



ADDETC – Área Departamental de Engenharia Eletrónica e Telecomunicações
e de Computadores

LEIM -Licenciatura Engenharia informática e multimédia

Processamento Imagem e Visão

Trabalho prático 2

Turma:

LEIM-51N

Trabalho realizado por:

Miguel Silvestre N°45101

Miguel Távora N°45102

Docente:

Pedro Jorge

Data: 14/02/2021

Índice

1. INTRODUÇÃO	1
2. DESENVOLVIMENTO	2
1.1 ESTIMAÇÃO DA IMAGEM DE FUNDO E DETECÇÃO PÍXEIS ATIVOS	2
1.2 BINARIZAÇÃO E UTILIZAÇÃO DE OPERADORES MORFOLÓGICOS	4
1.3 DETECÇÃO DE REGIÕES ATIVAS	6
1.4 EXTRAÇÃO DE PROPRIEDADES DAS REGIÕES	7
1.5 CLASSIFICAÇÃO E CORRESPONDÊNCIA DAS REGIÕES	8
1.6 VISUALIZAÇÃO DOS RESULTADOS	9
3. CONCLUSÃO	12
4. BIBLIOGRAFIA	15

Índice ilustrações

Figura 1 - Imagem diferença entre frames	2
Figura 2 - Imagem sem blur	3
Figura 3 - Imagem com blur	3
Figura 4 - Imagem com blur gaussiano	4
Figura 5 - Imagem binarizada	4
Figura 6 - Binarização da imagem sem operador morfológico	5
Figura 7 - Binarização da imagem com operador morfológico	5
Figura 8 - Imagem com operador morfológico	6
Figura 9 - Desenho dos contornos na imagem principal	6
Figura 10 - Contornos relevantes na imagem	7
Figura 11 - Classificação dos objetos	8
Figura 12 - Resultado da classificação 1	9
Figura 13 - Resultado da classificação 2	9
Figura 14 - Resultado da classificação 3	10
Figura 15 - Resultado da classificação 4	10
Figura 16 - Resultado da classificação 5	11

1. Introdução

O segundo trabalho prático da unidade curricular de Processamento de Imagem e Vídeo (PIV) tem como objetivo desenvolver um algoritmo capaz de detetar e classificar zonas do vídeo (sequência de imagens) onde ocorreu movimento.

A área de deteção de movimento em vídeo é uma área que tem vindo a crescer, pois permite uma melhor análise de dados, nomeadamente reconhecimento de pessoas ou objetos, enquanto que sensores de movimento permitem um maior enriquecimento quando comparado a sensores de movimento que permite somente a deteção movimento e não classificação.

O algoritmo desenvolvido simula um sistema de vigilância, capaz de detetar regiões ativas (zonas de movimento) num vídeo. O algoritmo deve ser capaz de detetar o movimento dos objetos que no caso do trabalho são carros, pessoas e outro tipo de objetos.

No acto de deteção dos objetos, no mesmo instante, deve classificar em três categorias distintas que são:

- PESSOA
- CARRO
- OUTRA

Para melhorar a visualização do movimento dos objetos será utilizada uma *bondingbox* e também o desenho do rasto do objeto durante a sequência de imagens.

Toda a implementação será utilizada a linguagem de programação python e a biblioteca de processamento de imagem utilizada será o OpenCV.

2. Desenvolvimento

1.1 Estimação da imagem de fundo e detecção píxeis ativos

Para a detecção de movimento na sequência de vídeo inicialmente é necessário estimar a imagem de fundo. A imagem de fundo é nomeadamente uma imagem no plano temporal onde os objetos existentes na cena não realizam qualquer tipo de movimento. Este fundo pode ser obtido por uma filtragem temporal, nomeadamente diferença entre frames, e por uma filtragem de mediana.

A filtragem temporal visa a remoção ou atenuação de frequências dentro do sinal em bruto, que não são de interesse. A dificuldade da aplicação destes filtros reside em quais são as frequências de interesse e quais as frequências são ruído. A filtragem temporal foi utilizada nomeadamente na diferença entre frames consecutivas, onde é utilizada uma imagem base que será a frame corrente e será comparada com a segunda frames (frame anterior) e onde os píxeis da segunda frame são subtraídos á frame corrente. Desta forma obtem-se assim uma imagem somente com a variação entre imagens, onde os pixéis brancos representam o movimento e os pixéis pretos é onde não existiu movimento. Esta diferença foi implementada pelo função do OpenCV designado `absdiff(frame1, frame2)`, onde retorna a diferença entre imagens.

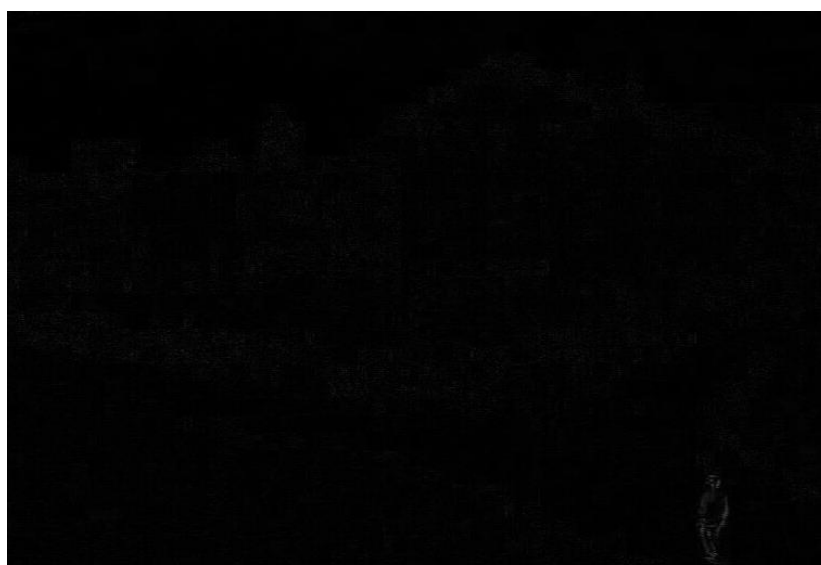


Figura 1 - Imagem diferença entre frames

A filtragem de mediana é uma técnica de filtragem digital não linear, utilizada frequentemente para remover ruído de uma imagem. A redução de ruído é feita na etapa de pré-processamento de uma imagem. Antes da utilização da filtragem de mediana foi utilizado o plano de cor vermelho, pois foram realizados testes e foi o plano de cor que permitiu obter os melhores resultados para a diferença entre frames. Será utilizada a filtragem de mediana no trabalho para o melhoramento da imagem que resulta na diferença entre imagens. A função que permite o melhoramento da imagem é `medianBlur()`. O resultado do melhoramento foi o que se segue:



Figura 2 - Imagem sem blur



Figura 3 - Imagem com blur

1.2 Binarização e utilização de operadores morfológicos

A binarização é a conversão de uma imagem com cores entre valores 0-255 onde são mapeados a partir de um valor pré-definido (*threshold*). Valores inferiores ao *threshold* são mapeados para 0 e valores iguais ou superiores ao *threshold* são postos a 255. A binarização do trabalho prático foi utilizado o *threshold* de 12, foi decidido este valor através de diversos testes e da visualização dos resultados obtidos, onde o melhor valor obtido foi neste caso o valor 12. A binarização foi feita através do método `threshold()` do OpenCV.

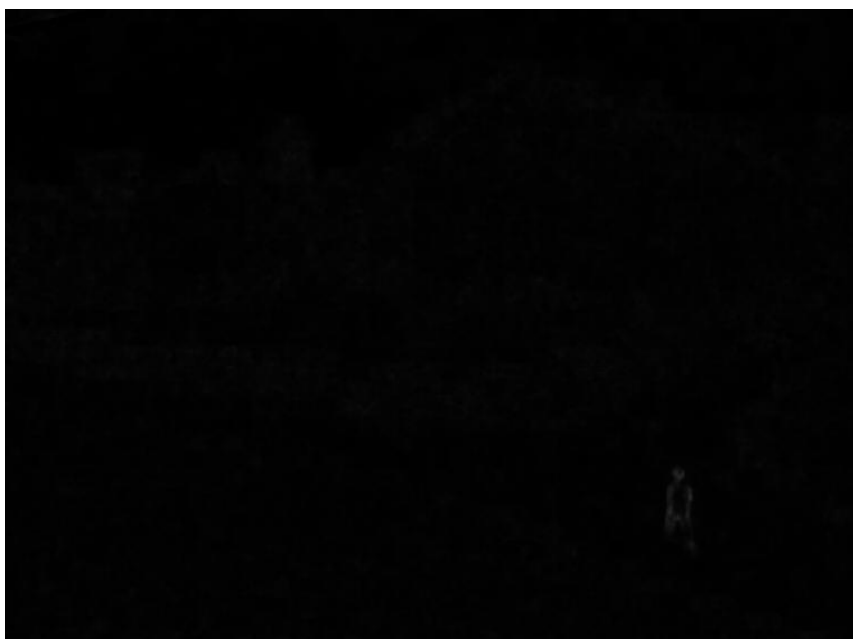


Figura 4 - Imagem com blur gaussiano



Figura 5 - Imagem binarizada

Para o melhoramento da imagem foram aplicados operadores morfológicos de maneira a conseguir melhorar e expandir os contornos obtidos na binarização. Para isso foi utilizado o operador morfológico dilatação que aumenta os contornos de maneira a ser mais visível, isto devido ao facto de os contornos quando convertidos para binário muitas das vezes são muito pequenos e torna-se difícil de posteriormente classificar. O kernel da dilatação foi utilizado um por omissão, visto que o objetivo é somente expandir as formas. Para isso foi utilizada a função `dilate()` do OpenCV.



Figura 6 - Binarização da imagem sem operador morfológico



Figura 7 - Binarização da imagem com operador morfológico

1.3 Detecção de regiões ativas

Para ser possível a deteção de objetos é necessário saber quais são as regiões ativas. As regiões ativas serão nomeadamente as zonas na imagem que representam um objeto, neste caso em movimento. Para a deteção dos objetos nas regiões ativas são encontrados através dos contornos dos objetos. Os contornos podem ser explicados como uma curva que une todos os pontos contínuos (ao longo do limite), com a mesma cor ou com a mesma intensidade. Os contornos são uma ferramenta útil para a análise e deteção da forma e também reconhecimento de objetos. A função do OpenCV que permite reconhecer os contornos designa-se `findContours()`.

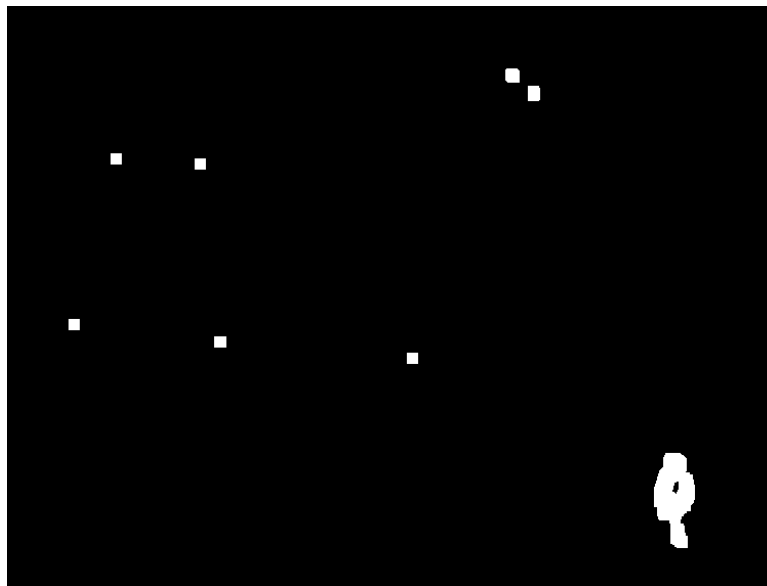


Figura 8 - Imagem com operador morfológico

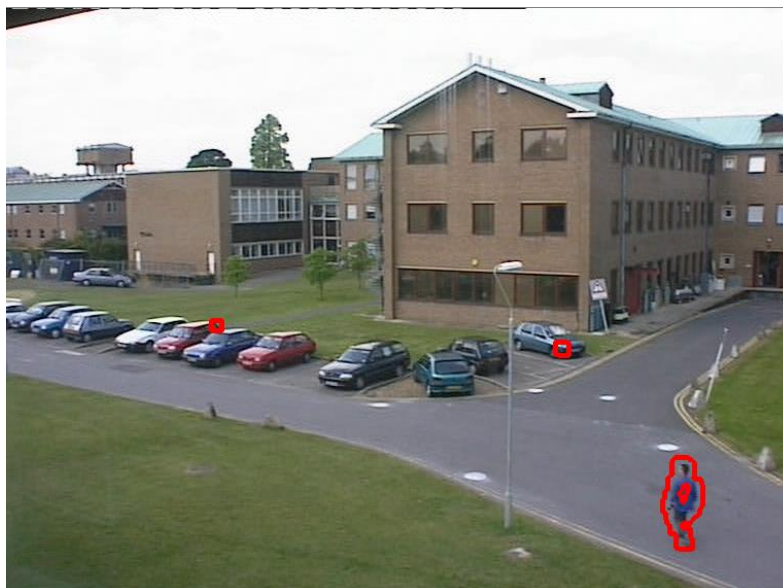


Figura 9 - Desenho dos contornos na imagem principal

1.4 Extração de propriedades das regiões

Para a extração de propriedades e características foram utilizados a área e também um retângulo em volta do contorno. Para obter a característica da área e do retângulo é necessário obter os contornos todos existentes na imagem binarizada. Para obter os contornos é percorrido todos os contornos criados na imagem pela função `findContours()` e são chamadas funções do OpenCV que permite ir buscar nomeadamente a área e também o rectângulo em volta do contorno.

A área do contorno será utilizada para a deteção de contornos de ruído e contornos de interesse, a partir de um determinado limiar é possível através da área saber se um contorno é um contorno de interesse ou não. Contornos que não possuem uma determinada área não são tidos em conta na análise de características, nomeadamente do retângulo do contorno. Esta característica é obtida pela função `contourArea()` do OpenCV.

O retângulo do contorno permite saber o *aspect-ratio* que corresponde a largura a dividir pela altura. Esta informação é conseguida através do quadrado desenhado em volta do contorno. Através de cada retângulo é possível obter a posição em x e y e a altura e largura, pela função `boundingRect()` do OpenCV, com a altura e largura é calculado o *aspect-ratio*. O *aspect-ratio* será a característica utilizada para diferenciar os diferentes objetos.

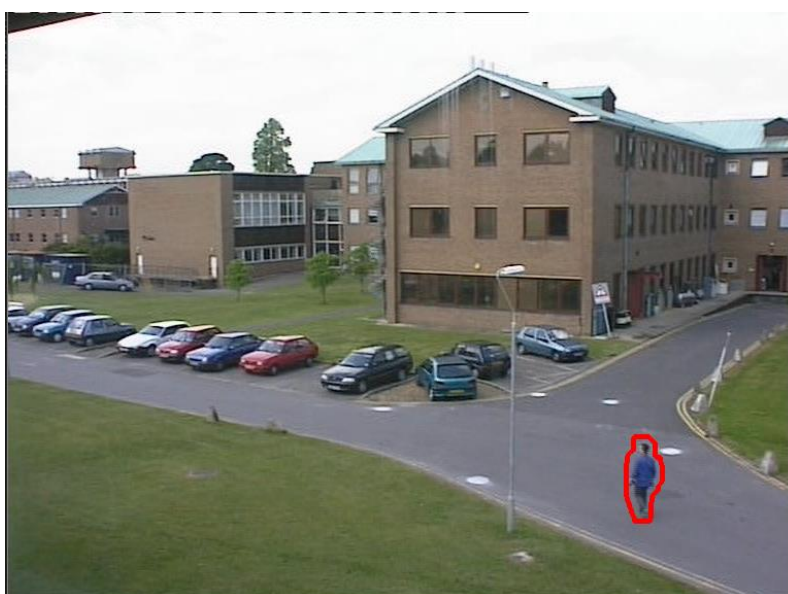


Figura 10 - Contornos relevantes na imagem

1.5 Classificação e correspondência das regiões

A classificação é feita baseado num classificador de distância ao centróide. Esta classificação utiliza como centróides as medianas do *aspect-ratio*. Foi utilizada a mediana porque é uma medida mais robusta quando comparada á média devido aos *out-liners* que acabaram por afetar muito o valor da média.

Para calcular as medianas foram vistas certas partes da gravação do vídeo onde apareciam uma pessoa, um carro ou outra. A partir destas partes da gravação foi guardado num ficheiro pickle estes valores de maneira que podessem ser analisados posteriormente com maior detalhe. O classificador é feito através da distância de Manhattan entre o valor do *aspect-ratio* e as medianas do *aspect-ratio* calculadas entre cada objeto. O valor que der menor de distância será então classificado então nessa determinada classe.



Figura 11 - Classificação dos objetos

1.6 Visualização dos resultados



Figura 12 - Resultado da classificação 1

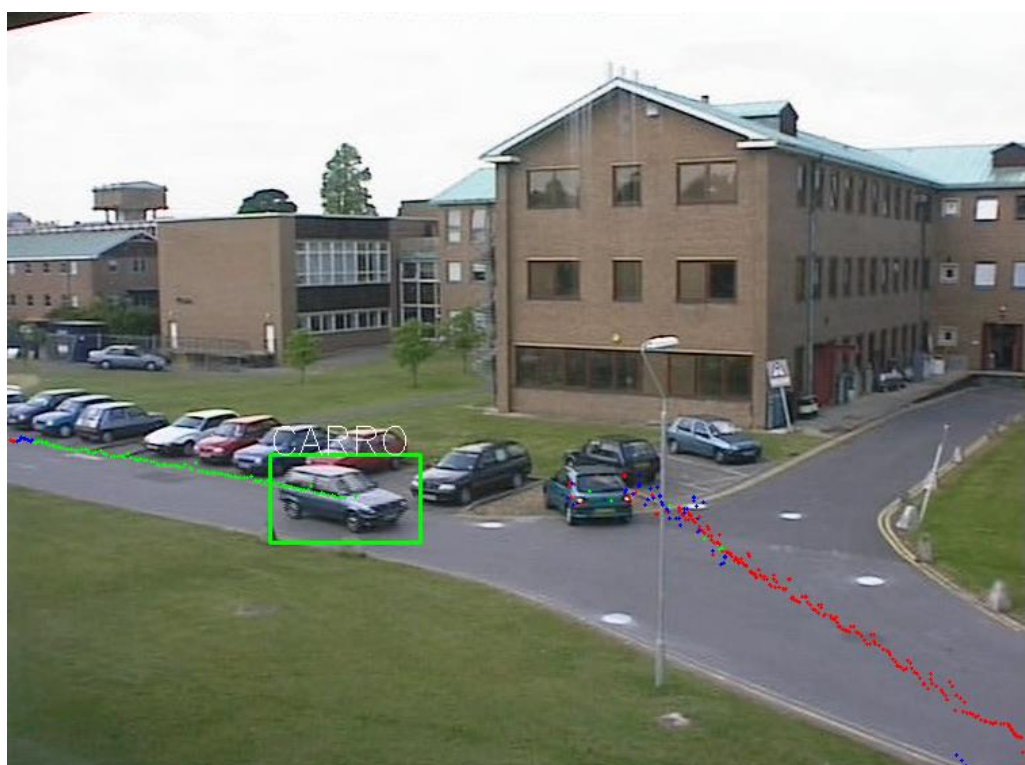


Figura 13 - Resultado da classificação 2

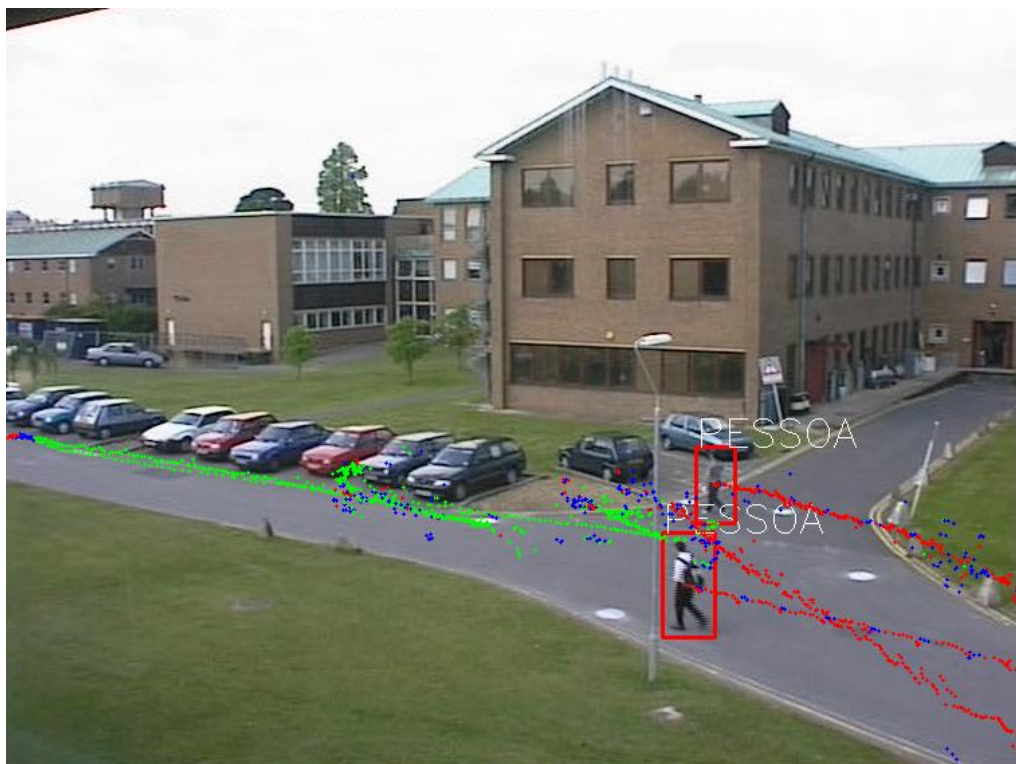


Figura 14 - Resultado da classificação 3

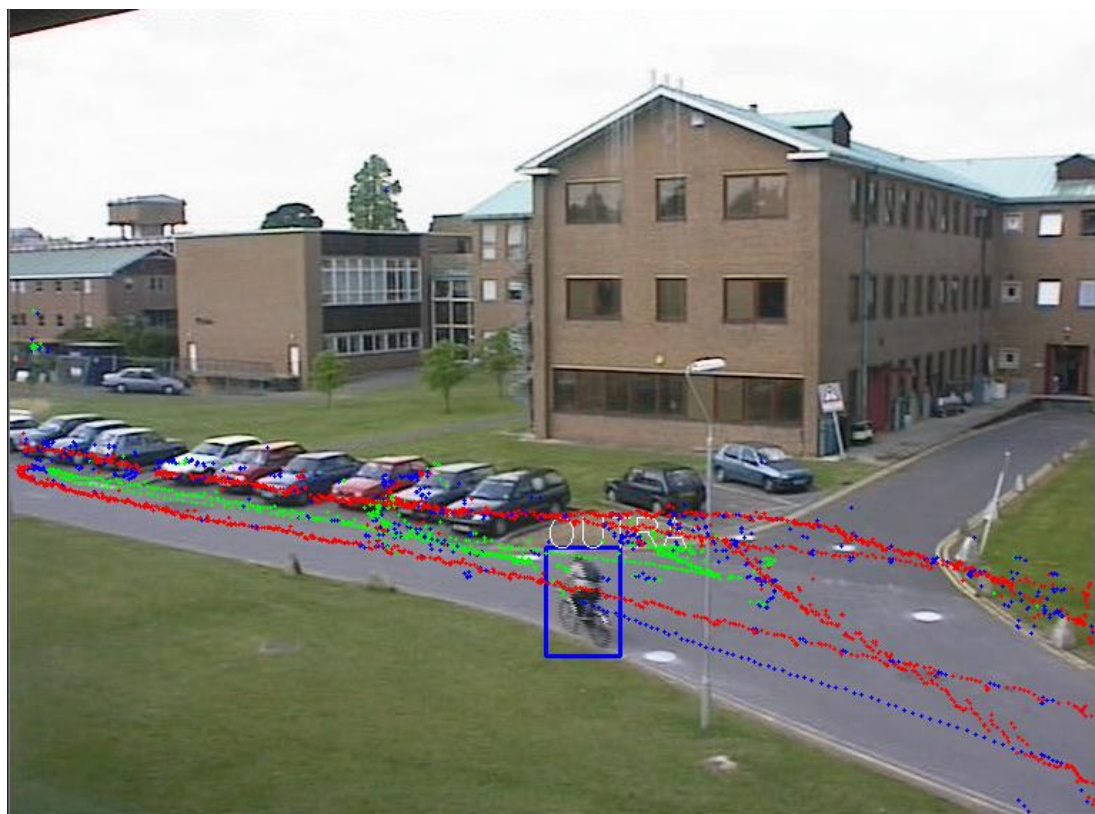


Figura 15 - Resultado da classificação 4

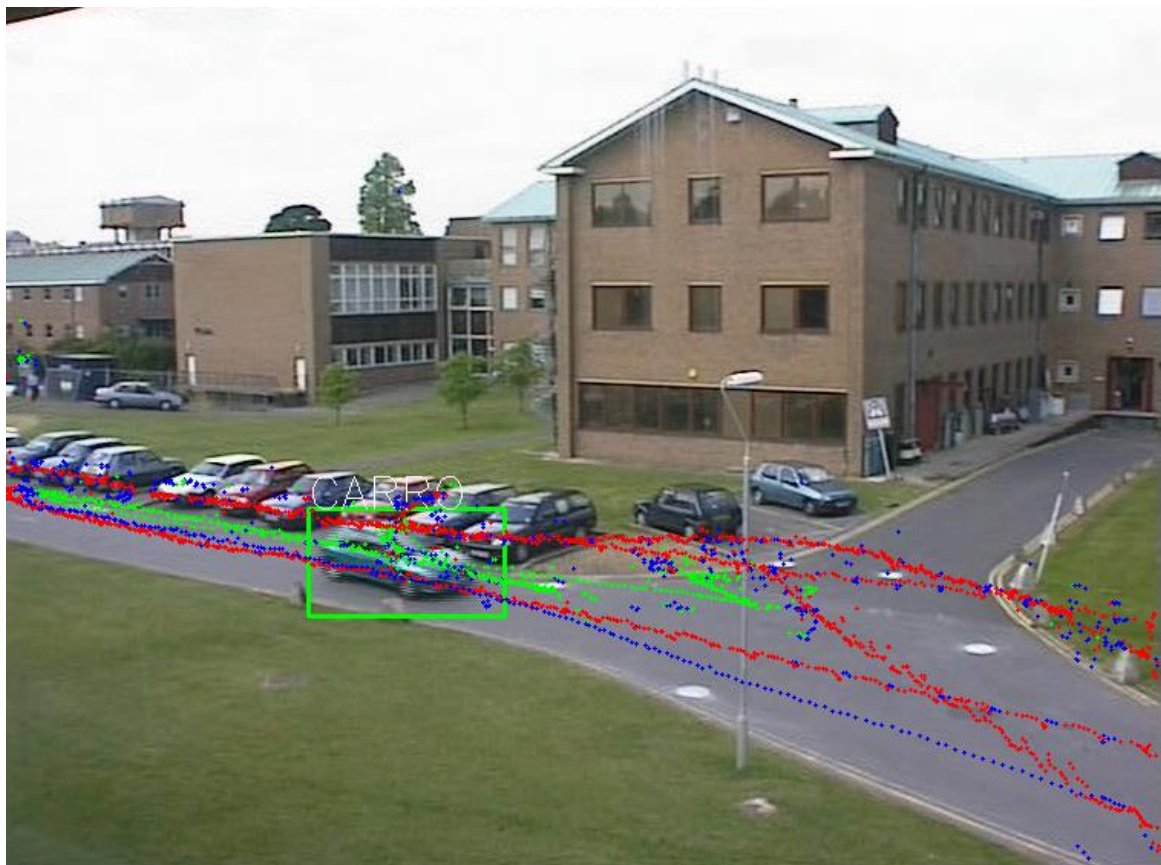


Figura 16 - Resultado da classificação 5

Perante os resultados é possível concluir que a classificação durante todo o vídeo em geral é bem feita, contudo existem alguns momentos onde os objetos realizam pouca movimentação e por isso não são classificados ou são classificados na classe errada. Durante a realização do trabalho o grupo tentou diversas metodologias para melhorar os resultados, contudo existem zonas do vídeo que pela falta de movimentação não é possível grandes resultados.

3. Conclusão

Com o desenvolvimento deste projeto o grupo adquiriu os seguintes conhecimentos:

- Como realizar processamento de imagens numa sequência de vídeos e as aplicações práticas do mesmo.
- Como detetar o movimento de objetos a partir da diferença consecutiva entre frames e como manipular essa diferença.
- Converter a diferença entre imagens para uma binária de maneira a conseguir uma melhor obtenção de características relevantes.
- Utilização de elementos estruturantes para melhorar a qualidade da imagem.
- Encontrar os contornos de uma imagem para ir buscar os píxeis onde existe movimentação entre frames consecutivas.
- Extrair propriedades para verificar se é a forma geométrica pretendida e obter as características relevantes para a classificação.
- Classificar baseado nas propriedades obtidas e desenho da classificação na imagem visualizada.

Durante a realização do trabalho prático foi compreendida a importância de conceitos como:

- Diferença entre frames para deteção de movimento de objetos.
- Escolha do melhor *contour features* e *properties* para classificação do objeto.
- Importância da calibração do classificador para uma boa classificação.

As maiores dificuldades sentiram-se na criação do classificador, onde foram testados diversos classificadores. A característica que possuía melhor classificação era o *aspect-ratio* e por isso foi utilizada essa característica.

Em relação aos resultados o vídeo classifica quase sempre bem exceto em algumas partes do vídeo que existe pouco movimento por parte do objeto. Este problema poderia ser resolvido através de um classificador mais sofisticado e um processamento de movimento para detenção de pouca movimentação entre frames consecutivas.

4. Bibliografia

https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html

https://docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html

https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html