

Detecção de Anomalias em Vias Públicas Usando Características Espaciais e um Classificador Sequencial Bidirecional

Fábio Ricardo Oliveira Bento* Raquel Frizera Vassallo**
Jorge Leonid Aching Samatelo**

* *Coordenadoria de Eletrotécnica, Instituto Federal do Espírito Santo, Guarapari, ES (e-mail: fbento@ifes.edu.br).*

** *Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, ES (e-mail: raquel@ele.ufes.br, jorge.samatelo@ufes.br)*

Abstract: Anomaly detection consists of identifying events that do not conform to an expected behavior pattern. In the area of law enforcement and security, anomalous event detection has application in the identification of suspicious behaviors. This paper addresses the problem of detecting suspicious behavior in public area monitoring videos. Our approach involves a convolutional neural network for spatial features extraction, followed by a time series classifier with an one-dimensional convolutional layer and stacked bidirectional recurrent neural networks. The experiments were performed on the UCSD Anomaly Detection Dataset, so we could compare our approach with other previous works. The obtained results were evaluated using , Area Under the Receiver Operating Characteristic Curve - AUC, Equal Error Rate - EER, and Area Under the Precision vs Recall Curve - AUPRC. During the experiments, the model obtained AUC above 95% and EER below 9%, which are results compatible with the current literature.

Resumo: Detecção de anomalias consiste na identificação de eventos que não estão em conformidade com um padrão de comportamento esperado. No contexto de segurança em vias públicas, a detecção automática de eventos anômalos através de vídeo, tem aplicação na identificação de comportamentos suspeitos. Nesse artigo é proposta uma abordagem para o problema da detecção automática de eventos anômalos em vídeos de vias públicas baseado em um modelo de redes neurais profundas *end-to-end*, composto de duas partes: um extrator de características espaciais baseado em uma rede neural convolucional pre-treinada, e um classificador de sequências temporais baseado em camadas recorrentes empilhadas. Realizamos experimentos nos conjuntos de dados *UCSD Anomaly Detection Dataset*. Os resultados foram avaliados com as métricas *Area Under the Receiver Operating Characteristic Curve - AUC*, *Area Under the Precision vs Recall Curve - AUPRC* e *Equal Error Rate - EER*. Durante os experimentos, o modelo obteve AUC acima de 95% e EER abaixo de 9%, os quais são resultados compatíveis com a literatura atual.

Keywords: Smart Cities, Computational Vision, Deep Learning, Anomaly Detection.

Palavras-chaves: Cidades Inteligentes, Visão Computacional, Aprendizagem Profunda, Detecção de Anomalias.

1. INTRODUÇÃO

Cidades Inteligentes podem ser definidas como uma área urbana, que proporciona desenvolvimento econômico sustentável e alta qualidade de vida, buscando excelência em: economia, mobilidade, meio ambiente, assistência à saúde, moradia e segurança pública (Montemayor et al., 2015).

Na área de segurança pública, os sistemas de visão computacional têm sido utilizados para detectar atividades anômalas, ou mesmo suspeitas (Ravanbakhsh et al., 2019; Singh et al., 2020). Anomalias são padrões em dados que

não estão de acordo com uma noção bem definida de comportamento e, portanto, são raros e sua detecção em vídeos é uma área de pesquisa ainda em aberto.

A atividade anômala também é conhecida como comportamento irregular, atividade suspeita, evento surpreendente, atividade incomum e assim por diante (Hendel et al., 2012). Nesse contexto, a habilidade de detectar eventos anômalos em vídeos tem inúmeras aplicações, incluindo identificação de acidentes, ocorrência de crimes e gestão de multidões. Todavia, se a detecção não for automatizada, os resultados dependem da direta intervenção humana. E, caso a demanda requerida de recursos humanos não possa ser atendida, um considerável volume de dados em vídeo será simplesmente armazenado, sem a devida análise

* Trabalho realizado com apoio da FAPES - Fundação de Amparo a Pesquisa e Inovação do Espírito Santo através do Projeto 577/2018, e da NVIDIA Corporation através da doação da GPU Titan V.

de seu conteúdo. Portanto, há evidentes oportunidades de pesquisa relacionadas ao problema de detecção automática de eventos anômalos em vídeos de vias públicas.

Nas últimas décadas, métodos baseados em aprendizado de máquina têm se mostrado eficazes em aplicações para detecção de anomalias, especialmente para a tarefa de detecção de eventos anômalos em vídeos de vias públicas. Pesquisas recentes têm como foco utilizar um subconjunto de modelos de aprendizado de máquina, conhecidos como redes neurais profundas, apresentando resultados superiores aos obtidos com a utilização de outros modelos clássicos de aprendizado. Nesse contexto, o presente artigo propõe modelar o problema em estudo como um classificador sequencial de dados implementado via uma arquitetura composta unicamente de redes neurais profundas, permitindo assim tornar a etapa de extração de características um problema de otimização dependente dos dados de entrada e das particularidades do modelo em si.

Para descrever o trabalho realizado, o restante desse artigo está organizado da seguinte maneira. A Seção 2 relata trabalhos relacionados. A Seção 3 detalha o modelo proposto. A Seção 4 apresenta um estudo comparativo através de experimentos, enquanto a Seção 5 traz as conclusões.

2. TRABALHOS RELACIONADOS

Na literatura, diferentes abordagens foram propostas para o problema em estudo, considerando-se dominantes as técnicas focadas em modelos de aprendizado de máquinas e visão computacional, como os descritos a seguir.

O aprendizado de características espaciais, temporais, e suas representações conjuntas, através de três *Stacked Denoising Autoencoders*, foi proposto por Xu et al. (2015). Em seguida, estimou pontuação de anomalias através de um *one-class Support Vector Machine* para cada um dos três tipos de características. Na sequência, fundiu os resultados da detecção em um único vetor de pesos para, finalmente, apresentar a decisão final.

Um *deep Gaussian Mixture Model*, foi proposto por Feng et al. (2017), para representar o problema. No entanto, esse tipo de modelo generativo tende a produzir uma alta taxa de falsos positivos, principalmente quando há considerável similaridade visual entre *frames* normais e anômalos.

Uma metodologia iterativa, que aprende características espaço-temporais usando uma rede convolucional 3D e, para treinar a rede de forma não supervisionada, emprega os resultados esparsos de *hand-crafted features*, foi desenvolvida por Chu et al. (2019). Entretanto, o desempenho do modelo é sensível à informação de movimentos não relacionados, como o da câmera, ou dos objetos de fundo.

Uma solução em três componentes: extração de características, representação esparsa e aprendizado de dicionário foi proposta por Zhou et al. (2019). Além disso, propuseram um algoritmo iterativo adaptativo, como uma nova versão de uma rede recorrente LSTM (*Long Short-Term Memory*).

Duas redes GAN (*Generative Adversarial Network*), que geram imagens a partir de representações de fluxo óptico, e vice-versa, foi a proposta de Ravanbakhsh et al. (2019). Contudo, pedestres não são reproduzidos com precisão.

Além disso, objetos anormais ou movimentos rápidos estão completamente ausentes nas reconstruções: os geradores não foram capazes de reconstruir os movimentos que nunca observaram durante o treinamento.

Um AOE (*Aggregation of Ensembles*), utilizando redes convolucionais CNN (*Convolutional Neural Network*) pré-treinadas e um conjunto de classificadores, foi proposto por Singh et al. (2020). Sua abordagem foi inspirada no conceito de que um conjunto de diferentes CNN ajustadas, representa vários níveis de semântica e, portanto, codificam um amplo conjunto de características robustas. Ou seja, AOE fornece robustez usando a diversidade no processo de detecção pois, em vez de um único modelo forte, vários modelos de menor porte são criados e combinados. Todavia, depois que os resultados são gerados, o desafio final é determinar a maneira pela qual diferentes pontuações podem ser combinadas.

Os trabalhos supracitados, realizaram experimentos no conjunto de dados *UCSD Anomaly Detection Dataset* de Li et al. (2014), referido como UCSDped, e utilizaram o AUC e EER (explicados na Seção 4.3) como métricas de desempenho, com o inconveniente de que nem todos os trabalhos apresentaram o mesmo protocolo de validação. Tal situação é contornada usando uma técnica de validação cruzada estratificada.

A diferença entre os trabalhos descritos e a proposta aqui apresentada, é que foi elaborado um modelo *end-to-end*, com resultado final único e conciso. Tal modelo aproveita as ideias apresentadas em Singh et al. (2020) onde são usados modelos CNN pré-treinados como extratores de características, e em Zhou et al. (2019), onde é usada uma rede recorrente para efetuar a tarefa de classificação. Também, devido aos tipos de camadas usadas, o modelo proposto dispõe de capacidade de aprendizado de dependências temporais (de movimento) de curto prazo, pois percorre as observações nas direções temporais direta e reversa, além de mitigar os efeitos da similaridade espacial entre *frames*. Os experimentos foram realizados com o UCSDped, no entanto, além das métricas AUC e EER, foi utilizada a métrica AUPRC explicada em detalhe na Seção 4.3.

3. PROPOSTA

O problema de detecção de anomalias em uma sequência de vídeo, comumente é modelado como um problema de classificação binária, onde, para cada *frame* de entrada, I_t , de um vídeo, é construído um modelo, f , que infere o respectivo rótulo l_t , ou seja,

$$l_t = f(I_t), \quad l_t \in \{0, 1\}. \quad (1)$$

Tal abordagem não leva em conta a ordem das observações, implicando que a informação temporal não seja considerada. Esta desvantagem pode ser superada ao modelar o problema como uma classificação de sequências do tipo *many-to-many*, onde, para um conjunto de *frames* de um vídeo extraídos em uma janela de tempo, $I_t, \dots, I_{t-(T-1)}$, é construído um modelo f que infere o rótulo para cada um dos *frames* de entrada, a dizer:

$$l_t, \dots, l_{t-(T-1)} = f(I_t, \dots, I_{t-(T-1)}), \quad l_* \in \{0, 1\}. \quad (2)$$

¹ Rótulo é a saída final de um modelo de classificação. No caso em estudo, anômalo (1) ou normal (0).

Nesse contexto, aqui é proposto construir o modelo sequencial, formado por dois componentes, o primeiro, um extrator de características espaciais, $Emb(\bullet)$, que transforma cada *frame* em um vetor de características denominado *embedding*², e o segundo, um classificador sequencial, $M(\bullet)$, que recebe uma sequência de *embeddings* correspondentes aos *frames* de entrada e de forma sequencial determina os rótulos de classe correspondentes a cada *frame*. Especificamente:

$$l_t, \dots, l_{t-(T-1)} = M(Emb(I_t), \dots, Emb(I_{t-(T-1)})).$$

Onde, $Emb(\bullet)$ é implementado via uma CNN pré-treinada e $M(\bullet)$ é implementado usando uma arquitetura baseada em camadas convolucionais unidimensionais e redes recorrente GRU (*Gated Recurrent Unit*). Já que ambos componentes são modelos neurais, pode-se afirmar que a proposta é uma arquitetura *end-to-end* e, portanto, podem ser otimizados de forma conjunta.

Para o caso de um classificador padrão (Equação (1)), o conjunto de treinamento é definido por um conjunto de pares entrada-saída, $\mathcal{T}_{org} = \{(I_i, l_i)\}_{i=1}^N$. No caso de um classificador sequencial (Equação (2)), o conjunto de treinamento também é configurado como tuplas entrada-saída, só que, tanto as entradas como as saídas são sequências de observações e alvos, ou seja, $\mathcal{T}_{seq} = \{(I_i, l_i)\}_{i=1}^{N'}$, onde, $I_i = \{I_{i,1}, \dots, I_{i,T}\}$ e $l_i = \{l_{i,1}, \dots, l_{i,T}\}$. Neste contexto, é necessário gerar a partir dos vídeos rotulados um conjunto de sequências, aplicando-se para tal caso, o método da janela deslizante. De forma resumida, esse método extrai sub-sequências a partir do conjunto de dados original. Assim, considerando um determinado tamanho de janela (T) e valor de deslocamento (d) a primeira sub-sequência de entrada, I_1 , é composta pelos *frames* 1 até T ; a segunda sub-sequência de entrada, I_2 , é composta pelos *frames* d até $T + d$; e assim por diante, conforme ilustrado na Figura 1. O mesmo procedimento é aplicado para a geração das sub-sequências de alvos. Cabe indicar que, o método de janelas deslizantes resulta na construção de uma quantidade de tuplas substancialmente maior ($N' \gg N$), contribuindo para superar a natural escassez de observações positivas do problema em estudo.

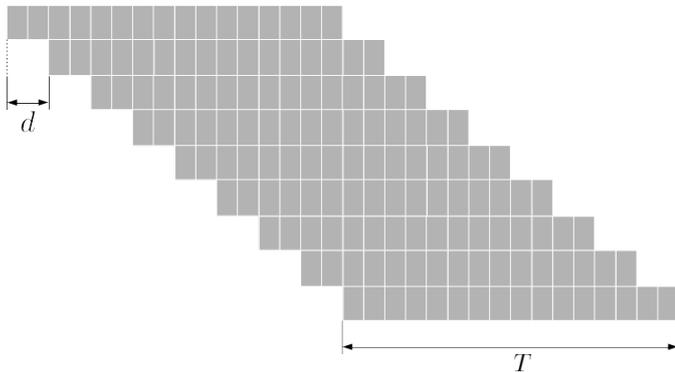


Figura 1. Extração de sub-sequências com deslocamento d e tamanho de janela T .

² Embedding é uma representação vetorial de baixa dimensão, aprendida a partir de variáveis com maior dimensionalidade.

3.1 Extrator de características espaciais

A primeira parte da proposta requer que seja selecionado um extrator de características espaciais. Este extrator será posteriormente utilizado para a construção de *embeddings* de cada um dos *frames*, de todos os vídeos, conforme ilustrado na Figura 2.

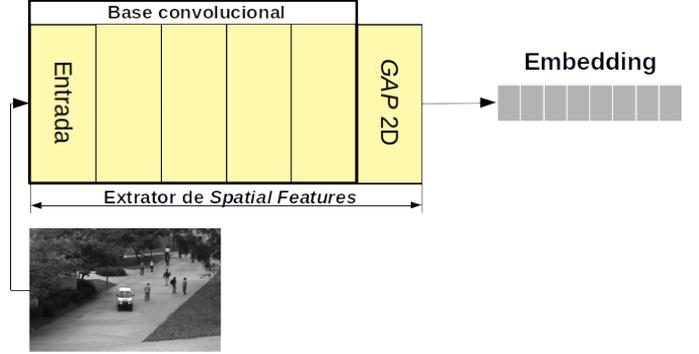


Figura 2. Conversão de *frames* em *embeddings* com o extrator de características espaciais.

Nesse contexto, o primeiro a fazer é realizar a escolha da arquitetura convolucional que será a base do extrator. Quatro arquiteturas CNN conhecidas da literatura foram avaliadas: VGG16 (Simonyan and Zisserman, 2014), Resnet50 (He et al., 2016), Xception (Chollet, 2017) e Densenet (Huang et al., 2017). A metodologia usada para a escolha baseia-se no desempenho de cada arquitetura frente ao problema de classificação de anomalia binário clássico (Equação (1)), ilustrado na Figura 3. Isso implica na modificação das arquiteturas originais, de modo que, a partir de cada CNN uma base convolucional é definida e sobre ela é adicionada um grupo de camadas que formam a saída do classificador binário. A base convolucional é integralmente aproveitada no extrator final, no entanto, as camadas adicionadas são desconsideradas.

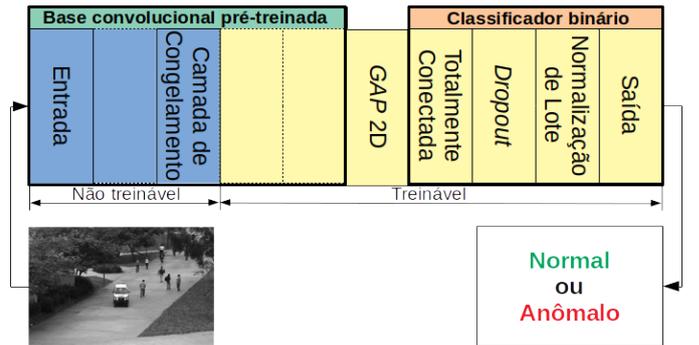


Figura 3. Detector de anomalias binário utilizado durante a escolha da arquitetura e treino da base convolucional do extrator de características espaciais.

De forma específica, a base convolucional é construída através de três passos. (i) Sobre cada arquitetura convolucional, inicializada com os respectivos pesos de treinamento no conjunto de dados *Imagenet* (Deng et al., 2009), duas camadas são definidas, a camada de congelamento e a camada de corte. Estas duas camadas dividem a rede em três partes, a primeira chamada de base não-treinável que vai desde a camada de entrada até a camada de

congelamento, a segunda denominada como base treinável que vai desde a camada de congelamento até a camada de corte, e a terceira que vai desde a camada de corte até a camada de saída. (ii) A última parte é eliminada, enquanto os pesos das bases não-treináveis são desabilitados, e os da base treinável são liberados para atualização durante o treinamento. (iii) Sobre a base convolucional já definida são adicionadas as camadas correspondentes ao classificador binário, efetuando-se depois o treinamento do modelo. A ideia desta metodologia é que a base não-treinável armazene o conhecimento adquirido do modelo pré-treinado, enquanto que, a base treinável permite a contextualização do modelo para o domínio do problema, através do ajuste dos pesos efetuado durante o treinamento.

O classificador binário é formado por: uma camada totalmente conectada com 64 neurônios com função de ativação ReLU, seguida de uma camada de *dropout* e de Normalização de Lote (Ioffe and Szegedy, 2015), finalizando com uma camada de saída de um neurônio com função de ativação sigmoide.

Finalmente na etapa de treinamento foi usada a função de perda *binary focal loss function* (Lin et al., 2017), e o conjunto de dados UCSDped, aplicando-se *grid-search* sobre os diferentes hiperparâmetros de cada modelo. De posse dos resultados obtidos, a arquitetura VGG16 foi escolhida como a base convolucional do extrator, considerando que as camadas de congelamento e corte foram, respectivamente `block4-pool` e `block5-pool`. Em termos proporcionais, 10,64% da arquitetura original da VGG16 foi aproveitada, dos quais 48,11% foi especializado para tarefa em questão.

É importante notar que o extrator de características espaciais após a remoção das camadas do classificador binário, ilustrado na Figura 4, é formado pela base convolucional mais uma camada GAP (*Global Average Pooling*), que recebe a saída da base convolucional e calcula a média global de cada um dos 512 mapas de características, gerando na saída do extrator um vetor unidimensional (*embedding*) com 512 características por cada *frame* de entrada.

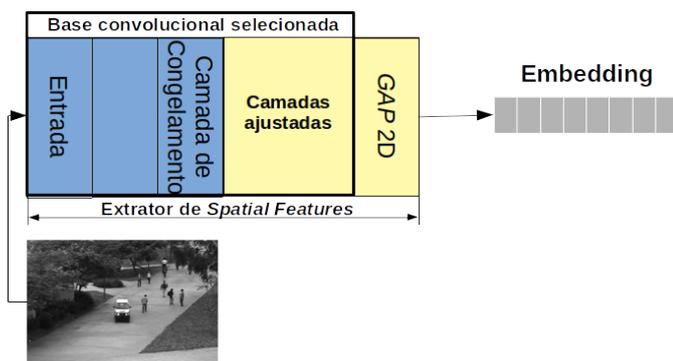


Figura 4. Esquema do extrator de características espaciais selecionado.

3.2 Classificador sequencial

O classificador sequencial é formado por dois tipos de camadas neurais: a primeira, uma camada convolucional unidimensional (CONV1D), utilizada para obter as informações mais relevantes dos *embeddings* gerados pelo

extrator de características espaciais e atenuar a presença de ruído; a segunda, um conjunto de camadas bidirecionais GRU (BGRU) empilhadas, que permitem aprendizado de dependências temporais de curto prazo, ao processar as observações nas direções direta e reversa. Resumindo, o classificador sequencial possui um total 176.381 parâmetros a serem determinados na etapa de treinamento, e é formado por uma camada CONV1D, aplicada de forma distribuída no tempo, seguida por duas camadas BGRU empilhadas e com saída densa única, com função de ativação sigmoide. A seguir alguns comentários sobre ambos tipos de camadas.

Uma CONV1D aplica vários filtros de convolução, que percorrem cada *embedding* de entrada, produzindo um mapa de características unidimensional por filtro. Com isso, durante treinamento, cada filtro aprenderá a detectar um único padrão sequencial muito curto (não maior que o próprio comprimento). Cada *embedding* de 512 elementos, é reduzido a um vetor de 20 elementos pela camada CONV1D. Considerando que, o classificador sequencial opera não com um único *embedding* se não com uma sequência deles, a camada CONV1D deve ser aplicada de forma distribuída no tempo, de modo que, seja aplicada a cada *embedding* da sequência de entrada.

Os mapas de características de saída da CONV1D são processados por BGRU empilhadas. Em uma BGRU, cada camada recorrente é duplicada, de forma que agora hajam duas camadas GRU lado a lado, fornecendo o vetor de entrada na sequência original para a primeira camada, e uma cópia invertida na segunda camada. Essa estratégia de reversão da ordem de entrada foi desenvolvida no intuito de melhorar o desempenho das redes recorrentes em geral (Schuster and Paliwal, 1997), e inicialmente teve maior aplicação no domínio do reconhecimento de fala. Isso ocorreu pois há evidências de que, em humanos, o contexto de toda a expressão é usado para interpretar o que está sendo dito, em vez de uma interpretação unidirecional (Graves and Schmidhuber, 2005). O presente trabalho, por sua vez, aplica BGRU empilhadas no contexto de visão computacional, objetivando: (i) permitir ao modelo aprender melhor as dependências contextuais de curto prazo, ao percorrer os *embeddings* dos *frames* nas direções temporais direta e reversa (Sutskever et al., 2014), e (ii) expandir a capacidade de aprendizado com mais unidades neurais através do empilhamento de camadas BGRU.

4. EXPERIMENTOS

Nesta seção é descrito o banco de dados utilizado nos experimentos; as características de implementação de *software* e *hardware* usadas; e os resultados obtidos na avaliação da abordagem proposta em função das métricas de desempenho em função das áreas sob as curvas ROC e *precision* vs. *recall*.

4.1 Conjunto de Dados

Os experimentos foram realizados no conjunto de dados UCSDped, o qual é orientado ao estudo de detecção de anomalias em vídeo. Os vídeos deste *dataset* foram adquiridos com uma câmera estacionária montada em uma elevação, com vista para passarelas de pedestres. A densidade da multidão capturada nos vídeos varia de esparsa

a muito lotada. Em situação considerada normal os vídeos contém apenas pedestres. Os eventos classificados como anormais incluem: (i) circulação de entidades não pedestres nas passarelas; (ii) padrões anômalos de movimento de pedestres. As anomalias mais comuns incluem motociclistas, skatistas, automóveis de pequeno porte e pessoas cruzando a passarela em sentido não usual, ou na grama que a cerca. Em alguns vídeos, pessoas em cadeira de rodas também foram registradas. Todas as anomalias ocorrem naturalmente, ou seja, não foram planejadas com o objetivo de montar o conjunto de dados.

O UCSDped foi dividido por seus autores em dois subconjuntos, Ped1 e Ped2, cada um correspondendo a uma cena diferente. As imagens capturadas em cada cena foram divididas em vários cliques de cerca de 200 *frames*. O subconjunto Ped1 é composto de vídeos de grupos de pessoas, sendo que algumas estão caminhando em direção à câmera, e outras estão caminhando no sentido contrário. Os vídeos desta seção também apresentam alguma distorção de perspectiva. O subconjunto Ped2, por sua vez, possui cenas de pedestres se movimentando perpendicularmente à câmera. Em termos quantitativos, Ped1 possui um total de 14000 *frames*, distribuídos em 70 vídeos com resolução de 238×158 pixels, enquanto o subconjunto Ped2 contém 4560 *frames*, distribuídos em 28 vídeos com 360×240 pixels de resolução.

O UCSDped possui dois tipos de anotações de referência (*ground-truth*): a nível de *frame* e a nível de pixel. Todos os vídeos do UCSDped possuem o primeiro tipo, que consiste de anotação binária indicando se alguma anomalia está presente (classificação positiva) em cada *frame*. Para o segundo tipo de anotação, apenas um subconjunto de 10 vídeos para Ped1 e 12 vídeos para Ped2 é fornecido com máscaras binárias (a nível de pixel) geradas manualmente. Essas máscaras identificam as regiões que contêm anomalias. Para o presente trabalho foram utilizadas as anotações do primeiro tipo (por *frame*), tendo em vista que elas abrangem todo o conjunto de dados. Os autores do UCSDped também subdividiram os vídeos de cada uma das duas cenas (Ped1 e Ped2) em dois diretórios: *train* e *test*. Com isso, os vídeos são identificados com nomes como “Train008”, ou “Test011”. É importante destacar que todos os vídeos armazenados nesses diretórios “*train*” contém apenas *frames* normais. Todos os vídeos nas pastas “*test*”, por outro lado, possui pelo menos algum *frame* anômalo. Essa organização sem amostras anômalas nos dados de treino, favorece soluções através de erro de reconstrução como, por exemplo, aquelas baseadas em redes GAN.

Para o presente trabalho, o UCSDped foi reorganizado de maneira a efetuar a validação da proposta via técnica *Stratified K-fold cross-validation*, com o objetivo de: (i) compor conjuntos de dados de treino e de teste, que contenham tanto amostras normais como anômalas; (ii) assegurar que, para cada *fold*³, não haja interseção entre conjuntos de vídeos treino e de teste, evitando assim que o modelo avaliado observe os dados de teste, o que invalidaria a análise do desempenho.

4.2 Implementação e Treino

O modelo proposto foi implementado e treinado utilizando o *framework* Tensorflow na sua versão 2.1.0. Os experimentos foram realizados em um computador com a seguinte configuração: (i) Sistema Operacional Linux, distribuição Ubuntu 16.04 Xenial, versão servidor; (ii) Processador Intel Xeon Silver 4214s(-HT-MCP-SMP-), 2.20GHz, com 48 núcleos; (iii) 64GB de RAM; (iv) 3TB de unidade de armazenamento permanente (disco rígido); (v) Placa gráfica *Nvidia Titan V*, com 12GB de memória dedicada.

Durante a etapa de treinamento do extrator de características espaciais mais o classificador sequencial, o conjunto de treinamento, \mathcal{T}_{seq} , foi formado por 8.256 tuplas de subsequências extraídas a partir dos vídeos dos conjuntos Ped1 e Ped2, considerando $T = 16$ e $d = 2$. Os hiperparâmetros usados no algoritmo de otimização foram: tamanho do *batch* de 64, um máximo de 50 épocas, otimizador Adam com taxa de aprendizado inicial de $6,05e-3$ decaindo 5% a cada época, e uma taxa de *dropout* de 0.5. Também foi utilizado o método de parada antecipada (*early stop*) para fazer com que o treinamento pudesse ser interrompido quando a acurácia média não apresentasse melhoria superior a 1% ao longo das últimas cinco épocas do treinamento.

4.3 Métricas

Considerando que o caso em estudo foi modelado como um problema de classificação binário, as métricas fazem uso das seguintes definições básicas: *VP* são os verdadeiros positivos (número de *frames* corretamente classificados pela técnica como anômalos), *VN* são os verdadeiros negativos (número de *frames* corretamente classificados pela técnica como normais), *FP* são os falsos positivos (número de *frames* classificados erroneamente pela técnica como anômalos), e *FN* são falsos negativos (número de *frames* classificados erroneamente pela técnica como normais).

No domínio de detecção de anomalias, é uma prática comum avaliar o detector utilizando a curva ROC (*Receiver Operating Characteristic Curve*), a qual envolve a taxa de verdadeiros positivos (*TVP*) e a taxa de falsos positivos (*TFP*). Nesta curva, um detector de boa qualidade deve estar o mais próximo possível do canto superior esquerdo do gráfico *TVP* vs. *TFP*, onde: $TVP = \frac{VP}{VP+FP}$ e $TFP = \frac{FP}{VP+FP}$. Uma visualização alternativa é a curva *Precision* vs. *Recall* - PR. *Precision* indica qual o percentual dos itens selecionados é verdadeiramente relevante (anômalo). *Recall*, por sua vez, representa qual percentual dos itens relevantes foi selecionado. Numericamente: $Precision = \frac{VP}{VP+FP}$ e $Recall = \frac{VP}{VP+FN}$. A interpretação da curva PR é um pouco diferente, pois um detector será tão melhor, quanto mais próxima sua curva estiver do canto superior direito. Nessa região da curva PR ocorre um melhor equilíbrio entre *Recall* e *Precision*.

Dados referentes a cenários com anomalias apresentam desequilíbrio entre classes. Nesse contexto a curva ROC pode não refletir aspectos importantes do desempenho do detector. Ou seja, uma boa curva ROC não garante sozinha a qualidade de um detector. A curva PR, por outro

³ Um *fold* é um conjunto de amostras de um conjunto de dados.

lado, expõe mais explicitamente deficiências de um modelo, mesmo que sua curva ROC tenha sido favorável. Portanto, é recomendável utilizar as curvas ROC e PR em conjunto para avaliar detectores de anomalias, tendo em vista o desbalanço característico dos dados. A fim de comparar o desempenho de diferentes modelos, são utilizadas as áreas abaixo dessas curvas: AUC para a curva ROC, e AUPRC para a curva PR.

Outra métrica utilizada no presente trabalho foi o EER (*Equal Error Rate*). O EER é numericamente igual ao *TFP* que ocorre no limiar da curva ROC, em que o *TFP* se iguala com o *TFN*. Qualitativamente o EER pode ser considerado um ponto de compromisso entre a taxa de falsos positivos e taxa de falsos negativos. O *TFN* corresponde a proporção de positivos que são previstos como negativos no teste, e é numericamente definido como $TFN = \frac{FN}{VP+FN}$. O EER pode ser também definido como o valor do *TFP* no ponto de intersecção entre a curva ROC e a reta correspondente a $TFP = 1 - TVP$.

Foi adotado o protocolo de avaliação de detecção de anomalias em nível de *frame*. Portanto, é avaliada a atribuição de um rótulo de anormalidade para cada *frame* de teste, com base na pontuação obtida ao final do modelo proposto, mediante a ultrapassagem de um limiar pré-definido. Este procedimento de avaliação é iterado usando uma faixa de limiares para construir as curvas ROC e PR. Considerando que os experimentos foram realizados com *Stratified K-fold cross-validation*, o cálculo das métricas respeita a organização em *folds* e os respectivos grupos de treino e de teste. Com isso, as métricas são calculadas individualmente para cada um dos *folds*, e o desempenho final do modelo é dados pela média dessas métricas (*Mean AUC* e *Mean AUPRC*). O EER é calculado sobre a curva de *Mean AUC*.

4.4 Resultados

As Figuras 5 e 6 apresentam as curvas ROC e PR de nosso modelo para o conjuntos de dados Ped1. As Figuras 7 e 8, por sua vez, apresentam as curvas ROC e PR para o conjunto de dados Ped2. As seguintes informações estão presentes nos gráficos: (i) a curva de cada *fold*; (ii) a curva média e a respectiva área abaixo da curva; (iii) o desvio padrão da curva média e respectiva faixa correspondente; (iv) a faixa de variação entre as curvas de *folds* com área mínima e a curva correspondente à curva média subtraída do desvio padrão; (v) a faixa de variação entre as curvas de *folds* com área máxima e a curva correspondente à curva média somada ao desvio padrão.

A partir da análise das Figuras 5 a 8 pode-se concluir que o modelo proposto possui desempenho satisfatório nos conjunto de dados de teste. No entanto, observa-se qualidade notadamente superior no conjunto Ped2 do que no Ped1. A possível razão é que a variância de densidade de pedestres de Ped2, é muito menor que a de Ped1, implicando em um desvio padrão de curvas médias muito inferior.

As Tabelas 1 e 2 mostram os resultados do modelo proposto em comparação a outros trabalhos nos conjunto de dados Ped1 e Ped2, respectivamente. Constatase que

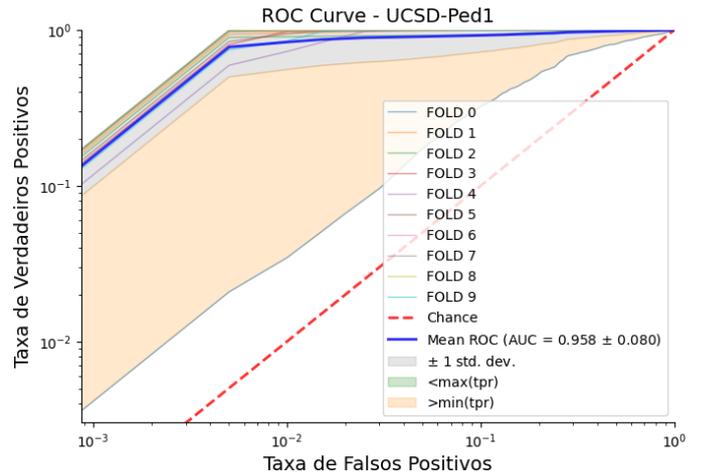


Figura 5. Curva ROC dos resultados do modelo proposto no conjunto de dados Ped1

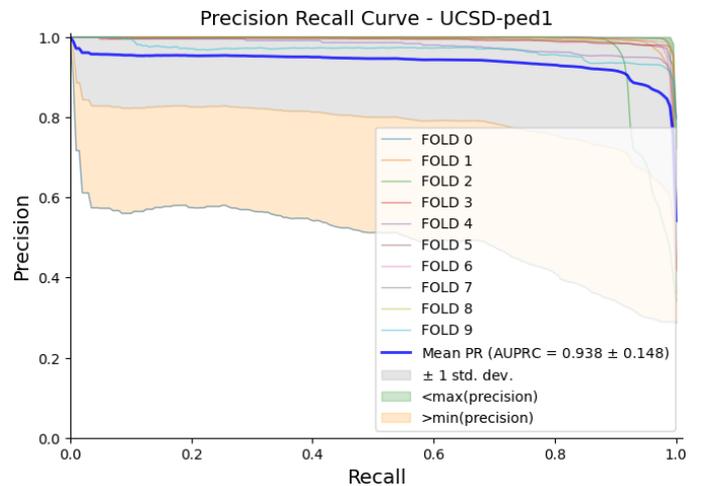


Figura 6. Curva Precision Recall dos resultados do modelo proposto no conjunto de dados Ped1

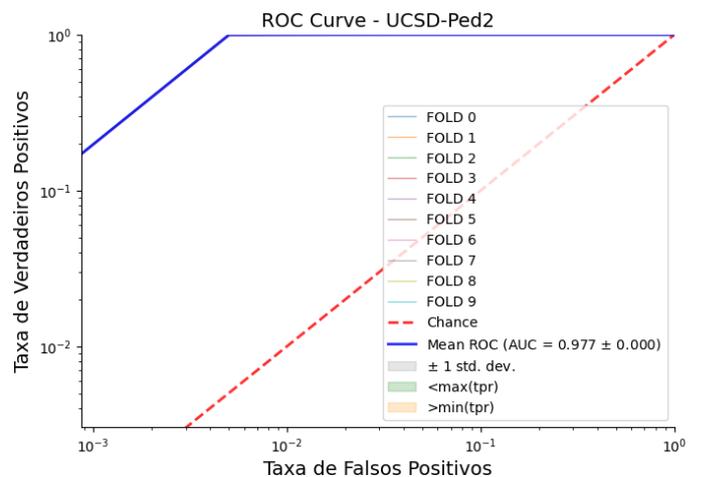


Figura 7. Curva ROC dos resultados do modelo proposto no conjunto de dados Ped2

nossa proposta alcançou resultados compatíveis com alguns dos melhores resultados atuais no UCSDped.

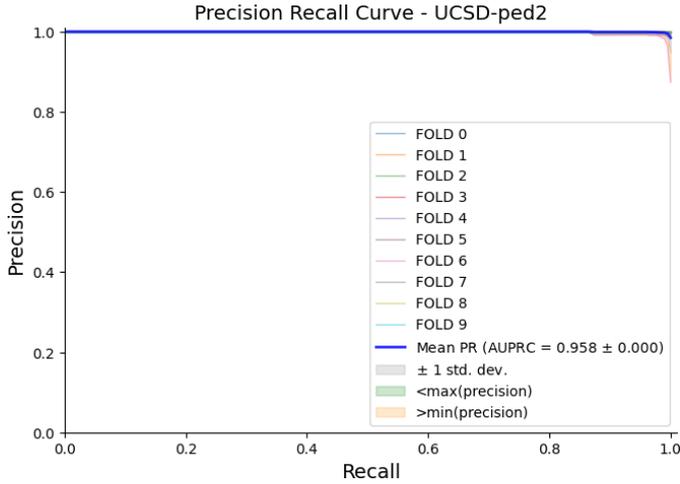


Figura 8. Curva Precision Recall dos resultados do modelo proposto no conjunto de dados Ped2

Tabela 1. Comparação dos resultados a nível de *frame* no conjunto de dados *UCSD-Ped1*.

	AUC	EER	AUPRC
Xu et al. (2015)	92,1%	16,0%	-
Feng et al. (2017)	92,5%	15,1%	-
Chu et al. (2019)	90,9%	16,2%	-
Khan et al. (2019)	81,1%	23,7%	-
Zhou et al. (2019)	83,5%	25,2%	-
Ravanbakhsh et al. (2019)	96,8%	7,0%	-
Singh et al. (2020)	94,6%	-	-
Proposto	95,8%	8,3%	93,8%

Tabela 2. Comparação dos resultados a nível de *frame* no conjunto de dados *UCSD-Ped2*.

	AUC	EER	AUPRC
Xu et al. (2015)	90,8%	17,0%	-
Feng et al. (2017)	-	-	-
Chu et al. (2019)	90,2%	17,3%	-
Khan et al. (2019)	93,8%	9,8%	-
Zhou et al. (2019)	94,9%	10,3%	-
Ravanbakhsh et al. (2019)	95,5%	11,0	-
Singh et al. (2020)	95,9%	-	-
Proposto	97,7%	3,1%	95,8%

4.5 Discussão

O modelo proposto possui, em suas camadas iniciais, o conhecimento adquirido da CNN VGG16, e nas últimas camadas, a especialização ao problema em estudo. A contribuição positiva desse procedimento fica evidente nos *heatmaps* obtidos através de mapas de ativação *Grad-CAM* da camada *block4-conv3* da base convolucional (Figuras 9 e 10). Nessas figuras, *y_pred* e *y_true* são, respectivamente, a pontuação de anomalia obtida ao final do modelo

proposto, e o valor de referência (*ground-truth*), do *frame* em questão. Estes *heatmaps* explicitam a anomalia presente em cada *frame*, indicando que o modelo é estimulado por características espaciais de objetos notáveis, como o veículo presente nos *frames* (Figura 9).

Observa-se, conforme ilustrado na Figura 10, que o modelo apresenta falsos positivos em situações em que pedestres minoritários transitam na direção oposta à da maioria, mesmo que em velocidade normal. Esse tipo de erro ocorre, notadamente, em *folds* cujo conjunto de treino é composto predominantemente por vídeos com baixa variância, tanto de densidade, quanto de direção de pedestres. Por essa razão, o modelo identifica equivocadamente nos *frames* da Figura 10, que a direção de movimento do pedestre minoritário implica em uma anomalia.

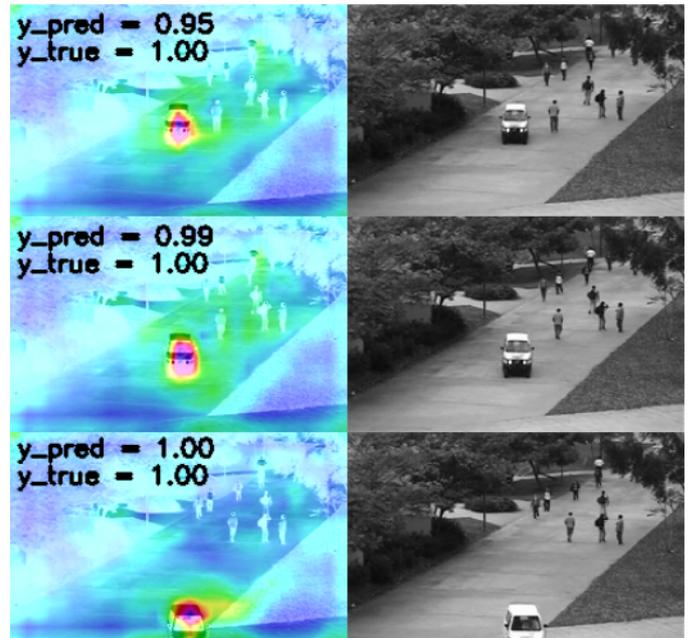


Figura 9. *Heatmap Grad-CAM* - Modelo estimulado por características espaciais de objetos

5. CONCLUSÃO

O considerável volume de vídeo capturado para detecção de eventos anômalos na área de segurança pública, está além da capacidade de análise de operadores humanos. Se, por outro lado, forem aplicadas técnicas focadas em modelos de aprendizado de máquinas e visão computacional, a qualidade da análise pode estar vinculada ao poder computacional disponível. Todavia, identificamos que um aspecto chave não está presente nas abordagens ora propostas: uma solução com resultado final conciso e único, na forma de um modelo *end-to-end*.

Nesse contexto, foi proposto construir um classificador de sequências *many-to-many*, com arquitetura *end-to-end*, composto por dois modelos neurais: (i) um extrator de características espaciais implementado via uma CNN pré-treinada; (ii) um classificador sequencial com arquitetura baseada em CONV1D e BGRUs. Nossos experimentos demonstram que, tanto o conhecimento adquirido da CNN

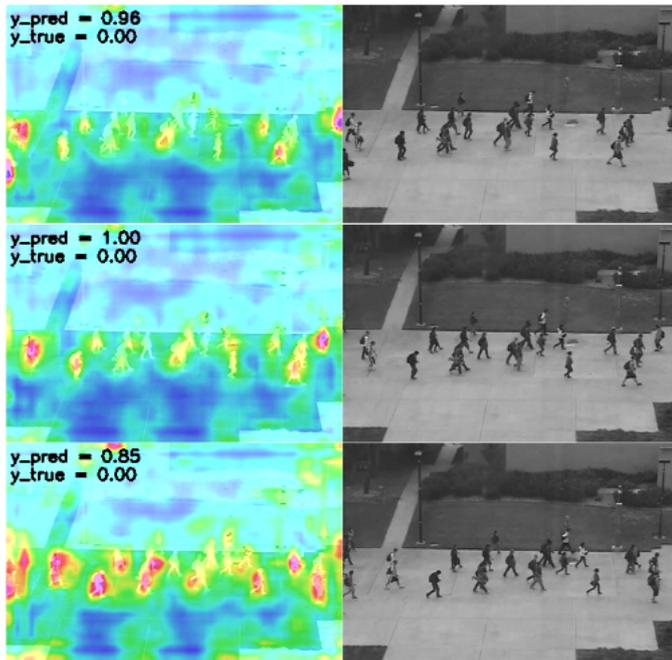


Figura 10. *Heatmap Grad-CAM - Presença de Falsos Positivos*

pré-treinada, quanto a posterior especialização ao problema, contribuíram positivamente para o desempenho do modelo.

A camada CONV1D favorece o aprendizado de características relevantes dos *embeddings* gerados pelo extrator de características espaciais, ao mesmo tempo em que pode atenuar a presença de ruído. As camadas BGRU, por sua vez, conferem capacidade de aprendizado de dependências de curto prazo, pois percorrem as observações nas direções temporais direta e reversa, mitigando os efeitos da similaridade espacial entre *frames*.

Acredita-se que, para aprimorar a abordagem proposta, trabalhos futuros sejam realizados com conjuntos de dados, que incluam vídeos capturados a partir de câmeras não estacionárias e com perspectiva variável.

REFERÊNCIAS

- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.
- Chu, W., Xue, H., Yao, C., and Cai, D. (2019). Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos. *IEEE Transactions on Multimedia*, 21(1), 246–255.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
- Feng, Y., Yuan, Y., and Lu, X. (2017). Learning deep event models for crowd anomaly detection. *Neurocomputing*, 219, 548–556.
- Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6), 602–610.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hendel, A., Weinshall, D., and Peleg, S. (2012). Identifying surprising events in video using bayesian topic models. In *Detection and Identification of Rare Audio-visual Cues*, 97–105. Springer.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Khan, M.U.K., Park, H., and Kyung, C. (2019). Rejecting motion outliers for efficient crowd anomaly detection. *IEEE Transactions on Information Forensics and Security*, 14(2), 541–556.
- Li, W., Mahadevan, V., and Vasconcelos, N. (2014). Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1), 18–32. Data retrieved from Statistical Visual Computing Laboratory - University of California - San Diego, <http://www.svcl.ucsd.edu/projects/anomaly/>.
- Lin, T.Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017). Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Montemayor, A.S., Pantrigo, J.J., and Salgado, L. (2015). Special issue on real-time computer vision in smart cities. *Journal of Real-Time Image Processing*, 10(4), 723–724.
- Ravanbakhsh, M., Sangineto, E., Nabi, M., and Sebe, N. (2019). Training adversarial discriminators for cross-channel abnormal event detection in crowds. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1896–1904. IEEE.
- Schuster, M. and Paliwal, K.K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Singh, K., Rajora, S., Vishwakarma, D.K., Tripathi, G., Kumar, S., and Walia, G.S. (2020). Crowd anomaly detection using aggregation of ensembles of fine-tuned convnets. *Neurocomputing*, 371, 188–198.
- Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.
- Xu, D., Ricci, E., Yan, Y., Song, J., and Sebe, N. (2015). Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*.
- Zhou, J.T., Du, J., Zhu, H., Peng, X., Liu, Y., and Goh, R.S.M. (2019). Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 14(10), 2537–2550.