UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ELTON CUSTÓDIO JÚNIOR

GERAÇÃO AUTOMÁTICA DE MAPAS DE DIFICULDADE PARA DATASETS

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2022

ELTON CUSTÓDIO JÚNIOR

GERAÇÃO AUTOMÁTICA DE MAPAS DE DIFICULDADE PARA DATASETS

Automatic Generation of Difficulty Maps for Datasets

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Informática”.

Orientador: Prof^{o}. Dr^{o}. Silvio Ricardo Rodrigues Sanches

Co-orientador: Prof^{o}. Dr^{o}. Pedro Henrique Bugatti

CORNÉLIO PROCÓPIO
2022

ELTON CUSTODIO JUNIOR

GERAÇÃO AUTOMÁTICA DE MAPAS DE DIFICULDADE PARA DATASETS

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Informática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Computação Aplicada.

Data de aprovação: 01 de Setembro de 2022

Dr. Silvio Ricardo Rodrigues Sanches, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Cleber Gimenez Correa, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Larissa Pavarini Da Luz, Doutorado - Faculdade de Tecnologia de Garça (Fatecga)

Dr. Pedro Henrique Bugatti, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Valdinei Freire Da Silva, Doutorado - Usp-Universidade de São Paulo

AGRADECIMENTOS

A Deus, pela saúde e determinação para não desanimar durante a realização deste trabalho. Aos professores do Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná de Cornélio Procópio/PR, por todo apoio e ensinamentos, que contribuíram para a elaboração deste trabalho.

Aos amigos, familiares e a minha noiva Analu Nakamura por todo suporte fornecido, em especial ao meu irmão Everton e a minha mãe Isabel pelo auxílio nesta caminhada. Aos professores Silvio Sanches e Pedro Bugatti, pela sábia orientação, competência e dedicação a este trabalho.

“O conhecimento serve para encantar as pessoas, não para humilhá-las”. (Mario Sergio Cortella)

RESUMO

CUSTÓDIO JÚNIOR, Elton. GERAÇÃO AUTOMÁTICA DE MAPAS DE DIFICULDADE PARA DATASETS. 61 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Mapas de dificuldade são estruturas que armazenam os níveis de dificuldade estimados para que um algoritmo de detecção de mudança classifique corretamente cada pixel dos quadros de um vídeo. Tais mapas são utilizados como principal informação na obtenção de uma medida objetiva, considerada o "nível de dificuldade" do vídeo. A geração de um mapa de dificuldade de um vídeo requer a utilização do *ground truth*. Criar um *ground truth* é um processo considerado trabalhoso, pois consiste na atribuição, muitas vezes de forma manual, de rótulos para todos os pixels de todos os quadros do vídeo. Nesta pesquisa, apresenta-se um método, que consiste em uma rede neural treinada, capaz de, sem o auxílio do *ground truth*, gerar novos mapas de dificuldade. Deste modo, espera-se que os mapas de dificuldades auxiliem na criação de *datasets* e na avaliação de algoritmos de detecção de mudanças.

Palavras-chave: detecção de mudança, mapas de dificuldade, *dataset*, segmentação de vídeos.

ABSTRACT

CUSTÓDIO JÚNIOR, Elton. AUTOMATIC GENERATION OF DIFFICULTY MAPS FOR DATASETS. 61 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Difficulty maps are structures that store estimated difficulty levels for a change detection algorithm to classify each pixel in a video's frames correctly. Such maps are essential to obtain an objective measure, considering the "difficulty level" of the video. Generating a video difficulty map requires using *ground truths*. Creating a *ground truth* is considered a laborious process as it involves assigning, often manually, labels to all pixels of all video frames. In this research, a method is presented, which consists of a trained neural network capable of estimating the difficulty level of a video without using a *ground truth*. We expected that researchers use the results of this work to generate new videos *datasets* to evaluate change detection algorithms.

Keywords: change detection, difficulty maps, *dataset*, video segmentation.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Os algoritmos de detecção de mudança têm como objetivo identificar áreas que se modificam dentro do campo de visão de uma câmera (UNIVERSITÉ DE SHERBROOKE, 2021). Detectar mudanças, principalmente movimentos, pode ser considerada uma etapa fundamental em aplicações como contagem de pessoas, análise de tráfego por vídeo, rastreamento de objetos e monitoramento de ambientes (GOYETTE et al., 2012). Esses algoritmos buscam identificar as regiões cujos *pixels* de interesse sofrem alterações ou se movimentam em relação ao modelo de fundo.

O desempenho dos algoritmos de detecção de mudanças está normalmente associado aos desafios dos ambientes que são apresentados no conteúdo da cena. Variações na iluminação, sombras e movimentos no fundo são exemplos de situações que dificultam a ação desses algoritmos (SANCHES et al., 2019). Métodos sofisticados, baseados nas mais diferentes abordagens (SOBRAL; VACAVANT, 2014), são desenvolvidos com o objetivo de superar esses desafios. Tais métodos são capazes de classificar corretamente a maioria dos *pixels* dos quadros capturados pela câmera e atribuir a cada um deles um rótulo indicando que pertencem a um elemento de interesse (região que se move) ou ao fundo da cena (WANG et al., 2014).

A avaliação do desempenho de um novo algoritmo de detecção de mudança consiste em 4 etapas principais: (i) executar o algoritmo para segmentar vídeos de um *dataset*, (ii) comparar os resultados com um *ground truth*, (iii) calcular um conjunto de métricas que representam o desempenho do algoritmo e (iv) comparar esse desempenho com os desempenhos dos algoritmos mais eficientes, que representam o estado-da-arte. A medida que os novos algoritmos gradativamente superam os desafios apresentados nas cenas dos vídeos dos *datasets*, torna-se necessário que

novos vídeos, que contenham novos desafios, sejam capturados e incluídos no conjunto.

Apesar dos desafios do ambiente real que são apresentados no conteúdo da cena fornecerem indícios de que os quadros de um determinado vídeo são “difíceis” de serem classificados, uma medida objetiva é necessária para estimar com maior precisão o nível dessa dificuldade. Para essa finalidade podem ser gerados mapas de dificuldades. Um mapa de dificuldade é uma estrutura que armazena os níveis de dificuldade exigidos para que um algoritmo de detecção de mudança classifique corretamente os *pixels* dos quadros de um vídeo (SILVA, 2021). É possível utilizar esses mapas para obter uma medida objetiva da dificuldade que pode ser associada a um determinado vídeo.

O problema em utilizar os mapas para selecionar novos vídeos que apresentem níveis de dificuldade superiores aos do conjunto original de um *dataset* é que a abordagem para geração dessas estruturas requer a utilização de *ground truths* desses novos vídeos. A geração de *ground truths* é um processo trabalhoso, uma vez que envolve a atribuição de rótulos para todos os *pixels* de todos os quadros (SANCHES et al., 2021b). O desenvolvimento de um método que seja capaz de, sem o auxílio do *ground truth*, estimar o nível de dificuldade de um vídeo pode ser importante para que os pesquisadores possam atualizar constantemente os *datasets*, incluindo vídeos que representem novos desafios aos algoritmos do estado-da-arte. Para tal, treinaremos uma rede neural artificial alimentada com os mapas de dificuldades descritos por (SILVA, 2021).

1.1 OBJETIVO

O objetivo deste trabalho consiste em desenvolver um método para gerar os mapas de dificuldade e estimar o nível de dificuldade de um vídeo, ainda que não estejam disponíveis os *ground truths* correspondentes aos quadros desses vídeos.

1.2 ORGANIZAÇÃO DO TEXTO

Para que favoreça seu entendimento, este trabalho está organizado da forma que segue: o Capítulo 1 apresenta os conceitos e definições, destacando a importância da avaliação do desempenho de algoritmos de detecção de mudanças. No Capítulo 2 abordam-se a revisão bibliográfica dos conceitos que serão discutidos no decorrer do trabalho. No Capítulo 3 descreve-se a abordagem utilizada, desde a criação dos mapas de dificuldade até o treinamento da rede neural artificial. O Capítulo 4 apresenta os resultados obtidos e finalmente, reservou-se o Capítulo 5 para as discussões dos resultados e conclusões.

2 REVISÃO BIBLIOGRÁFICA

Neste Capítulo são apresentados em detalhes os conceitos relacionados aos mapas de dificuldade e suas principais aplicações. Apresenta-se também um referencial teórico envolvendo as principais técnicas que pretende-se utilizar no desenvolvimento da solução proposta nesta pesquisa.

2.1 MAPAS DE DIFICULDADE

Os mapas de dificuldade são estruturas utilizadas para duas finalidades: (i) a avaliação de algoritmos de detecção de mudanças e (ii) a estimação do nível de dificuldade que um algoritmo encontrará na classificação dos *pixels* de um determinado vídeo que pertence a um *dataset* (Seção 2.2.1). Para cada quadro do vídeo, gera-se um mapa de dificuldade, que é responsável por armazenar um conjunto de valores que correspondem ao nível de dificuldade estimado de cada pixel do quadro. O conceito de mapa de dificuldade foi apresentado nos trabalhos de Silva (2021) e Silva et al. (2021).

No processo de geração dos mapas utiliza-se recursos como um conjunto de vídeos, o *ground truth* (Seção 2.2.1) e resultados da segmentação dos vídeos por algoritmos de detecção de mudanças. Tais resultados são máscaras em que os *pixels* pertencentes aos elementos de interesse e os pertencentes ao fundo da cena possuem rótulos distintos. A Figura 1 mostra um quadro do vídeo “Pedestrians”, obtido do *dataset* CDNet 2012 (GOYETTE et al., 2012), o *ground truth* correspondente ao quadro e os resultados da segmentação de três algoritmos do estado da arte.

Considerando valores normalizados, um algoritmo de detecção de mudança gera uma máscara $S \in \{0,1\}^{l \times c}$ como resultado, onde 1 é o rótulo dos *pixels* da região

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

Figura 1: (a) quadro de um vídeo de um *dataset*, (b) *ground truth*, (c),(d) e (e) resultados da segmentação na forma de máscara, obtidos de algoritmos do estado da arte (SANCHES et al., 2021b)

de interesse (áreas em que houve mudança), 0 é rótulo dos *pixels* do plano de fundo e $l \times c$ é a resolução do quadro do video (linha $\times$ coluna). Os rótulos do *ground truth* $G \in [0,1]^{l \times c}$ são o valor 0 (*pixels* pertencentes ao plano de fundo) e o valor 1 (*pixels* pertencentes ao elemento de interesse).

O *ground truth* também pode possuir rótulos para regiões como “sombras” e “regiões indeterminadas”, que estão localizadas normalmente ao redor de elementos de interesse (UNIVERSITÉ DE SHERBROOKE, 2021). Essa é a região onde não é possível visualmente identificar se os *pixels* pertencem ao fundo ou ao elemento de interesse. Uma matriz $R \in \{0,1\}^{l \times c}$ deve ser definida para indicar os *pixels* que pertencem à região de interesse do *ground truth*, que são apenas os *pixels* rotulados como elemento de interesse ou plano de fundo.

Segundo a abordagem proposta por Silva et al. (2021), deve-se executar vários algoritmos para segmentar os vídeos do *dataset*. 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 *dataset* 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 utilizando diferentes tons da cinza. A Figura 2 apresenta um exemplo de um quadro que pertence a um *ground truth* G, um quadro de uma matriz R (que indica os *pixels* de interesse) e 4 quadros com os resultados de 4 diferentes algoritmos ($n = 4$) de detecção de mudança (máscaras S).

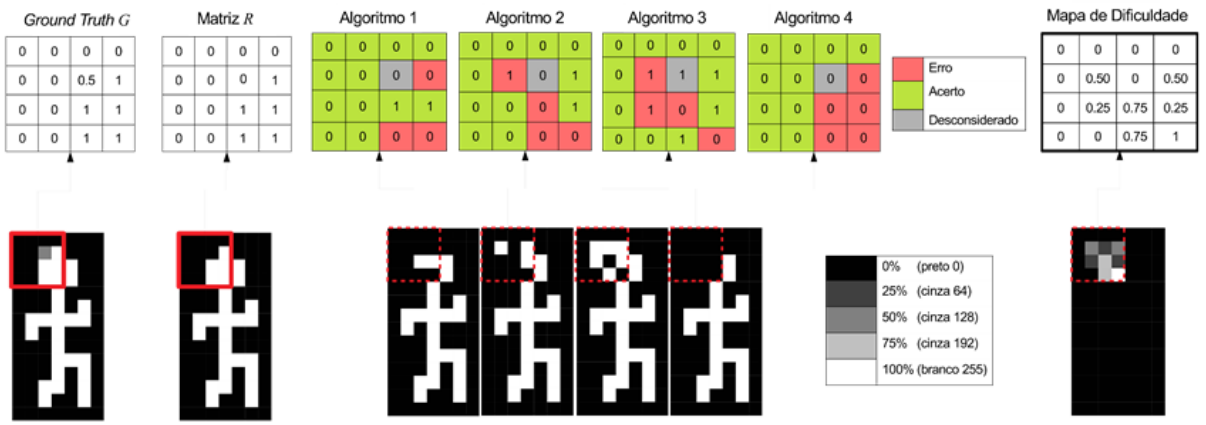

Figura 2: Um mapa de dificuldade com $n = 4$. Nesse exemplo, o *ground truth* tem *pixels* rotulados como plano de fundo (rótulo = 0), primeiro plano (rótulo = 1) e região indeterminada (rótulo = 0.5) (SILVA, 2021)

As regiões delimitadas pelos retângulos vermelhos dentro do *ground truth*, da matriz R, das máscaras S e do mapa de dificuldade são apresentadas numericamente pelas matrizes posicionadas acima de cada um desses elementos. Nas matrizes correspondentes às máscaras (algoritmos 1, 2, 3 e 4), os *pixels* vermelhos representam os erros dos algoritmos, os *pixels* verdes representam os acertos dos algoritmos e o pixel cinza representa uma região indeterminada, que será desconsiderada na geração do mapa.

Nas máscaras S, os algoritmos atribuíram a cor branca aos *pixels* que classificaram como pertencentes aos elementos de interesse e a cor preta aos *pixels* que classificaram como pertencentes ao plano de fundo da cena contida no quadro. No *ground truth*, onde os rótulos são atribuídos manualmente, existe ainda o rótulo cinza, que representa a região indeterminada. Apenas os *pixels* rotulados com o valor 1 na matriz R (região de interesse) são considerados.

No exemplo, o mapa de dificuldade que utiliza os resultados dos 4 algoritmos é constituído por 5 níveis de dificuldade $(n + 1)$, sendo, nível 0 (*pixel* com nenhuma dificuldade) e 1 (*pixel* com a mais alta dificuldade). Além disso, a dificuldade atribuída aos demais *pixels* é determinada pela porcentagem de algoritmos que erraram sua classificação (SILVA et al., 2021). No exemplo da Figura 2, quando apenas um algoritmo falha (25%) o *pixel* é rotulado com o nível 0,25, quando 3 algoritmos falham (75%) rotula-se o *pixel* com o nível 0,75 e assim, sucessivamente. Cada nível no exemplo é representado por uma cor diferente, 0% (preto 0), 25% (cinza

64), 50% (cinza 128), 75% (cinza 192) e 100% (branco 255). O Algoritmo 1 mostra o pseudocódigo que implementa a abordagem mostrada na Figura 2.

Algorithm 1 Pseudocódigo para gerar Mapas de Dificuldade (SILVA, 2021)

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**
          $R_{vid(i).frame(j).pixel(k).label} \leftarrow 1$ **then**
          $D_{vid(i).frame(j).pixel(k).level} \leftarrow D_{vid(i).frame(j).pixel(k).level}$+1

A Figura 3 mostra exemplos de mapas de dificuldade gerados de alguns quadros dos vídeos (*fall* e *cubicle*), pertencentes ao CDNet 2014. Silva et al. (2021) utilizaram 30 algoritmos ($n = 30$), gerando 31 níveis de dificuldade ($n + 1$) nesses mapas. As máscaras S desses algoritmos foram escolhidas aleatoriamente no *site* CDNet 2014 (UNIVERSITÉ DE SHERBROOKE, 2019). No *site* existe um *ranking* que apresenta os algoritmos que obtiveram os melhores desempenhos entre os que utilizaram seus vídeos para avaliação.

2.2 AVALIAÇÃO DE ALGORITMOS DE DETECÇÃO DE MUDANÇAS

Um conjunto grande de aplicações, entre elas as voltadas para vigilância por vídeo, necessitam detectar objetos em movimento em determinada cena (CHEUNG; KAMATH, 2005; TIAN et al., 2012; SENIOR et al., 2010; CARRANZA et al., 2003). A primeira etapa nesse processo é a separação dos elementos em movimento, denominados primeiro plano (*foreground*) ou elemento de interesse, da parte estática, chamada plano de fundo (*background*). O conceito mais utilizado para detectar objetos que se movimentam em vídeos é a subtração de fundo (ELGAMMAL, 2014).

Segundo Bouwmans (2014), esse processo consiste em três etapas: (I) inicialização do fundo utilizando n quadros para obter a imagem de fundo sem os

(a)

(b)

Figura 3: Exemplos de mapas de dificuldades. (a) quadro 2021 do vídeo *cubicle* (primeira coluna) e quadro 2021 do seu mapa de dificuldade correspondente (segunda coluna). (b) quadro 1513 do vídeo *fall* (primeira coluna) e quadro 1513 do mapa de dificuldade correspondente (segunda coluna)

objetos em movimento; (II) a detecção do objeto em movimento é realizada por meio da detecção do elemento de interesse, que consiste em classificar os *pixels* em elemento de interesse ou plano de fundo, comparando o modelo gerado do fundo e o quadro atual; (III) manutenção do fundo, que consiste em atualizar o modelo do fundo ao longo do tempo. As etapas (II) e (III) são executas repetidamente.

Diversos métodos de detecção de mudanças baseados em subtração de fundo foram propostos na literatura. A Tabela 1 apresenta algumas dessas abordagens e uma classificação proposta por Bouwmans (2014).

Uma etapa importante no processo de desenvolvimento desses algoritmos é a avaliação dos seus desempenhos. Um novo algoritmo de detecção de mudanças deve ser comparado com outros, de forma a deixar clara sua superioridade em relação aos algoritmos que representam o estado-da-arte. O método tradicional de avaliar algoritmos de detecção de mudança, que é bem aceito pela comunidade científica, se baseia na comparação dos resultados da segmentação dos quadros de um vídeo pelo

Tabela 1: Classificação das abordagens utilizadas pelos algoritmos de detecção de mudanças proposta por Bouwmans (2014)

Abordagens	**Categorias**
Modelos Tradicionais	Modelos básicos Modelos estatísticos Modelos de agrupamentos (*clusters*) Redes neurais Modelos de estimativa
Modelos Recentes	Modelos estatísticos avançados Modelos *fuzzy* Modelos de subespaço discriminativo e misto Modelos de subespaço de kernel Modelos robustos de subespaço (via supressão de *outliers*) Modelos de subespaço robustos Rastreamento de subespaço Minimização de classificação baixa Modelos esparsos Modelos de tensores robustos Detecção de *outliers* Transformar modelos de domínio

algoritmo com um *ground truth* do mesmo vídeo.

Além da tradicional, uma nova forma de avaliar algoritmos foi proposta em Silva (2021). Segundo o autor, a utilização dos mapas de dificuldade no processo de desenvolvimento possibilita identificar algoritmos considerados promissores (SILVA, 2021).

2.2.1 AVALIAÇÃO DE ALGORITMOS UTILIZANDO O *GROUND TRUTH*

A forma tradicional de avaliar o desempenho dos algoritmos de detecção de mudanças consiste em 4 etapas principais: (i) executar o novo algoritmo para segmentar vídeos de um *dataset*, extraindo os elementos de interesse, (ii) comparar os resultados da segmentação com os *ground truths* correspondentes, gerados para esses vídeos, (iii) calcular as medidas de desempenho do algoritmo, aplicando métricas que usam os resultados da comparação anterior, e (iv) comparar o desempenho do novo algoritmo com os desempenhos dos algoritmos do estado-da-arte (SANCHES et al., 2021b).

Os *datasets* são conjuntos de vídeos disponibilizados *online* cuja utilização possibilita que os desempenhos de diferentes algoritmos sejam comparados, uma vez que os resultados de suas execuções são obtidos utilizando o mesmo conjunto de vídeos. O conteúdo desses vídeos normalmente apresentam as aplicações para os quais o algoritmo de subtração de fundo foi desenvolvido (SANCHES et al., 2019).

A geração de um *dataset* é um processo trabalhoso, principalmente em função do esforço necessário para criação do *ground truth*. Para cada quadro do vídeo original deve ser gerado um quadro correspondente em que os *pixels* que pertencem ao elemento de interesse são rotulados com uma determinada cor, geralmente a cor branca (RGB 255,255,255), e os *pixels* que pertencem ao *background* são rotulados com uma cor diferente, normalmente a cor preta (RGB 0,0,0). A atribuição de rótulos normalmente é feita de forma manual (SANCHES et al., 2021b).

Existem ainda *datasets* em que são rotuladas outras regiões do quadro, por exemplo, sombras, regiões fora da área de interesse na cena e regiões em que não é possível identificar visualmente se o *pixel* pertence ao fundo ou ao elemento de interesse (ocorre com frequência nas bordas do elemento de interesse). Os *pixels* dessas regiões são rotulados com cores diferentes das atribuídas ao fundo e ao elemento de interesse (geralmente tons de cinza).

A comparação do resultado da execução de determinado algoritmo sobre os vídeos do *dataset* com seu quadro correspondente do *ground truth* possibilita quantificar os erros de classificação de *pixels* cometidos pelo algoritmo.

Segundo Sanches et al. (2021a), dada a norma de uma matriz $\|A\| = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|$, um verdadeiro positivo (*pixel* que pertence ao elemento de interesse corretamente classificado como elemento de interesse) é definido como $TP = \|G \odot S \odot R\|$, onde $\odot$ é o produto *entrywise*. Um falso negativo (*pixel* que pertence ao fundo erroneamente classificado como elemento de interesse) é definido como $FP = \|(1-G) \odot S \odot R\|$, um verdadeiro negativo TN (*pixel* que pertence ao fundo classificado corretamente como fundo) é definido como $TN = \|(1-G) \odot (1-S) \odot R\|$ e um falso negativo (*pixel* que per pertence a um elemento de interesse classificado erroneamente como fundo) é definido como $FN = \|G \odot (1-S) \odot R\|$.

Utilizando esses valores, pode-se calcular algumas métricas, como a precisão (Pr), de acordo com a equação

$$Pr = \frac{TP}{TP+FP}, \tag{1}$$

a revocação (Re), definida pela equação

$$Re = \frac{TP}{TP+FN}, \tag{2}$$

a especificidade (Sp), definida como

$$Sp = \frac{TN}{TN+FP} \tag{3}$$

e a *f-measure* ($F1$), que é uma métrica bastante utilizada para representar o desempenho de algoritmos de detecção de mudança (GOYETTE et al., 2012; UNIVERSITÉ DE SHERBROOKE, 2021). A $F1$, que é baseada nas métricas Pr e Re pode ser obtida por meio da equação

$$F1 = \frac{2 \times Pr \times Re}{Pr+Re}. \tag{4}$$

A métrica $F1$ também é conhecida como *Dice Coefficient* (DSC) (ZIJDENBOS et al., 1994) e avalia a intersecção das duas regiões como uma razão para a área total de ambas, de tal forma que, quando houver total concordância, a medida equivale a 1 e quando não existe é 0. Em outras palavras, os valores de concordância variam de 0 a 1.

A DSC pode ser considerado uma medida de similaridade sobre conjuntos, de forma que, dados dois conjuntos X e Y, a definição da métrica pode ser representada pela equação

$$DSC = \frac{2|X \cap Y|}{|X|+|Y|} \tag{5}$$

Quando aplicado a dados *booleanos*, utilizando a definição de TP, FP e FN, obtêm-se a equação

$$DSC = \frac{2 \times TP}{2 \times TP + FP + FN}. \tag{6}$$

2.2.2 AVALIAÇÃO DE ALGORITMOS UTILIZANDO MAPAS DE DIFICULDADE

Um mapa de dificuldade pode ser utilizado como uma ferramenta que auxilia a obtenção de uma nova medida de desempenho de um algoritmo. Avaliar um algoritmo por meio de um mapa consiste em comparar os quadros desse mapa com o *ground truth* G e com os resultados dos algoritmos, apresentados na forma de máscaras S. O *ground truth* é necessário para que seja identificada a região do quadro em que o erro ocorreu (elemento de interesse ou plano de fundo), pois essa informação não está contida no mapa. O mapa de dificuldade armazena a frequência de erros dos algoritmos em um determinado *pixel*, sem especificar a região que o *pixel* pertence.

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. 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

$$TP_D = \|G \odot S \odot D \odot R\|, \tag{7}$$

$$TN_D = \|(1-G) \odot (1-S) \odot D \odot R\|, \tag{8}$$

$$FP_D = \|(1-G) \odot S \odot D \odot R\| \tag{9}$$

e

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

TP_D, TN_D, FP_D e FN_D são utilizados para calcular a precisão (Pr_D), revocação (Re_D), especificidade (Sp_D) e *f-measure* ($F1_D$). Essas métricas representam o desempenho de um algoritmo de detecção de mudança em relação ao mapa de dificuldade.

O objetivo principal de avaliar algoritmos em relação aos mapas é identificar algoritmos promissores (SILVA, 2021). Esses algoritmos são aqueles que, mesmo

que cometam erros na classificação de *pixels* que são corretamente classificados pela maioria dos algoritmos, são capazes de classificar *pixels* em que os algoritmos do estado-da-arte normalmente erram (SILVA, 2021).

2.3 NÍVEIS DE DIFICULDADE DE VÍDEOS E CONJUNTOS REPRESENTATIVOS

A principal finalidade da geração de mapas de dificuldade é sua utilização para estimar níveis de dificuldade em vídeos de *datasets*. Um nível de dificuldade de um vídeo pode ser definido como o esforço que um algoritmo de detecção de mudanças presumidamente terá que realizar para classificar corretamente os *pixels* do vídeo em questão (SANCHES et al., 2021b).

O objetivo da obtenção dessa medida é a possibilidade de selecionar conjuntos de vídeos representativos de um *dataset*. Esses conjuntos são representados por um subconjunto dos vídeos originais e são capazes de avaliar um algoritmo de detecção de mudança com a mesma eficiência do *dataset* original, porém, com um menor número de vídeos. É possível estimar o nível de dificuldade de um vídeo e selecionar conjuntos representativos sem utilizar mapas de dificuldade. As principais abordagens para essas finalidades são apresentadas nas Seções seguintes.

2.3.1 ESTIMAÇÃO DO NÍVEL DE DIFICULDADE EM VÍDEOS BASEADA EM MAPAS

Além da avaliação de algoritmos, um mapa de dificuldade também pode ser utilizado para avaliar vídeos de um *dataset*, mais especificamente, para identificar os níveis de dificuldade desses vídeos. Esse nível de dificuldade está relacionado com o esforço necessário para classificar os *pixels* dos quadros desse vídeo. O conteúdo da cena define o esforço que um vídeo exige de um algoritmo.

Utilizando as informações dos mapas de dificuldades, é possível estimar o nível de dificuldade L de um vídeo para avaliar algoritmos. Esse valor pode ser utilizado para organizar as categorias dos *datasets* de acordo com essa característica. *Datasets* que contenham vídeos com diferentes valores de L podem diferenciar com

maior precisão os algoritmos mais eficientes dos menos eficientes.

Dado o número de *pixels* válidos Nvp para o j^{th} quadro de um *ground truth* G

$$Nvp_j = \sum_{i=1}^{l*c} p(i) \tag{11}$$

$$p(i) = R(i) \tag{12}$$

o nível de dificuldade L de uma sequência de quadros k pode ser obtido de acordo com a equação

$$L(k) = \sum_{j=start}^{end} \sum_{i=1}^{N_{vp}} f \times \frac{d(i,j,D_k)}{N_{vp_j}} \tag{13}$$

onde $start$ é o quadro inicial, end é o quadro final, D_k é o mapa de dificuldade da sequência de quadros k, f é uma constante para reduzir o tamanho da escala dos valores que representam o nível de dificuldade (definido empiricamente como 0,1) e $d(i,j,D_k)$ é o nível de dificuldade armazenado do pixel i do quadro j do mapa de dificuldade D_k.

O valor L pode ser calculado para um conjunto de quadros, para um vídeo completo ou para um *dataset*. Alguns *datasets* agrupam seus vídeos de acordo com algum desafio específico (por exemplo, plano de fundo dinâmico, sombras e tremulação da câmera) (UNIVERSITÉ DE SHERBROOKE, 2021). O nível de dificuldade também pode ser calculado para cada um desses grupos.

2.3.2 ESTIMAÇÃO DE NÍVEIS DE DIFICULDADE BASEADA NA MEDIANA

Primeiro, os N algoritmos de detecção do estado da arte segmentam todos os vídeos V para gerar $TP_{(n,v)}$, $FP_{(n,v)}$, $TN_{(n,v)}$ e $FN_{(n,v)}$, onde n é o id do algoritmo ($n = (1,2,\ldots,N)$) e v é o id do vídeo ($v = (1,2,\ldots,V)$). Em seguida são agrupados todos os valores de TP obtidos, considerando os vídeos e calculando a mediana desses valores dentro de cada grupo para obter $TP_{(v)}$ (ou seja, denota o verdadeiro positivo do vídeo v) . De forma similar, é necessário obter $FP_{(v)}$, $TN_{(v)}$ e $FN_{(v)}$. A taxa de falso positivo ($FPR_{(v)}$) e a taxa de falso negativo ($FNR_{(v)}$) podem ser calculadas de acordo

com as Equações 14 e 15 , respectivamente

$$FPR_{(v)} = \frac{FP_{(v)}}{FP_{(v)} + TN_{(v)}} \tag{14}$$

$$FNR_{(v)} = \frac{FN_{(v)}}{TP_{(v)} + FN_{(v)}}. \tag{15}$$

A métrica *f-measure* ($F1$) é calculada pela equação 16.

$$F1_{(v)} = \frac{2 \times Pr_{(v)} \times Re_{(v)}}{Pr_{(v)} + Re_{(v)}} \tag{16}$$

onde $Pr_{(v)}$ (Equação 17) e $Re_{(v)}$ (Equação 18) são as métricas de *precisão* e *revocação* para v vídeos, respectivamente.

$$Pr_{(v)} = \frac{TP_{(v)}}{TP_{(v)} + FP_{(v)}} \tag{17}$$

$$Re_{(v)} = \frac{TP_{(v)}}{TP_{(v)} + FN_{(v)}}. \tag{18}$$

O nível de dificuldade do vídeo v é definida pela Equação 19.

$$L_{(v)} = 1 - F1_{(v)} \tag{19}$$

2.4 REFERENCIAL TEÓRICO PARA A ABORDAGEM PROPOSTA

Os mapas de dificuldade gerados utilizando a abordagem mostrada na Seção 2.1 requerem resultados de algoritmos do estado-da-arte na forma de máscaras, vídeos e *ground-truths* para serem gerados. O objetivo desta pesquisa é desenvolver um método capaz de gerar esses mapas de forma automática, sem o auxílio de resultados de algoritmos e *ground-truths* de vídeos. Os conceitos relacionados às técnicas que pretende-se utilizar no desenvolvimento da abordagem proposta são apresentadas nesta Seção.

2.4.1 APRENDIZAGEM PROFUNDA

Segundo (BEZERRA, 2016) a aprendizagem profunda (*deep learning*) é uma subárea da aprendizagem de máquina que estuda técnicas para simular o comportamento do cérebro humano em algumas tarefas, como o reconhecimento de fala, o processamento da linguagem natural e o reconhecimento visual. Os algoritmos de aprendizagem profunda têm como objetivo produzir representações hierárquicas dos dados de entrada, utilizando camadas de processamento sequencial em uma rede neural artificial (RNA) (BEZERRA, 2016). De forma ordenada, o aprendizado de características (*features*) dos níveis elevados da hierarquia são gerados pela combinação de características dos níveis mais baixos (GOODFELLOW et al., 2016).

Alguns dos métodos reconhecidamente eficientes de aprendizagem profunda envolvem RNAs. As RNAs possuem a capacidade de aquisição e manutenção do conhecimento e são definidas como um conjunto de unidades de processamento. Tais redes são caracterizadas por possuírem neurônios artificias interligados por um grande número de interconexões (SILVA et al., 2010).

Uma RNA pode ser classificada conforme sua arquitetura, variando o modo pelo qual suas unidades de processamento estão conectadas. Essas conexões influenciam diretamente nos tipos de problemas que essas redes podem solucionar e na forma em que elas são treinadas (BEZERRA, 2016).

2.4.2 REDES CONVOLUCIONAIS

As redes neurais convolucionais (*convolution neural networks* – CNN) também são inspiradas em sistemas nervosos de seres vivos, estritamente no funcionamento do córtex visual (ZEILER; FERGUS, 2014). Segundo Bezerra (2016), as CNNs são baseadas nas seguintes etapas:

- Campos receptivos locais (*local repective fields*);
- Compartilhamento de pesos (*shared weights*);
- Convolução (*convolution*);

- Subamostragem (*subampling* ou *pooling*).

Ao contrário das redes neurais tradicionais, que são completamente conectadas, uma CNN faz uso de conectividade local. Considerando $L^{(1)}$ a primeira camada intermediária, cada unidade de $L^{(1)}$ está conectada a uma quantidade limitada de unidades da camada de entrada $L^{(0)}$. Essas unidades conectadas formam um campo receptivo local da própria unidade (BEZERRA, 2016), que é encarregado de detectar características visuais elementares. Essas características podem ser combinadas nas camadas subsequentes para detectar características mais complexas.

Em uma CNN, as unidades de uma determinada camada são organizadas em conjuntos separados. Para cada conjunto, é determinado um mapa de características (*feature map*) em que cada unidade está conectada a uma parte diferente da camada anterior (campo receptivo). Todas as unidades de um mapa de características compartilham os mesmos parâmetros, conhecidos como filtros (LECUN et al., 2015).

Essas unidades dentro de um mapa servem como detectores de uma mesma característica e cada uma delas está conectada em regiões distintas da imagem. Deste modo, uma camada oculta é segmentada em diversos mapas de características, dos quais cada unidade de um mapa tem o objetivo de realizar a mesma operação sobre a imagem de entrada. Essa operação é aplicada em regiões especificas da imagem (BEZERRA, 2016).

A convolução é realizada para cada uma das unidades de um mapa de característica. Na área de processamento de imagens, uma operação de convolução aplica o produto de Hadamard entre a matriz de *pixels* I da imagem e a matriz do núcleo da convolução (*convolution kernel)* (BEZERRA, 2016). A subamostragem tem o objetivo de reduzir a dimensionalidade de um mapa de característica, produzindo um outro com dimensões menores (GU et al., 2018). As técnicas de subamostragem comumente utilizadas são o *max poling*, o *average pooling* e a norma do conjunto (*L2-pooling*).

2.4.2.1 ARQUITETURA DA REDE PIX2PIX

Entre as características das Redes Generativas Adversárias Condicionais (*cGANs*) está a capacidade de transformar uma imagem em outra. Essas redes podem converter, por exemplo, imagens em tons de cinza em imagens coloridas e desenhos em imagens humanas realistas.

A utilização da rede *pix2pix* neste trabalho baseou-se nos resultados obtidos no trabalho de Tyleček e Šára (2013), que utizou o *dataset Center for Machine Perception* (CMP). Os autores utilizaram mapas de rótulos que foram gerados manualmente, em que foram atribuídos cores de acordo com o objeto representado (paredes, janelas e portas). Os autores utilizaram o mapa de rótulos para gerar novas fachadas.

No presente trabalho, a rede é treinada para gerar os mapas de dificuldades a partir das imagens reais. Como os mapas são construídos em tons de cinza e cada tonalidade representa um nível de dificuldade, a rede deve aprender identificar tais *pixels* em diferentes vídeos e em diversas categorias de vídeos.

A rede é composta por um gerador, baseado na arquitetura da rede *U-Net*, e um discriminador, representado por um classificador *PatchGAN* convolucional. Assim como na arquitetura *U-Net*, o gerador consiste em um codificador (*downsampler*) e um decodificador (*upsampler*). Cada bloco do codificador é composto pelas etapas *Convolution, Batch normalization, Leaky ReLU*. Cada bloco decodificador é composto pelas etapas *Transposed convolution, Batch normalization, Dropout* (aplicado aos primeiros 3 blocos), *ReLU* e conexões de salto entre o codificador e o decodificador.

Enquanto as *Generative Adversarial Networks (GANs)* aprendem com uma perda que se adapta aos dados, as *GANs* condicionais (*cGANs*) desenvolvem uma perda resultante da comparação da saída da rede com o *ground truth*. A perda do gerador é uma perda de entropia cruzada sigmoide das imagens geradas e uma matriz de uns (*gan_loss*). Também é utilizada a perda *L1*, que é o erro médio absoluto entre a imagem gerada e o *ground thuth*, o que permite que a imagem gerada se torne estruturalmente semelhante à imagem de destino. A perda total Lg do gerador é dada

pela equação

$$Lg = gan\_loss + LAMBDA * L1, \tag{20}$$

onde *LAMBDA* é definido em Isola et al. (2017) como $LAMBDA = 100$.

O discriminador *cGAN* é um classificador convolucional *PatchGAN*, que classifica cada *patch* (região da imagem) de imagem como real ou falso. Cada bloco do discriminador é compostor pelas etapas *Convolution*, *Batch normalizaiton* e *Leaky ReLU*. A saída da última camada é do tipo (*batch_size*, 30,30,1) na qual cada *patch* de 30×30 da imagem de saída classifica uma porção de 70×70 da imagem de entrada. O discriminador recebe duas entradas, a imagem de entrada e a imagem de destino, que devem ser classificadas como verdadeiras. Além disso, são recebidas também a imagem de entrada e a imagem gerada (a saída do gerador), que devem ser classificadas como falsa.

A função de perda do discriminador tem duas entradas: as imagens reais e as imagens geradas. A perda das imagens reais (*real_loss*) é uma perda cruzada da entropia sigmóide das imagens reais e uma matriz de uns (uma vez que estas são as imagens reais). A perda as imagens geradas (*generated_loss*), também são perdas de entropia sigmóide cruzadas, e uma matriz de zeros (uma vez que estas são as imagens falsas). A perda total do discriminador, é calculada pela soma de ambas as perdas.

2.4.2.2 TREINAMENTO DA REDE PIX2PIX

Conforme apresentado em Isola et al. (2017), faz-se necessário aplicar algumas etapas de pré-processamento antes do treinamento da rede. São elas: o redimensionamento de cada imagem para que tenham a resolução 256×256 *pixels*; inverter aleatoriamente a imagem horizontalmente (espelhamento aleatório) e normalizar as imagens para que os valores dos *pixels* estejam no intervalo $[-1, 1]$.

No procedimento para o treinamento, o gerador recebe a imagem de entrada e gera sua respectiva saída. O discriminador, por sua vez, recebe, além da imagem de entrada, a saída do gerador como suas respectivas entradas e a *target image* (a saída que a rede deve gerar). Em seguida é calculada a perda do gerador e

do discriminador. Posteriormente, calcula-se os gradientes de perda em relação às variáveis do gerador e do discriminador (entradas) e aplica-se o otimizador *Adam* para a descida do gradiente, resultando no laço de treinamento mostrado na Figura 4

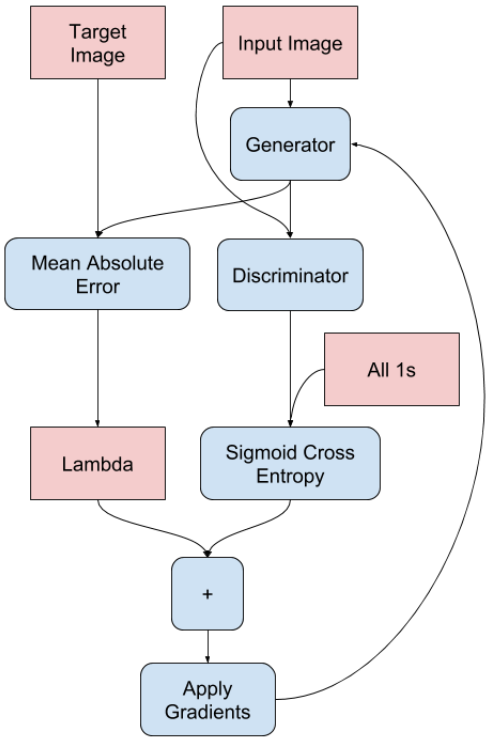

Figura 4: Treinamento do gerador (Isola et al. (2017))

O gerador deve receber a *input image* e gerar sua respectiva saída. O discriminador recebe a *input image* e a saída do gerador como suas respectivas entradas, além da *target image*. Em seguida é calculada a perda do gerador e do discriminador e, posteriormente, calcula-se os gradientes de perda em relação às variáveis do gerador e do discriminador (entradas) e aplica-se o otimizador *Adam*. O laço de treinamento do discriminador é exibido na Figura 5.

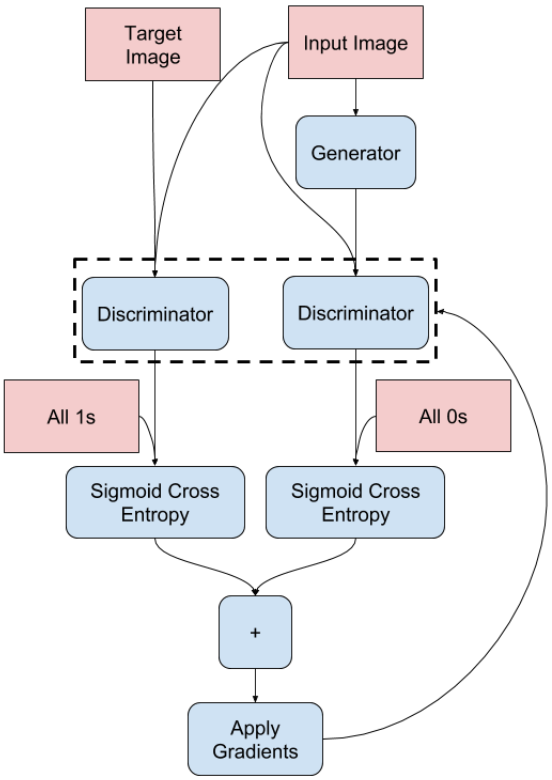

Figura 5: Treinamento do discriminador (Isola et al. (2017))

3 ABORDAGEM PROPOSTA

A geração de mapas de dificuldade sem o auxílio de *ground thruts* é uma tarefa que pode ser sintetizada em algumas etapas principais. Tais etapas são apresentada na Figura 6.

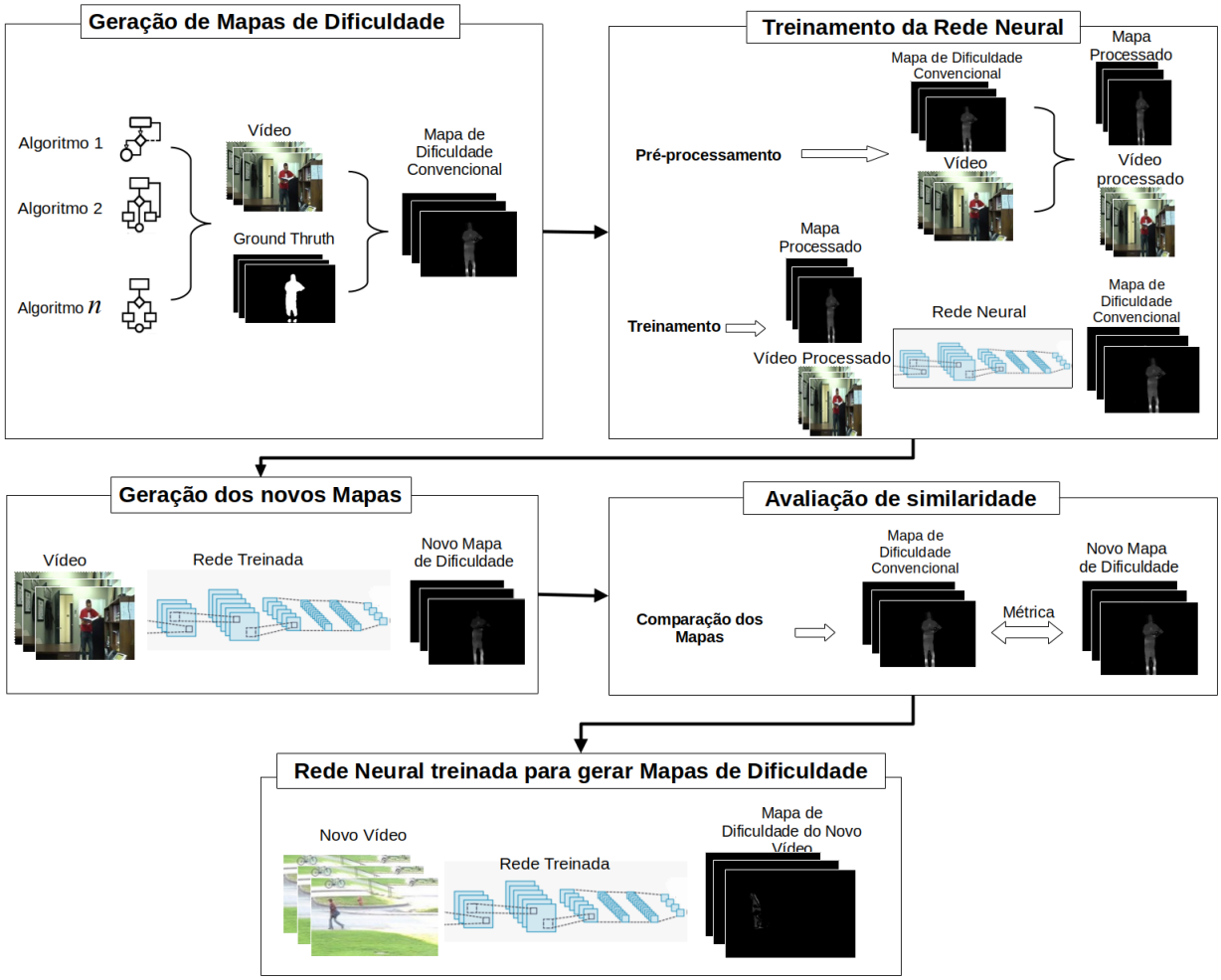

Figura 6: Etapas da abordagem proposta

A primeira etapa do desenvolvimento do método proposto “Geração dos mapas de dificuldade” envolve a criação de mapas de dificuldade utilizando a

abordagem tradicional Silva (2021). Os mapas gerados dessa forma são utilizados nesta pesquisa para o treinamento de uma rede neural e para a métrica de avaliação.

A etapa seguinte, denominada “Treinamento da Rede Neural”, trata do treinamento da rede utilizando os vídeos de um *dataset*, os quadros dos mapas de dificuldade gerados na etapa anterior. Nessa etapa, é realizado o pré-processados dos dados de forma que cada quadro se adapte aos requisitos exigidos para a entrada da rede neural. A formatação de cada quadro depende da arquitetura de rede escolhida.

Em seguida, a etapa “Geração de novos Mapas” consiste na geração automática de mapas de dificuldade por meio da rede treinada. Esses mapas são comparados com os gerados na primeira etapa (utilizando abordagem convencional) para verificar se existe similaridade. Nesta etapa, denominada “Avaliação de similaridade”, utilizamos a métrica *F1* (*DSC*) para quantificar a similaridade dos mapas gerados.

Uma vez que os resultados da aplicação da métrica sejam satisfatórios (os mapas sejam similares), a rede treinada pode ser utilizada para gerar novos mapas de dificuldades de novos vídeos, mesmo que esses vídeos não possuam *ground truths*. Essa é a etapa denominada “Rede Neural treinada para gerar Mapas de Dificuldade” exibida na Figura 6.

3.1 EXPERIMENTOS

Nesta seção, são apresentados os experimentos realizados, que consistem na aplicação do método proposto para gerar mapas de dificuldades sem o auxílio do *ground thuth*. Entende-se como etapas principais dos experimentos, a geração dos mapas de dificuldade por meio da abordagem tradicional e treinamento da rede neural.

3.1.1 GERAÇÃO DOS MAPAS DE DIFICULDADE

Inicialmente, é necessário a geração dos mapas de dificuldade dos vídeos de um *dataset* utilizado por pesquisadores da área para avaliar algoritmos de

detecção de mudança. Os vídeos dos *datasets* normalmente possuem *ground truths* correspondentes, o que é necessário para criação dos mapas utilizando a abordagem tradicional. Nesta pesquisa, os mapas gerados dessa forma são utilizados na etapa de treinamento da rede.

A criação dos mapas de dificuldade utilizados no treinamento utiliza a abordagem apresentada em Silva et al. (2021). Para isso, são necessários resultados da aplicação de algoritmos de detecção de mudanças do estado-da-arte para segmentar vídeos de um *dataset*. Tais resultados são imagens obtidas da comparação do resultado de cada algoritmo com o *ground truth* de cada vídeo.

As imagens que contêm os resultados de algoritmos, aqui chamadas de máscaras S, além dos vídeos e *ground truths* foram obtidas no *site* do *dataset* CDNet 2014 (UNIVERSITÉ DE SHERBROOKE, 2021). O vídeos do CDNet 2014 são divididos em categorias, cada uma delas representando um desafio diferente para os algoritmos de detecção de mudança. A Tabela 2 mostra os nomes dos vídeos do CDNet e suas respectivas quantidades de quadros.

No *site*, é disponibilizado também um *ranking* que mostra os algoritmos que obtiveram os melhores desempenhos entre os que utilizaram seus vídeos para avaliação. As métricas PFR, FNR, Pr, Re, Sp, PWC e $F1$ são utilizadas para representar os desempenhos nesse *ranking*. Entre os algoritmos listados no *ranking* na data de acesso ao *site* (25 de janeiro de 2021), as máscaras S de 30 deles, escolhidos aleatoriamente e listados na Tabela 3, foram utilizadas para geração dos mapas.

Conforme discutido e exemplificado na Seção 2.1, um mapa gerado utilizando 30 algoritmos ($n = 30$) contém 31 níveis de dificuldade ($n + 1$). Uma imagem cujos valores dos *pixels* variam entre 0 (*pixel* sem dificuldade) e 240 (*pixel* 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.

Tabela 2: Quantidade de quadros por vídeos do CDNet 2014

Categoria	Vídeo	Quadros	Vídeo	Quadros
BadWeather	blizzard	1923	skating	1550
(7823)	snowFall	2850	wetSnow	1500
Baseline	highway	1231	office	1481
(4413)	pedestrians	800	PETS2006	901
CameraJitter	badminton	351	boulebard	1711
(3134)	sidewalk	401	traffic	671
Dynamic	boats	6100	canoe	390
Background	fall	3001	fountain01	785
(13277)	fountain02	1000	overpass	2001
Intermittent	abandonedBox	2051	parking	1401
Object Motion	sofa	2251	streetLight	3026
(12111)	tramstop	1881	winterDriveway	1501
Low Framerate	port_0_17fps	1000	tramCrossroad_1fps	250
(2600)	tunnelExit_035fps	1000	turnpike_0_5fps	350
Night	bridgeEntry	750	busyBoulvard	1015
Videos	fluidHighway	482	streetCornerAtNight	2200
(6137)	tramStation	1250	winterStreet	440
PTZ	continuousPan	550	intermittentPan	1044
(2546)	twoPositionPTZCam	637	zoomInZoomOut	315
Shadow	backdoor	1601	bungalows	1401
(14105)	busStation	951	copyMachine	2901
	cubicle	6301	peopleInShade	950
Thermal	corridor	4901	diningRoom	3001
(18055)	lakeSide	5501	library	4301
	park	351		
Turbulence	turbulence0	2000	turbulence1	1400
(6100)	turbulence2	2000	turbulence3	700

3.1.2 TREINAMENTO DA REDE NEURAL

O treinamento de uma rede neural é uma tarefa que consiste em algumas etapas. A primeira delas é o pré-processamento das imagens de entrada em que cada quadro deve ser preparado de forma que se adapte aos requisitos exigidos para a entrada da rede.

Nos experimentos realizados neste trabalho, a arquitetura de rede neural que produziu melhores resultados é a PIX2PIX. Para que sejam utilizadas como entrada da rede, as imagens devem ser redimensionadas para $256x256$ pixels. Além do redimensionamento da imagem, um processo chamado *jittering* aleatório (redimensionar a imagem para um tamanho maior, recortar aleatoriamente a imagem

Tabela 3: Algoritmos utilizados para gerar os mapas de dificuldade

Algoritmo	Referência
WeSamBE	Jiang e Lu (2018)
SuBSENSE	St-Charles et al. (2015b)
SharedModel	Yingying Chen et al. (2015)
FTSG	Wang et al. (2014)
CwisarDRP	De Gregorio e Giordano (2017)
C-EFIC	Allebosch et al. (2016)
Multimode BS	Sajid e Cheung (2017)
EFIC	Allebosch et al. (2015)
CwisarDH	Gregorio e Giordano (2014)
Multimode BS Version 0	Maddalena e Petrosino (2010)
Spectral-360	Sedky et al. (2014)
SBBS	Varghese e G (2017)
BMOG	Martins et al. (2017)
AAPSA	Ramírez-Alonso e Chacón-Murguía (2016)
IUTIS-1	Bianco et al. (2017)
GraphCutDiff	Miron e Badii (2015)
Mahalanobis distance	Benezeth et al. (2010)
SC_SOBS	Maddalena e Petrosino (2012)
RMoG	Varadarajan et al. (2013)
KDE - ElGammal	Elgammal et al. (2000)
GMM - Stauffer & Grimson	Stauffer e Grimson (1999)
CP3-online	Liang et al. (2015)
GMM - Zivkovic	Zivkovic (2004)
Euclidean distance	Benezeth et al. (2010)
BSPVGAN	Zheng et al. (2019)
FgSegNet-S (FPM)	Lim e Keles (2018)
IUTIS-3	Bianco et al. (2017)
IUTIS-5	Bianco et al. (2017)
PAWCS	St-Charles et al. (2015a)
WisenetMD	Lee et al. (2019)

para o tamanho de destino do modelo e virar aleatoriamente nas imagens) é necessário para a formatação dos dados de entrada da rede. A Figura 7 mostra a etapa de pré-processamento aplicada sobre um quadro do vídeo *blizzard*, que pertence ao *dataset* CDNet 2014. Para o treinamento da rede é necessário combinar imagem de entrada e de saída em uma mesma imagem.

Realizado o pré-processamento, inicia-se o processo de treinamento da rede, conforme descrito em 2.4.2.2. Nessa etapa, os quadros de vídeo foram divididos em dois grupos: 80% para o treinamento e 20% para teste. Isso representa 72216

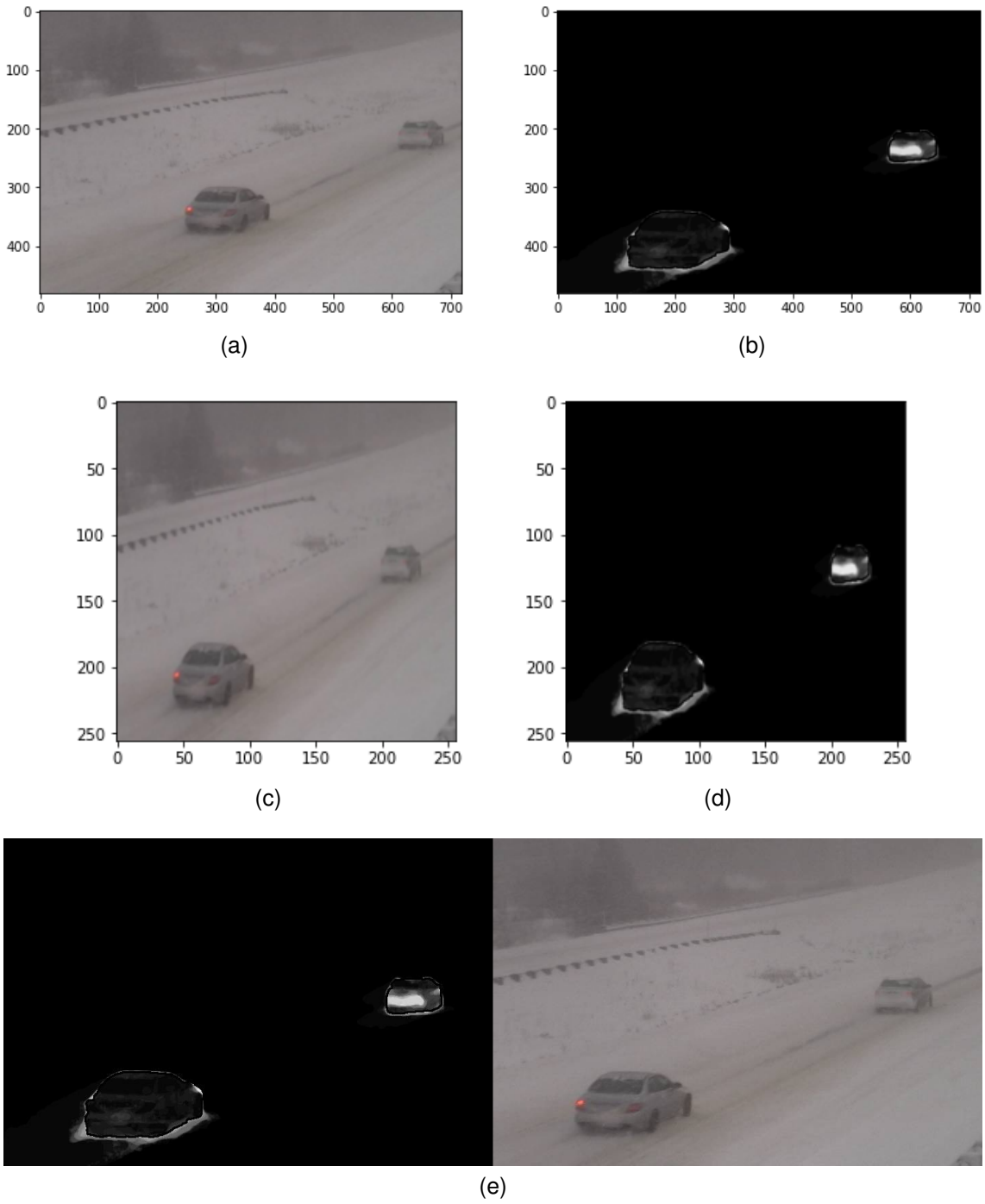

Figura 7: Exemplo da etapa de pré-processamento aplicada sobre um quadro do vídeo blizzard. (a) quadro de entrada, (b) quadro de entrada do mapa de dificuldade, (c) quadro de entrada pré-processado, (d) quadro de saída do mapa pré-processado e (e) entrada para a rede.

imagens para o treinamento e 18082 imagens para teste, totalizando 90298 imagens. As configurações utilizadas para o treinamento da rede PIX2PIX são apresentadas na Tabela 7, do Apêndice A.

Uma vez treinada a rede neural, essa rede foi utilizada para gerar mapas de dificuldade dos vídeos do *dataset* CDNet 2014. Em seguida, a métrica $F1$ (DSC) foi aplicada para comparar esses mapas com os gerados pela abordagem de Silva et al. (2021). As análises dos resultados obtidos do experimento são apresentadas na Seção 3.2

3.2 RESULTADOS

Os experimentos utilizando a abordagem proposta foram realizados com o objetivo de identificar o desempenho da rede utilizada, medido pela comparação do mapa de dificuldade de um determinado vídeo, gerado pela rede *PIX2PIX* com um mapa gerado utilizando a abordagem apresentada em Silva et al. (2021), que utiliza o *ground thuth* no processo de criação dos mapas.

Uma limitação do método proposto está relacionado com a perda de informações, resultado do processo de redimensionamento dos quadros do mapa de dificuldade para adequá-los à entrada da rede neural *PIX2PIX*. O redimensionamento foi relizado por meio da interpolação do vizinho mais próximo (*Nearest-neighbor interpolation)*. Por exemplo, o quadro 1960 do vídeo *skating*, que pertence a categoria *badweather* possui dimensões 540×360 *pixels*. O mapa de dificuldade do quadro utilizado como entrada (que é gerado pelo método tradicional) possui um total de 304 *pixels* com nível de dificuldade de 25%, 231 *pixels* com nível de dificuldade de 50%, 317 *pixels* com nível de dificuldade de 75% e 118 *pixels* com o maior nível de dificuldade (100%). O mesmo quadro, após o pré-processamento, apresentou 8 *pixels* com nível de dificuldade 25%, 10 *pixels* com nível de dificuldade 50%, 21 *pixels* com nível de dificuldade 75% e nenhum *pixel* com nível de dificuldade 100%.

Em seguida, comparou-se o mapa de dificuldade redimensionado com a saída da rede, buscando quantificar os acertos da abordagem proposta. A saída da rede obteve 18 *pixels* com nível de dificuldade 25%, 9 *pixels* com 50% de dificuldade, 11 *pixels* com nível de dificuldade 75% e nenhum *pixel* com nível de dificuldade 100%. A Tabela 4 mostra a quantidade de *pixels* com os níveis de dificuldade 25%, 50%, 75% e 100% antes e depois de processados utilizando a abordagem proposta para a

geração de mapas de dificuldade.

Tabela 4: Quantidade de *pixels* do quadro 1960 do vídeo *skating* conforme os níveis de dificuldade.

Nível de Dificuldade	Mapa de dificuldade	Mapa redimensionado	Saída (Rede)
25%	304	8	18
50%	231	10	9
75%	317	21	11
100%	118	0	0

A Figura 8 mostra o resultado visual do mapa de dificuldade do quadro 782 do vídeo *sofa*, gerado utilizando a abordagem proposta. É apresentado, além do quadro 782, o mapa de dificuldade gerado pela abordagem de Silva et al. (2021) (utilizado como *ground truth*). Estas são consideradas as entradas da rede. Em seguida é apresentado um quadro que mostra a saída da rede (imagem predita) que representa o mapa de dificuldade gerado pela abordagem proposta.

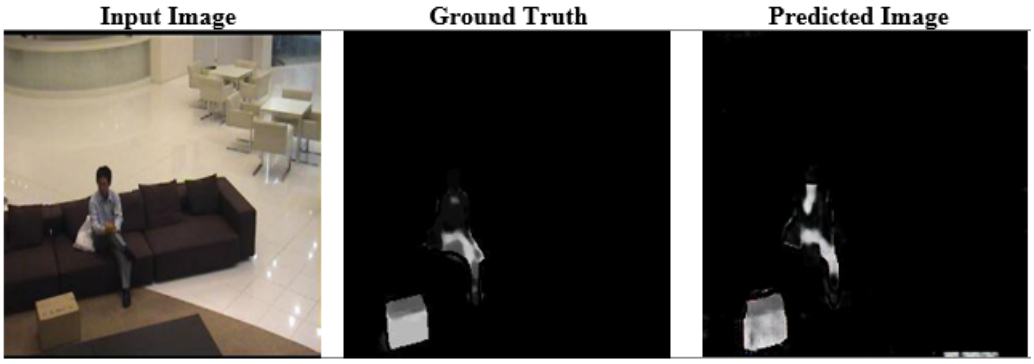

Figura 8: Resultado, quadro 782 do vídeo *sofa*, sua respectivas entrada, *ground truth* e a saída da rede.

Outro fator importante verificado durante os experimentos iniciais encontrado está relacionado com a formação dos conjuntos de treinamento e de teste. Uma vez que existem muitos quadros que não possuem *pixels* com todos os níveis de dificuldade, i.e. quadros que não possuem o elemento de interesse, o agrupamento dos quadros (e consequentemente dos vídeos) na proporção 80% e 20% (ou qualquer outra) deve ser realizada de forma que ambos os grupos contenham a região de interesso e dessa forma quadros com níveis de dificuldade. Sendo assim, foi necessário a remoção dos quadros que não possuem a região de interesse.

O *dataset* CDNet 2014, utilizado nos experimentos realizados neste trabalho, está dividido em 11 categorias, e cada uma delas possuem 4, 5 ou 6 vídeos

(Tabela 2). Os resultados da comparação dos mapas de dificuldade gerados pelo método convencional com os gerados pelo método proposto são apresentados a seguir, conforme o treinamento da rede.

Na Figura 9 é possível visualizar a evolução do treinamento na avaliação dos vídeos relacionados à categoria *BadWeather*. Ainda na época 5, os resultados de *F1* são superiores a 0,85. Apenas no vídeo *blizzard* ocorre maiores variações, permanecendo as demais no intervalo entre 0,85 e 0,98.

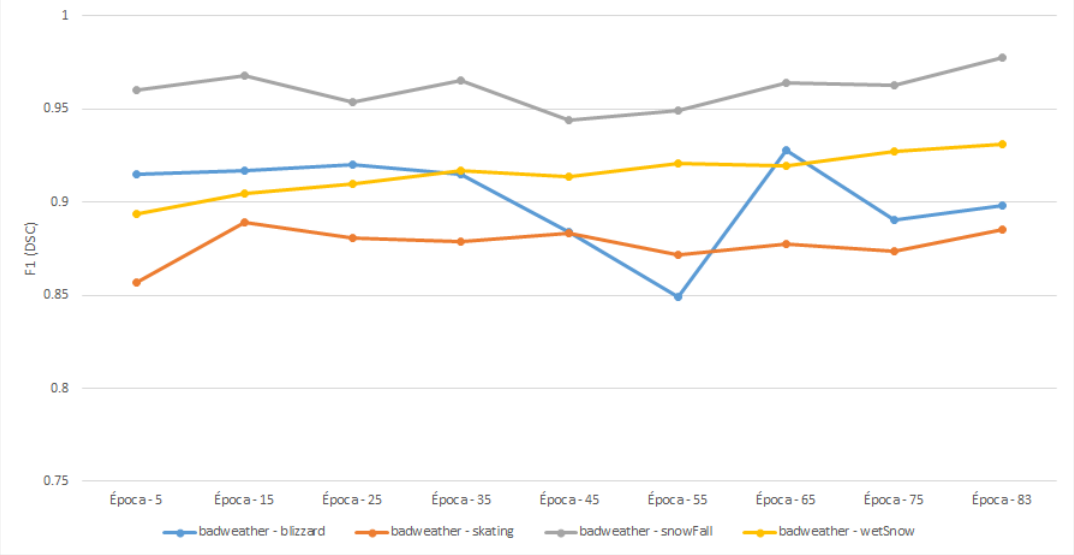

Figura 9: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *badweather*.

Na Figura 10 é apresentada a evolução dos resultados quando se utiliza os vídeo da categoria *baseline*. Como pode ser observado, os dados se comportam de maneira semelhante ao que ocorre na categoria *BadWeather*. O desvio padrão calculado é 0.047.

Nos vídeos da categoria *Camera Jitter*, apenas o resultado obtido utilizando o vídeo *traffic* obteve um valor abaixo de 0,7 do cálculo da métrica *F1*. Os resultados dessa avaliação pode ser visualizado na Figura 11.

A categoria *Dynamic Background* possui seis vídeos. Desses, em apenas dois a métrica *F1* obteve valor abaixo de 0.8. Nota-se também que os valores obtidos não variaram conforme o número de épocas era aumentado. Os quadros do vídeo *canoe* obtiveram baixos índices da época 5 até a 83 (Figura 12).

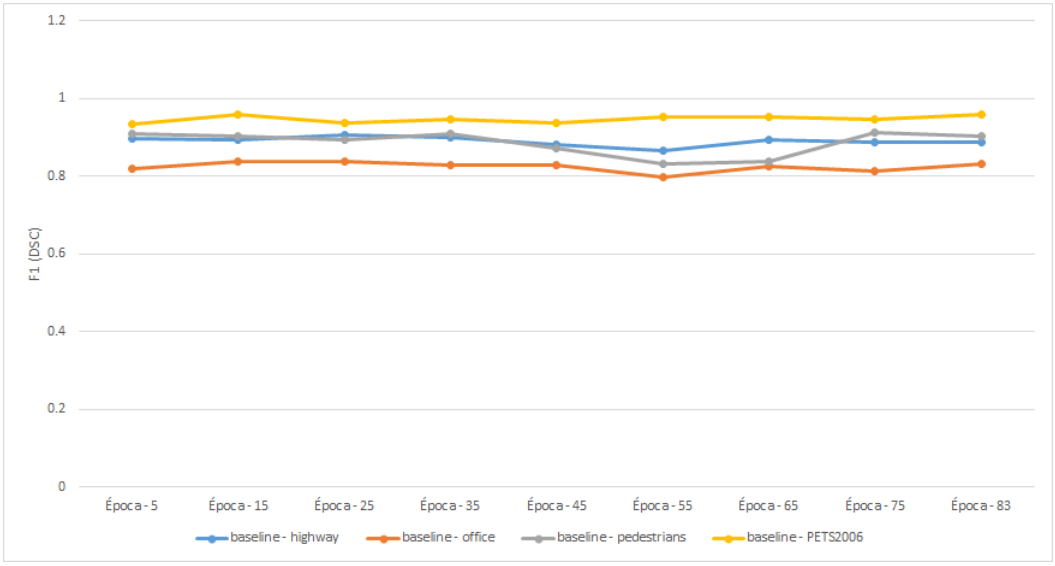

Figura 10: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *baseline*.

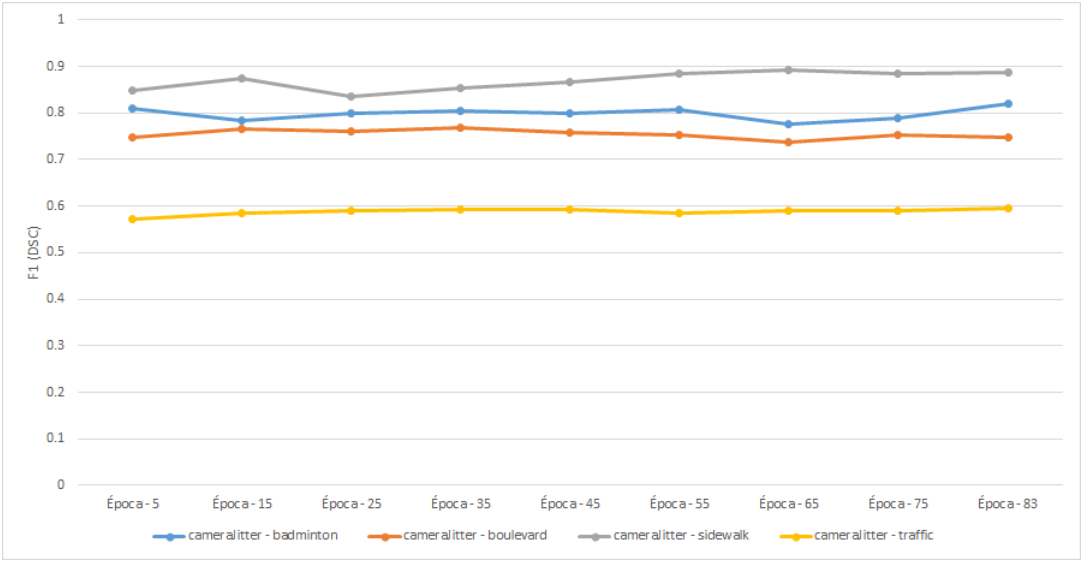

Figura 11: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *camerajitter*.

A geração de mapas utilizando quatro dos seis vídeos da categoria *Intertermmittent Object Motion* obtiveram resultados satisfatórios, uma vez que os valores de $F1$ foram acima de 0,95. Os demais vídeos variaram de 0.84 e 0.9, no decorrer das épocas, conforme pode ser observado na Figura 13.

Na Figura 14 são apresentados os resultados da geração de mapas de dificuldade utilizando os vídeos da categoria *Low Framerate*. Os vídeos *tramCrossroad_1fps* e *tunnelExit_0_35fps* obtiveram valores entre 0,89 e 0.95 do

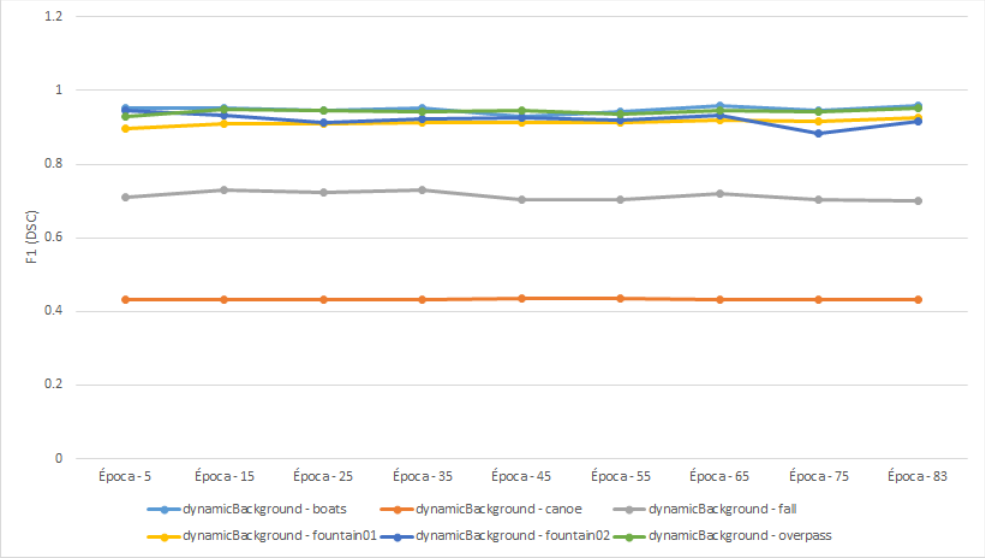

Figura 12: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *Dynamic Background*.

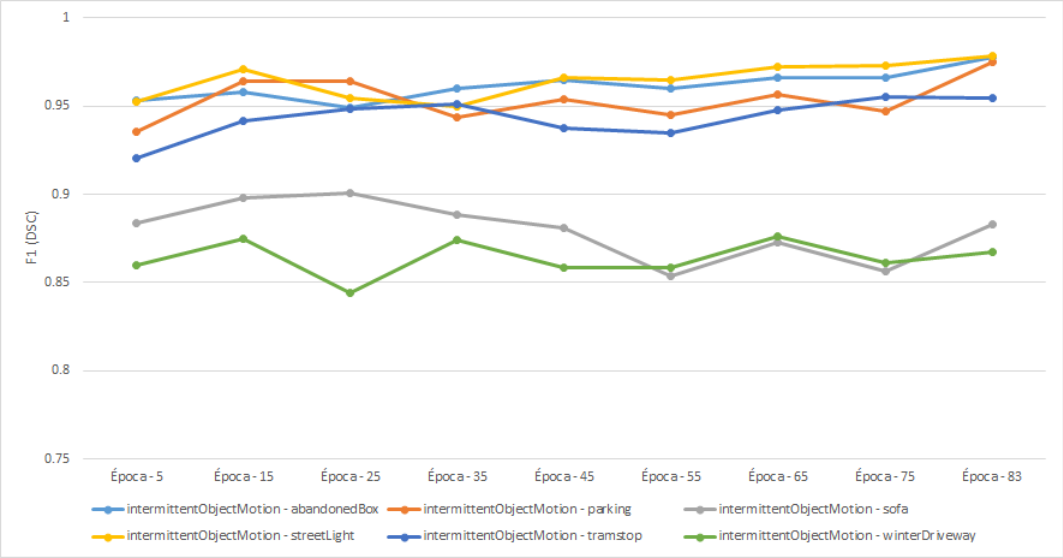

Figura 13: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *Intermittent Object Motion*.

cálculo de $F1$. Nos vídeos *port_0_17fps* e *turnpike_0_5fps*, os valores de $F1$ permaneceram entre entre 0,73 e 0,82.

Os vídeos da categoria *Night Videos* obtiveram uma média de 0,88 no cálculo da métrica *F1* e um desvio padrão de 0,046. Os resultados dos mapas gerados a partir de vídeos dessa categoria também apresentou baixa variação no decorrer do treinamento da rede, obtendo valores acima de 0,8 a partir da quinta época. A Figura 15 os resultados dessa avaliação.

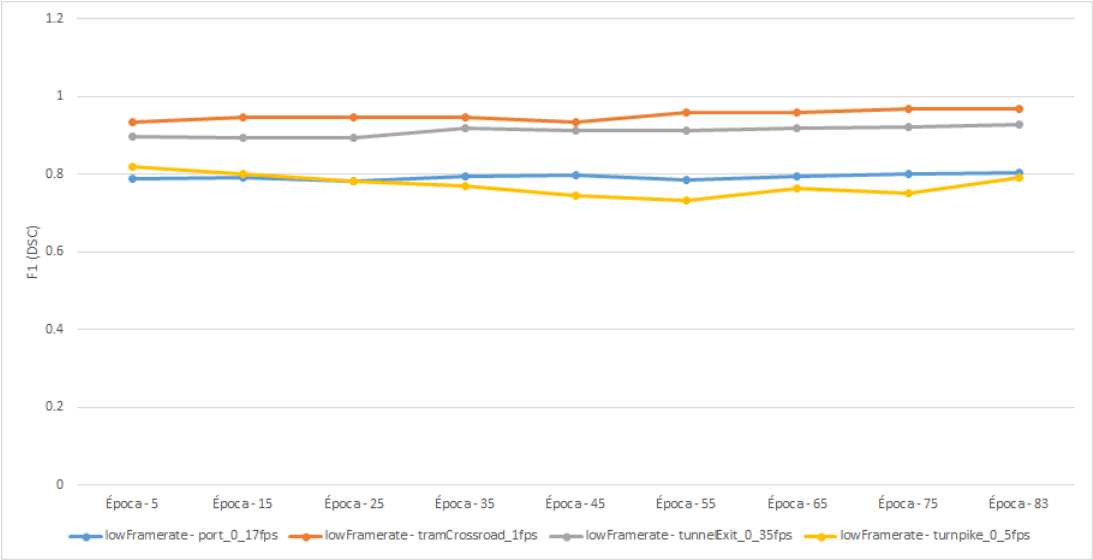

Figura 14: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *Low Framerate*.

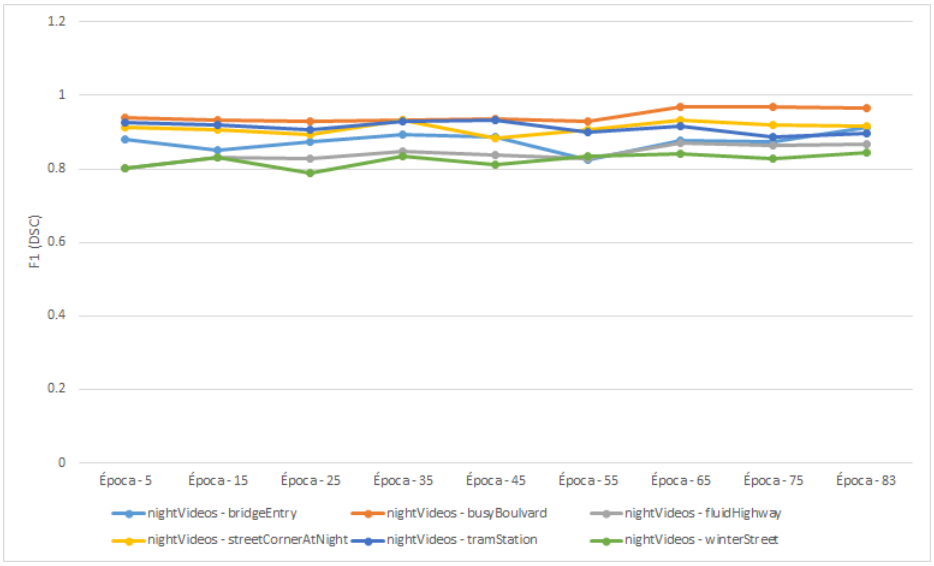

Figura 15: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *Night Videos*.

Os mapas gerados a partir dos vídeos da categoria *PTZ* obtiveram resultados pouco satisfatórios. Apenas quando se utilizou o vídeo *twopositionPTZcam*, os resultados apresentaram valores de *F1* acima de 0,4, o que indica baixa similaridade entre a saída da rede e o *ground truth* (Figura 16).

Em relação à categoria *Shadow* os mapas gerados apresentaram boa similaridade para a maioria dos vídeos. Apenas no vídeo *copymachine*, a métrica $F1$ ficou abaixo de 0,85. Os resultados considerando os vídeos da categoria *Shadow*

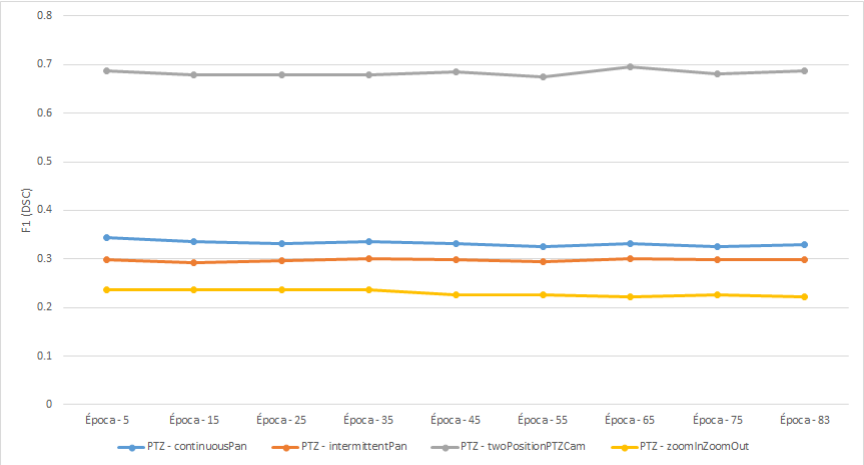

Figura 16: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *PTZ*.

são apresentados na Figura 17.

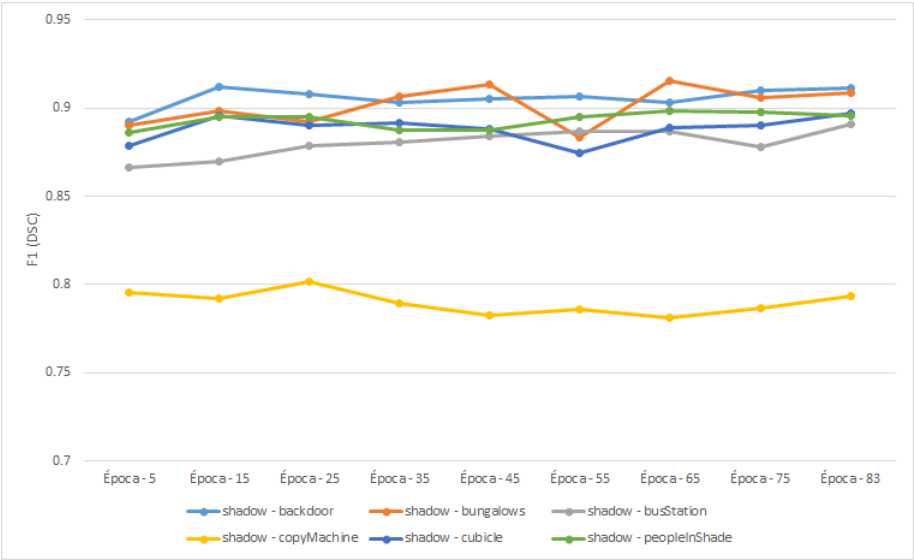

Figura 17: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *shadow*.

Os resultados obtidos utilizando os vídeos da categoria *thermal* podem ser visualizados na Figura 18. A geração dos mapas apresentou bons resultados a partir da quinta época, com exceção dos vídeos *diningroom* e *library*, em que os valores de *F1* foram mais baixos.

Na Figura 19 é possível observar a variação que ocorre nos resultados, quando utiliza-se o vídeo *turbulence3*, entre as épocas 35 e 55. Os melhores resultados obtidos da categoria *Turbulence* foram nos mapas gerados utilizando o vídeo *turbulence2*, em que os valores de $F1$ chegou a 0,98, o que representa alta

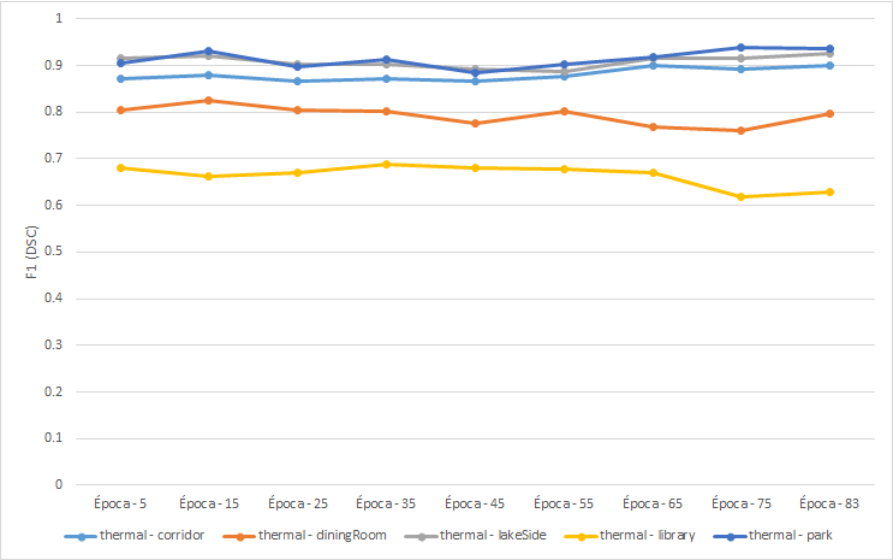

Figura 18: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *thermal*.

similaridade entre a saída da rede e o *ground truth*.

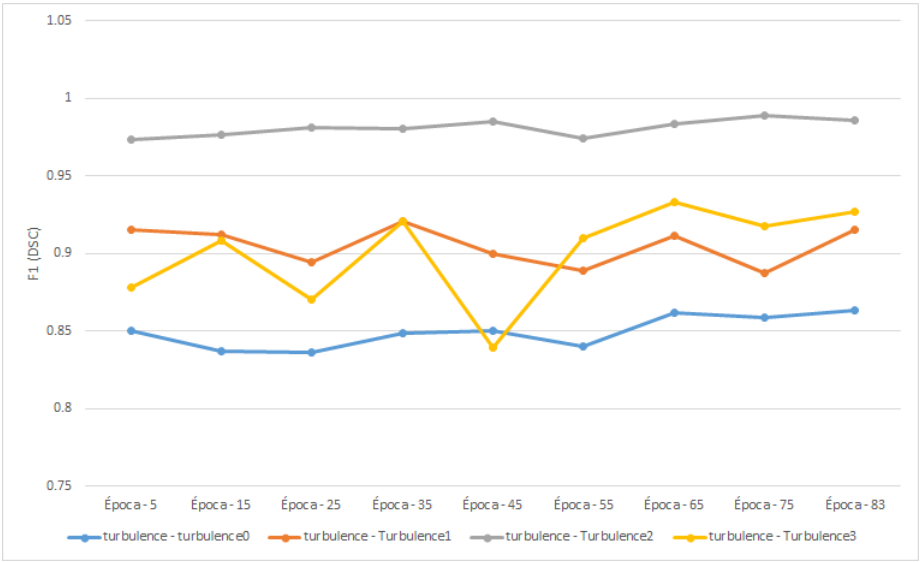

Figura 19: Valores de *F1* de acordo com o número de épocas quando são utilizados os vídeos da categoria *Turbulence*.

Para apresentação detalhada dos resultados da geração dos mapas de dificuldade, a Tabela 5 mostra os valores da métrica $F1$ calculados considerando a utilização de cada vídeo do *dataset* CDNet 2014. São apresentados, ainda, os valores de $F1$ obtidos da utilização de todos os vídeos de uma mesma categoria. Nessa análise, considerou-se os resultados do treinamento com 83 épocas.

Como pode ser observado, a categoria de vídeo que apresentou resultados

Tabela 5: Cálculo da métrica $F1$ de cada vídeo e de cada categoria do *dataset*

Categoria	Vídeo	$F1$ (Vídeo)	$F1$ (Categoria)
badweather	blizzard	0.8981987098594765	0.9307242884041783
	skating	0.8850039082188760	
	snowFall	0.9773182919150905	
	wetSnow	0.9311812337239583	
baseline	highway	0.8867896076156060	0.8858617577849135
	office	0.8316145360670507	
	pedestrians	0.9036122322082519	
	PETS2006	0.9579179458196650	
cameraJitter	badminton	0.8201837673993178	0.7418000236390129
	boulevard	0.7485984157195245	
	sidewalk	0.8878806785300926	
	traffic	0.5956546359592014	
dynamicBackground	boats	0.9578577260502049	0.8783363569128275
	canoe	0.4326711801382215	
	fall	0.7005816061365823	
	fountain01	0.9273013947116342	
	fountain02	0.9173534393310547	
	overpass	0.9508687540182746	
intermittentObjectMotion	abandonedBox	0.9774323994805924	0.9425277112686541
	parking	0.9752940113434164	
	sofa	0.8827498408484618	
	streetLight	0.9781293963441754	
	tramstop	0.9545025408109873	
	winterDriveway	0.8671706697077450	
lowFramerate	port_0_17fps	0.8052789306640625	0.8669860546405499
	tramCrossroad_1fps	0.9697811889648438	
	tunnelExit_0_35fps	0.9291104888916015	
	turnpike_0_5fps	0.7923686436244419	
nightVideos	bridgeEntry	0.9145058186848959	0.910538452844278
	busyBoulvard	0.9640154157366071	
	fluidHighway	0.8676060942030445	
	streetCornerAtNight	0.9152016379616478	
	tramStation	0.8962892456054687	
	winterStreet	0.8449022119695490	
PTZ	continuousPan	0.3300682761452415	0.39321157418045344
	intermittentPan	0.2988682797080592	
	twoPositionPTZCam	0.6862478256225586	
	zoomInZoomOut	0.2210666717044891	
shadow	backdoor	0.9113557049047167	0.878151482776203
	bungalows	0.9083288878308496	
	busStation	0.8909582966909358	
	copyMachine	0.7936114838119218	
	cubicle	0.8973644840443162	
	peopleInShade	0.8955495733963815	
thermal	corridor	0.8997054571526981	0.8269894801729447
	diningRoom	0.7958141714086548	
	lakeSide	0.9271600846265469	
	library	0.6287400160378557	
	park	0.9369463853433099	
turbulence	turbulence0	0.8633304214477540	0.9229758215732262
	turbulence1	0.9157214573451451	
	turbulence2	0.9863248825073242	
	turbulence3	0.9269026620047432	

menos satisfatórios foi a *PTZ*. O termo *PTZ* refere-se a união de três funcionalidades em uma câmera (*PAN, TILT, ZOOM)*. *PAN*, movimento panorâmico horizontal, *TILT*, movimento vertical e *ZOOM*, movimento de aproximação ou distanciamento.

Segundo (Wang et al., 2014), os vídeos dessa categoria são considerados um desafio maior para os algoritmos de detecção de mudanças, quando comparados com vídeos de uma câmera estática. Possivelmente, em razão da descontinuidade das cenas, os erros de classificação cometidos podem variar de acordo de com o algoritmo, o que pode gerar mapas de dificuldades menos precisos. Na Figura 20 é possível observar a diferença entre o *ground truth* (mapa gerado pela abordagem tradicional) e a saída da rede.

Figura 20: Resultado - Quadro 725 do vídeo zoomInZoomOut, sua respectivas entrada, *ground truth* e a saída da rede.

Os melhores resultados da utilização dos vídeos da categoria *Turbulence* foram obtidos a partir do vídeo *turbulence2*. Na Figura 21, observa-se que a região que contêm pixels com níveis de dificuldade altos é pequena. O valor da métrica $F1$ chegou a 0,98. De acordo com os mapas de dificuldades do vídeo *turbulence2*, ele possui em média doze pixels de difícil classificação em cada imagem. Por outro lado, o vídeo *zoomInZoomOut* apresenta em média 4753 pixels em cada imagem com alguma dificuldade de classificação. Vale ressaltar que esses valores correspondem aos mapas de dificuldades sem o redimensionamento.

Em relação à análise que considera os vídeos agrupados por categoria, a que o mapa gerado pelo método proposto é mais similar ao gerado pelo método convencional foi a *Intermittent Object Motion* (Figura 22). Essa categoria é a que contêm vídeos destinados a avaliar algoritmos que se adaptam à mudanças no fundo

Figura 21: Resultado – Quadro 2332 do vídeo *turbulence2*, sua respectivas entrada, *ground truth* e a saída da rede.

da cena.

Dos resultados da utilização dos vídeos pertencentes à categoria *Intermittent Object Motion*, o menor valor de $F1$ foi obtido da utilização do vídeo *winterDriveway*. O vídeo possui a maior média de pixels com algum nível de dificuldade, com aproximadamente 946 *pixels* por quadro. Vale ressaltar que as categorias e vídeos não possuem a mesma quantidade de quadros, conforme mostrado na Tabela 2. Essa característica pode explicar as diferenças nos resultados obtidos quando as categorias e vídeos são comparados.

Os resultados apresentados até o momento foram obtidos obedecendo uma divisão que considerou, para cada vídeo, de 80% dos quadros para treino e 20% dos quadros para teste. No experimento seguinte, os grupos de treino e teste consideram o vídeo com todos os seus quadros, mantendo-se a proporção 80% para treino e 20% dos vídeos para teste. O agrupamento considerou que, pelo menos, um vídeo de cada categoria fosse incluído no conjunto de teste. A Tabela 6 apresenta os resultados obtidos do novo experimento, que resultou em 71355 quadros no conjunto de treino e 18946 no de teste.

Tabela 6: Cálculo da métrica $F1$ do conjunto de teste separado por vídeo.

Vídeos	**F1 (DSC)**
skating	0.88032939295615
PETS2006	0.93667378991875
traffic	0.57172565488631

fountain01	0.91187108519730
overpass	0.90118560714760
parking	0.94871479608943
winterDriveway	0.84644892500052
tramCrossroad_1fps	0.90632824707031
turnpike_0_5fps	0.77606628417969
bridgeEntry	0.89913970947266
fluidHighway	0.79741991605007
continuousPan	0.34844499067827
backdoor	0.88622450247770
bungalows	0.89278817568227
diningRoom	0.79845043310758
park	0.85794567314648
turbulence1	0.92579958234515

Nota-se que o melhor desempenho em termos de geração dos mapas de dificuldade permaneceu nos experimentos que utilizou os vídeos da categoria *Intermittent Object Motion*, especialmente quando se considera o vídeo *parking*. Do mesmo modo, os mapas com menores similaridades foram obtidos da utilização dos vídeos categoria *PTZ*. O mapa gerado a partir do vídeo *continuousPan* obteve melhores resultados do que utilizando a abordagem de treino anterior. Os demais vídeos apresentaram desempenho similar ao do experimento anterior. Neste experimento foram executadas 58 épocas.

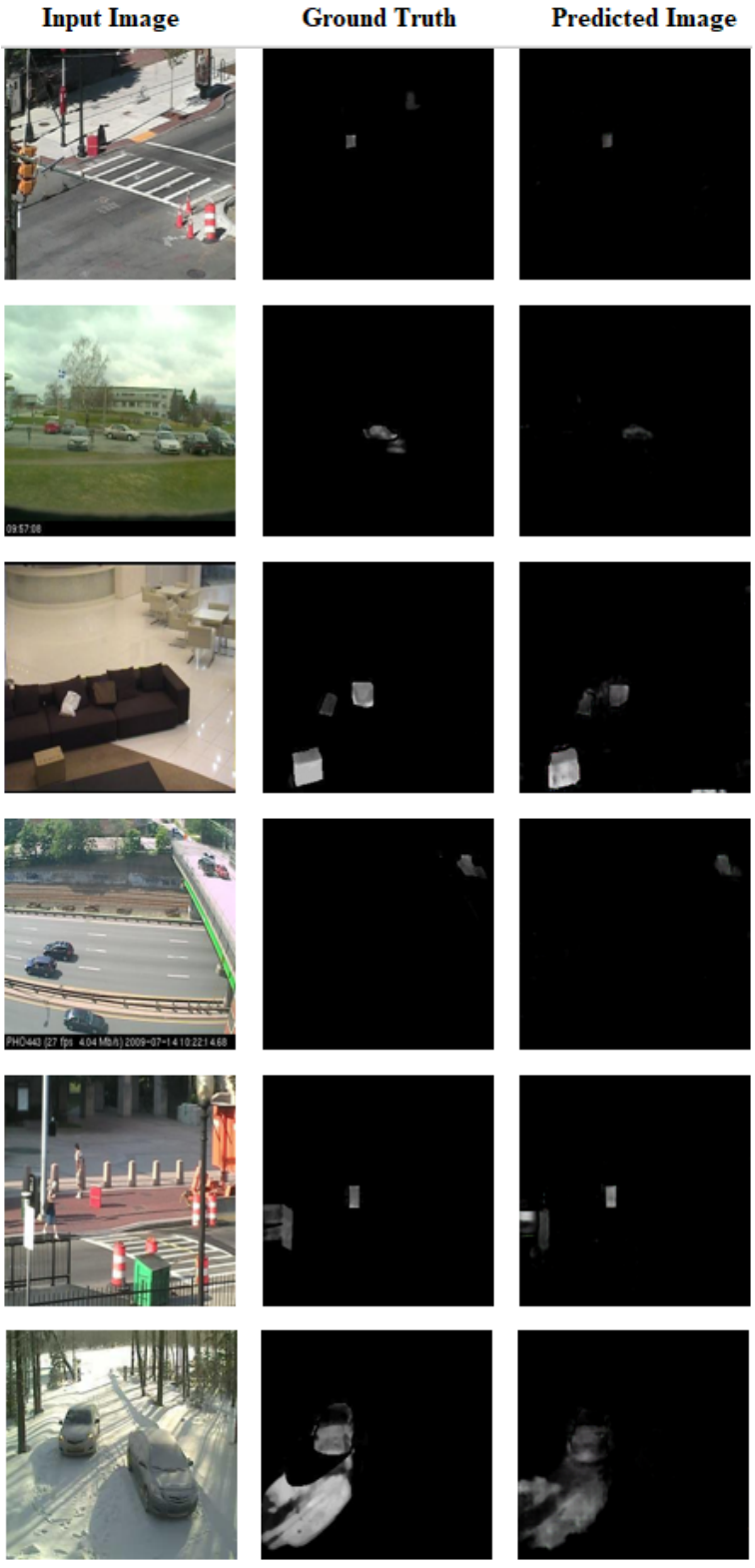

Figura 22: Resultado dos vídeos da categoria *Intermittent Object Motion*.

4 CONSIDERAÇÕES FINAIS

O presente trabalho apresenta um método para geração de mapas de dificuldade de vídeos. Esses mapas são gerados automaticamente, sem o auxilio do *ground truth*. O método desenvolvido consiste em uma rede neural treinada, que pode auxiliar pesquisadores da área na tarefa de criar novos vídeos de *datasets*, uma vez que a tarefa mais trabalhosa na criação desses conjuntos de vídeos é a atribuição de rótulos para gerar os *ground truths*.

Os resultados dos experimentos aqui realizados mostraram que os mapas gerados pelo rede neural são, em sua maioria, similares aos criados utilizando a abordagem tradicional (SILVA, 2021; SILVA et al., 2021) para criação dessas estruturas. Os mapas de alguns vídeos utilizados (do *dataset* CDNet 2014), que pertencem a uma categoria específica (*PTZ*), apresentaram pouca similaridade com os respectivos mapas gerados pela abordagem tradicional. Com exceção de vídeos com as mesmas características desses, o método proposto mostrou-se capaz de criar mapas de dificuldades mesmo quando um *ground truth* não está disponível.

Como trabalhos futuros, pretende-se desenvolver um método para selecionar vídeos os quais podem ser gerados mapas de dificuldades com qualidade e testar outros modelos de redes neurais artificias para geração de mapas.

REFERÊNCIAS

ALLEBOSCH, G.; DEBOEVERIE, F.; VEELAERT, P.; PHILIPS, W. Efic: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints. In: BATTIATO, S.; BLANC-TALON, J.; GALLO, G.; PHILIPS, W.; POPESCU, D.; SCHEUNDERS, P. (Ed.). **Advanced Concepts for Intelligent Vision Systems**. Cham: Springer International Publishing, 2015. p. 130–141. ISBN 978-3-319-25903-1.

ALLEBOSCH, G.; HAMME, D. V.; DEBOEVERIE, F.; VEELAERT, P.; PHILIPS, W. C-efic: Color and edge based foreground background segmentation with interior classification. In: BRAZ, J.; PETTRÉ, J.; RICHARD, P.; KERREN, A.; LINSEN, L.; BATTIATO, S.; IMAI, F. (Ed.). **Computer Vision, Imaging and Computer Graphics Theory and Applications**. Cham: Springer International Publishing, 2016. p. 433–454. ISBN 978-3-319-29971-6.

BENEZETH, Y.; JODOIN, P.-M.; EMILE, B.; LAURENT, H.; ROSENBERGER, C. Comparative study of background subtraction algorithms. **Journal of Electronic Imaging**, SPIE, v. 19, n. 3, p. 1 – 12, 2010.

BEZERRA, E. Introdução à aprendizagem profunda. **Artigo–31º Simpósio Brasileiro de Banco de Dados–SBBD2016–Salvador**, 2016.

Bianco, S.; Ciocca, G.; Schettini, R. Combination of video change detection algorithms by genetic programming. **IEEE Transactions on Evolutionary Computation**, v. 21, n. 6, p. 914–928, Dec 2017. ISSN 1941-0026.

BIANCO, S.; CIOCCA, G.; SCHETTINI, R. How far can you get by combining change detection algorithms? In: BATTIATO, S.; GALLO, G.; SCHETTINI, R.; STANCO, F. (Ed.). **Image Analysis and Processing - ICIAP 2017**. Cham: Springer International Publishing, 2017. p. 96–107. ISBN 978-3-319-68560-1.

BOUWMANS, T. Traditional and recent approaches in background modeling for foreground detection: An overview. **Computer science review**, Elsevier, v. 11, p. 31–66, 2014.

CARRANZA, J.; THEOBALT, C.; MAGNOR, M. A.; SEIDEL, H.-P. Free-viewpoint video of human actors. **ACM transactions on graphics (TOG)**, ACM New York, NY, USA, v. 22, n. 3, p. 569–577, 2003.

CHEUNG, S.-C. S.; KAMATH, C. Robust background subtraction with foreground validation for urban traffic video. **EURASIP Journal on Advances in Signal Processing**, Springer, v. 2005, n. 14, p. 726261, 2005.

De Gregorio, M.; GIORDANO, M. Wisardrp for change detection in video sequences. In: **25th European Symposium on Artificial Neural Networks, Computational**

Intelligence and Machine Learning (ESANN 2017). [S.l.: s.n.], 2017. p. 453–458. ISSN 978-287587039-1.

ELGAMMAL, A. **Background Subtraction: Theory and Practice**. [S.l.]: Morgan & Claypool Publishers, 2014. 83 p. ISBN 1627054405, 9781627054409.

ELGAMMAL, A.; HARWOOD, D.; DAVIS, L. Non-parametric model for background subtraction. In: VERNON, D. (Ed.). **Computer Vision — ECCV 2000**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 751–767. ISBN 978-3-540-45053-5.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. [S.l.]: MIT press Cambridge, 2016.

GOYETTE, N.; JODOIN, P. M.; PORIKLI, F.; KONRAD, J.; ISHWAR, P. Changedetection.net: A new change detection benchmark dataset. In: **2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2012. p. 1–8. ISSN 2160-7508.

Gregorio, M. D.; Giordano, M. Change detection with weightless neural networks. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 409–413. ISSN 2160-7508.

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J. et al. Recent advances in convolutional neural networks. **Pattern Recognition**, Elsevier, v. 77, p. 354–377, 2018.

ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1125–1134.

Jiang, S.; Lu, X. Wesambe: A weight-sample-based method for background subtraction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 28, n. 9, p. 2105–2115, Sep. 2018.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LEE, S.-h.; LEE, G.-c.; YOO, J.; KWON, S. Wisenetmd: Motion detection using dynamic background region analysis. **Symmetry**, v. 11, n. 5, p. 1–15, 2019. ISSN 2073-8994.

LIANG, D.; KANEKO, S.; HASHIMOTO, M.; IWATA, K.; ZHAO, X. Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes. **Pattern Recognition**, v. 48, n. 4, p. 1374 – 1390, 2015. ISSN 0031-3203.

LIM, L. A.; KELES, H. Y. Foreground segmentation using convolutional neural networks for multiscale feature encoding. **Pattern Recognition Letters**, v. 112, p. 256 – 262, 2018. ISSN 0167-8655.

MADDALENA, L.; PETROSINO, A. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. **Neural Computing and Applications**, v. 19, n. 2, p. 179–186, Mar 2010. ISSN 1433-3058.

Maddalena, L.; Petrosino, A. The sobs algorithm: What are the limits? In: **2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2012. p. 21–26. ISSN 2160-7516.

MARTINS, I.; CARVALHO, P.; CORTE-REAL, L.; ALBA-CASTRO, J. L. Bmog: Boosted gaussian mixture model with controlled complexity. In: ALEXANDRE, L. A.; SÁNCHEZ, J. S.; RODRIGUES, J. M. F. (Ed.). **Pattern Recognition and Image Analysis**. Cham: Springer International Publishing, 2017. p. 50–57. ISBN 978-3-319-58838-4.

Miron, A.; Badii, A. Change detection based on graph cuts. In: **2015 International Conference on Systems, Signals and Image Processing (IWSSIP)**. [S.l.: s.n.], 2015. p. 273–276. ISSN 2157-8702.

RAMíREZ-ALONSO, G.; CHACóN-MURGUíA, M. I. Auto-adaptive parallel som architecture with a modular analysis for dynamic object segmentation in videos. **Neurocomputing**, v. 175, p. 990 – 1000, 2016. ISSN 0925-2312.

Sajid, H.; Cheung, S. S. Universal multimode background subtraction. **IEEE Transactions on Image Processing**, v. 26, n. 7, p. 3249–3260, July 2017. ISSN 1941-0042.

SANCHES, S. R. R.; OLIVEIRA, C.; SEMENTILLE, A. C.; FREIRE, V. Challenging situations for background subtraction algorithms. **Applied Intelligence**, v. 49, n. 5, p. 1771–1784, May 2019. ISSN 1573-7497.

SANCHES, S. R. R.; SAITO, P. T. M.; BUGATTI, P. H.; FREIRE, V. Identifying levels of difficulty in videos used to evaluate change detection algorithms. **Soft Computing**, Springer, v. 0, n. Under Review, p. 1–12, 2021.

SANCHES, S. R. R.; SEMENTILLE, A. C.; AGUILAR, I. A.; FREIRE, V. Recommendations for evaluating the performance of background subtraction algorithms for surveillance systems. **Multimedia Tools and Applications**, Springer, v. 80, n. 3, p. 4421–4454, 2021.

Sedky, M.; Moniri, M.; Chibelushi, C. C. Spectral-360: A physics-based technique for change detection. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 405–408. ISSN 2160-7508.

SENIOR, A. W.; TIAN, Y.; LU, M. Interactive motion analysis for video surveillance and long term scene monitoring. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2010. p. 164–174.

SILVA, C. M.; ROSA, K. A. I.; BUGATTI, P. H.; SAITO, P. T. M.; CORRêA, C. G.; YOKOYAMA, R. S.; SANCHES, S. R. R. Method for selecting representative videos for change detection datasets. **Multimedia Tools and Applications**, Springer, v. 0, n. 0, p. 1–14, 2021.

SILVA, C. M. da. **Avaliação de Datasets e de Algoritmos de Detecção de Mudança utilizando Mapas de Dificuldade**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2021.

SILVA, I. D.; SPATTI, D.; FLAUZINO, R. **Redes Neurais Artificiais para engenharia e ciências aplicadas: curso prático, Artliber Editora Ltda, São Paulo, SP, Brasil**. [S.l.], 2010.

SOBRAL, A.; VACAVANT, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. **Computer Vision and Image Understanding**, Elsevier Inc., v. 122, p. 4–21, 2014. ISSN 10773142.

St-Charles, P.; Bilodeau, G.; Bergevin, R. A self-adjusting approach to change detection based on background word consensus. In: **2015 IEEE Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2015. p. 990–997. ISSN 1550-5790.

St-Charles, P.; Bilodeau, G.; Bergevin, R. Subsense: A universal change detection method with local adaptive sensitivity. **IEEE Transactions on Image Processing**, v. 24, n. 1, p. 359–373, Jan 2015. ISSN 1941-0042.

Stauffer, C.; Grimson, W. E. L. Adaptive background mixture models for real-time tracking. In: **Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)**. [S.l.: s.n.], 1999. v. 2, p. 246–252 Vol. 2. ISSN 1063-6919.

TIAN, Y.; SENIOR, A.; LU, M. Robust and efficient foreground analysis in complex surveillance videos. **Machine vision and applications**, Springer, v. 23, n. 5, p. 967–983, 2012.

TYLEČEK, R.; ŠÁRA, R. Spatial pattern templates for recognition of objects with regular structure. In: **Proc. GCPR**. Saarbrucken, Germany: [s.n.], 2013.

UNIVERSITÉ DE SHERBROOKE. **Results for CD.net 2014**. 2019. http://jacarini.dinf.usherbrooke.ca/results2014/. Accessed 19 Mar 2020.

UNIVERSITÉ DE SHERBROOKE. **Change Detection 2014 - A video database for testing change detection algorithms**. 2021. https://www.kaggle.com/datasets/maamri95/cdnet2014. Accessed 18 Ago 2022.

Varadarajan, S.; Miller, P.; Zhou, H. Spatial mixture of gaussians for dynamic background modelling. In: **2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance**. [S.l.: s.n.], 2013. p. 63–68. ISSN null.

VARGHESE, A.; G, S. Sample-based integrated background subtraction and shadow detection. **IPSJ Transactions on Computer Vision and Applications**, v. 9, n. 1, p. 25, Dec 2017. ISSN 1882-6695.

Wang, R.; Bunyak, F.; Seetharaman, G.; Palaniappan, K. Static and moving object detection using flux tensor with split gaussian models. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2014. p. 420–424. ISSN 2160-7508.

WANG, Y.; JODOIN, P.-M.; PORIKLI, F.; KONRAD, J.; BENEZETH, Y.; ISHWAR, P. Cdnet 2014: An expanded change detection benchmark dataset. In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2014. p. 387–394.

Yingying Chen; Jinqiao Wang; Hanqing Lu. Learning sharable models for robust background subtraction. In: **2015 IEEE International Conference on Multimedia and Expo (ICME)**. [S.l.: s.n.], 2015. p. 1–6. ISSN 1945-788X.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 818–833.

ZHENG, W.; WANG, K.; WANG, F.-Y. A novel background subtraction algorithm based on parallel vision and bayesian gans. **Neurocomputing**, 2019. ISSN 0925-2312.

ZIJDENBOS, A. P.; DAWANT, B. M.; MARGOLIN, R. A.; PALMER, A. C. Morphometric analysis of white matter lesions in mr images: method and validation. **IEEE transactions on medical imaging**, IEEE, v. 13, n. 4, p. 716–724, 1994.

Zivkovic, Z. Improved adaptive gaussian mixture model for background subtraction. In: **Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.** [S.l.: s.n.], 2004. v. 2, p. 28–31 Vol.2. ISSN 1051-4651.

APÊNDICE A – CONFIGURAÇÕES DA REDE

Tabela 7: Configurações utilizadas para o treinamento da rede PIX2PIX

Configurações da rede	
batch_size	10
beta1	0.5
checkpoints_dir	./checkpoints
continue_train	True
crop_size	256
dataroot	./datasets/dataset_CdNet
dataset_mode	aligned
direction	BtoA
display_env	main
display_freq	400
display_id	1
display_ncols	4
display_port	8097
display_server	http://localhost
display_winsize	256
epoch	latest
epoch_count	28
gan_mode	vanilla
gpu_ids	0
init_gain	0.02
init_type	normal
input_nc	3
isTrain	True
lambda_L1	100.0
load_iter	0
load_size	286
lr	0.0002
lr_decay_iters	50
lr_policy	linear
max_dataset_size	inf
model	pix2pix
n_epochs	100
n_epochs_decay	100
n_layers_D	3

Configurações da rede	
name	dataset_CdNet_pix2pix
ndf	64
netD	basic
netG	unet_256
ngf	64
no_dropout	False
no_flip	False
no_html	False
norm	batch
num_threads	4
output_nc	3
phase	train
pool_size	0
preprocess	resize_and_crop
print_freq	100
save_by_iter	False
save_epoch_freq	5
save_latest_freq	5000
serial_batches	False
update_html_freq	1000
use_wandb	False
verbose	False

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PPGI
programa de pós-graduação
em informática

@ppgicp

www ppgi-cp@utfpr.edu.br
portal.utfpr.edu.br/cp/ppgi