

Avaliação do Uso de RNSP na Detecção de Mudanças em Imagem

L. M. VIANA, COPPE / UFRJ, 2019

Resumo—O objetivo deste trabalho é analisar o desempenho das Redes Neurais sem Peso Wisard na detecção de mudanças em imagens. Neste experimento, comparamos o desempenho de uma variação das Redes Neurais sem Peso Wisard (*CwisarDRP*), com as Redes Neurais Profundas (*Deep Learning*), usadas no desafio proposto pela “CDnet 2014 – Change Detection.Net”. Neste estudo, comparamos o modelo de Redes Neurais Profundas, “FgNetSeg v2 – Foreground Segmentation Network version 2”, que obteve 99,39% de média de acerto no desafio CDnet 2014, com a Rede Neural sem Peso *CwisarDRP* que obteve 78.80% de acerto no mesmo desafio.

Index Terms—Learning systems, Supervised learning, Machine learning, Neural Networks, Deep Learning, Weightless Neural Network.

I. INTRODUÇÃO

O objetivo deste trabalho foi avaliar o desempenho das Redes Neurais sem Peso Wisard na detecção de mudanças em imagens de vídeo, além de efetuar uma comparação entre estes modelos de redes neurais com as Redes Neurais Profundas (*Deep Learning*), analisando o desempenho obtido por estes modelos na análise do *dataset* proposto pela “CDnet 2014 – Change Detection.Net”.

II. CONCEITOS INICIAIS

A. Redes Neurais sem Peso Wisard

As Redes Neurais sem Peso Wisard procuram mapear os bits de uma imagem em um vetor de endereçamento de memória, com o objetivo de armazenar o aprendizado obtido durante a fase de treinamento da rede neural.

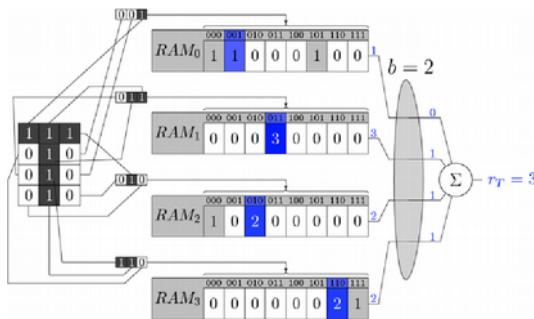


Figura 1: Modelo de mapeamento de Redes Neurais sem Peso Wisard

A Figura 1, apresenta o modelo de mapeamento de uma imagem em uma Rede Neural sem Peso Wisard. Neste exemplo, temos a imagem da letra “T”, representada em uma tela com 3x4 (12 pixels).

Para fazer o mapeamento da imagem na Rede Neural sem Peso Wisard, mapeamos de forma aleatória os pixels da imagem em um padrão de endereçamento. No exemplo, foi utilizado um endereçamento de 3 bits, permitindo que até 4 padrões de endereçamento com 2^3 RAMs x 8 bytes/RAM = 256 bytes, sejam usados para armazenar a frequência em que cada pixel da imagem é ativado na fase de treinamento de uma Rede Neural sem Peso Wisard.

Denominamos de “Discriminador”, o modelo Wisard aplicado na identificação de um padrão. No exemplo, um Discriminador é usado para o aprendizado e reconhecimento da letra “T”, e um outro Discriminador será usado no aprendizado e reconhecimento da letra “R”, e assim por diante.

B. Redes Neurais sem Peso *CwisarDRP*

Para o desafio CDnet 2014, foi implementado uma variação do modelo de Rede Neural sem Peso Wisard, denominado *CwisardDRP* (Figura 2)

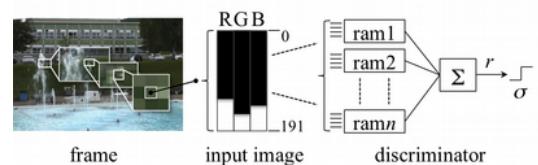


Figura 2: Modelo de mapeamento de Redes Neurais sem Peso *CwisardDRP*

Neste modelo, cada pixel da imagem é dividido em seu componente RGB, que classifica as intensidades de Vermelho, Verde e Azul de cada pixel da imagem em um padrão de endereçamento de 192 bits.

Para reduzir a memória necessária ao armazenamento do padrão de cada pixel da imagem, foram usadas 24 RAMs para cada pixel da imagem. Deste modo, foram necessários cerca de 9,5 milhões de RAMs para armazenar o padrão de uma imagem de 720 x 576 pixels.

C. Redes Neurais Profundas (*Deep Learning*)

As Redes Neurais Profundas (*Deep Learning*), são modelos de redes neurais fortemente conectadas que apresentam uma camada de entrada, duas ou mais camadas ocultas e uma camada de saída (*Figura 3*).

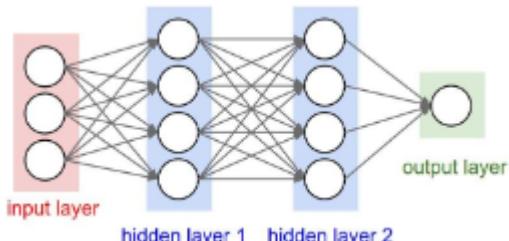


Figura 1: Modelo de redes neurais profundas

As Redes Neurais Convolucionais, são modelos de Redes Neurais Profundas largamente utilizadas na classificação de imagens e áudio. Este modelo de rede neural procura separar em cada camada as características que definem as imagens, permitindo a identificação dos dados. A *Figura 2*, apresenta o resultado de um modelo de Rede Neural Convolucional aplicada ao reconhecimento de imagens.



Figura 2: Modelo de redes neurais convolucionais

D. Redes Neurais Convolucionais

Ao analisarmos imagens de 256 cores usando redes neurais, podemos arrumar os elementos da imagem em um vetor. Desta forma, uma imagem de 28×28 pixels e 256 cores, seria representada por um vetor com 784 bytes, e utilizariamos na camada inicial da rede neural 784 entradas (*Figura 4*).

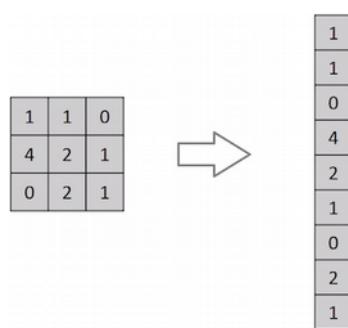


Figura 4: Organização dos pixels da imagem em vetor

Quando estudamos imagens com maior quantidade de cores, como imagens RGB, o tamanho do vetor necessário para representar a imagem é de 2352 bytes. Além disso, ao representar uma imagem por um vetor de dados, perdemos as características associadas a cada componente de cor, gerando um modelo com menor taxa de assertividade.

Desta forma, sugeriram as redes neurais convolucionais, que separam os dados de acordo com a profundidade de cor aplicada as imagens, assegurando um melhor desempenho na classificação das imagens (*Figura 5*).

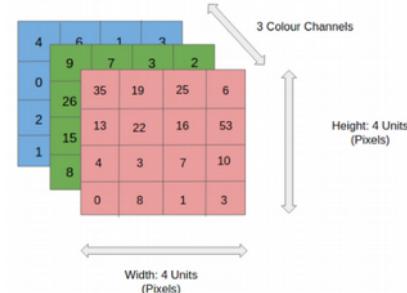


Figura 5: Organização dos pixels da imagem por canal de cor

E. Camadas Convolucionais (Kernel)

Cada camada convolucional do modelo de redes neurais implementa uma função de filtro denominada Kernel, que procura extrair os detalhes da imagem. Esta operação é realizada na imagem e reduz a quantidade de informações para um número específico de elementos. A *Figura 5* apresenta o processo de filtro aplicado por uma camada convolucional.

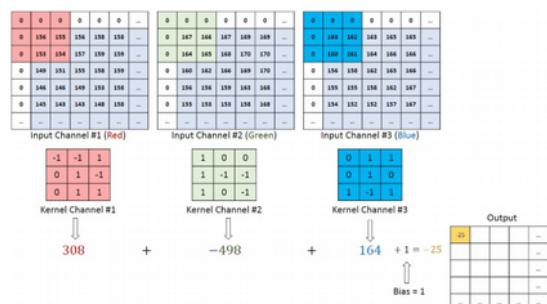


Figura 5: Processo de filtro aplicado pela camada de convolução

O filtro é aplicado em cada canal de cor, e o resultado obtido da camada de convolução é uma matriz com um número reduzido de elementos. No processo de convolução, as operações mais comuns aplicadas pelos filtros, são operações de identificação de valores médios ou máximos da camada de entrada.

III. EXPERIMENTOS

Neste experimento, comparamos o desempenho do modelo de Rede Neural Profunda, “FgNetSeg v2 – Foreground Segmentation Network version 2” com o modelo de Rede Neural sem Peso CwisarDRP no processo de identificação de mudanças em imagens de vídeo.

A. Detecção de Mudança em Imagens de Vídeo

O processo de identificação de mudanças em imagens, é um processamento fundamental na análise automatizada de imagens de vídeo. Com o objetivo de melhorar os processos de identificação de mudanças em imagens de vídeo, foi criado em 2012 e repetido em 2014, um desafio com o objetivo de envolver os pesquisadores no processo de desenvolvimento de melhorias neste tipo de análise.

Na edição CDnet 2014, foi proposta uma solução de detecção e mudança utilizando uma variação da Rede Neural sem Peso Wisard (CwisarDRP), que obteve excelente resultado.

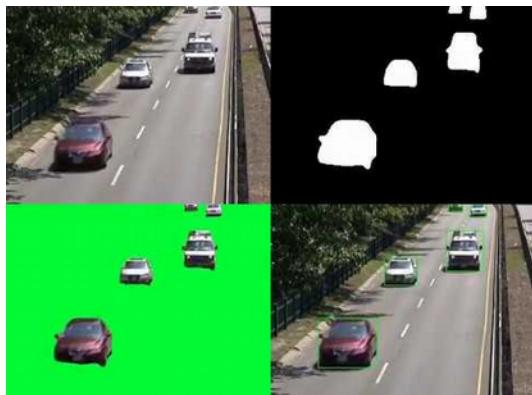


Figura 6: Processo de detecção de mudanças em imagens de vídeo

A Figura 6, apresenta o processo de detecção de mudanças em imagens de vídeo. Neste processo, é realizado a identificação das mudanças nas imagens e gerado a imagem binária em preto e branco. Em seguida, é realizada a combinação da imagem binária em preto e branco com a imagem real para separar os objetos em movimento do fundo da imagem. Depois, é feita a classificação dos objetos em movimento.

Neste experimento, analisamos duas metodologias utilizadas no desafio CDnet 2014 (Tabela 1).

Métodologia FgSegNet_v2: Foreground Segmentation Network version 2
Autores Long Ang Lim, Ankara University Hacer Yalim Keles, Ankara University
Código Fonte: https://github.com/lim-anggun/FgSegNet_v2
Metodologia CwisarDRP: Change Detection with Weightless Neural Networks
Autores: Massimo De Gregorio, Istituto di Cibernetica “E. Caianiello” Maurizio Giordano, Istituto di Calcolo e Retiad Alte Prestazioni
Código Fonte: https://github.com/giordamaug/CwisarDRP

Tabela 1: Metodologias usadas no desafio CDnet 2014.

O dataset apresentado no desafio CDnet 2014 contém 11 categorias de vídeos com 4 a 6 sequências de vídeos em cada categoria (Tabela 2). Cada sequência de vídeo está separada em um diretório contendo as imagens JPEG de cada frame do vídeo. O arquivo "temporalROI.txt" contém os índices dos frames inicial e final da faixa de vídeo utilizada no cálculo do resultado.

Baseline	highway, pedestrians, office, PETS2006
Camera Jitter	badminton, traffic, boulevard, sidewalk
Bad Weather	skating, blizzard, snowFall, wetSnow
Dynamic Background	boats, canoe, fall, fountain01, fountain02, overpass
Intermittent Object Motion	abandonedBox, parking, sofa, streetLight, tramstop, winterDriveway
Low Framerate	port_0_17fps, tramCrossroad_1fps, tunnelExit_0_35fps, turnpike_0_5fps
Night Videos	bridgeEntry, busyBoulevard, fluidHighway, streetCornerAtNight, tramStation, winterStreet
PTZ	continuousPan, intermittentPan, twoPositionPTZCam, zoomInZoomOut
Shadow	backdoor, bungalows, busStation, copyMachine, cubicle, peopleInShade
Thermal	corridor, diningRoom, lakeSide, library, park
Turbulence	turbulence0, turbulence1, turbulence2, turbulence3

Tabela 2: Conteúdo do dataset apresentado no desafio CDnet 2014.

A Figura 7, apresenta dois exemplos de sequências de vídeos presentes do dataset do desafio CDnet 2014. Na sequência superior temos três imagens obtidas da categoria Bad Weather – Skating, e na sequência inferior, nós temos três imagens obtidas da categoria Thermal – Dining Room.



Figura 7: Exemplos de imagens contidas no dataset CDnet 2014

B. Experimentos com o modelo FgSegNet v2

O modelo “FgSegNet v2: Foreground Segmentation Network version 2”, foi implementação em Python usando Keras e Tensorflow-gpu. Duas bibliotecas usadas na implementação de modelos de Rede Neural Convolucional Profunda (*Deep Learning*), cujo modelo aplicado foi otimizado para identificação do Background (*Figura 8*).

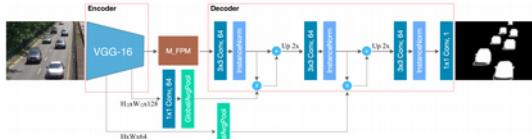


Figura 8: Modelo FgNetSeg v2 usado no desafio CDnet 2014

Este modelo é composto por uma rede neural inicial, baseada no modelo VGG-16, que identifica diferentes níveis de *features* da imagem (64, 128, 256 e 512 *features*). Em seguida, a imagem passa por um segundo modelo de rede neural, M_FPM, que efetua uma análise com base em diferentes escala da imagem. Na última etapa do processo os resultados obtidos passam através de uma terceira rede neural que decodifica os dados e produz a imagem binária em preto e branco.

Para obter o melhor resultado no desafio CDnet 2014, foi utilizado uma base de treinamento com 200 *frames*, e foram realizadas 1000 épocas de treinamento para gerar um modelo de rede neural com grande precisão. O modelo gerado foi usado na avaliação das imagens de teste com excelente resultado na classificação (99,39%).

Porém, um modelo como este demora muito tempo para processar 1000 épocas de treinamento analisando 200 *frames* de vídeo, e nos experimentos que realizei, pude verificar que com apenas 1 época de treinamento e 25 *frames* obtidos aleatoriamente os resultados ainda não são muito bons, mas após 10 épocas de treinamento e usando os mesmos 25 *frames*, o resultado já é superior a 90%.



Figura 9: Resultado do modelo FgNetSeg v2 com Bad Weather - Skating

Na *Figura 9*, podemos observar o resultado obtido com o modelo FgSegNet v2 para a sequência de imagens de vídeo da categoria *Bad Weather – Skating*.

Categories	Avg Precision
BadWeather	99.39
LowFramerate	90.05
NightVideos	98.38
PTZ	98.49
Turbulence	97.27
Baseline	99.78
DynamicBackground	99.34
CameraJitter	99.79
IntermitentObjectMotion	99.55
Shadom	99.52
Thermal	99.49
Overall	98.23

Tabela 3: Resultado do modelo FgSegNet v2 no desafio CDnet 2014.

No desafio CDnet 2014 o modelo FgSegNet v2 obteve o melhor resultado em todas as categorias, obtendo uma média geral de 98,23% (*Tabela 3*). Porém, o elevado tempo de processamento e o fato deste algorítimo ter sido fortemente adaptado a estrutura dos dados do *dataset* com o objetivo de alcançar o melhor resultado no desafio, este algorítimo não pode ser facilmente aplicado em situações de análise de vídeo com detecção de mudanças em tempo real.

C. Experimentos com o modelo CwisarDRP

O modelo CwisarDRP implementa uma variação da Rede Neural sem Peso Wisard, onde cada pixel da imagem é separado nas componentes de cores RGB, com intensidades de Vermelho, Verde e Azul, usando 3 Discriminadores de 192 bits de endereçamento para cada pixel da imagem.

Para reduzir a quantidade de RAMs, cada Discriminador mapeia apenas 24 RAMs, reduzindo a quantidade de RAMs para 9,5 milhões em uma imagem de 720 x 576 pixels.

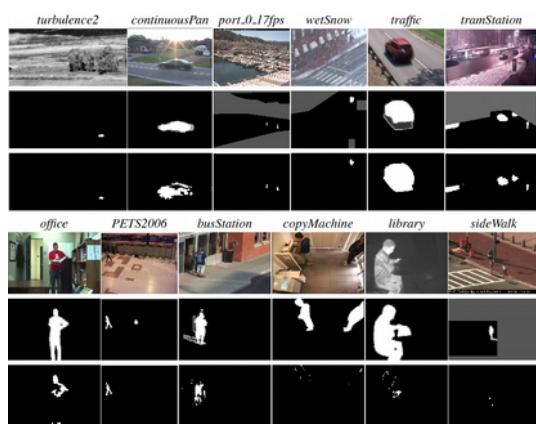
A implementação deste modelo foi realizada em C/C++ e usa as bibliotecas OpenCL e OpenMP para obter melhor desempenho no processamento das imagens com o uso de GPUs e do processamento multicore das CPUs atuais.

Neste modelo, o aprendizado é realizado continuamente, e durante todo o processamento das imagens. Podemos verificar que logo após os primeiros 50 *frames* já obtemos um excelente resultado. Porém, após muito tempo de exposição das imagens os Discriminadores começam a ficar muito poluídos, reduzindo a precisão do modelo. Desta forma, verificamos a necessidade de usar mais Discriminadores por pixel ou implementar um mecanismo de esquecimento das imagens antigas, para manter constante a assertividade do modelo.



Figura 10: Resultado do modelo CwisarDRP com Bad Weather - Skating

Na *Figura 10*, podemos observar o resultado obtido com o modelo CwisarDRP para a sequencia de imagens de vídeo da categoria *Bad Weather – Skating*.

Figura 11: Resultado do modelo CwisarDRP com o *dataset* CDnet 2014

A *Figura 11*, apresenta o resultado do modelo CwisarDRP para o *dataset* CD net 2014. Podemos observar o bom resultado deste modelo na *Tabela 4*.

Categories	Avg Precision
BadWeather	87.18
LowFramerate	70.45
NightVideos	54.47
PTZ	32.00
Turbulence	92.73
Baseline	93.47
DynamicBackground	87.23
CameraJitter	87.13
IntermitentObjectMotion	85.43
Shadom	85.51
Thermal	91.16
Overall	78.80

Tabela 4: Resultado do modelo CwisarDRP no desafio CDnet 2014.

No desafio CDnet 2014 o modelo CwisarDRP obteve um bom resultado, com uma média geral de 78,80% (*Tabela 4*). Na categoria *Baseline*, este modelo alcançou 93,47% de assertividade.]

Este modelo também apresentou um excelente tempo de processamento, realizando o aprendizado e análises das imagens em tempo real, abrindo oportunidade de uso deste modelo em aplicações que precisam de maior rapidez no aprendizado e classificação dos dados.

IV. CONCLUSÃO

O modelo FgSegNet_v2 foi implementado usando algoritmos e técnicas bastante utilizadas no processamento de imagens, e na detecção de mudanças e reconhecimento de *background* em imagens de vídeo (Keras e Tensorflow-gpu).

Este algoritmo obteve excelente resultado com o *dataset* CDnet 2014, alcançando 98,23% de precisão. Porém, este resultado foi obtido usando uma base de treinamento previamente concebida e um modelo otimizado para os dados analisados.

Além disso, para obter um elevado grau de precisão, o modelo foi treinado inicialmente usando 200 *frames* e 1000 épocas de treinamento para cada sequencia de vídeo do *dataset* CDnet 2014.

Nesta abordagem encontramos algumas barreiras, como a dificuldade prática de obter treinamento prévio de imagens de vídeo, e o tempo de treinamento necessário para obter o resultado apresentado no desafio CDnet 2014, de 98,23% de média geral.

Deste modo, apesar do resultado da CwisarDRP no desafio CDnet 2014 ter sido de apenas 78,80% de assertividade média, este é um algoritmo simples para identificação de mudanças e reconhecimento de *background* em imagens de vídeo, que possui um aprendizado rápido e contínuo, e que se adapta as mudanças que ocorrem nas sequencias de imagens dos vídeos.

Porém, o modelo CwisarDRP ainda possui alguns desafios para melhorar a precisão na detecção de mudanças, e podemos sugerir incialmente que seja realizado a avaliação do desempenho deste algoritmo com o uso de uma base de treinamento inicial, para permitir o treinamento prévio do modelo.

Devemos avaliar também técnicas de esquecimento das imagens antigas, e o uso de mais de um Discriminador por pixel, com o objetivo de tentar minimizar o problema de poluição dos Discriminadores com a exposição das imagens.

REFERENCIAS

- [1] Y. S. Abu-Mostafa, M. Magdon-Ismail, H. Lin Authors, “e-Chapter 7 – Neural Networks” in Learning From Data a Short Course, Country: Pasadena, CA, USA, 2012, ch. 7.
- [2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011, https://scikit-learn.org/stable/user_guide.html
- [3] Keras: The Python Deep Learning library, F. Chollet et al, 2015, <https://keras.io/>
- [4] TensorFlow, J. Dean, G. Corrado, A. Ng., 2011, <https://www.tensorflow.org/>
- [5] Y. LeCun, “Learning Process in an Asymmetric Threshold Network”, 1986
- [6] Y. LeCun, “A Theoretical Framework for Back-Propagation”, 1988
- [7] D. Eigen, J. Rolfe, R. Fergus, Y. LeCun, “Understanding Deep Architectures using a Recursive Convolutional Network”, 2014
- [8] M. Gregorio, M. Giordano, “WiSARD rp for Change Detection in Video Sequences”, 2017
- [9] L. A. Lim, H. Y. Keles, “Learning Multi-scale Features for Foreground Segmentation”, 2018
- [10] M. Gregorio, M. Giordano, ‘Change Detection with Weightless Neural Networks”, 2014
- [11] CDnet: ChangeDetection.Net, 2012-2014, <http://changedetection.net/>