APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE VIOLAÇÃO DE CAIXAS ELETRÔNICOS

MACHINE LEARNING FOR VIOLATION DETECTION ON AUTOMATED TELLER MACHINE

Dennis Rojas Pereira[1]
Fabio Andrijauskas[2]
Universidade São Francisco – *Campus* Itatiba
dennisrojaspereira@gmail.com | fabio.andrijauskas@gmail.com

[1]Alunos do Curso de Engenharia da Computação
[2] Professor Orientador

RESUMO. O crescimento da violência, principalmente em bancos, fez com que a demanda por sistemas de monitoramento confiáveis, efetivos e com baixo custo aumentasse. Em diversos casos a análise é manual onde um ser humano monitora câmeras de vigilância 24 horas, este trabalho apresenta um sistema inteligente utilizando aprendizado de máquina e técnicas de visão computacional para reconhecer padrões e fornecer eventos a partir de determinados comportamentos para uma melhoria da segurança em ambientes bancários. Foram selecionados mecanismos para rastreamento de objetos e reconhecimento de anomalias sendo utilizado o algoritmo de atualização em segundo plano com o auxílio da ferramenta de subtração. A aplicação foi desenvolvida na linguagem de programação Python, utilizando a IDE Eclipse em conjunto com a biblioteca OpenCV. Foram obtidos resultados na detecção de movimento e classificação satisfatórias quando utilizados em vídeos com resolução baixa.O algoritmo foi testado atingindo 78% em uma amostra de 128 segundos.

PALAVRAS-CHAVE: Visão Computacional, Aprendizado de Máquina, Vigilância, Rastreamento, Subtração.

ABSTRACT. The growth of violence mainly in banks have made the demand for reliable, effective and low-cost monitoring systems grow up. Many cases has in mind that the process is currently made by a human analyzing surveillance cameras all day, this work presents an intelligent system using machine learning and computer vision techniques to recognize patterns and provide events from certain behaviors for improved security in banking environments. From the analysis of the difficulties and techniques used by researchers were selected mechanisms for object tracking and anomaly recognition being used the algorithm of updating in the background with the aid of the tool of subtraction.The application was developed in the Python programming language, using the Eclipse IDE in conjunction with the OpenCV library. Satisfactory results have been obtained when using low resolution videos and further studies are needed to improve the technique developed for more complex images.The algorithm was tested with a percentage of 78% accuracy in sampling of 128 seconds.

KEYWORDS: Computer Vision, Machine Learning, Surveillance, Tracking, Subtraction.

INTRODUÇÃO

A criminalidade tem se agravado dia após dia no Brasil, os estudos sobre violência e crime no país ganharam fôlego a partir dos anos 70 e, desde então, sofisticaram o debate sobre os significados que tais conceitos têm assumido na sociedade. (GUIRARDI e MANOLESCU, 2009). Foi necessário que o setor de segurança se tornasse prioritário na esfera pública e privada resultando no investimento elevado em tecnologia. Um componente amplamente utilizado é o monitoramento em sistemas de segurança servindo apenas para coleta de evidências após o crime e produção de dados estatísticos.Nos sistemas tradicionais, o procedimento clássico de inspeção consiste na avaliação por um operador humano das imagens geradas por um Circuito Fechado de Televisão (STIVANELLO *et al.*,2014).

Segundo Santos *et al.*(2017), os sistemas CTV são compostos por câmeras fixas, instaladas em ambientes internos ou corredores e até mesmo em ambientes externos como ruas e praças, seus objetos de interesse são pessoas ou veículos que geralmente não aparecem sozinhos no campo de visão. A busca de sistemas de monitoramentos confiáveis, efetivos e de baixo custo onde operadores humanos têm a tarefa de avaliar a imagem em tempo real em um Circuito Fechado de Televisão(CFTV) no ambiente monitorado. Dadas as dificuldades e o custo elevado para manter um expediente 24 horas considera-se importante o estudo e desenvolvimento de sistemas inteligentes capazes de gerar eventos automaticamente. (STIVANELLO *et al.*,2012).

Conforme Santos(2017) *et al.* apud Yilmaz *et al*.(2006) e Casagrande (2016), com isso a demanda do mercado aumentou por softwares capazes de identificar, classificar, rastrear e analisar trajetórias e comportamentos anômalos para apontar possíveis eventos suspeitos têm se tornado prioritária. Marques (1999) a área de processamento de imagens vem sendo objeto de crescente interesse por permitir viabilizar grande número de aplicações em duas categorias bem distintas: (1) o aprimoramento de informações pictóricas para interpretação humana; e (2) a análise automática por computador de informações extraídas de uma cena.

Programar computadores para enxergar o ambiente ao nosso redor e torná-los capazes de reconhecer e extrair informações de objetos podem parecer tarefas complexas que necessitam de fortes conhecimentos matemáticos e de programação (BARELLI, 2017). Os sistemas de visão computacional têm ajudado cada dia mais sendo essencial e presente no nosso cotidiano onde Barelli (2017) cita veículos autônomos, robôs industriais ou em equipamentos hospitalares capazes de diagnosticar doenças automaticamente em exames por imagem.

O objetivo deste trabalho é utilizar técnicas de aprendizado de máquina para a detecção de padrões anormais que podem levar a alguma prevenção de crime bancário.

LEVANTAMENTO BIBLIOGRÁFICO

Conforme González (2011) e Woods (2011), uma das primeiras aplicações aconteceu na década de 1920 na transmissão de imagens por um cabo submarino entre Londres e Nova York. Esta transmissão adiantou o processo que era mais de uma semana em menos de três horas para o transporte da imagem, tendo uma codificação no envio e decodificação no recebimento. Após as inovações de hardware, a visão computacional surge como uma ciência que estuda e desenvolve tecnologias permitindo onde máquinas enxerguem e extraiam características do meio através de imagens. Estas informações permitem reconhecer, manipular e processar dados sobre os objetos que compõem a imagem capturada.(BARELLI,2017 apud BALLARD e BROWN,1982). Entretanto, inúmeras tarefas

que realizamos manualmente podem ser automatizadas por sistemas de Visão Computacional. Os sistemas de monitoramento por imagem, capazes de detectar pessoas em um ambiente, como os robôs industriais, preparados para montar automóveis, são exemplos comuns da aplicação da Visão Computacional em nosso cotidiano (BARELLI,2017). Santos et al.(2017) explica que é comum encontrar trabalhos que citam ou discorrem sobre os desafios de rastreamento de objetos e o interesse pela interação automática é evidente pois não depende de nenhum ato humano porém o principal desafio é o rastreamento conforme exemplificado na Figura 1.

Figura 1 – Desafios do rastreamento de objetos. (Fonte: Santos, 2017)

A etapa de aquisição tem como função converter uma imagem em uma representação numérica adequada para o processamento digital subseqüente onde armazenamento de imagens digitais é um dos maiores desafios no projeto de sistemas de processamento de imagens, em razão da grande quantidade de bytes necessários para tanto. (Marques Filho;Vieira Neto,2009). Nos trabalhos de Stivanello et. al(2014) e Santos et. al(2017) e Wu(2009) são utilizadas a biblioteca OpenCV, que consiste em uma interface que contém mais de 300 funções de programação em C e não depende de bibliotecas externas e todos seus algoritmos são baseados em estruturas de dados dinâmicos agrupados que tem alta flexibilidade e mais da metade deles foram otimizados para processadores Intel(WU,2009).

Nos trabalhos de Stivanello et. al(2014) e Santos et. al(2017) e Wu(2009) são utilizadas a biblioteca OpenCV, que consiste em uma interface que contém mais de 300 funções de programação em C e não depende de bibliotecas externas e todos seus algoritmos são baseados em estruturas de dados dinâmicos agrupados que tem alta flexibilidade e mais da metade deles foram otimizados para processadores Intel(WU,2009). Stivanello(2009) utiliza as técnicas de identificação de presença e a estimação de posição métrica de objetos ou pessoas em um ambiente monitorado. Sua segmentação consiste no agrupamento dos pontos presentes na imagem percebida baseado em proximidade, similaridade e continuidade tendo como objetivo o isolamento dos elementos presentes na cena observada possuindo dificuldade maior em cenários dinâmicos onde as variações no ambiente devem ser consideradas.

Com isso é utilizado um processo de subtração de intensidades onde a imagem é comparada com uma imagem referência contendo apenas o fundo da cena finalizando o processo destacando somente os elementos em movimento na cena possibilitando o isolamento de cada objeto na cena e estimação de coordenadas (Stivanello et al;2014). As técnicas foram aplicadas na linguagem C++ com a biblioteca OpenCV e foram feitos ensaios em uma sequência de vídeos com pessoas transitando em um corredor e no processo de

subtração o algoritmo considerou a pessoa e a sombra conforme demonstrado na Figura 2 que destaca o indivíduo identificado por técnica de subtração.

Figura 2 – Técnica de subtração.(Fonte:Stivanello.,2012).

Santos *et al.*(2017) apud Choi *et al.*(2015) dividem seu projeto em detecção, rastreamento e associação de objetos seu trabalho. Na etapa de detecção utilizam o modelo de mistura de gaussianas para gerar o plano de fundo e regiões de interesse são obtidas através de método de subtração com uma imagem modelo. Essas regiões passam por classificadores LBP-AdaBoost e HOGSVM, baseados em características de texturas. Na etapa de rastreamento é usado o filtro de partículas para estimar a posição dos objetos e detecções utilizadas atualizando o rastreador e com isso chegando na etapa de associação de dados. Santos *et al.*(2017) apresentou um algoritmo de rastreamento de múltiplos objetos baseados em extração de fundo e filtro de Kalman propondo simplicidade e obtendo resultados satisfatórios. Os algoritmos de rastreio devem executar uma sequência de etapas da melhor forma possível, pois pode haver uma propagação de erros entre elas comprometendo seu desempenho envolvendo tarefas de pré-processamento, modelagem e extração de plano de fundo e pós processamento.

Após isto Santos *et al.*(2017) em seu pré processamento, faz com que os quadros sejam convertidos em tons de cinza, reduzindo a quantidade de dados a ser processado e aplicando um filtro gaussiano para eliminar ruídos. Sendo assim, cada quadro passa pelo extrator de plano de fundo ViBe onde a segmentação é a partir do primeiro quadro e classifica os pixels em duas regiões chamadas de plano de fundo(*background*) e possíveis pontos de interesse (*foreground*) gerando uma matriz binária que é pós processada removendo aglomerados de pixels. Na sua próxima etapa chamada de previsão estados futuros dos objetos é utilizado o filtro de Kalman de um quadro para o outro em uma alta taxa de quadros por segundos associando os dados e identificando um mesmo objeto quadro-a-quadro em meio aos outros objetos detectados. Estes processos ficam em um laço e somente se encerram quando o último vídeo é processado sendo exportados os registros dos objetos no formato: número do quadro, rótulo do objeto, posição X do centróide, posição Y do centróide e Largura e altura (SANTOS *et al*,2017).

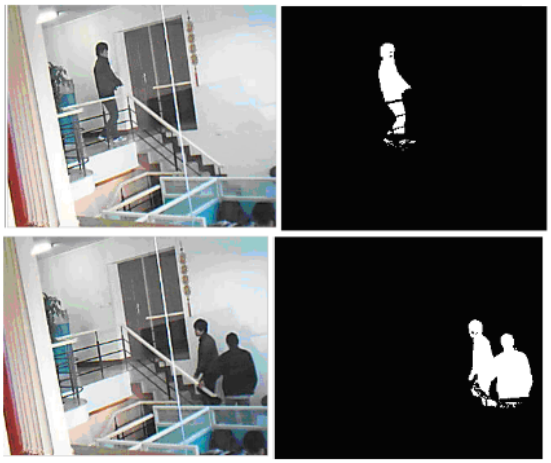
No trabalho de Wu(2009) ele projeta um sistema de processamento de imagem e monitoramento de ambientes com base no OpenCV. Sendo assim o sistema tem uma forte adaptabilidade para variedades nas regiões de monitoramento e capaz de garantir que as imagens de fundo sejam baseadas na atualização de áreas estáticas em tempo real. Utilizando também a ferramenta de subtração de fundo porém seu fundo é dinâmico mudando em um processo que roda em segundo plano em seu sistema conforme demonstrado na Figura 3.

Figura 3 – Teste parte da manhã com seu quadro de imagem padrão atualizada em tempo real.(Fonte:WU ,2009).

As etapas seguidas de pré-processamento, processamento, pós-processamento e análise e seu monitoramento consiste em método de análise baseado em corpo humano, método baseado nas características temporais e espaciais e método baseado em estatística de imagem (WU,2009).

O método baseado em características temporais e espaciais primeiramente extrai a zona do corpo humano em cada quadro e então tenta interpretar e identificar o comportamento humano com várias características de forma de sequência de contorno humano (WU,2009) conforme visualizado na Figura 4.

Figura 4 – Resultados da identificação de humanos no ambiente.(Fonte:Wu ,2009).



O fluxo do sistema de Wu (2009), onde o processamento de imagem de tela de sistema de monitoramento é composto de hardware e software. A parte do hardware incluirá camera, placa de vídeo de captura e um computador. Seria aplicado o processo ADC nas imagens e enviaram estes resultados para o computador que obteria as informações de movimento de corpo por meio de algoritmos relacionados ao computador e analisados pelo software.

Este software consiste em algoritmos de detecção e rastreamento de movimento, que são as partes centrais de todo o sistema, esta imagem de vídeo enviada pela câmera passaria através de um pré-processamento, processamento, pós-processamento e análise antes que os resultados da disposição sejam revelados, sendo uma combinação entre a plataforma OpenCV e algoritmos de detecção do corpo humano, rastreamento e determinação de comportamento. A parte crítica do projeto é a detecção de movimento humano pois o modelo da imagem de fundo pode ser alterado por oclusões ou iluminação do ambiente sendo necessário utilizar o esquema de Wu (2009) que faz com que esta imagem seja dinâmica, sendo alterada ao longo do tempo. Sendo a base essencial o algoritmo de subtração para o projeto.

Para classificação do objeto, Viola e Jones(2001) em seu método possuem duas categorias onde tudo que deve dar positivo e tudo que deve dar negativo. No caso seu algoritmo é utilizado para encontrar faces em uma imagem, sendo assim possui um conjunto de imagens de faces com um intervalo muito amplo de condições levando em consideração iluminação, escala, pose e angulo da camera. Na Figura 5, é possível verificar exemplo de um dataset com imagens positivas, onde a maioria das imagens são faces.

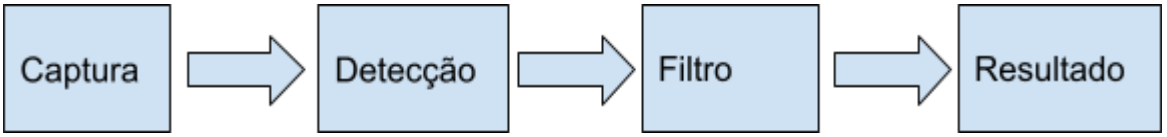
Figura 5 – Resultados positivos para faces.(Fonte:VIOLA;JONES ,2001).

METODOLOGIA

O projeto consiste em um sistema inteligente de processamento de vídeo bancário integrando visão computacional e reconhecimento de padrões fornecendo eventos adequados para melhorar a segurança sendo desenvolvido em Python. Conforme utilizado por Santos (2017) o rastreamento e reconhecimento do algoritmo compara o frame atual com o anterior, aplicando filtro de escala de cinza nas imagens e calculando a porcentagem de diferença pela subtração.

A classificação na imagem utiliza o Haar Cascade do OpenCV e seus arquivos de classificação foram obtidos do repositório https://github.com/opencv/opencv/blob/master/data/haarcascades/ onde foram utilizados os metodos *Full Body*, *Lower Body* e *Upper Body* gerando poligonos baseado em resultados, recortando a imagem e armazenando em formato JPG em pastas separadas. A Figura 6 mostra as etapas a serem executadas que inicia por uma captura de quadro em quadro, em seguida detecta a diferença da imagem anterior e a imagem atual onde a imagem anterior é definida como referência para ter o rastreamento de objetos, aplicando filtros de cores em escala de cinza e executando a classificação destacando partes interessadas.

Figura 6 – Diagrama da metodologia a ser aplicada. (Fonte: Próprio autor)



A aplicação foi desenvolvida na linguagem de programação Python, utilizando a IDE Eclipse em conjunto com a biblioteca OpenCV.

RESULTADOS E DISCUSSÃO

O algoritmo foi testado em um vídeo de explosão de um caixa eletrônico disponível no Youtube pela TV Barra Velha, possuindo uma grande quantidade de frames com ruídos, fundos pouco destacados e objetos sobrepostos exigindo a necessidade de aprimoramentos no tratamento da imagem a fim de diminuir a identificação de falsos positivos. Com o algoritmo foi possível obter a extração da porcentagem de diferença entre os últimos frames registrados e o processamento do OpenCV utilizando Haar Cascade possibilitando a detecção e o mapeamento de coordenadas da imagem conforme Figura 7.

Figura 7 – Código fonte em Python. (Fonte: Próprio autor)

```
1  '''Created on 10 de out de 2018 @author: Dennis '''
2  import cv2 ;import numpy as np ;import time ;import winsound
3  face_cscFull = cv2.CascadeClassifier("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\haarcascade_fullbody.xml")
4  face_cscLower = cv2.CascadeClassifier("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\haarcascade_lowerbody.xml")
5  face_cscUpper = cv2.CascadeClassifier("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\haarcascade_upperbody.xml")
6  vidcap = cv2.VideoCapture("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\banco_curto.mp4")
7  count = 0; list_first_frames = []; id_image = 0
8  while True:
9      ret, original_frame = vidcap.read()
10     if not ret: break
11     temp = original_frame
12     if(len(list_first_frames) == 3):
13         ultima, penultima, antepenultima = list_first_frames[0], list_first_frames[1], list_first_frames[2]
14         d1 = cv2.absdiff(antepenultima, penultima);d2 = cv2.absdiff(penultima, ultima);
15         res = d1.astype(np.uint8);percentage = round((np.count_nonzero(res) / res.size) * 100, 4)
16         if ( percentage > 80): winsound.PlaySound('C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\alarme.wav', winsound.SND_FILENAME)
17         if (percentage >= 0):
18             temp = cv2.cvtColor(temp, cv2.COLOR_BGR2GRAY)
19             full =   face_cscFull.detectMultiScale(image=temp, scaleFactor=1.1, minNeighbors=7)# minSize=(10,10), maxSize=(150,500))
20             lower = face_cscLower.detectMultiScale(image=temp, scaleFactor=1.1, minNeighbors=4)# minSize=(10,10), maxSize=(150,500))
21             upper = face_cscUpper.detectMultiScale(image=temp, scaleFactor=1.1, minNeighbors=4)#, minSize=(10,10), maxSize=(150,500))
22             for (x, y, w, h) in full:
23                 cv2.rectangle(original_frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
24                 cv2.putText(original_frame, 'Corpo', (x,y-10), cv2.FONT_HERSHEY_COMPLEX, 0.75, (255,0,0), 1)
25                 cv2.imwrite("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\video\\full\\full%d.jpg" % id_image, original_frame[y:y+h,x:x+w]);id_image += 1
26             for (x, y, w, h) in lower:
27                 cv2.rectangle(original_frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
28                 cv2.putText(original_frame, 'Inferior', (x, y-10), cv2.FONT_HERSHEY_COMPLEX, 0.75, (0, 255, 0), 1)
29                 cv2.imwrite("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\video\\lower\\lower%d.jpg" % id_image, original_frame[y:y+h,x:x+w]);id_image += 1
30             for (x, y, w, h) in upper:
31                 cv2.rectangle(original_frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
32                 cv2.putText(original_frame, 'Superior', (x, y-10), cv2.FONT_HERSHEY_COMPLEX, 0.75, (0, 0, 255), 1)
33                 cv2.imwrite("C:\\Users\\Dennis\\eclipse-workspace-tcc\\Tcc\\video\\upper\\upper%d.jpg" % id_image, original_frame[y:y+h,x:x+w]);id_image += 1
34     if len(list_first_frames) <= 2:
35         list_first_frames.append(temp)
36     elif len(list_first_frames) == 3:
37         list_first_frames = []
38     count += 1;cv2.imshow('', original_frame);time.sleep(0.04);
39     if cv2.waitKey(1) & 0xFF == ord('q'):
40         break
41 cv2.waitKey(0);cv2.destroyAllWindows()
42
43
```

Nas linhas entre 2 e 7 são importadas as bibliotecas necessárias e os arquivos são de classificação são carregados em uma variável pela função *CascadeClassifier*. O vídeo também é carregado pela função *VideoCapture* do OpenCV. Na linha 8 até a 40 é feito um laço de repetição que roda até o fim do vídeo extraindo quadro por quadro armazenando a imagem penúltima, última e atual.

As linhas 13,14 e 15 com as imagens em memória obtém a porcentagem de cada mudança pela função *absDiff* que calcula o valor absoluto de cada elemento da matriz da imagem usando também a biblioteca Numpy destinada para computação científica permitindo fazer operações em arrays. Na linha 16 é feita uma condição caso tenha uma diferença maior que 80% dispara um alarme utilizando a biblioteca WinSound. Da linha 18 até a 38, é aplicado o método de detecção de objetos *Haar Cascade* detectando e obtendo coordenadas classificadas em 3 listas contendo o corpo inteiro, a parte inferior e a parte superior desenhando um retângulo nas áreas detectadas e inserindo um texto pelo tipo de classificação. Entre a linha 38 e 40 , caso seja pressionado a tecla Q o algoritmo sai do *loop* e a tela é fechada.

A detecção pode ser dividida em partes inferiores, superiores e corpo inteiro de uma pessoa desenhando um traçado em forma de retângulos sobre a área detectada. Nos testes realizados, a maioria dos movimentos foram detectados porém houve falhas na detecção quando as pessoas estavam muito distantes da câmera ou utilizavam roupas que se assemelhavam ao fundo da imagem.

Conforme a tabela 1, na amostragem de 118 segundos a movimentação detectada teve um acerto em 92 segundos onde indicou também detectou 3 eventos de falsa movimentação e 23 momentos onde não foi detectado obtendo um acerto de 78%.

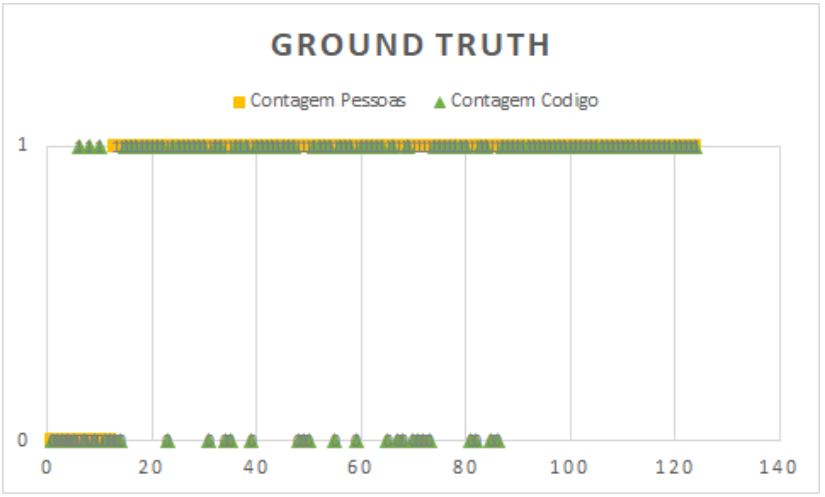
Tabela 1 – Falsos Positivos; Falsos Negativos;Acertos;Erros**.**

Tipo	Valor
Falsos Positivos	3
Falsos Negativos	23
Erros	26
Acertos	92

Fonte: Próprio autor.

Durantes os testes realizados, pode-se observar que os traçados não apareciam pois eram tão rápidos que a renderização da forma geométrica ficou imperceptível ao olho humano. Ao colocar um tempo de espera entre os desenhos foi possível visualizar os contornos em volta dos objetos detectados. O método utilizado para a avaliação de acertos é o *Ground Truth* que é usado em vários campos para se referir às informações fornecidas por observação direta, em oposição a informações fornecidas por inferência onde foram contabilizados por olho humano se houve movimentação e após isto comparado com a sinalização do algoritmo.

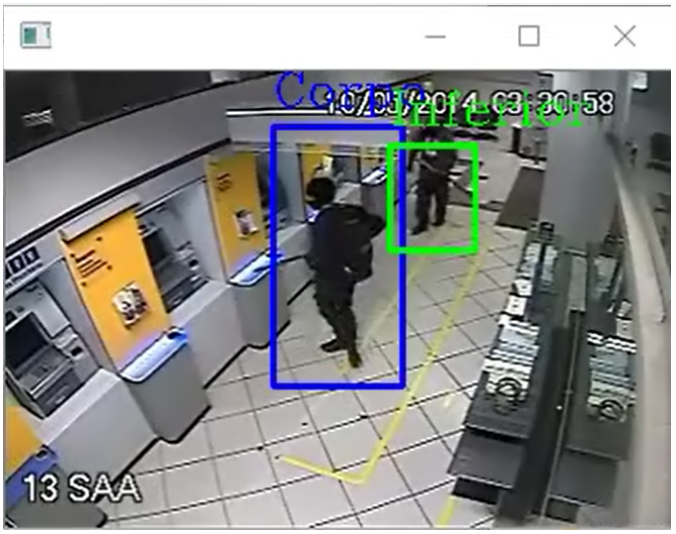
Figura 8 – Ground Truth da classificação.(Fonte: Próprio Autor)



É possível notar que os falsos positivos acontecem nos primeiros frames, a imagem contém um caixa eletrônico com o seu visor alterando a iluminação gerando uma detecção falha classificando como uma pessoa. Santos (2017) sugere que além de testar o algoritmo em outras sequências de vídeo, é necessário incorporar, o uso de descritores locais para aumentar a robustez na associação de dados. A partir da solução desenvolvida é possível detectar os objetos em movimento, de forma que a área pode ser calculada e extraída para armazenamento em pastas. O algoritmo é capaz de realizar essas tarefas de forma satisfatória para vídeos de resolução baixa porém quanto maior for a qualidade do vídeo melhor o resultado esperado.

A classificação em tempo real pode ser observada na Figura 9 onde são destacadas na cor azul o corpo inteiro e em verde a parte inferior de um corpo.

Figura 9 – Classificação em tempo real.(Fonte:Próprio Autor)



As coordenadas são utilizadas para recortar partes da imagem gerando um novo arquivo em JPG sequencial e em pastas distintas pelo tipo da classificação conforme as Figuras 11, 12 e 13.

Figura 10 – Classificação do corpo inteiro em tempo real.(Fonte:Próprio Autor)

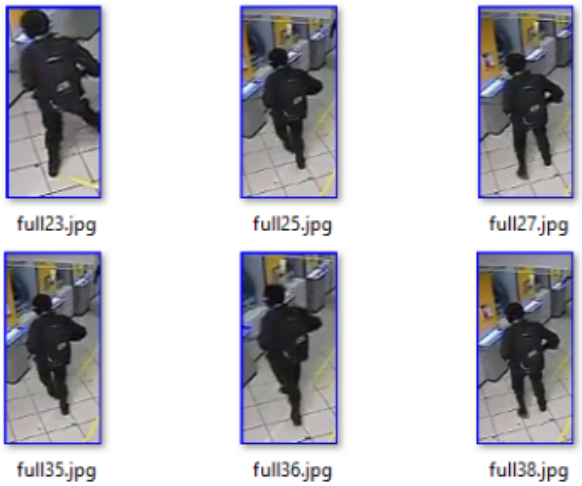


Figura 11 – Classificação do corpo inteiro em tempo real.(Fonte:Próprio Autor)

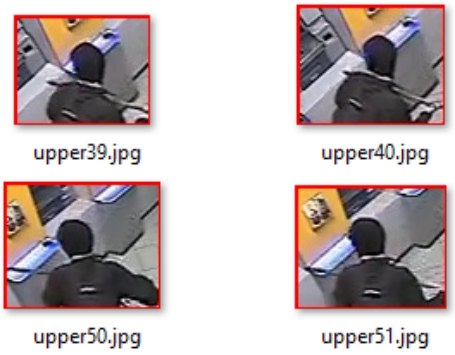
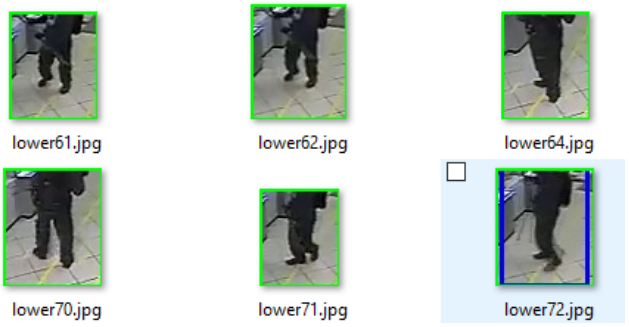


Figura 12 – Classificação da parte inferior do corpo em tempo real.(Fonte:Próprio Autor)



CONCLUSÕES

O monitoramento em tempo real é uma atividade essencial para àrea de segurança, impulsionando o desenvolvimento de aplicações promissoras, controlando o acesso em determinadas áreas, identificando objetos específicos em certas cenas e detectando a presença de anomalias a fim de gerar alarmes auxiliando o operador. Neste trabalho foi proposto um sistema de detecção automática por evento, videos reais de roubos com explosões de caixa-eletrônicos foram analisados utilizando um algoritmo em Python. As tarefas principais desenvolvidas foram a detecção, rastreamento, compreensão e identificação de comportamentos anômalos. A detecção de objetos em movimento envolveu processos de modelagem, segmentação de movimento e classificação de objetos sendo aplicadas as técnicas de subtração de fundo, diferenciação em porcentagem de cada frame e identificação por *Haar Cascade* do OpenCV.

Os resultados foram satisfatórios onde foi encontrado um acerto de 78%, a maioria dos falsos alarmes ou erros gerando a ocorrência de falsos alarmes é justificado devido ao posicionamento das câmeras, iluminação no ambiente e ruídos onde câmeras com resolução melhor facilitaria a classificação e aumentaria a precisão. O sistema teve um resultado considerável com vídeos de baixa resolução sendo útil no auxílio do operador para vigilância de ambientes contribuindo com uma prestação de serviço ideal e aumentando a segurança. Para projetos futuros seria testar o software em *Raspberry Pi* e ver sua performance por se tratar de um microcomputador acoplado a uma câmera USB e tentar aumentar o número de acertos para mais que 90%.

REFERÊNCIAS BIBLIOGRÁFICAS

GUIRADRI, E. R.; MANOLESCU, M., K.;. **Criminalidade e Violência no Brasil**. Artigo para XIII Encontro Latino Americano de Iniciação Científica e IX Encontro Latino Americano de Pós-Graduação – Universidade do Vale do Paraíba.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. ISBN 8574520098.

GONZALEZ, R. C.; WOODS, R. E. 3rd.. **Processamento Digital de Imagens**. Pearson,2011.ISBN 978-0131687288.

STIVANELLO,E. ; MASSON,J.; CASAS,B.; SOARES,F.; CARLESSO,G.**Monitoramento Automático de Ambientes Através de Visão Computacional**.Instituto Federal de Catarina,2014.

BARELLI, Felipe. **Introdução à visão Computacional**.São Paulo,2018.ISBN 9788594188571.

POUBEL, Victor. **As câmeras de vigilância no combate à criminalidade**. 2017. Disponível em <https://extra.globo.com/casos-de-policia/papo-federal/as-cameras-de-vigilancia-no-combate-criminalidade-21406282.html>. Acesso em 23 de Abril de 2018.

MACIEL, L.M.S.; VIEIRA, M.B. **Reconhecimento de ações humanas utilizando histogramas de gradiente e vetores de tensores localmente agregados**.2012

LOPES, D.M.; SOBIERANSKI, A.C.; COMUNELLO,Eros; WANGENHEIM, A.V. **Monitoramento Automatizado de Ambientes**.Disponível em: <https://siaiap32.univali.br/seer/index.php/acotb/article/view/6588/3734>. Acesso Em: 17 maio 2018.

SANTOS, P.T.; STEMMER, R.M.;CASAGRANDE,J.H.B.**Rastreamento de Múltiplos Objetos em cenas de videovigilância baseado no algoritmo de extração vibe e filtro de Kalman**. Disponível em <https://www.ufrgs.br/sbai17/papers/paper_203.pdf>. Acesso em 17 maio de 2018.

TV BARRA VELHA.**Explosão caixas eletrônicos Banco do Brasil - Balneário Piçarras de SC**. Disponível em <https://www.youtube.com/watch?v=4JNRq3xpsbo>. Acesso em 17 maio de 2018.