FICE 8ª A FEIRA DE INICIAÇÃO CIENTÍFICA E EXTENSÃO 05 E 06 DE SETEMBRO

DETECÇÃO DE ANOMALIAS EM VÍDEOS UTILIZANDO AUTOCODIFICADORES CONVOLUCIONAIS

Letícia K Moreira[1] *; Manassés Ribeiro*[2]

INTRODUÇÃO

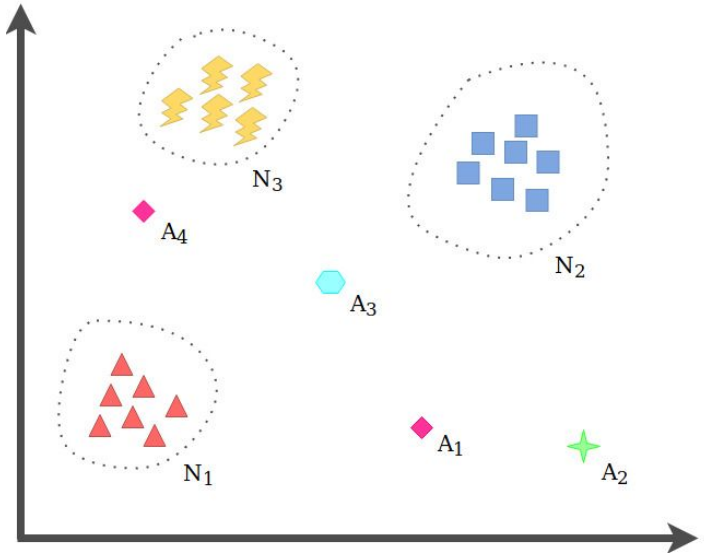
A detecção de anomalias é a tarefa de classificar dados que não estejam em acordo com um determinado contexto "normal". Conceitualmente, o contexto normal é formado por amostras de dados que sejam conhecidas e que ocorrem com certa frequência. Por outro lado, as amostras que raramente ocorrem, ou que nunca tenham ocorrido em um determinado contexto, são consideradas amostras "anormais". A detecção de anomalias é uma abordagem tipicamente usada quando a quantidade de dados disponíveis para a classe negativa é insuficiente para a construção de modelos de Reconhecimento de Padrões (RP) tradicionais, que exigem quantidade suficiente de amostras de ambas as classes normal e anormal (PIMENTEL et al., 2014).

A detecção de anomalias pode ser abordada como um problema de classificação de uma classe (que em inglês é conhecido por (*one-class classification*). Os termos detecção de anomalias e classificação de uma classe são tratados muitas vezes na literatura como sendo sinônimos, de modo que neste trabalho optou-se pelo uso do termo *detecção de anomalias* como padrão. Nesta classe de problemas um modelo computacional é construído para descrever os dados considerados normais e, tipicamente, a classificação é dada pelo distanciamento entre a amostra analisada e o contexto considerado normal. A Figura 1 apresenta um exemplo didático de um problema clássico de detecção de anomalia onde os conjuntos N_1, N_2 e N_3 representam conceitos normais e A_1, A_2, A_3 e A_4 representam anomalias, respectivamente.

[1] Aluna do Instituto Federal Catarinense, Campus Videira. Curso de Bacharelado em Ciência da Computação. E-mail: leticia.moreira@yandexl.com

[2] Professor Orientador do Instituto Federal Catarinense, Campus Videira. Curso de Bacharelado em Ciência da Computação. E-mail: manasses.ribeiro@ifc.edu.br

Figura 1: Exemplo didático de detecção de anomalias, onde *N* são normais e *A* anormais.



Fonte: Autoria própria

Uma das área de grande aplicação dos métodos de detecção de anomalias é na Visão Computacional (VC). No contexto de VC, a detecção de anomalias pode ser aplicada para a resolução de problemas envolvendo imagens ou vídeos que dependam da análise humana para classificação, como por exemplo, em vídeos de segurança e vigilância. Já os métodos de aprendizado profundo, por sua vez, têm cada vez mais alcançado resultados otimistas na literatura recente, em especial as Redes Neurais Convolucionais (RNC) (PERLIN e LOPES, 2016). A desvantagem das RNC está no fato de que a otimização destas redes é supervisionada, ou seja, é necessário que todas as classes de um contexto sejam conhecidas para que o treinamento possa ocorrer. Como em problemas de detecção de anomalias no mundo real nem sempre é possível obter amostras de ambas as classes (normal e anormal), as RNC tornam-se inviáveis para a modelagem de problemas onde apenas uma classe é conhecida (a classe normal) (RIBEIRO et al., 2018).

Por outro lado, os Autocodificadores (AC), que também são métodos de aprendizado profundo, podem ser aprendidos por meio de treinamento não supervisionado. Para este tipo de treinamento não é necessário o conhecimento prévio de ambas as classes, da mesma forma que não necessita que as amostras estejam anotadas. Os métodos de treinamento não supervisionados são especialmente interessantes para problemas de detecção

de anomalias, uma vez que nestes problemas apenas a classe normal é amplamente conhecida. Contudo, uma limitação dos ACs tradicionais, para uso envolvendo o contexto de imagens e vídeos, é o fato de que sua estrutura é toda voltada para trabalhar com dados unidimensionais (1D). Como imagens (e os vídeos que são formado por um conjunto de imagens) são bidimensionais (2D), os ACs tradicionais tornam-se desinteressantes por desprezarem informações de orientação espacial. Entretanto, assim como a RNC é a versão convolucional das redes neurais, os Autocodificadores Convolucionais (ACC) são a versão convolucional dos ACs. A grande vantagem dos métodos convolucionais é sua capacidade de capturar as características espaciais das imagens e vídeos (MASCI et al., 2011).

Os ACCs são organizados em camadas de codificação e decodificação, que permitem uma representação latente (representação de baixa dimensão) dos dados de entrada em sua camada intermediária (camada do meio da arquitetura). Como os ACC não necessitam de informação de rótulo das amostras de entrada, podem ser úteis na modelagem de problemas de detecção de anomalias (RIBEIRO et al., 2018). O uso de ACCs para problemas de detecção de anomalias em vídeo vêm sendo bastante explorado na literatura recente, como por exemplo pode ser visto em Ribeiro et al. (2018), Guo et al. (2017), Hasan et al. (2016) e Masci et al. (2011). Neste sentido, o problema abordado neste trabalho consiste no uso do ACC como ferramenta para a detecção de anomalias em vídeos.

OBJETIVOS

Objetivo Geral: Estudar o uso do autocodificador convolucional para a detecção de anomalias em vídeos.

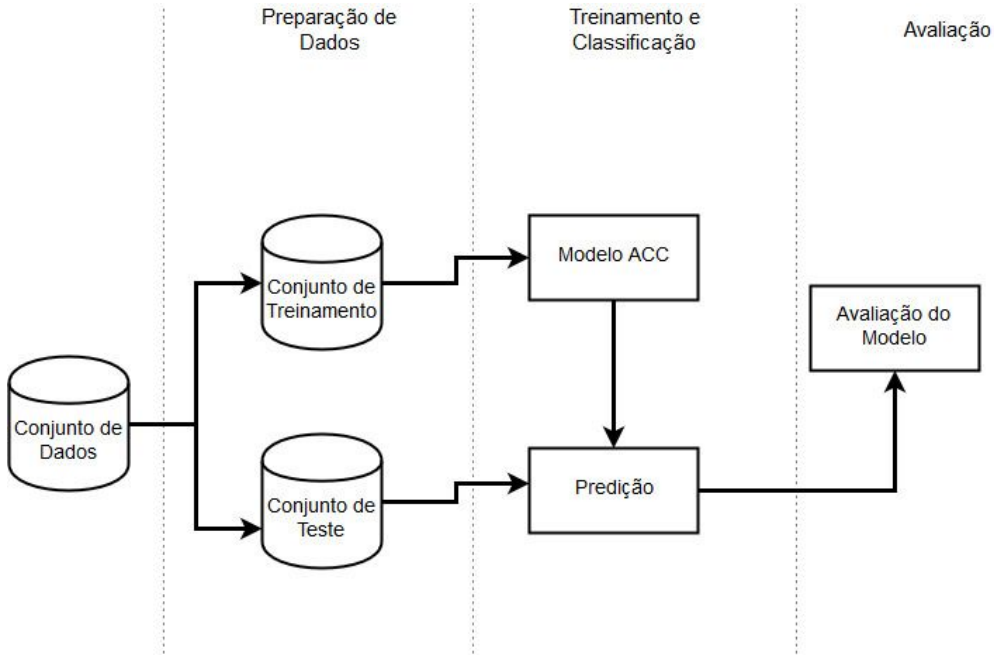
Objetivos Específicos

1. Sugerir uma metodologia para abordar o problema de detecção de anomalias em vídeos utilizando um autocodificador convolucional;
2. Verificar o uso do erro de reconstrução como pontuação para a detecção de anomalias;
3. Realizar estudo de caso em um conjunto de dados disponível publicamente.

PROCEDIMENTOS METODOLÓGICOS

Nesta seção serão apresentados os métodos sugeridos para contornar o problema proposto neste trabalho. Os métodos sugerido são baseados nos trabalhos de Ribeiro et al. (2018) e Hasan et al. (2016), e são dividido em etapas que seguem desde a preparação dos dados até a avaliação do desempenho de classificação do modelo. A visão geral do método é apresentada na Figura 2, que é composta por cada uma das partes que serão detalhadas na sequência.

Figura 2: Visão geral do Método



Fonte: Autoria própria

A primeira etapa do método é a preparação dos dados, que consiste em converter as imagens, se coloridas, para escala de cinza, redimensionar o tamanho das imagens, padronizar a escala dos dados em valores de mínimos e máximos e preparar os conjuntos de treinamento e teste. Os conjuntos de dados são formados por vídeos, portanto o primeiro passo é discretizar os vídeos extraindo os quadros (frames), que são as imagens dos vídeos. Após o processo de discretização todas as imagens do vídeo são convertidas para escala de cinza, redimensionadas para 236 × 156 e escaladas no intervalo [0..1]. Para o subconjunto de

treinamento em problemas de detecção de anomalias são utilizados apenas amostras consideradas normais. Para o subconjunto de teste, amostras de ambas as classes, normais e anormais, devem estar presentes, assim como a rotulação direta de todas as amostras.

A arquitetura sugerida para o ACC é composta por 35 camadas no total, sendo 17 de codificação, 17 de decodificação e a camada da representação latente (*bottleneck*) que fica no meio da arquitetura. Tanto na parte de codificação, quanto de decodificação, as três últimas camadas são totalmente conectadas (*fully-connected*), assim como a camada latente. As camadas de entrada e saída possuem dimensões iguais. Os detalhes das camadas e suas configurações são apresentadas na Tabela 1.

Codificador			Decodificador		
Camada	Filtro	Dimensão	Camada	Filtro	Dimensão
Entrada	-	236 x 156 x 1	Densa-Reversa 6	-	168
Convolução 1	3 x 3	236 x 156 x 128	Densa-Reversa 5	-	504
ReLU 1	-	236 x 156 x 128	Densa-Reversa 4	-	2016
BatchNormalization 1	-	236 x 156 x 128	Unpooling 3	3 x 3	63 x 39 x 32
Pooling 1	2 x 2	118 x 78 x 128	Cropping2D 3	2 x 0	59 x 39 x 32
Convolução 2	3 x 3	118 x 78 x 64	BatchNormalization-Reverso 3	-	59 x 39 x 32
ReLU 2	-	118 x 78 x 64	ReLU-Reverso 3	-	59 x 39 x 32
BatchNormalization 2	-	118 x 78 x 64	Deconvolução 3	3 x 3	59 x 39 x 64
Pooling 2	2 x 2	59 x 39 x 64	Unpooling 2	2 x 2	118 x 78 x 64
Convolução 3	3 x 3	59 x 39 x 32	BatchNormalization-Reverso 2	-	118 x 78 x 64
ReLU 3	-	59 x 39 x 32	ReLU-Reverso 2	-	118 x 78 x 64
BatchNormalization 3	-	59 x 39 x 32	Deconvolução 2	3 x 3	118 x 78 x 128
ZeroPadding 1	2 x 0	63 x 39 x 32	Unpooling 1	2 x 2	236 x 156 x 128
Pooling 3	3 x 3	21 x 13 x 32	BatchNormalization-Reverso 1	-	236 x 156 x 128
Densa 4	-	2016	ReLu-Reverso 1	-	236 x 156 x 128
Densa 5	-	504	Deconvolução 1	3 x 3	236 x 156 x 1
Densa 6	-	168	Saída		236 x 156 x 1
Latente	-	50			

Tabela 1: Detalhes das camadas da arquitetura do ACC

O modelo é otimizado com uma variação do método de erro médio quadrático, denominado Erro Médio Quadrático Logaritmo (EMQL), que é calculado entre os valores da entrada e saída da arquitetura. Os dados originais são inseridos na camada de entrada do ACC, e propagam-se pela arquitetura de camada em camada, seguindo o fluxo da esquerda para a direita. As camadas do codificador tem como objetivo reduzir o tamanho da entrada, na camada latente são representadas das características da imagem de entrada, em dimensionalidade reduzida, e nas camadas do decodificador os dados são reconstruídos. O EMQL é calculado de acordo com a transformação logarítmica dos valores reconstruídos na camada de saída e seus valores originais da camada de entrada e determina a capacidade de

aprendizado do modelo. Quanto menor for o valor do EMQL, significa que melhor otimizado o modelo está, sendo de mesma forma, um valor de EMQL alto, significa que o modelo não está otimizado de maneira satisfatória. Os parâmetros da rede neural são ajustados utilizando o método de retropropagação de erro (*backpropagation*) (RUMELHART et al., 1986).

Após a otimização do modelo é realizado o processo de classificação utilizando o subconjunto de dados de teste, os quais não participaram do processo de treinamento. Esse subconjunto de dados possui amostras tanto normais como anormais. Para a classificação é sugerido o uso do erro de reconstrução (RIBEIRO et al., 2018; HASAN et al., 2016) (no caso deste trabalho o EMQL) para gerar uma pontuação (*score*) que possa servir como parâmetro de classificação para cada uma das imagens do subconjunto de teste. A premissa é que, se uma amostra do subconjunto de testes for do tipo normal, o EMQL será baixo e consequentemente será reconhecida pelo modelo, enquanto se a amostra for anormal o EMQL será maior. Neste trabalho é utilizado o erro de reconstrução normalizado NRE(t) (RIBEIRO et. al (2018), que é composto pelo EMQL escalado no intervalo [0..1] e suavizado por uma média móvel, para a quadro *t* discretizado do vídeo. Assim, para valores de NRE(t) próximo de 0 as amostras são consideradas normais e para valores próximo de 1 para amostras anormais.

Para que a classificação seja de fato realizada, é necessário definir-se o valor do limiar de classificação que possa discriminar as imagens entre normais e anormais. Os valores de NRE(t) abaixo do limiar de classificação são classificados como normais, enquanto que valores acima do limiar de classificação são classificados como anormais. A definição do limiar de classificação não é uma questão trivial em problemas de detecção de anomalias e neste trabalho optou-se por utilizar um limiar de classificação definido a *posteriori*, que é obtido por meio do EER na curva ROC. A curva ROC, por sua vez, é construída utilizando os valores dos NREs calculadas para cada um dos quadros do vídeo do conjunto de teste.

Por fim, é realizada a avaliação geral de desempenho de classificação do modelo, que é realizada utilizando ambas as Taxas de Verdadeiros Positivos (TVP), conhecido como sensibilidade, e de Verdadeiros Negativos (TVN), conhecida como especificidade e a área abaixo da curva ROC (AUC). Enquanto que o TVP opera sobre a taxa de acertos das

amostras interpretadas como normais, o TVN opera sobre a taxa de acertos das amostras interpretadas como anormais.

EXPERIMENTOS, RESULTADOS E DISCUSSÕES

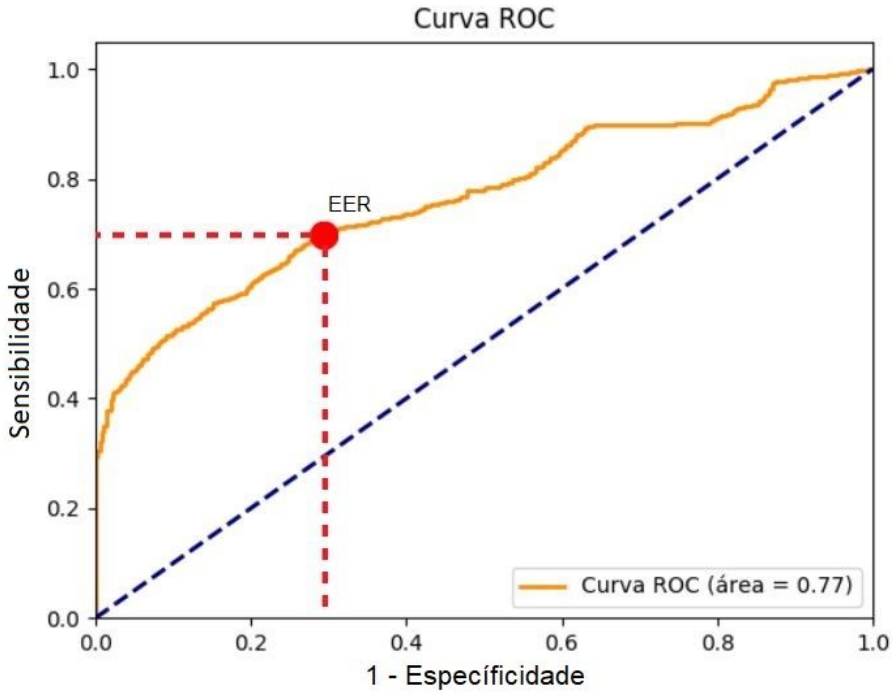
Para os experimentos propostos neste trabalho foi utilizado o UCSD-PED2 (MAHADEVAN et al., 2010), que é conjunto de dados de vídeo disponibilizado publicamente desenvolvido especificamente para detecção de anomalias. O UCSD-PED2 é composto por vídeos de pedestres se locomovendo paralelamente à câmera de vigilância e possui 16 videoclipes de treinamento e 12 videoclipes de teste, onde cada videoclipe possui cerca de 200 quadros. Assim, após discretizados os videoclipes, estão disponíveis aproximadamente 3200 imagens para treinamento e 2400 imagens para teste. As imagens, após discretizadas, foram redimensionados para o tamanho 236 x 156. Por se tratar de um problema de detecção de anomalias, o treinamento foi realizado apenas sobre o subconjunto de dados de treinamento, o qual contém apenas amostras normais. Para a implementação do modelo, foi utilizado a biblioteca Keras (CHOLLET, 2015), versão 2.2.4, rodando sobre o *framework* Tensorflow (ABADI et al, 2015), versão 1.14.

Para a otimização foram utilizadas 1000+1 épocas de treinamento, sendo que durante a otimização uma parte do conjunto de treinamento é separada para validação do modelo. O processo de validação é importante passo para verificar se o modelo está otimizando sem que ocorra o sobreajustamento (*overfitting*). O sobreajustamento ocorre quando o modelo fica muito ajustado para os dados de treinamento mas apresenta desempenho ruim na classificação do teste.

A definição o limiar de classificação para problemas de classificação em detecção de anomalias não é um processo trivial, uma vez que apenas as amostras da classe positiva estão presentes. Portanto, como primeiro estudo é sugerido este experimento para encontrar um limiar de classificação que possa ser acessado de maneira automatizada e também verificar a capacidade de classificação deste limiar. Neste estudo é utilizado o valor do EER, que é acessado a partir da curva ROC, como limiar de classificação. A curva ROC é construída com base nos valores dos NREs calculados para cada uma das imagens (quadros

do vídeo) do conjunto de teste. Nesta abordagem a definição do limiar acontece (*a posteriori*). Pelo gráfico da curva ROC, que é mostrada na Figura 3, é possível verificar os melhores valores de TVP e TNV acessados pelo EER. Após a classificação utilizando o limiar acessado pelo EER, os valores de TVP e TNV foram **0,7044** e **0,7032**, respectivamente. Os valores da matriz de confusão gerados foram **1.159**, **107**, **255**, e **489** para verdadeiros positivo, falsos positivo, verdadeiros negativo e falsos negativo, respectivamente.

Figura 3: Curva ROC

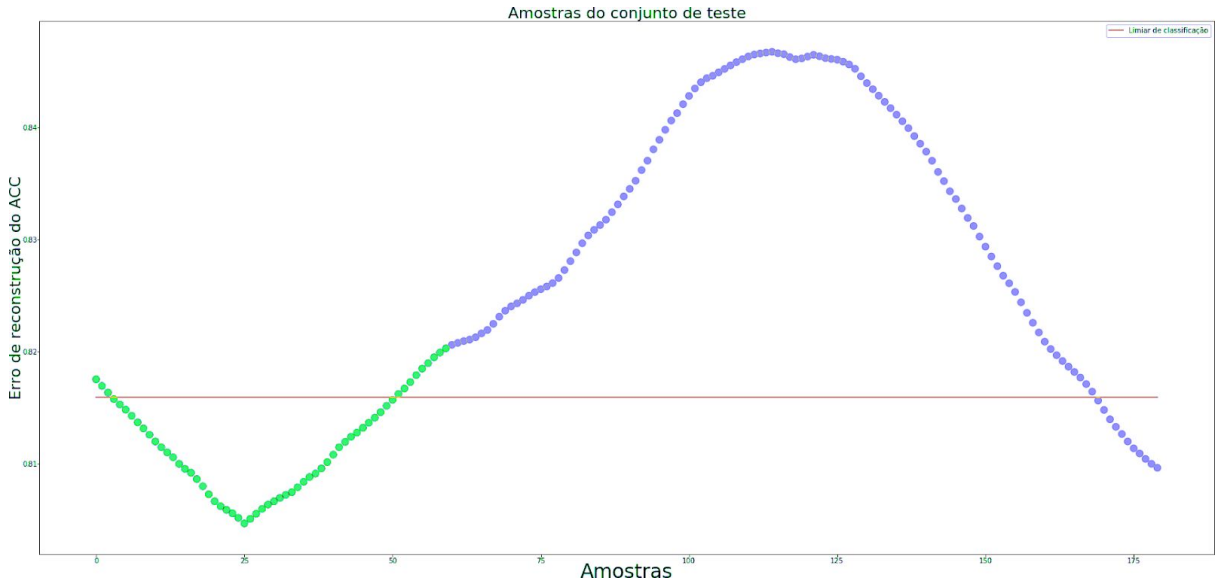


Fonte: Autoria própria.

Como segundo experimento foi inspecionado visualmente o desempenho de classificação do modelo sugerido. Para isto foi gerado gráfico plotando os pontos obtidos com o erro de reconstrução normalizado NRE calculado para cada imagem do vídeo. Para as amostras consideradas normais o NRE é plotado na cor lilás, enquanto o NRE para as amostras consideradas anormais são plotadas em verde e o limiar de classificação é plotado na cor vermelha. A Figura 4 mostra a análise visual para o vídeo 7, onde é possível notar que todas as amostras normais foram de fato classificadas como normais. No entanto, quando ocorreu a transição entre as amostras normais e anormais (por volta dos quadros 45 a 65), os quadros anormais foram classificados como normais. Estes resultados sugerem que o modelo

pode apresentar algumas dificuldades na classificação de amostras que se encontram nas transições entre normais e anormais. Como estas transições acontecem suavemente o modelo leva um certo tempo até conseguir capturar corretamente a diferenciação do movimento.

Figura 4: NRE plotados para os quadros normais e anormais do vídeo 7 do conjunto de testes
Fonte: Autoria própria



Fonte: Autoria Própria

CONCLUSÃO

Neste artigo fora apresentado métodos para a detecção de anomalias em vídeos utilizando ACCs. A literatura acerca do problema foi revisitada e utilizando como base alguns dos trabalhos relevantes da área, sugeriu-se um modelo para abordar o problema detecção de anomalias em vídeos. A proposta sugerida inclui um meio para realizar a classificação das amostras utilizando o próprio erro de reconstrução do ACC, bem como uma maneira de definição do limiar de classificação acessando o EER da curva ROC. Por fim, experimentos utilizando o conjunto de dados UCSD-PED2 foram realizados e os resultados foram analisados e discutidos. No primeiro experimento levou-se em consideração a classificação das amostras utilizando como limiar de classificação o valor acessado pelo EER da curva ROC, que foi obtido por meio do erro de reconstrução do ACC. No segundo

experimento foram analisados o comportamento de classificação em um dos vídeos do conjunto de teste.

Como possíveis trabalhos futuros, pode-se apontar o uso de diferentes parâmetros para otimização e treinamento do modelo e modificações no tamanho e tipos de camadas que o compõe. Também pode-se direcionar o estudo para um meio de definir o limiar de classificação considerando que em problemas do mundo real apenas estão disponíveis, em quantidade, amostras normais.

AGRADECIMENTOS

A autora L. K. Moreira agradece ao Instituto Federal Catarinense e o CNPq pela bolsa de estudos. Todos os autores agradecem à NVIDIA pela doação da GP-GPU utilizada neste trabalho.

REFERÊNCIAS

ABADI, M. et al. **TensorFlow: Large-scale machine learning on heterogeneous systems, 2015**. Software available from tensorflow.org.

CHOLLET, F. (2015) **keras, GitHub**. https://github.com/fchollet/keras

GUO, Xifeng \& LIU, Xinwang \& ZHU, En \& YIN, Jianping. (2017). **Deep Clustering with Convolutional Autoencoders.** 373-382. 10.1007/978-3-319-70096-039.

HASAN, M. et al. **Learning temporal regularity in video sequences. In: Conference on Computer Vision and Pattern Recognition.** Piscataway, NJ: IEEE, 2016. p. 733–742.

MAHADEVAN, V. et al. **Anomaly detection in crowded scenes. In: Conference on Computer Vision and Pattern Recognition.** Piscataway, NJ: IEEE, 2010. p. 1975–1981.

MASCI, Jonathan et al. **Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction.** Istituto Dalle Molle di Studi Sull'intelligenza Artificiale: (IDSIA). Lugano, Suiça, p. 52-60. 2011.

PERLIN, H. A.; LOPES, H. S. **Extracting human attributes using a convolutional neural network approach.** Pattern Recognition Letters, v. 68, n. 2, p. 250–259, 2015.

PIMENTEL, Marco A.f. et al. **A review of novelty detection.** Elsevier. Oxford, p. 215-249. jan. 2014.

RIBEIRO, Manassés; LAZZARETTI, André Eugênio; LOPES, Heitor Silvério. **A study of deep convolutional auto-encoders for anomaly detection in videos.** Elsevier. [s.i], p. 13-22. ago. 2017.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning representations by back-propagating errors.** Nature, v. 323, p. 533–536, Oct 1986.