most state-of-the-art algorithms cannot classify (promising algorithms). Identifying such algorithms is essential since improving their performance means facing challenges already overcome by existing approaches.**

***Index Terms*—Algorithm evaluation, change detection, difficulty map, video segmentation.**

I. Introdução

Vigilância por vídeo, ambientes inteligentes e recuperação de conteúdo são exemplos de sistemas que utilizam algoritmos de detecção de mudança (*change detection*) na imagem monitorada [1]–[5]. Tais algoritmos identificam regiões (conjunto de pixels) que sofrem modificações ou se movem em relação aos pixels que representam o plano de fundo da cena [6]. A detecção de mudança é pré-requisito para muitas aplicações de visão computacional e processamento de vídeo [6].

A avaliação do desempenho de um algoritmo de detecção de mudança é uma tarefa fundamental em que se deve mostrar claramente que um novo algoritmo apresentado é superior em desempenho aos encontrados na literatura. As etapas da avaliação de desempenho de um algoritmo consistem basicamente na sua execução para segmentar um conjunto de dados compostos por vídeos, chamado *dataset*, e comparar os resultados com um *ground truth*. Um *ground truth* é um conjunto de quadros rotulados manualmente por um especialista, que permite identificar o resultado ideal da segmentação. Dessa forma, os resultados da segmentação dos algoritmos, que normalmente contêm erros de classificação de pixels, são obtidos e comparados com o *ground truth*. Em seguida, são calculadas métricas que representam o desempenho do algoritmo (Acurácia, *F-Measure*, Revocação, Precisão, etc).

Quando vários algoritmos utilizam o mesmo conjuntos de vídeos e as mesmas métricas no processo de avaliação, é possível comparar seus desempenhos, pois os resultados desses algoritmos são obtidos considerando as mesmas ferramentas e métodos. Essa forma tradicional de avaliar desempenho é bem aceita pela comunidade científica. No entanto, a avaliação realizada dessa forma não possibilita identificar as regiões dos quadros em que os erros de classificação ocorrem.

Quando muitos algoritmos erram nas mesmas regiões de determinado quadro de um mesmo vídeo, presume-se que o nível de dificuldade para classificar aquele conjunto de pixels é alto [1]. Uma vez que os algoritmos de detecção de mudança se baseiam nas mais diferentes abordagens (redes neurais, algoritmos genéticos, lógica *fuzzy*, etc) [7], pode haver um método capaz de classificar corretamente os pixels de dificuldade alta, mas que pode falhar em regiões do quadro que são classificadas corretamente pela maioria dos algoritmos do estado-da-arte. As métricas tradicionais não são capazes de identificar esse comportamento do algoritmo.

Um algoritmo com essa característica pode ser considerado promissor, pois a tarefa de melhorar o seu desempenho consiste em enfrentar desafios que foram superados por soluções que podem ser encontradas na literatura. Isso significa que se um pesquisador pretende combinar algoritmos ou melhorar o desempenho de um algoritmo existente para que este supere o desempenho dos disponíveis na literatura, ele pode ter seu trabalho facilitado se o algoritmo escolhido for promissor. Como identificar um algoritmo promissor? Para auxiliar os pesquisadores, neste trabalho, apresenta-se uma métrica, baseada no uso de mapa de dificuldade, capaz de identificar algoritmos de detecção de mudança promissores. No melhor do nosso conhecimento, métricas com essa finalidade não são encontradas na literatura.

II. Avaliação de Algoritmos

Para mostrar o desempenho de seus algoritmos, alguns pesquisadores adotam a estratégia de criar seus próprios vídeos e *ground truths*. Essa estratégia, no entanto, não permite comparar o desempenho de um novo algoritmo com o desempenho dos algoritmos que representam o estado-da-arte. Para facilitar essa comparação, a maioria dos pesquisadores utiliza conjunto de vídeos disponíveis na literatura [8].

S. Sanches, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, silviosanches@utfpr.edu.br

C. Corrêa, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, clebergimenez@utfpr.edu.br

B. Brum, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, beatrizbrum2009@hotmail.com

P. Bugatti, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, pbugatti@utfpr.edu.br

P. Saito, Universidade Federal de São Carlos, São Carlos, Brasil, priscila-saito@ufscar.br

C. Silva, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, claudinei.moreira@fema.edu.br

E. Custódio Junior, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Brasil, elton_junior@outlook.com.

A avaliação de um novo algoritmo de detecção de mudança compreende quatro etapas: (i) executar o algoritmo para segmentar vídeos de um conjunto, (ii) comparar os resultados da segmentação com um *ground truth*, (iii) calcular uma métrica ou um conjunto de métricas que representam o desempenho do algoritmo, e (iv) comparar o desempenho do novo algoritmo com os desempenhos dos algoritmos que representam o estado-da-arte. Esse tipo de avaliação é aplicada tanto durante a fase de desenvolvimento, para testar versões intermediárias, quanto depois de finalizado o desenvolvimento.

O recurso mais importante no processo de avaliação de algoritmos é o conjunto de vídeos (*dataset*). Além dos vídeos, normalmente estão disponíveis *ground truths*, resultados de algoritmos que representam o estado-da-arte e ferramentas de auxílio aos pesquisadores. A utilização de conjuntos de vídeos conhecidos permite a comparação do desempenho de vários algoritmos uma vez que algoritmos diferentes são executados e produzem resultados da segmentação considerando um mesmo conjunto de vídeos [1]. conjunto de vídeos para avaliar algoritmos de detecção de mudanças normalmente contém vídeos com cenas que simulam situações típicas do ambiente de uma determinada aplicação [9].

Outro componente importante na avaliação de algoritmos de detecção de mudança são as métricas. Muitos conjuntos de vídeos oferecem aos pesquisadores ferramentas que calculam essas métricas e os resultados obtidos são utilizados para comparar um novo algoritmo com o estado-da-arte.

Considerando valores normalizados, os resultados dos algoritmos de detecção de mudança geram uma máscara $S \in \{0,1\}^{l \times c}$, onde 0 é o rótulo dos pixels classificados como fundo, 1 é o rótulo dos pixels classificados como elemento de interesse e $l \times c$ é a resolução do quadro [10]. Os rótulos do *ground truth* $G \in [0,1]^{l \times c}$ são 0 e 1, que representam os pixels pertencentes ao plano do fundo e os pertencentes a um elemento de interesse, respectivamente. Adicionalmente, existem rótulos para regiões como sombras, regiões fora da área de interesse na cena e regiões desconhecidas [9].

Dada a norma da matriz $\|A\| = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|$, os verdadeiros positivos (pixels classificados corretamente como pertencentes ao elemento de interesse) são definidos como $TP = \|G \odot S\|$, onde $\odot$ é o produto *entrywise*. Os falsos positivos (pixels do plano de fundo classificados incorretamente como pertencentes a um elemento de interesse) são definidos como $FP = \|(1-G) \odot S\|$. Os verdadeiros negativos (pixels do plano de fundo classificados corretamente como pertencentes ao plano de fundo) são definidos como $TN = \|(1-G) \odot (1-S)\|$. E, por fim, os falsos negativos (pixels do elemento de interesse incorretamente classificados como pertencentes ao plano de fundo) são definidos como $FN = \|G \odot (1-S)\|$ [10].

As métricas taxa de falsos positivos (*False Positive Rate* – FPR) e taxa de falsos negativos (*False Negative Rate* – FNR), por exemplo, podem ser calculadas de acordo com as equações 1 e 2.

$$FPR = \frac{FP}{FP + TN} \tag{1}$$

$$FNR = \frac{FN}{TP + FN}. \tag{2}$$

Outras métricas utilizadas para avaliar o desempenho de algoritmos de detecção de mudanças são Precisão (Pr), Revocação (Re), Acurácia (Ac), Especificidade (Sp), Percentual de Erros de Classificaçnao (*Percentage of Wrong Classifications* - PWC) e *F-Measure* ($F1$), definidas pelas Equações 3–8, respectivamente. A $F1$, baseada nas métricas Pr e Re, é uma das métricas mais utilizadas para representar o desempenho de algoritmos de detecção de mudança [6], [9].

$$Pr = \frac{TP}{TP + FP}, \tag{3}$$

$$Re = \frac{TP}{TP + FN}, \tag{4}$$

$$Ac = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

$$Sp = \frac{TN}{TN + FP}, \tag{6}$$

$$PWC = \frac{100 \times (FN + FP)}{TP + FN + FP + TN}, \tag{7}$$

$$F1 = \frac{2 \times Pr \times Re}{Pr + Re}. \tag{8}$$

Essas métricas, apesar de utilizadas pela comunidade científica, não consideram a localização espacial dos erros de classificação de pixels cometidos pelos algoritmos. Quando um pesquisador pretende modificar um algoritmo existente na literatura com o objetivo de melhorar seu desempenho, ele pode ter seu trabalho facilitado se o algoritmo escolhido classificar pixels que a maioria erra, mesmo que o seu desempenho geral seja similar ou menor aos desempenhos dos algoritmos do estado-da-arte. Neste trabalho, é proposta uma métrica que identifica algoritmos com essa característica (algoritmos promissores).

III. MÉTODOS

A abordagem proposta no presente trabalho consiste em inicialmente gerar uma estrutura chamada mapa de dificuldade [10]. O mapa de dificuldade armazena o nível de dificuldade exigido para um algoritmo classificar corretamente os pixels de um quadro de vídeo. A métrica $F1_D$, utilizada para avaliar novos algoritmos de detecção de mudança, é calculada considerando o *ground truth* ponderado pelo mapa. A abordagem proposta é exibida na Figura 1.

Primeiramente, descreve-se nesta seção o processo de geração de um mapa de dificuldade e, em seguida, apresenta-se a forma de utilizar o mapa para gerar a métrica $F1_D$.

A. Processo de Geração do Mapa de Dificuldade

Um mapa de dificuldade é uma estrutura que armazena o nível de dificuldade exigido para um algoritmo classificar corretamente os pixels de um quadro de vídeo [10]. Esse nível de dificuldade, que é a informação base das métricas propostas neste trabalho, deve ser gerada utilizando os resultados de diversos algoritmos, preferencialmente representantes do estado-da-arte, na forma de uma máscara S. Com esses resultados, é possível identificar o nível de dificuldade de um pixel contando quantos algoritmos do grupo classificaram incorretamente esse

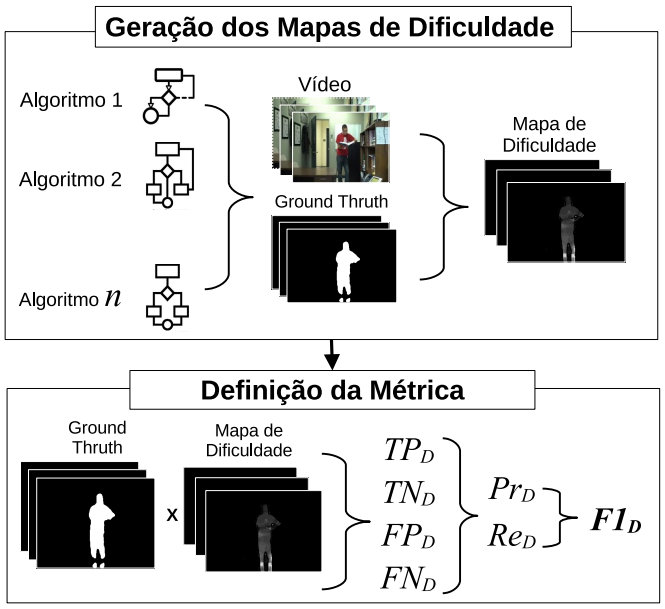


Fig. 1. Abordagem para geração da métrica $F1_D$.

pixel. Dessa forma, para cada quadro de um vídeo, gera-se um mapa de dificuldade correspondente, que é responsável por armazenar esses valores [10].

Um *ground truth* G pode possuir rótulos para regiões como "sombras" e "regiões indeterminadas", que estão localizadas normalmente ao redor de elementos de interesse [9]. Essa é a região onde não é possível visualmente identificar se os pixels pertencem ao fundo ou ao elemento de interesse. A matriz $R \in \{0,1\}^{l \times c}$ foi definida para que essas regiões sejam desconsideradas na geração dos mapas. R armazena as posições apenas dos pixels que pertencem à região de interesse do *ground truth*, que são apenas os pixels rotulados como elemento de interesse ou plano de fundo.

Para gerar um mapa de dificuldade, executa-se vários algoritmos para segmentar vídeos de um conjunto. Em seguida, as máscaras S que contêm os resultados dos algoritmos são comparadas com o *ground truth* G para identificar os pixels classificados incorretamente. O nível de dificuldade de um pixel é dado pelo número de algoritmos que classificaram incorretamente o pixel. Para cada quadro de cada vídeo de um conjuntos é gerado um mapa de dificuldade, que é definido como $D \in [0,1]^{l \times c}$ e armazena o nível de dificuldade para classificar cada pixel.

O número de algoritmos utilizado determina a quantidade de níveis de dificuldade contidos no mapa. Para n algoritmos, $n+1$ níveis de dificuldade são representados no mapa utilizando diferentes tons de cinza. O Algoritmo 1 mostra o pseudocódigo que gera um mapa de dificuldade [10].

B. Métrica para Avaliação de Algoritmos

Um mapa de dificuldade pode ser utilizado como uma ferramenta que auxilia a obtenção de uma nova medida de desempenho de um algoritmo de detecção de mudança. Avaliar um algoritmo por meio de um mapa consiste em comparar os quadros do *ground truth* G com os resultados dos algoritmos apresentados na forma de máscaras S ponderados pelo mapa de dificuldade.

O *ground truth* permite que os falsos positivos FP sejam diferenciados dos falsos negativos FN e que os verdadeiros positivos TP sejam diferenciados dos verdadeiros negativos TN (essa informação não pode ser obtida dos mapas). Uma vez identificado o tipo do erro, os níveis de dificuldade dos pixels que estão armazenados no mapa possibilitam calcular os valores TP_D, TN_D, FP_D e FN_D por meio das equações

$$TP_D = \|G \odot S \odot D \odot R\|, \tag{9}$$

$$TN_D = \|(1-G) \odot (1-S) \odot D \odot R\|, \tag{10}$$

$$FP_D = \|(1-G) \odot S \odot D \odot R\| \tag{11}$$

e

$$FN_D = \|G \odot (1-S) \odot D \odot R\| \tag{12}$$

TP_D, TN_D, FP_D e FN_D são utilizados para calcular as métricas propostas neste trabalho baseando-se nas equações 1-8. São obtidos, então, a Taxa de Falsos Positivos (FPR_D), a Taxa de Falsos Negativos (FNR_D), Precisão (Pr_D), Revocação (Re_D), Especificidade (Sp_D), Percentual de Classificações Incorretas (PWC_D) e F-Measure ($F1_D$). Essas métricas representam o desempenho de um algoritmo de detecção de mudança em relação ao mapa de dificuldade e podem ser utilizadas para identificar algoritmos promissores, aqueles que classificam pixels os quais a maioria dos algoritmos não são capazes de classificar.

IV. Experimentos e Análise dos Resultados

Nesta seção são apresentados as principais etapas dos experimentos realizados: geração dos mapas e definição da métrica. A análise dos resultados obtidos dos experimentos são também apresentados nesta seção.

A. Geração dos Mapas

Vídeos contendo cenas de uma aplicação que utiliza algoritmos de detecção de mudança, os *ground truths* e os resultados da segmentação desses vídeos produzidos por algoritmos do estado-da-arte (máscaras S) são os elementos necessários para gerar um mapa de dificuldade (Algoritmo 1) [10]. Esses recursos foram obtidos do *site* do CDNet 2014 [9].

As máscaras S, vídeos e *ground truths* G disponíveis no *site* do conjuntos de vídeos CDNet 2014 [9] foram utilizados para gerar os mapas de dificuldades. No *site*, também é disponibilizado um *ranking* que mostra o desempenho de vários algoritmos de detecção de mudança. Os vídeos são agrupados em categorias, que contém vídeos que apresentam um tipo de desafio ou ocorrência que dificultam a ação do algoritmo de detecção de mudança (tremulação da câmera, vídeos com baixa taxa de quadros, plano de fundo dinâmico, sombras, etc). As categorias e seus respectivos números de vídeos são: *Baseline* (4), *Dynamic Background* (6), *Camera Jitter* (4), *Shadows* (6), *Interm. Object Motion* (6), *Thermal* (5), *Bad Weather* (4), *Low Framerate* (4), *Night Videos* (6), *PTZ* (4) e *Turbulence* (4).

Algorithm 1 Pseudocódigo para gerar Mapas de Dificuldade [10]

Entradas: S (resultado da segmentação), R (região de interesse), G (*ground truth*), V (número de vídeos), Q (número de quadros), $(l \times c)$ (número de pixels) e n (número de algoritmos)
Saídas: D (estrutura que armazena mapas de dificuldade)

$D_{vid.frame.pixel.level} \leftarrow 0$
for i $\leftarrow$ 1 **to** V **do**
  for j $\leftarrow$ 1 **to** Q **do**
    for k $\leftarrow$ 1 **to** $(l \times c)$ **do**
      for m $\leftarrow$ 1 **to** n **do**
        if $S_{vid(i).frame(j).pixel(k).alg(m).label} \neq G_{vid(i).frame(j).pixel(k).alg(m).label}$ **and** $R_{vid(i).frame(j).pixel(k).label} = 1$ **then**
          $D_{vid(i).frame(j).pixel(k).level} \leftarrow D_{vid(i).frame(j).pixel(k).level}$+1

O *ranking* do CDNet 2014 apresenta os algoritmos que obtiveram os melhores desempenhos entre os que utilizaram seus vídeos para avaliação, de acordo com as métricas FPR, FNR, Pr, Re, Sp, PWC e $F1$. Um levantamento realizado em Sanches et. al [8] mostrou que a $F1$ foi utilizada para representar o desempenho em 70% dos trabalhos que apresentavam um novo algoritmo de detecção de mudança, o que a torna a métrica mais adotada pelos pesquisadores.

Um total de 40 algoritmos disponíveis no *site* do CDnet 2014 foram utilizados em nosso experimento. Na data do acesso, o *site* disponibilizava os resultados de 43 algoritmos, mas 3 deles estavam com os *links* quebrados. Dividiu-se os 40 algoritmos em três grupos: desempenho baixo ($F1 < 0,75$), desempenho médio ($0,75 \geq F1 < 0,9$) e desempenho alto ($F1 \geq 0,9$). O conjunto de máscaras escolhidas para geração do mapa (30 máscaras) e o conjunto utilizado na etapa de validação (10 máscaras) foram selecionadas aleatoriamente dentro de cada grupo de desempenho. Dessa forma, garantiu-se que resultados de algoritmos com desempenhos baixos, médios e altos estivessem presentes nos conjuntos de forma equilibrada.

As máscaras S selecionadas para geração dos mapas pertencem aos algoritmos WeSamBE, SuBSENSE, SharedModel, FTSG, CwisarDRP, C-EFIC, Multimode BS, EFIC, CwisarDH, Multimode BS Version 0, Spectral-360, SBBS, BMOG, AAPSA, IUTIS-1, GraphCutDiff, Mahalanobis distance, SC_SOBS, RMoG, KDE (ElGammal), GMM (Stauffer & Grimson), CP3-online, GMM (Zivkovic), Euclidean distance, BSPVGAN, FgSegNet-S (FPM), IUTIS-3, IUTIS-5, PAWCS e WisenetMD. Os autores dos algoritmos citados são exibidos no *site* do CDNet 2014 [9].

Conforme discutido na seção III-A, um mapa gerado utilizando 30 algoritmos ($n = 30$) contém 31 níveis de dificuldade ($n + 1$). Para facilitar a visualização na forma de imagens, optou-se por armazenar cada nível de dificuldade nos mapas em intervalos de 8 valores, sendo que cada nível equivale a um tom de cinza (nível 0 = tom de cinza 0, nível 1 = tom de cinza 8, nível 2 = tom de cinza 16 ... nível 30 = tom de cinza 240). Uma imagem cujos valores dos pixels variam entre 0 (pixels sem dificuldade) e 240 (pixels com dificuldade máxima) representa um quadro de um mapa. Para facilitar os cálculos, os valores que representam os níveis de dificuldade foram normalizados para ajustarem-se no intervalo entre 0 e 1 nos experimentos. Na Figura 2 são mostrados exemplos de mapas de dificuldade que correspondem a um quadro dos vídeos (*wetSnow* e *canoe*), pertencentes ao CDNet 2014. Cada pixels nesses mapas pertencem a um nível de dificuldade que varia entre 0 e 30.

Fig. 2. Exemplos de mapas de dificuldades. (a) Quadro 500 do vídeo wetSnow (primeira coluna) e quadro 1024 do mapa de dificuldade correspondente (segunda coluna). (b) Quadro 1024 do vídeo canoe (primeira coluna) e quadro 1024 do mapa de dificuldade correspondente (segunda coluna).

B. Cálculo da Métrica $F1_D$

Como formas de explorar as informações dos mapas de dificuldade na avaliação de algoritmos de detecção de mudanças, a abordagem proposta neste trabalho utiliza o mapa como uma ferramenta que possibilita considerar apenas pixels "difíceis" de serem classificados no processo de avaliação. Essa estratégia tem como objetivo identificar os algoritmos que são capazes de classificar esses pixels, mesmo que seu desempenho geral não seja superior aos desempenhos de algoritmos do estado-da-arte. Os resultados desse tipo de avaliação permitem identificar abordagens que podem ser consideradas "promissoras", visto que os desafios para melhorá-las foram superados por outros algoritmos e, por esse motivo, são teoricamente menores que os desafios para melhorar algoritmos cujos erros de classificação são similares ao do estado-da-arte.

Nos experimentos realizados para avaliar a nova métrica proposta, foram utilizados 10 algoritmos de detecção de mudança, diferentes daqueles adotados para gerar os mapas de dificuldade. O objetivo é identificar se alguns desses algoritmos são promissores. Novamente, as máscaras S que contêm os resultados de segmentação dos algoritmos foram obtidos do *site* do CDNet 2014 [11]. Na Tabela I são mostrados os 10 algoritmos utilizados nesta etapa.

Utilizando as Equações 9-12, os valores TP_D, TN_D, FP_D e FN_D foram obtidos para cada um dos 10 algoritmos

TABELA I
ALGORITMOS CUJOS DESEMPENHOS FORAM AVALIADOS UTILIZANDO OS MAPAS DE DIFICULDADE

Algoritmos	Referências
M4CDV2.0	[12]
SWCD	[13]
IUTIS-2	[14]
AMBER	[15]
Cascade CNN	[16]
DeepBS	[17]
FgSegNet-v2	[18]
Multiscale BG Model	[19]
BSUV-Sem	[20]
BSUV-Net	[21]

utilizando os mapas de dificuldade D (gerados a partir dos 30 algoritmos), os *ground truths* G e as matrizes R que contêm a região de interesse de cada vídeo. Em seguida, foram calculadas as métricas Pr_D, Re_D e $F1_D$, por meio das Equações 3, 4 e 8.

No levantamento realizado em Sanches et. al [8], os autores executaram uma revisão sistemática para identificar as principais etapas e componentes utilizados na avaliação do desempenho de um novo algoritmo de detecção de mudança. Como resultado da análise do levantamento, foram apresentadas uma série de recomendações na tentativa de padronizar o processo de avaliação de desempenho de algoritmos desse tipo. A pesquisa mostra ainda que a métrica $F1$ é utilizada para representar o desempenho na maioria dos artigos científicos que apresentam novos algoritmos de detecção de mudança. Por esse motivo essa é a métrica considerada nesta análise. Nesta etapa foram calculados os desempenhos dos algoritmos depois de executados para segmentar cada vídeo do CDNet 2014.

C. Resultados e Análise

Inicialmente, para demonstrar que existe diferença entre as métricas $F1$ e $F1_D$, foi aplicado teste de Wilcoxon. Os resultados mostraram uma diferença estatisticamente significativa entre as métricas, com $w = 164368$ e $p-value = 1,593e-06$. O teste de Wilcoxon mostrou-se adequado porque, além de existirem duas amostras, os resíduos não seguem uma distribuição normal.

A constatação de que as abordagens existentes não convergem em todas as situações, em termos de dificuldades encontradas pelos algoritmos em relação à classificação dos pixels dos quadros, também pode ser observada pela análise da correlação entre as métricas $F1$ e $F1_D$ mostrada na Figura 3.

Existem categorias de vídeo em que a correlação entre as métricas não é forte. Isso significa que os pixels (ou conjunto de pixels) que apresentam maior dificuldade para serem classificados não são os mesmos para todos os algoritmos testados. As Figuras 3a, 3b e 3c mostram exemplos de correlação (coeficiente de Kendall) apenas moderada entre as métricas $F1$ e $F1_D$ considerando os vídeos das categorias *Baseline*, *Low Framerate* e *Night Videos*. Por outro lado, uma correlação bastante forte pode ser observada quando se avaliam os algoritmos nos vídeos das categorias *Shadow* (Figura 3d), *PTZ* (Figura 3e) e *Bad Weather* (Figura 3f).

A Tabela II mostra exemplos de situações em que algoritmos apresentam desempenho superior quando avaliados com a métrica $F1_D$, se comparados com o seus desempenhos quando avaliados com a métrica $F1$. As colunas representam respectivamente o nome do algoritmo, a categoria em que o vídeo pertence no CDNet 2014; o nome do vídeo; as métricas $F1$ e $F1_D$ calculadas da execução do algoritmo sobre o vídeo; e a diferença entre os valores das métricas $F1_D$ e $F1$. Foram listados os 20 resultados em que a diferença entre $F1_D$ e $F1$ são maiores.

Os resultados indicam que, em alguns casos, o desempenho do algoritmo considerando a métrica $F1_D$ é melhor do que quando se considera a métrica $F1$. Isso significa que, em algumas cenas, tais algoritmos classificam pixels que a maioria dos algoritmos do estado-da-arte não são capazes de classificar. O algoritmo BSUV-Net, destacado na Tabela II, apresenta essa característica uma vez que aparece 7 vezes lista, seguido do algoritmo BSUV-Sem, que aparece 4 vezes. Em termos práticos, os erros de classificação cometidos por esses algoritmos quando segmentam o vídeo *port_0_17fps*, por exemplo, não são os mesmos cometidos por outros algoritmos do estado-da-arte. Teoricamente, as soluções para melhorar seus desempenhos nas cenas em que sua eficiência é baixa podem estar presentes em outros algoritmos. Melhorar a lógica de algoritmos promissores inspirando-se em soluções existentes ou combinando-os com algoritmos do estado-da-arte pode ser uma estratégia interessante.

V. CONCLUSÕES

A avaliação do desempenho de um algoritmo de detecção de mudança deve mostrar a superioridade desse algoritmo em relação ao estado-da-arte. Avaliar um algoritmo consiste na sua execução para segmentar os vídeos de um conjuntos e na comparação dos resultados com um *ground truth*.

Neste artigo foi apresentada uma métrica, baseada em mapas de dificuldade, para obter o desempenho de algoritmos de detecção de mudança em relação a esses mapas. Algoritmos que apresentam bom desempenho quando avaliados pela nova métrica podem ser considerados promissores. Os resultados mostraram que existem na literatura algoritmos promissores, pois alguns deles possuem a característica de classificar pixels que a maioria dos algoritmos do estado-da-arte não são capazes de classificar. A identificação desses algoritmos pode contribuir com o desenvolvimento de outros mais eficientes uma vez que melhorar o desempenho de um algoritmo promissor significa enfrentar desafios já superados por abordagens existentes. Pesquisadores que pretendem combinar algoritmos ou desenvolver um novo algoritmo tendo como base um existente podem ter seu trabalho facilitado se o algoritmo escolhido for promissor.

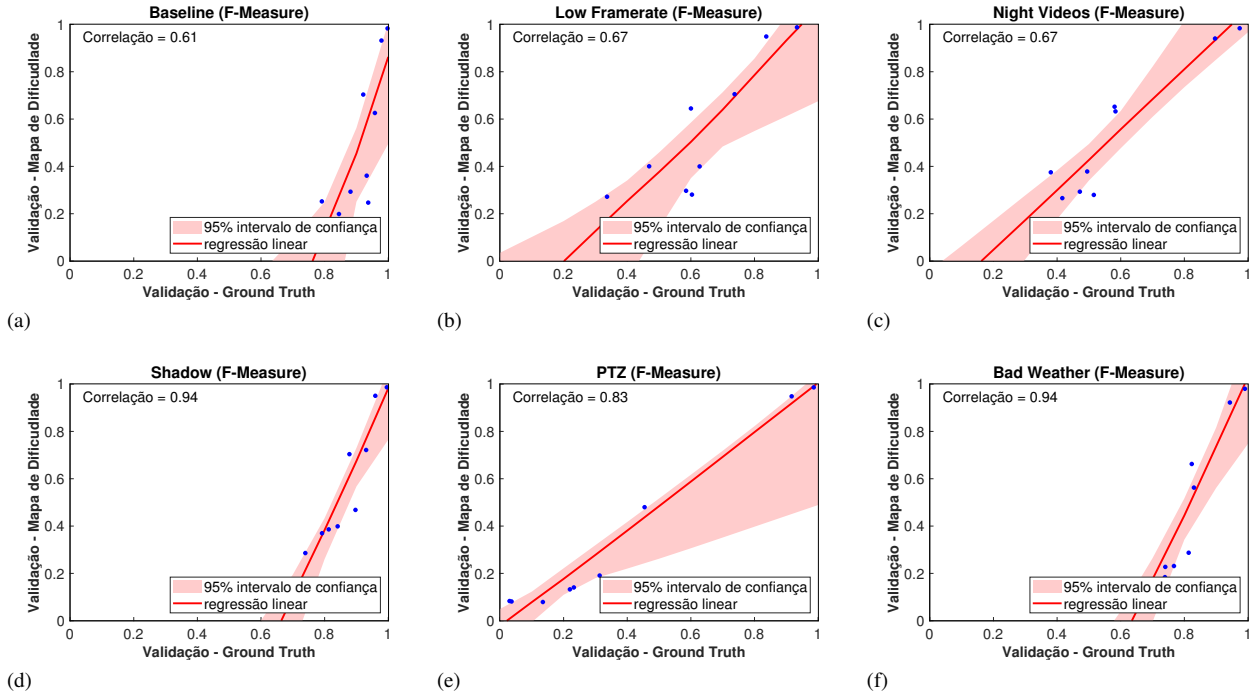


Fig. 3. Exemplos de correlação entre os resultados obtidos na avaliação de desempenho comparando o uso do *ground truth* tradicional (métrica $F1$) e o uso dos mapas de dificuldade (métrica $F1_D$). Em (a), (b) e (c) existe moderada correlação entre as métricas. Em (d), (e) e (f) a correlação entre as métricas é forte.

TABELA II

RESULTADOS DA EXECUÇÃO DOS ALGORITMOS SOBRE OS VÍDEOS DO CDNET 2014 ORDENADOS PELA DIFERENÇA ENTRE AS MÉTRICAS $F1_D$ E $F1$.

Algoritmo	**Categoria**	**Vídeo**	$F1$	$F1_D$	**Diferença**
BSUV-Sem	Low Framerate	port_0_17fps	0,0086	0,8956	0,8870
BSUV-Net	Low Framerate	port_0_17fps	0,0090	0,7874	0,7784
BSUV-Sem	Intermittent Object Motion	streetLight	0,2228	0,8838	0,6610
DeepBS	PTZ	intermittentPan	0,2063	0,7651	0,5588
BSUV-Net	Intermittent Object Motion	streetLight	0,2321	0,7890	0,5569
BSUV-Net	Dynamic Background	fountain01	0,3391	0,8834	0,5443
BSUV-Sem	PTZ	intermittentPan	0,4332	0,9463	0,5132
DeepBS	Night Videos	busyBoulvard	0,3209	0,8185	0,4976
BSUV-Net	Turbulence	turbulence0	0,4426	0,9381	0,4954
BSUV-Net	PTZ	continuousPan	0,2350	0,7255	0,4906
SWCD	PTZ	continuousPan	0,2465	0,6908	0,4442
SWCD	Intermittent Object Motion	winterDriveway	0,4839	0,9259	0,4420
DeepBS	Intermittent Object Motion	tramstop	0,4754	0,8996	0,4241
Cascade CNN	Intermittent Object Motion	winterDriveway	0,5312	0,9203	0,3891
Multiscale BG Model	Intermittent Object Motion	winterDriveway	0,2143	0,5583	0,3440
Multiscale BG Model	Dynamic Background	boats	0,4792	0,8140	0,3348
BSUV-Net	Night Videos	fluidHighway	0,6578	0,9756	0,3178
IUTIS-2	Low Framerate	tunnelExit_0_35fps	0,4981	0,7939	0,2958
BSUV-Net	Night Videos	winterStreet	0,6124	0,9018	0,2894
BSUV-Sem	Night Videos	bridgeEntry	0,6089	0,8906	0,2818

REFERÊNCIAS

[1] S. R. R. Sanches, C. Oliveira, A. C. Sementille, and V. Freire, "Challenging situations for background subtraction algorithms," *Applied Intelligence*, vol. 49, pp. 1771–1784, May 2019.

[2] S. R. R. Sanches, R. Nakamura, V. Freire Silva, and R. Tori, "Bilayer segmentation of live video in uncontrolled environments for background substitution: An overview and main challenges," *IEEE Latin America Transactions*, vol. 10, no. 5, pp. 2138–2149, 2012.

[3] J. L. Aching Samatelo and E. Ottoni Teatini Salles, "A new change detection algorithm for visual surveillance system," *IEEE Latin America Transactions*, vol. 10, no. 1, pp. 1221–1226, 2012.

[4] V. B. d. O. Barth, R. de Oliveira, M. A. de Oliveira, and V. E. do Nascimento, "Vehicle speed monitoring using convolutional neural networks," *IEEE Latin America Transactions*, vol. 17, p. 1000–1008, Nov. 2019.

[5] M. Valdeos, A. S. Vadillo Velazco, M. G. Pérez Paredes, and R. M. Arias Velásquez, "Methodology for an automatic license plate recog-

nition system using convolutional neural networks for a peruvian case study," *IEEE Latin America Transactions*, vol. 20, p. 1032–1039, Mar. 2022.
[6] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, June 2012.
[7] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.
[8] S. R. R. Sanches, A. C. Sementille, I. A. Aguilar, and V. Freire, "Recommendations for evaluating the performance of background subtraction algorithms for surveillance systems," *Multimedia Tools and Applications*, vol. 80, no. 3, p. 4421–4454, 2021.
[9] Université de Sherbrooke, "ChangeDetection.NET – a video database for testing change detection algorithms," 2022. http://www.changedetection.net. Accessed 10 Out 2022.
[10] C. M. Silva, K. A. I. Rosa, P. H. Bugatti, P. T. M. Saito, C. G. Corrêa, R. S. Yokoyama, and S. R. R. Sanches, "Method for selecting representative videos for change detection datasets," *Multimed Tools and Applications*, vol. 81, no. 3, pp. 3773–3791, 2022.
[11] UNIVERSITÉ DE SHERBROOKE, "Results for CD.net 2014," 2019. http://jacarini.dinf.usherbrooke.ca/results2014/. Accessed 10 Out 2022.
[12] K. Wang, C. Gou, and F.-Y. Wang, "M4cd: A robust change detection method for intelligent visual surveillance," 2018. https://arxiv.org/abs/1802.04979. Cornell University. Accessed 12 Nov 2019.
[13] S. Isik, K. Özkan, S. Günal, and O. N. Gerek, "Swcd: a sliding window and self-regulated learning-based background updating method for change detection in videos," *Journal of Electronic Imaging*, vol. 27, no. 2, pp. 1–11, 2018.
[14] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?," in *Image Analysis and Processing - ICIAP 2017* (S. Battiato, G. Gallo, R. Schettini, and F. Stanco, eds.), (Cham), pp. 96–107, Springer International Publishing, 2017.
[15] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 401–404, June 2014.
[16] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66 – 75, 2017. Scene Background Modeling and Initialization.
[17] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, pp. 635 – 649, 2018.
[18] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, Aug 2019.
[19] X. Lu, "A multiscale spatio-temporal background model for motion detection," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 3268–3271, Oct 2014.
[20] M. Braham, S. Piérard, and M. V. Droogenbroeck, "Semantic background subtraction," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 4552–4556, 2017.
[21] M. O. Tezcan, P. Ishwar, and J. Konrad, "Bsuv-net: A fully-convolutional neural network for background subtraction of unseen videos," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2763–2772, 2020.

Silvio Ricardo Rodrigues Sanches received the B.Sc. and M.Sc. degree all from the Centro Universitário Eurípides de Marília, MarÍia, Brazil, in 2003 and 2007 respectively. He received the Ph.D. degree from the University of São Paulo (USP) at Escola Politécnica, São Paulo, Brazil, in 2013. Currently he is a Professor at Federal University of Technology – Paraná (UTFPR), Cornélio Procópio, PR, Brazil). His research interests include Computer Vision with emphasis on video segmentation.

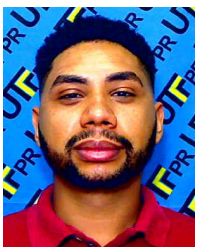
Cléber Gimenez Corrêa graduated in Data Processing from São Paulo Technology College (FATEC), Ourinhos, Brazil (2002). He has a M.Sc. degree in Computer Science from University Center Eurípides Soares da Rocha (UNIVEM), Marília, Brazil (2008), and a Ph.D. degree from the Escola Politécnica of the University of São Paulo, São Paulo, Brazil (2015). His research interests are human-computer interaction, Virtual Reality and software testing.

Beatriz Regina Brum graduated in Mathematics from the Federal Technological University of Paraná (2011). She has a M.Sc. degree in Biostatistics from the State University of Maringá (2018). Her research interests are teaching mathematics and analysis of longitudinal data.

Pedro Henrique Bugatti Associate Professor at the Department of Computing, Federal University of Technology - Parana (UTFPR), Brazil. Ph.D. in Computer Science (2012) in Computer Science from the University of São Paulo (ICMC-USP). Master's Degree in Computer Science from the University of São Paulo (ICMC-USP) in 2018. Bachelor's Degree in Computer Science from the Euripides Soares de Rocha (UNIVEM) in 2006. Research interests include deep learning, image analysis, machine learning, content-based image retrieval.

Priscila Tiemi Maeda Saito Associate Professor at the Department of Computing, Federal University of Sao Carlos (DC-UFSCar), Sao Carlos, Brazil. Ph.D. in Computer Science (2014) at the University of Campinas (IC-UNICAMP). Master's Degree in Computer Science from the University of São Paulo (ICMC-USP) in 2010. Bachelor's Degree in Computer Science from the Euripides Soares de Rocha (UNIVEM) in 2008. Research interests include image analysis, machine learning, and content-based image retrieval.

Claudinei Moreira da Silva graduate at Tecnologia Em Processamento de Dados from Instituto Municpal de Ensino Superior de Assis (1999). He has an M.Sc. degree in Informatic from Programa de Pós Graduação em Informática (PPGI - UTFPR). He has experience in Computer Science, focusing on Computer Vision.



Elton Custódio Junior Degree in Mathematics from the State University of Northern Paraná (UENP) in 2018. Master's student in Programa de Pós Graduação em Informática (PPGI - UTFPR). His research interests include Computer Vision.