



UNIVERSIDADE EDUARDO MONDLANE

Faculdade de Engenharia

Departamento de Engenharia Electrotécnica

Curso de Engenharia Electrónica

Trabalho de Licenciatura

**APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO DE UM
SEMÁFORO INTELIGENTE PARA OS CRUZAMENTOS DA CIDADE DE MAPUTO**

Discente:

Junga, Abel Mateus

Supervisor:

Mestre, Roxan Cadir, Eng^a

Maputo, Abril de 2022



UNIVERSIDADE EDUARDO MONDLANE

Faculdade de Engenharia

Departamento de Engenharia Electrotécnica

Curso de Engenharia Electrónica

Trabalho de Licenciatura

**APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO DE UM
SEMÁFORO INTELIGENTE PARA OS CRUZAMENTOS DA CIDADE DE MAPUTO**

Discente:

Junga, Abel Mateus

Supervisor:

Mestre, Roxan Cadir, Eng^a

Maputo, Abril de 2022

GUIA DE AVALIAÇÃO DO RELATÓRIO ESCRITO



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA _____

F1 - GUIA DE AVALIAÇÃO DO RELATÓRIO ESCRITO

Nome do estudante: _____

Referência do tema: _____ Data: ___/___/___

Título do tema: _____

1. Resumo					
1.1. Apresentação dos pontos chaves no resumo (clareza, organização, correlação com o apresentado)	1	2	3	4	5
Secção 1 subtotal (max: 5)					

2. Organização (estrutura) e explanação										
2.1. Objectivos	1	2	3	4	5					
2.2. Introdução, antecedentes e pesquisa bibliográfica	1	2	3	4	5	6	7	8	9	10
2.3. Metodologias	1	2	3	4	5	6	7	8	9	10
2.4. Resultados, sua análise e discussão	1	2	3	4	5	6	7	8	9	10
2.5. Conclusões e aplicação dos resultados (recomendações)	1	2	3	4	5	6	7	8	9	10
Secção 2 subtotal(max: 45)										

3. Argumentação										
3.1. Criatividade e originalidade	1	2	3	4	5					
3.2. Rigor	1	2	3	4	5					
3.3. Análise crítica, evidência e lógica	1	2	3	4	5	6	7	8	9	10
3.4. Relação objectivos/ métodos/ resultados/conclusões	1	2	3	4	5					
3.5. Relevância	1	2	3	4	5					
Secção 3 subtotal(max: 30)										

4. Apresentação e estilo da escrita					
4.1. Legibilidade e organização	1	2	3	4	5
4.2. Ilustração e qualidade das figuras e tabelas	1	2	3	4	5
4.3. Estilo da escrita (fluência do texto, uso da língua e gramática)	1	2	3	4	5
4.4. Fontes bibliográficas (citação correcta, referências, etc)	1	2	3	4	5
Secção 4 subtotal(max: 20)					

Total de pontos (max: 100)		Nota (=Total*0,2)	
-----------------------------------	--	--------------------------	--

Nota: Quando exista a componente gráfica (desenhos técnicos), a nota acima é multiplicada por 0,8 cabendo os restantes 20% do peso à referida parte gráfica.

GUIA DE AVALIAÇÃO DA APRESENTAÇÃO ORAL E DEFESA (PELO JÚRI)



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA _____

F2 – GUIA DE AVALIAÇÃO DA APRESENTAÇÃO ORAL E DEFESA

Nome do estudante: _____

Referência do tema: _____ Data: ___/___/___

Título do tema: _____

1. Introdução										
1.1. Apresentação dos pontos chaves na introdução (Contexto e importância do trabalho)	1	2	3	4	5	6	7	8	9	10
Secção 1 subtotal(max: 10)										

2. Organização e explanação										
2.1. Objectivos	1	2	3							
2.3. Metodologia	1	2	3	4						
2.4. Resultados, sua análise e discussão	1	2	3	4	5	6	7	8	9	10
2.5. Conclusões e aplicação dos resultados (recomendações)	1	2	3	4	5	6	7	8		
Secção 2 subtotal(max: 25)										

3. Estilo da apresentação										
3.1. Uso efectivo do tempo	1	2	3	4	5					
3.2. Clareza, tom, vivacidade e entusiasmo	1	2	3	4	5					
3.3. Uso e qualidade dos audio-visuais	1	2	3	4	5					
Secção 3 subtotal(max: 15)										

4. Defesa										
4.1. Exactidão nas respostas	1	2	3	4	5	6	7	8	9	10
4.2. Domínio dos conceitos	1	2	3	4	5	6	7	8	9	10
4.3. Confiança e domínio do trabalho realizado	1	2	3	4	5	6	7	8	9	10
4.4. Domínio do significado e aplicação dos resultados	1	2	3	4	5	6	7	8	9	10
4.5. Segurança nas intervenções	1	2	3	4	5	6	7	8	9	10
Secção 3 subtotal(max: 50)										

Total de pontos (max: 100)		Nota (=Total*0,2)	
-----------------------------------	--	--------------------------	--

FICHA DE AVALIAÇÃO DA ATITUDE DO ESTUDANTE (PELO SUPERVISOR)



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA _____

FICHA DE AVALIAÇÃO DA ATITUDE DO ESTUDANTE (Auxiliar para o supervisor)

Nome do estudante: _____

Referência do tema: _____ Data: ___/___/___

Título do tema: _____

Indicador	Classificação				
	1	2	3	4	5
Atitude geral (manteve uma disposição positiva e sentido de humor)	1	2	3	4	5
Dedicação e comprometimento (Deu grande prioridade ao projecto e aceitou as responsabilidades prontamente)	1	2	3	4	5
Independência (realizou as tarefas independentemente, como prometido e a tempo)	1	2	3	4	5
Iniciativa (viu o que devia ter sido feito e fê-lo sem hesitar e sem pressões do supervisor)	1	2	3	4	5
Flexibilidade (disponibilidade para se adaptar e estabelecer compromissos)	1	2	3	4	5
Sensibilidade (ouviu e tentou compreender as opiniões dos outros)	1	2	3	4	5
Criatividade (contribuiu com imaginação e novas ideias)	1	2	3	4	5
Total de pontos (max: 35)					

Valor do classificador	Cotação obtida	Significado
	1	Não aceitável (0 a 9 valores)
	2	Suficiente (10 a 13 valores)
	3	Bom (14 a 16 valores)
	4	Muito Bom (17 a 18 valores)
	5	Excelente (19 a 20 valores)

Total de pontos (max: 35) _____

Nota (=Total*20/35) _____

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer a Deus, pela saúde e força para a realização deste trabalho.

Aos meus familiares, especialmente aos meus tios nomeadamente, Lina António Dimbane Pode, António Máquina e Deolinda Máquina, que sempre foram a minha fonte de inspiração e me apoiaram de forma incondicional, acreditando no meu potencial.

A minha supervisora Roxan Cadir, pelo incentivo e todo conhecimento transmitido ao longo desses anos.

E por fim aos meus colegas e amigos, que diretamente ou indiretamente, fizeram parte desta grande caminhada, apoiando-me e incentivando-me a batalhar.

RESUMO

Segundo o Instituto Nacional de Estatística (Estatísticas de Acidentes de Viação, 2020)¹, em Moçambique e no mundo, tem-se verificado um aumento rápido de importação e uso de automóveis. Como resultado, aumentaram os problemas de tráfego nestes últimos anos e o semáforo controlador actual tem limitações porque usa um hardware predefinido que não tem a flexibilidade de modificação em tempo real. Devido aos intervalos fixos do tempo de verde, laranja e vermelho o tempo de espera é longo. Este trabalho visa apresentar o desenvolvimento de um controle de semáforo, usando inteligência artificial, que melhora a organização do tráfego, regulando os tempos de abertura do semáforo de acordo com o fluxo de veículos, com o intuito de reduzir ao máximo o atraso veicular. Em comparação com o controlador de semáforo em modo fixo, este novo sistema é mais eficiente e flexível.

Palavras-chaves: Tráfego, semáforo, inteligência artificial.

ABSTRACT

According to the national statistics institute, in Mozambique and worldwide, there has been a rapid increase in the importation and use of automobiles. As a result, traffic problems have increased in recent years and the current traffic light controller has limitations because it uses a predefined hardware that does not have the flexibility of real-time modification. Due to the fixed intervals of green, orange and red time the waiting time is long. This paper aims to present the development of a traffic light control, using artificial intelligence, that improves traffic organization by regulating the traffic light opening times according to the flow of vehicles, in order to reduce vehicular delay as much as possible. Compared to the fixed-mode traffic light controller, this new system is more efficient and flexible.

Key-words: Traffic, traffic light, artificial intelligence.

¹ http://www.ine.gov.mz/estatisticas/estatisticas-sectoriais/acidentes-de-viacao/estatisticas-de-acidentes-de-viacao-2020/at_download/file, pag. 11-13.

ÍNDICE

AGRADECIMENTOS	VI
RESUMO	VII
ÍNDICE DE TABELAS	III
ÍNDICE DE FIGURAS	IV
LISTA DE ABREVIATURAS E SIGLAS	V
DECLARAÇÃO DE HONRA	VI
1. INTRODUÇÃO.....	1
1.1. Introdução.....	1
1.2. Definição da Problemática.....	2
1.3. Justificativa.....	2
1.4. Objectivos.....	2
1.4.1. Objectivo Geral	2
1.4.2. Objectivos Específicos.....	2
1.5. Metodologia.....	2
1.6. Estrutura do Trabalho	3
2. REVISÃO DE LITERATURA	4
2.1. Conceitos básicos	4
2.1.1. Intersecções	4
2.1.2. Semáforos	6
2.1.3. Regulagem de Semáforos	7
2.2. Processamento de Imagem e Visão Computacional.....	11
2.2.1. Open CV (Open-Source Computer Vision).....	11
2.2.2. Inteligência Artificial	12
2.2.3. Aprendizado de Máquina.....	12
2.2.4. Neurónios	12
2.2.5. Redes Neurais Convolucionais (CNN).....	13
2.2.6. YOLO (You Only Look Once)	18

3.	IMPLEMENTAÇÃO	22
4.	TESTES E ANÁLISE DOS RESULTADOS	25
4.1.	Testes de detecção e contagem de veículos	25
4.2.	Análise dos Resultados	27
5.	CONCLUSÕES E RECOMENDAÇÕES	28
	REFERÊNCIAS BIBLIOGRÁFICAS	29
	ANEXOS	A
	APÊNDICE	G

ÍNDICE DE TABELAS

Tabela 1: Valores de Fluxo de Saturação para larguras de vias inferiores a 5.5 metros	7
Tabela 2: Teste de detecção de veículos usando o modelo YOLOv3-tiny	26
Tabela 3: Teste de detecção de veículos usando o modelo YOLOv3.....	26

ÍNDICE DE FIGURAS

Figura 1: Movimento Conflitantes em um cruzamento	5
Figura 2: Semáforo.....	6
Figura 3: Estrutura de um neurónio	13
Figura 4: Rede neural convolucional	14
Figura 5: Exemplos de filtros de convolução	15
Figura 6: Acções que ocorrem na camada de convolução.....	16
Figura 7: Tipos de agrupamento	17
Figura 8: Arquitectura completa de uma CNN	18
Figura 9: Divisão em grade	19
Figura 10: Caixa delimitadora e confiança	20
Figura 11: Combinação das três técnicas	21
Figura 12: Fluxograma do programa desenvolvido	23
Figura 13: Detecção de veículos	23
Figura 14: Linha de colisão	24
Figura 15: Imagem do vídeo obtido na internet.....	25
Figura 16: Modelo YOLOv3-tiny	26
Figura 17: Modelo YOLOv3.....	27

LISTA DE ABREVIATURAS E SIGLAS

COCO	Common Object in Context
CNN	Convolutional Neural Network
DENATRAN	Departamento Nacional de Trânsito
IOU	Intersection Over Union
OpenCv	Open-Source Computer Vision
ReLU	Rectified Linear Unit
YOLO	You Only Look Once

DECLARAÇÃO DE HONRA

Eu, Abel Mateus Junga declaro por minha honra que o presente Trabalho de Licenciatura é exclusivamente de minha autoria, não constituindo cópia de nenhum trabalho realizado anteriormente e as fontes usadas para a realização do trabalho encontram-se referidas na bibliografia.

Assinatura: _____

1. INTRODUÇÃO

1.1. Introdução

A gestão do tráfego tornou-se um problema grave hoje em dia. Por causa do aumento rápido da importação e uso de automóveis, houve um grande crescimento do tráfego.

Com o aumento do tráfego surgem inúmeros problemas, como grandes congestionamentos, violação das regras de trânsito, etc. A má gestão do congestionamento de tráfego também resulta em longos tempos de espera, perda de combustível e dinheiro. Portanto, é necessário ter um controle de tráfego rápido, económico e eficiente. Uma das formas de melhorar o fluxo de tráfego e a segurança do transporte do sistema actual é aplicar o controle inteligente às infraestruturas rodoviárias.

Os problemas mais comuns dos semáforos convencionais são:

Grandes congestionamentos

Com o aumento do número de veículos nas estradas, aumentaram os congestionamentos nas cidades. Isto acontece nos principais cruzamentos geralmente nas horas de ponta (pela manhã e ao fim do dia laboral). O principal efeito disso é o aumento da perda de tempo das pessoas na estrada. A solução para este problema é desenvolver um sistema inteligente capaz de definir mais tempo à via com maior fluxo de veículos e menos tempo parar a via com menor fluxo em tempo real.

Esperas desnecessárias

Em certos cruzamentos, mesmo quando não há tráfego, os condutores devem esperar. Porque a luz do semáforo permanece vermelha durante um período de tempo predefinido, e os condutores devem esperar até que a luz fique verde. A solução para este problema é desenvolver um sistema inteligente capaz de manter a via com fluxo aberta e manter fechada a via sem fluxo de carros em tempo real.

1.2. Definição da Problemática

O elevado índice de veículos nas estradas e dos transeuntes que não obedecem a passagem pela passadeira, assim como as ruas que não foram projectadas para uma grande quantidade de veículos como se vê nos dias de hoje, causam grandes congestionamentos, principalmente nas horas de ponta. Estes factores podem também causar acidentes por falta de espaço na via pública, criando assim esperas desnecessárias e dificuldade de locomoção dos veículos nas estradas.

1.3. Justificativa

Diante deste problema, foi desenvolvido este projecto para melhorar o controlo do trânsito, potencializando o fluxo de veículos, minimizando o congestionamento e evitando esperas desnecessárias em relação ao tempo de abertura do semáforo.

1.4. Objectivos

1.4.1. Objectivo Geral

Projectar um semáforo inteligente com recurso à inteligência artificial.

1.4.2. Objectivos Específicos

- Fazer uma investigação sobre os actuais modelos de semáforos;
- Implementar um sistema para ser utilizado por semáforos na determinação do fluxo de veículos;

1.5. Metodologia

Para a realização deste trabalho, foi dividido o trabalho da seguinte maneira:

- Pesquisa bibliográfica em livros, artigos, tutoriais e *sites* na internet sobre os conceitos de Engenharia do Tráfego, intersecções entre outros, que tratam do assunto em causa, com a finalidade de buscar maior entendimento do funcionamento e desempenho;
- Elaboração do diagrama do sistema, que servirá para orientar a implementação.

- Desenvolvimento de algoritmos necessários para execução do projecto. Todo código será feito no ambiente de desenvolvimento integrado *Google Colab*, usando a linguagem de programação de *python*, com *framework* de rede neural *Tensorflow* e *Open Cv*, com modelos pré-treinados da COCO;
- Testes da simulação no ambiente virtual para avaliar o desempenho e verificar o atendimento dos requisitos.
- Infelizmente a recolha de dados no Conselho Municipal da Cidade de Maputo, para auxiliar na modelagem do projecto, não foi atendida.

1.6. Estrutura do Trabalho

Este trabalho apresenta a seguinte estrutura:

O capítulo 1 apresenta a definição do problema, a justificativa, os objectivos gerais e específicos do trabalho, bem como a metodologia usada.

O capítulo 2 contém a revisão literária, ou seja, visa apresentar as principais teorias necessárias para a compressão dos elementos propostos na solução do problema.

O capítulo 3 mostra os detalhes sobre implementação do sistema proposto.

O capítulo 4 visa apresentar os resultados dos testes feitos de detecção e contagem de veículos, os seus erros e o resultado de desempenho do hardware.

O capítulo 5 e último deste trabalho, contém as considerações finais, visando mostrar a correspondência com os objectivos definidos, e as recomendações para os trabalhos futuros.

2. REVISÃO DE LITERATURA

2.1. Conceitos básicos

Neste capítulo, serão apresentados os aspectos mais importantes para a elaboração do projecto.

Aproximações de um cruzamento são trechos de via que convergem para uma intersecção.

Ciclo é o tempo em que a sequência verde, amarelo e vermelho aparece por completo.

Fase é a parte do ciclo reservada à uma combinação qualquer de movimentos de tráfego que recebem, simultaneamente, o direito de passagem.

Atraso é o tempo perdido num semáforo, quando mantido pelo sinal vermelho.

2.1.1. Intersecções

Uma intersecção é uma junção de nível onde dois ou mais fluxos de veículos se encontram ou se cruzam.

Em um cruzamento entre duas ou mais vias, como mostra a figura 1, existem movimentos que não podem ser realizados simultaneamente, pois são conflitantes entre si. (DENATRAN, 1984)

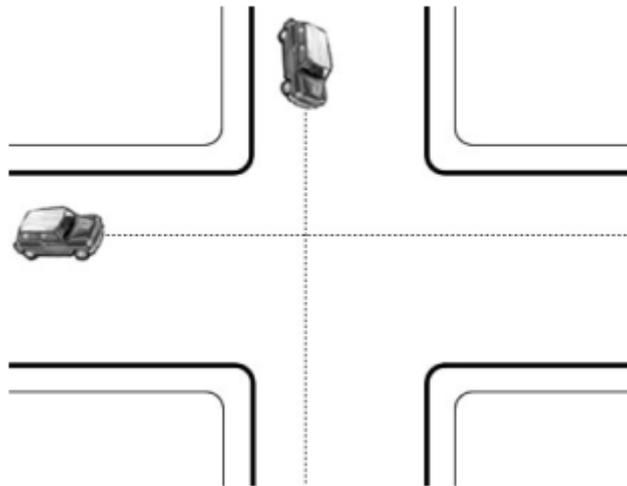


Figura 1: Movimento Conflitantes em um cruzamento

Fonte: DENATRAN

Segundo (DENATRAN 1984), em vias que apresentam baixo fluxo de tráfego, os conflitos entre vários veículos são facilmente resolvidos pela regra do primeiro a chegar é o primeiro a atravessar. No entanto, em ruas um pouco mais movimentadas essa regra não é obedecida, sendo necessário estabelecer regras de prioridade entre as aproximações do cruzamento para permitir a travessia da intersecção.

Existem três modos convencionais de controle de intersecções que são nomeadamente:

Controlo Manual: o controlo manual como o nome sugere requer mão-de-obra para controlar o tráfego. A polícia de trânsito é designada para uma área necessária para controlar o tráfego. A polícia de trânsito carrega uma luz de sinalização e um apito para controlar o tráfego.

Controlo Automático: o semáforo automático é controlado por temporizadores. No semáforo, um valor numérico constante é carregado no cronómetro, as luzes são ligadas e desligadas automaticamente com base no valor do temporizador.

Sensores Eletrónicos: este método consiste em colocar alguns detectores ou sensores de proximidade na estrada. Este sensor fornece dados sobre o tráfego na estrada, e de acordo com esses dados do sensor, os sinais de trânsito são controlados.

2.1.2. Semáforos

Semáforo é um dispositivo de sinal de trânsito que controla o tráfego de veículos e pedestres nas estradas, através de indicações luminosas.

O semáforo inteligente é um sistema com inteligência para captar imagens e uma grande quantidade de informações, e consegue entender o fluxo do trânsito em tempo real para decidir, por quanto tempo este deve ficar fechado ou aberto, baseado no diagnóstico que faz as vias que monitora ao longo dos dias (ARAUJO, 2006).

O semáforo geralmente é composto por três focos de luzes de cores distintas, nomeadamente, a cor verde, a cor amarela e a cor vermelha, como mostra a figura 2. Cada uma delas tem um significado, conforme descrito a seguir:

Cor verde: indica aos condutores de veículos que o cruzamento está livre para a passagem, ou seja, podem seguir em frente.

Cor amarela: indica aos condutores de veículos que a passagem está prestes a ser fechada, ou seja, é uma chamada de atenção.

Cor vermelha: indica aos condutores de veículos que a passagem pelo cruzamento está proibida, ou seja, deve parar o veículo, antes de entrar no cruzamento.

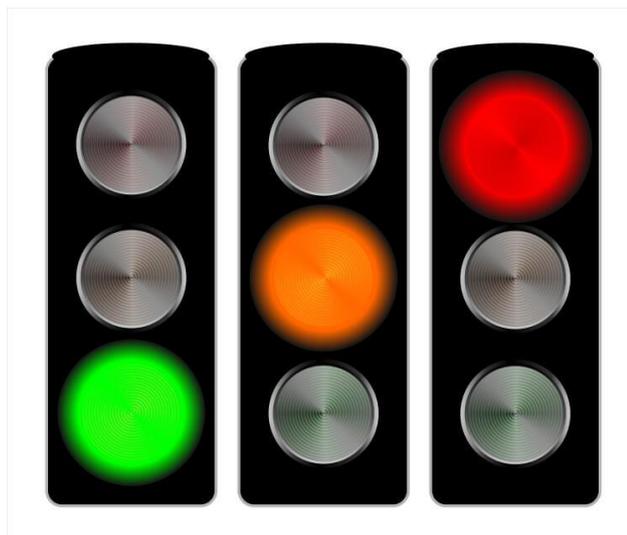


Figura 2: Semáforo

Fonte: <https://images.app.goo.gl/uBe4TrXkU8GBhtTL9>

2.1.3. Regulagem de Semáforos

Para os semáforos de tempos não fixos é necessário fazer a verificação constantemente da demanda nas aproximações e calcular o tempo em verde que as vias necessitam. Para o cálculo do verde é necessário saber a quantidade de veículos que estão na via.

Dentre vários métodos de dimensionamento de semáforos insolados foi proposto o Método de Webster como técnica padrão. Foi escolhido este método por se tratar de um método completo e detalhado, que possibilita a determinação dos tempos de verde e do ciclo óptimo, causando menor atraso possível nos cruzamentos.

2.1.1.1. Fluxo de Saturação

Segundo (DENOTRAN, 1984), fluxo de saturação é definido como sendo o fluxo que seria obtido se houvesse uma fila de veículos na aproximação e a ela fossem dados 100% de tempo de verde do cruzamento (escoamento ininterrupto).

$$S = 525L \quad (1)$$

Onde:

S é o fluxo de saturação (veículo/horas);

L é a largura da aproximação (metros).

A equação 1 é válida para larguras de vias compreendidas entre 5.5 metros e 18.0 metros. Para valores inferiores a 5.5 metros, o fluxo de saturação é dado pela tabela 1.

Tabela 1: Valores de Fluxo de Saturação para larguras de vias inferiores a 5.5 metros

L(m)	3	3.3	3.6	3.9	4.2	4.5	4.8	5.2
S(veq/h)	1850	1875	1900	1950	2075	2250	2475	2700

2.1.1.2. Fluxo de aproximação

Fluxo de aproximação é o número de veículos que passam em uma determinada via, durante um dado período de tempo considerado.

$$q_i = \frac{n}{t} \quad (2)$$

Onde:

q_i é fluxo de aproximação (Veículo/horas);

n é o número de veículos;

t é o tempo (horas).

2.1.1.3. Taxa de ocupação

A taxa de ocupação é uma medida absoluta da solicitação de tráfego em uma aproximação, e é dada como sendo a relação entre o fluxo de aproximação e o fluxo de saturação.

$$y_i = \frac{q_i}{S_i} \quad (3)$$

Onde:

y_i é a taxa de ocupação;

q_i é o fluxo de aproximação (Veículo/horas);

S_i é o fluxo de saturação (Veículo/horas).

A taxa de ocupação crítica

A taxa de ocupação crítica é a taxa máxima entre aproximações de uma fase.

$$y_i^* = y_i, \max \quad (4)$$

Onde:

y_i^* é a taxa de ocupação crítica para cada fase;

$y_{i, \max}$ é a taxa de ocupação máxima entre as aproximações de uma fase.

Taxa de ocupação da intersecção

A taxa de ocupação da intersecção é o somatório das taxas de ocupação crítica de cada fase.

$$Y = \sum y_i^* \quad (5)$$

Onde:

Y é a taxa de ocupação da intersecção;

y_i^* é a taxa de ocupação crítica para cada fase.

2.1.1.4. Tempo perdido

Uma parte do início de cada período verde e uma parte de cada mudança amarela, não são utilizáveis pelos veículos. A soma desses dois valores compreende ao tempo perdido total.

$$T_p = I_1 + I_2 \quad (6)$$

Onde:

T_p é o tempo perdido total (segundos);

I_1 é o tempo perdido inicial (segundos);

I_2 é o tempo perdido final (segundos);

2.1.1.5. Tempo de ciclo ótimo

Tempo de ciclo ótimo é o tempo em que a sequência verde, amarelo e vermelho aparece por completo.

$$C_o = \frac{1.5T_p + 5}{1 - Y} \quad (7)$$

Onde:

C_o é o tempo de ciclo ótimo;

T_p é o tempo perdido (segundos)

Y é a taxa de ocupação.

Considera-se um tempo de ciclo mínimo de 25 segundos por questões de segurança, e máximo tempo de ciclo de 120 segundos para minimizar o atraso e as frustrações dos motoristas.

2.1.1.6. Tempo de verde

Tempo de verde efectivo

Tempo de verde efectivo é o tempo de verde necessário para escoar com o fluxo de saturação o número máximo possível de veículos acumulados por ciclo. (DENATRAN, 1984)

$$g_{ef} = y_i^* \left(\frac{C_o - T_p}{Y} \right) \quad (8)$$

Onde:

g_{ef} é o tempo de verde efectivo;

y_i^* é a taxa de ocupação máxima para cada fase;

C_o é o tempo de ciclo ótimo (segundos);

T_p é o tempo perdido (segundos);

Y é a taxa de ocupação da interseção.

Tempos de verde real

$$g = g_{ef} + T_p - t_a \quad (9)$$

Onde:

g é o tempo de verde actual (segundos);

g_{ef} é o tempo de verde efectivo (segundos);

T_p é o tempo perdido (segundos);

t_a é o tempo de amarelo (segundos), geralmente tem valor entre 2 e 3 segundos.

2.2. Processamento de Imagem e Visão Computacional

Visão computacional é um campo de inteligência artificial que permite ao computador interpretar e entender o mundo visual através de informações significativas de imagens digitais, vídeos e outras entradas visuais, realizando acções ou recomendações com base nessas informações. A visão computacional treina máquinas capazes de identificar e classificar objectos e a reagir a eles da mesma forma que os humanos. Mas tem que fazer isso em muito menos tempo com câmara, dados e algoritmos.

2.2.1. Open CV (Open-Source Computer Vision)

OpenCV é a biblioteca de código aberto líder para visão computacional, processamento de imagens e aprendizado de máquina, e agora apresenta aceleração de GPU para operação em tempo real.²

O OpenCV é lançado sob uma licença BSD e, portanto, é gratuito para uso académico e comercial. Possui interfaces C++, C, Python e Java e suporta Windows, Linux, Mac OS, iOS e Android. O OpenCV foi projectado para eficiência

² <https://developer.nvidia.com/opencv>, 23 de fevereiro de 2022.

computacional e com forte foco em aplicações de tempo real. Escrito em C/C++ otimizado, a biblioteca pode tirar proveito do processamento multi-core. Em todo o mundo, o OpenCV tem mais de 47 mil utilizadores e número estimado de downloads superior a 6 milhões. O uso varia de arte interactiva a inspeção de minas, costura de mapas na web ou através de robótica avançada.³

2.2.2. Inteligência Artificial

Desde o desenvolvimento do computador digital na década de 1940, foi demonstrado que os computadores podem ser programados para realizar tarefas muito complexas, por exemplo, como descobrir provas para teoremas matemáticos ou jogar xadrez com grande proficiência.⁴

Inteligência artificial é a capacidade de um computador digital ou robô controlado por computador para realizar tarefas comumente associadas a seres inteligentes. O termo é frequentemente aplicado aos sistemas de processos intelectuais característicos dos humanos, com a capacidade de raciocinar descobrir significados, generalizar ou aprender com a experiência passada.⁵

2.2.3. Aprendizado de Máquina

Aprendizado de máquina é um campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados (Arthur Samuel, 1959).

2.2.4. Neurónios

Os neurónios ou células nervosas, são células do sistema nervoso que estão relacionadas com a propagação do impulso nervoso. Como mostra a figura 2, os neurónios apresentam três partes básicas: os dendritos, o axónio e o corpo celular.

³ <https://developer.nvidia.com/opencv>, 23 de fevereiro de 2022.

⁴ <https://www.socialismocriativo.com.br/inteligencia-artificial-usos-nos-sistemas-produtivos-e-impactos-sobre-o-mundo-do-trabalho/>, 24 de fevereiro de 2022

⁵ <https://www.socialismocriativo.com.br/inteligencia-artificial-usos-nos-sistemas-produtivos-e-impactos-sobre-o-mundo-do-trabalho/>, 24 de fevereiro de 2022

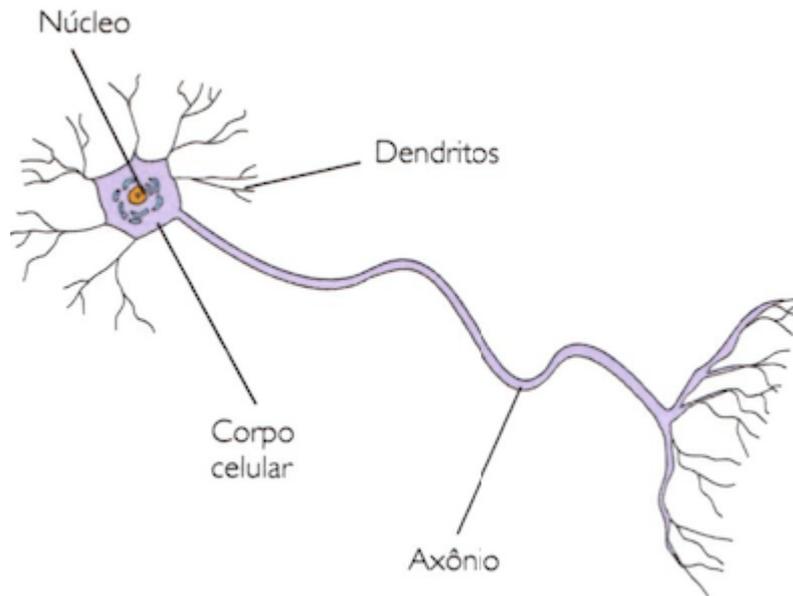


Figura 3: Estrutura de um neurônio

Fonte: <https://images.app.goo.gl/XbtQXVwcA7w98WdG6>⁶

Um sinal de um neurônio é passado ao terminal do axônio e aos dendritos de outro neurônio conectado, que o recebe e o passa pelo corpo celular, axônio e o terminal e assim por diante.

2.2.5. Redes Neurais Convolucionais (CNN)

Uma rede neural convolucional (CNN ou ConvNet), é uma arquitetura de rede para aprendizado profundo que aprende diretamente dos dados, eliminando a necessidade de extração manual de recursos. É usado para reconhecimento de imagens, classificações de imagens, detecções de objetos, reconhecimento de rostos. Eles também podem ser bastante eficazes para classificar dados que não são de imagem, como áudio, séries temporais e dados de sinal.

2.2.5.1.1. Como funcionam as CNN's?

Uma rede neural convolucional pode ter dezenas ou centenas de camadas que aprendem a detectar diferentes características de uma imagem. Filtros são aplicados a cada imagem de treinamento em diferentes resoluções, e a saída de cada imagem

⁶ Visto em 14 de dezembro de 2021.

convolvida é usada como entrada para a camada seguinte. Os filtros podem começar como recursos muito simples, como brilho e bordas, e aumentar em complexidade para recursos que definem exclusivamente o objeto.

Uma CNN é composta por uma camada de entrada, uma camada de saída e muitas camadas ocultas entre elas, como mostra a figura abaixo.

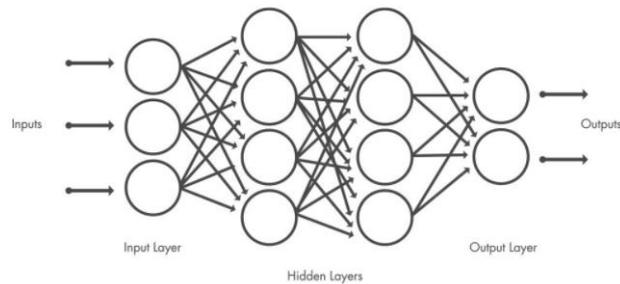


Figura 4: Rede neural convolucional

Fonte: <https://images.app.goo.gl/RDAkbWqUx3hLLEML6>⁷

Essas camadas executam operações que alteram os dados com a intenção de aprender recursos específicos dos dados. Três das camadas mais comuns são: convolução, ativação ou ReLU e pooling.

Camada de Convolução

Convolução é a primeira camada a extrair recursos de uma imagem de entrada. A convolução preserva a relação entre os pixels aprendendo os recursos da imagem usando pequenos quadrados de dados de entrada.

Um filtro, também chamado de *kernel*, também é uma matriz com valores. Seus valores dependem do objetivo desse filtro. Os filtros mais conhecidos são filtros para detecção de bordas (*edge detection*), desfoque (*blur*), nitidez (*sharpen*). (Victor Rodrigues, 2019)

⁷ Visto em 16 de fevereiro de 2022.

Detecção de bordas	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Nitidez	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Desfoque	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Figura 5: Exemplos de filtros de convolução

Fonte: Victor Rodrigues, 2019⁸

Considere uma matriz 7 x 7 cujos valores de pixel de imagem são 0, 1, 2 e matriz de filtro 3 x 3 como mostrado na figura 6.

Um filtro é uma matriz usada para operações de multiplicação de matrizes, para cada caso, o elemento do filtro é multiplicado pelo elemento da mesma posição na qual o filtro está sendo aplicado em cada região naquele instante, depois é realizada a soma dos produtos obtidos, restando um único valor que representa o quanto a região foi activada pelo filtro.

⁸ <https://vitorborbarodrigues.medium.com/conhecendo-a-vis%C3%A3o-do-computador-redes-neurais-convolucionais-e1c2b14bf426>, 16 de fevereiro de 2022.

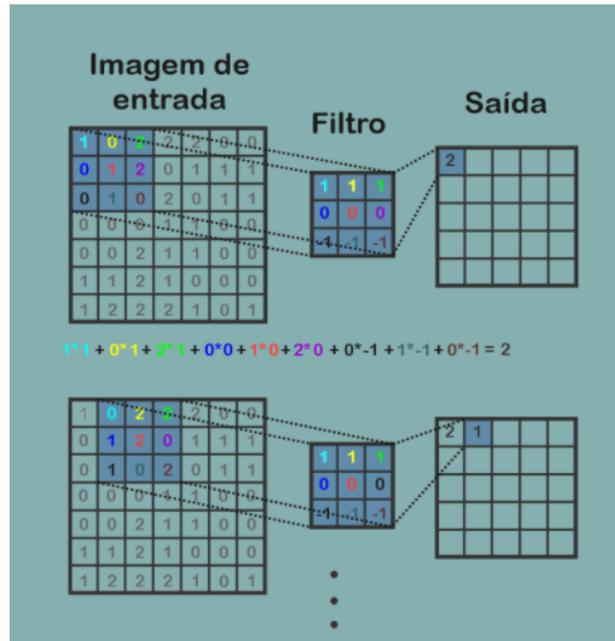


Figura 6: Acções que ocorrem na camada de convolução

Fonte: <https://images.app.goo.gl/17xsaZCg9iviMshG6>⁹

Camada de Unidade Linear Rectificada (ReLU)

A camada de activação é uma função de activação que permite um treinamento mais rápido e eficaz mapeando valores negativos para zero e mantendo valores positivos. Isso às vezes é chamado de activação, porque apenas os recursos activados são transportados para a próxima camada.

Por exemplo, suponha que um valor de célula deva ser idealmente 1, porém tem um valor de 0,85, e uma vez que não se pode alcançar uma probabilidade de 1 em CNN, logo, aplica-se uma função de activação. Se o valor da célula for maior que 0,7, a função de activação atribua o valor de 1, caso contrário, atribua o valor de 0. Desta forma, pode-se facilmente obter uma imagem com características nítidas.

Agrupamento

A secção de camadas de agrupamento reduz o número de parâmetros quando as imagens são muito grandes, simplifica a saída reduzindo o número de parâmetros

⁹ Visto em 16 de fevereiro de 2022.

que a rede precisa aprender, mantendo assim o processo de treinamento efectivo do modelo.

O agrupamento pode ser de vários tipos, como por exemplo, agrupamento máximo, agrupamento da média e agrupamento de soma.

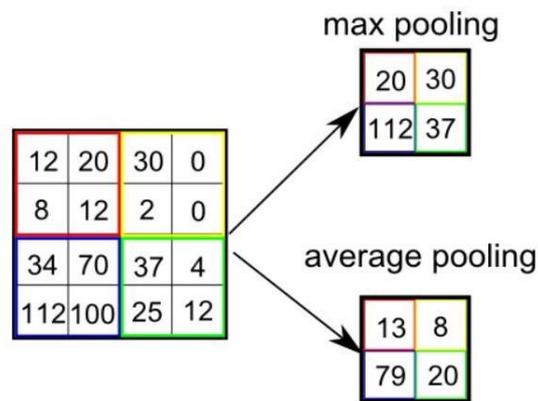


Figura 7: Tipos de agrupamento

Fonte: Sumit Saha, 2018¹⁰

O agrupamento máximo pega o maior elemento do mapa de recursos rectificado. O agrupamento médio calcula a média do mapa de recursos e o agrupamento de soma, faz a soma de todos os elementos na chamada do mapa de recursos.

¹⁰ <https://medium.com/towards-data-science/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, 16 de fevereiro de 2022.

Essas operações são repetidas em dezenas ou centenas de camadas, com cada camada aprendendo a identificar diferentes recursos.

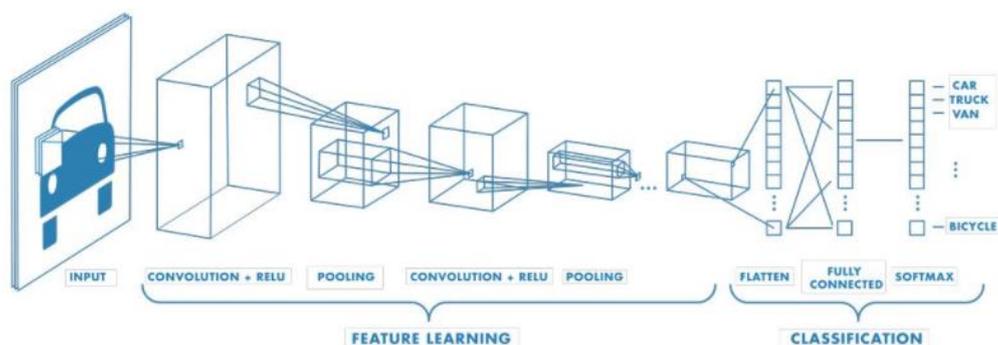


Figura 8: Arquitetura completa de uma CNN

Fonte: <https://images.app.goo.gl/Fq7DRnGTHHsjJ6n66>¹¹

Depois de passar dos recursos de aprendizado, como mostra a figura 8, segue-se para uma Rede Neural regular para fins de classificação.

Camada Totalmente Conectada - Classificação das Camadas

A penúltima camada é uma camada totalmente conectada que gera um vector de K dimensões onde K é o número de classes que a rede poderá prever. Este vector contém as probabilidades para cada classe de qualquer imagem ser classificada.¹²

A camada final da arquitetura CNN usa um modelo camada de classificação capaz de distinguir entre características dominantes e certas características de baixo nível em imagens e classificá-las usando a técnica de classificação Softmax.

2.2.6. YOLO (You Only Look Once)

YOLO é uma abreviação para o termo *You Only Look Once*. Este é um algoritmo que usa redes neurais para a detecção e reconhecimento de vários objectos em uma imagem (em tempo real). A detecção de objectos no YOLO é feita como um

¹¹ Visto em 16 de fevereiro de 2022.

¹² <https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>, 16 de fevereiro de 2022.

problema de regressão e fornece as probabilidades de classe das imagens detectadas.

Como funciona o algoritmo YOLO?

O algoritmo YOLO funciona usando as três técnicas a seguir:

- Blocos residuais;
- Regressão da caixa delimitadora;
- Intersecção sobre União (IOU).

Blocos Residuais

Primeiro, a imagem é dividida em várias grades. Cada grade tem uma dimensão de $S \times S$. A imagem a seguir mostra como uma imagem de entrada é dividida em grades.

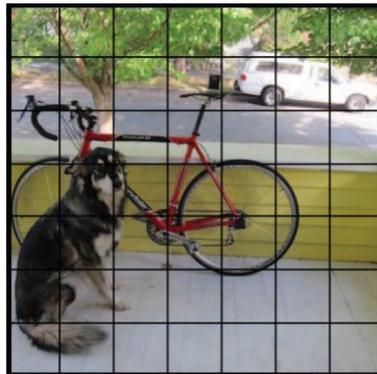


Figura 9: Divisão em grade

Fonte: Redmon et al., 2016

Na imagem acima, existem muitas células de grade de igual dimensão. Cada célula da grade detectará objectos que aparecem dentro delas. Por exemplo, se um centro de objecto aparecer dentro de uma determinada célula da grade, essa célula será responsável por detectá-lo.

Regressão da caixa delimitadora

Cada caixa delimitadora consiste em 5 previsões: x , y , w , h e confiança. As coordenadas (x, y) representam o centro da caixa em relação aos limites da célula da grade, a largura (w) e altura (h) são previstos em relação a toda a imagem. Uma

caixa delimitadora é um contorno que destaca um objecto em uma imagem, como mostra a figura abaixo.



Figura 10: Caixa delimitadora e confiança

Fonte: Redmon et al., 2016

YOLO usa uma única regressão de caixa delimitadora para prever a altura, largura, centro e classe de objectos. A imagem acima, representa a probabilidade de um objecto aparecer na caixa delimitadora.

Intersecção sobre união (IOU)

Intersecção sobre união é um fenômeno na detecção de objectos que descreve como as caixas se sobrepõem. YOLO usa IOU para fornecer uma caixa de saída que envolve os objectos perfeitamente.

Cada célula da grade é responsável por prever as caixas delimitadoras e suas pontuações de confiança. O IOU é igual a 1 se a caixa delimitadora prevista for igual à caixa real. Esse mecanismo é usado para eliminar as caixas delimitadoras que não são iguais à caixa real.

Combinação das três técnicas

A imagem abaixo mostra como as três técnicas são aplicadas para produzir os resultados finais de detecção.

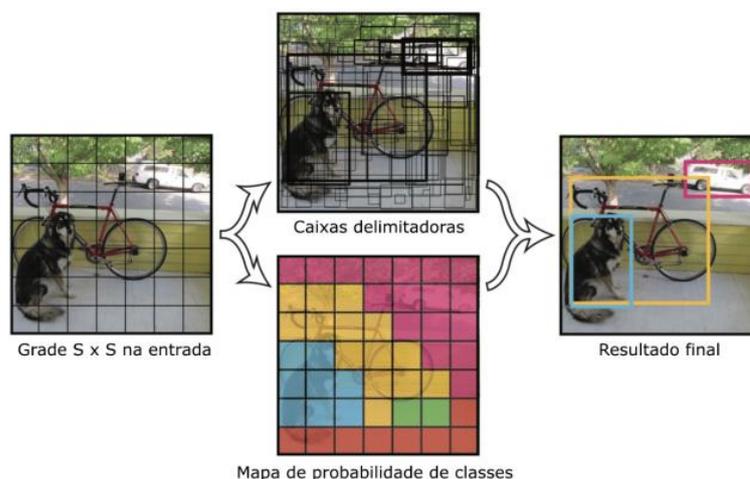


Figura 11: Combinação das três técnicas

Fonte: Redmon et al., 2016

Primeiro, a imagem é dividida em células de grade. Cada célula da grade prevê caixas delimitadoras e fornece suas pontuações de confiança. As células predizem as probabilidades de classe para estabelecer a classe de cada objecto.

Podemos notar pelo menos três classes de objectos: um carro, um cão e uma bicicleta. Todas as previsões são feitas simultaneamente usando uma única rede neural convolucional.

A intersecção sobre união garante que as caixas delimitadoras previstas sejam iguais às caixas reais dos objectos. Esse fenómeno elimina caixas delimitadoras desnecessárias que não atendem às características dos objectos (como altura e largura). A detecção final consistirá em caixas delimitadoras exclusivas que se encaixam perfeitamente nos objetos.

3. IMPLEMENTAÇÃO

Hardware Utilizado

Para realizar todos os testes, o hardware utilizado foi um computador de marca DELL Inspiron 5558, com as seguintes especificações:

- Processador: Intel® core™ i5-6200U CPU @ 2.30GHz
- Memória RAM: 8GB
- GPU: AMD Radeon™ R5 M335 e Intel® HD Graphics 520

Implementação

A detecção de objectos usando algoritmos baseados em aprendizagem profunda reduz a necessidade de serem implementadas etapas de aplicação de filtros de processamento de imagens e segmentação de objectos.

Os algoritmos necessários para execução do projecto todo será feito no ambiente de desenvolvimento integrado Jupyter Notebook, usando a linguagem de programação *python*, com *framework* YOLO e rede neural da *Darknet* e *Open Cv*, com modelos pré-treinados da COCO.

Modelagem do Sistema

Os veículos serão detectados através das câmaras com o uso da inteligência artificial. Isso funcionará como entrada para o sistema. Esta entrada do sinal indicará o comprimento dos veículos na estrada. O sistema gera sinais de saída para vermelho, laranja e verde, e monitorar seus horários levando em consideração o número de veículos em cada via. A verificação do fluxo de veículos será feita a cada 30 segundos. Caso não haja o fluxo de veículos em uma determinada via, não será atribuído a permissão para a abertura do semáforo nessa via. Se o fluxo de tráfego detectado é normal, então os tempos do sinal vermelho, laranja e verde é normal, mas se o tráfego detectado é médio, então o tempo do sinal verde é automaticamente aumentado em alguns segundos pelo sistema e se o tráfego detectado for alto, o tempo de sinal verde é aumentado ainda mais conforme o fluxo de veículos detectados.

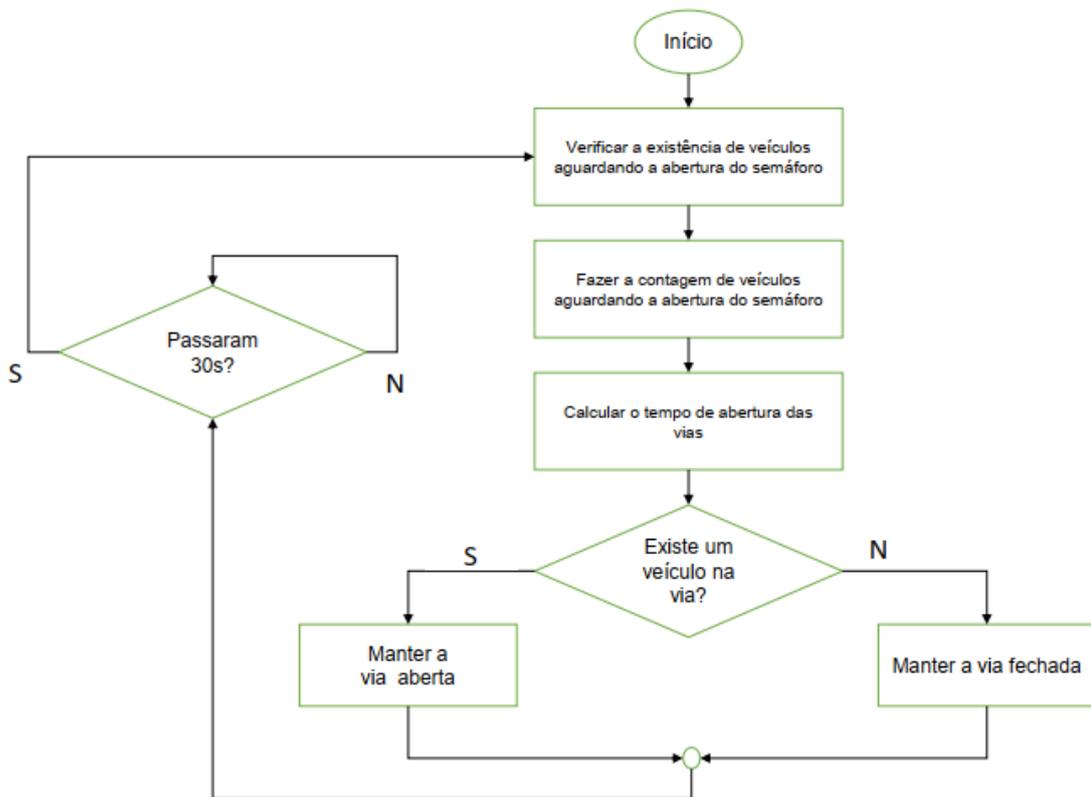


Figura 12: Fluxograma do programa desenvolvido

Fonte: autor

Inicialmente, para realizar a detecção de veículos será usado o algoritmo *YOLO* e a biblioteca *OpenCV*, a figura 13 mostra o resultado da detecção de veículos.

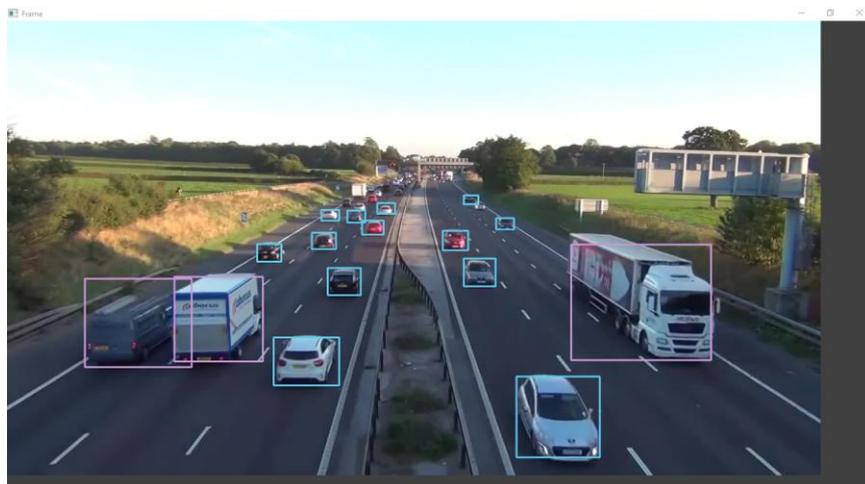


Figura 13: Detecção de veículos

Fonte: Autor

Depois de realizar a detecção de veículos, o próximo passo diz respeito a contagem dos mesmos. Para tanto, definiu-se uma região de contagem onde verifica-se a passagem do veículo por ela. Esta região é nada mais do que uma linha de colisão (linha verde, mostrada na figura 14) ao qual são verificadas as posições X e Y superiores e inferiores do veículo cruzaram esta linha. Caso as coordenadas tenham cruzado a linha de colisão, incrementa-se a quantidade de totais veículos.

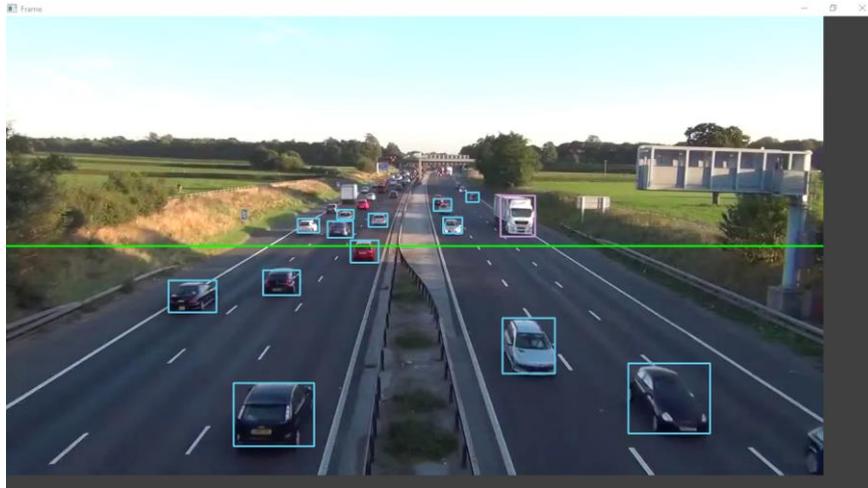


Figura 14: Linha de colisão

Fonte: Autor

4. TESTES E ANÁLISE DOS RESULTADOS

Neste capítulo irão se apresentar os resultados dos testes e sua análise.

4.1. Testes de detecção e contagem de veículos

Neste teste, utilizou-se vídeo obtido da internet de uma autoestrada, com 1 minuto e 1 segundos de duração, na qual temos dois sentidos. O vídeo foi obtido na internet devido a impossibilidade de ter acesso a um semáforo real e afixação da câmara.



Figura 15: Imagem do vídeo obtido na internet

Fonte: Internet¹³

Inicialmente foi processado o vídeo em alta resolução, para que se tivesse a ideia da quantidade de carros que foram detectados no vídeo usando o modelo pré-treinado YOLOv3-tiny.

A tabela 2, apresenta a quantidade de veículos que passaram na via e os que foram detectados. De acordo com a mesma tabela, verificaram-se muitos erros na detecção dos veículos e conseqüentemente a sua contagem, como mostra a figura 14, o que dificulta o uso desse modelo para o controle do tráfego.

Por esse motivo, foi necessário o uso do modelo YOLOv3 para o processamento do vídeo. Este modelo reduziu a velocidade de processamento, mas aumentou de forma significativa a detecção de veículos, reduzindo quase para zero o erro de detecção segundo a tabela 3, como mostra figura 15. O que mostra que a contagem e detecção de veículos foi precisa e correcta.

¹³ <https://www.youtube.com/watch?v=NcaGFp76BTY>, 26 de fevereiro de 2022

Tabela 2: Teste de detecção de veículos usando o modelo YOLOv3-tiny

Modelo \ Veículo	Em Trânsito	Detectados pelo modelo	Erro
Yolov3-tiny	20	15	5

Fonte: Autor

Tabela 3: Teste de detecção de veículos usando o modelo YOLOv3

Modelo \ Veículo	Em Trânsito	Detectados pelo modelo	Erro
Yolov3	20	19	1

Fonte: Autor



Figura 16: Modelo YOLOv3-tiny

Fonte: Autor

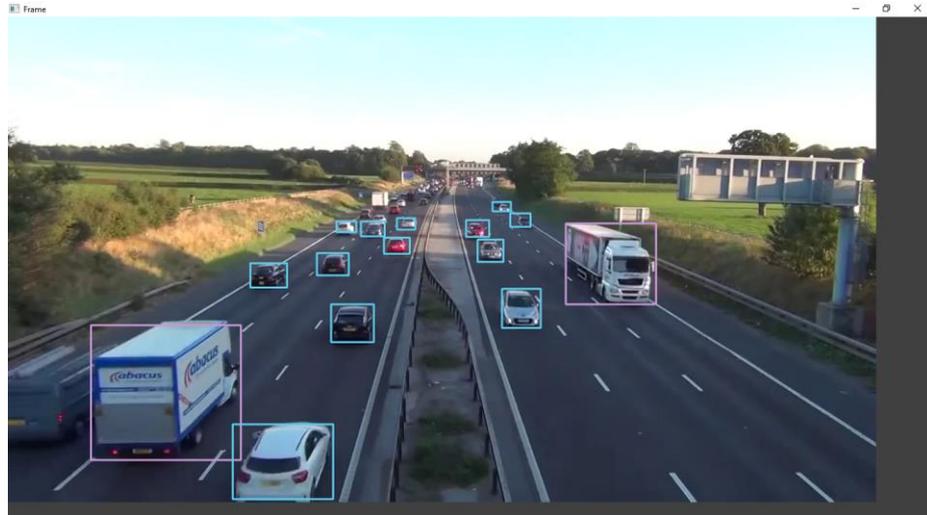


Figura 17: Modelo YOLOv3

Fonte: Autor

4.2. Análise dos Resultados

Estes testes não foram feitos em tempo real, pois devido ao processador e baixa performance do laptop, houve um baixo desempenho, utilizando o vídeo de alta resolução. O modelo YOLOv3-tiny teve um tempo de processamento do vídeo de aproximadamente 3 minuto e 81 milissegundos, enquanto que o modelo YOLOv3 teve um tempo de processamento do vídeo de aproximadamente 22 minutos e 52 segundos.

Como todo modelo pré-treinado em detecção e classificação, é inevitável que ocorram nele alguns erros, no caso deste sistema, não temos erros graves, como o caso de o veículo estar obstruído por um outro veículo, sendo reconhecido mais a frente em outro posicionamento.

5. CONCLUSÕES E RECOMENDAÇÕES

Utilizar um controlador automático de tráfego, ou seja, de tempo fixo, não é a melhor opção de escolha quando se trata de cruzamentos com um tráfego que varia muito diariamente. Nesse contexto usar um controlador com câmaras, é a melhor opção, pois os cálculos do tempo são feitos e aplicados em tempo real.

Os resultados deste trabalho foram satisfatórios, desde a detecção, contagem e determinação do tempo do verde. A solução proposta, o sistema pode ser adaptado conforme a necessidade do problema.

O sistema proposto define os tempos de verde, amarelo e vermelho em tempo real de forma inteligente de acordo com o fluxo de veículos e garante que a via com maior fluxo tenha maior tempo em relação a via com menor fluxo de veículos. Reduzindo assim os atrasos indesejados, tempo de espera desnecessário e o congestionamento.

Como vimos no capítulo 4, os testes foram feitos em um laptop com processador de baixo desempenho e performance, o processamento dos vídeos não foi em tempo real. Para solucionar este problema pode-se utilizar um hardware com alta performance e desempenho, ou utilizar câmaras que possam captar vídeos com menor resolução onde o sistema é capaz de detectar os veículos.

Como recomendação para trabalhos futuros, criar uma rede de semáforos que se comunicam, sincronizando assim os tempos de verde, e também projectar um sistema que dê prioridade a veículos prioritários.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 Algoritmo You Only Look Once:
<https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>, 16 de fevereiro de 2022;
- 2 Algoritmo You Only Look Once:
<https://pjreddie.com/darknet/yolo/>, 16 de fevereiro de 2022;
- 3 ARAUJO, S C. 2006, Controlador de tráfego: semáforo inteligente;
- 4 Cruzamentos e Intersecções:
[https://www.stringfixer.com/pt/Crossroads_\(junction\)](https://www.stringfixer.com/pt/Crossroads_(junction)), 26 de novembro de 2021;
- 5 DENATRAN, 1984, Coleção de Serviços de Engenharia - Manual de Semáforos, volume 4, Brasília;
- 6 Inteligência Artificial:
<https://www.britannica.com/technology/artificial-intelligence>, 7 de dezembro de 2021;
- 7 KOONCE, Peter et al. 2008, Traffic Signal Timing Manual, Washington DC;
- 8 REDMON, Josef et al. 2016, You Only Look Once;
- 9 Rede Neural Convolutacional:
<https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>, 16 de fevereiro de 2022;
- 10 Rede Neural Convolutacional:
<https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/>, 16 de fevereiro de 2022;
- 11 Rede Neural Convolutacional:
<https://datawow.io/blogs/interns-explain-cnn-8a669d053f8b>, 16 de fevereiro de 2022;
- 12 Rede Neural Convolutacional:
<https://vitorborbarodrigues.medium.com/conhecendo-a-vis%C3%A3o-do-computador-redes-neurais-convolucionais-e1c2b14bf426>, 17 de fevereiro de 2022;

- 13 Rede Neural Convolutacional:
<https://medium.com/towards-data-science/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, 17 de fevereiro de 2022;
- 14 Neurónios:
<https://www.google.com/amp/s/mundoeducacao.uol.com.br/amp/biologia/neuronios.htm>, 8 de dezembro de 2021;
- 15 Visão Computacional:
<https://www.ibm.com/eg-en/topics/computer-vision>, 15 de fevereiro de 2022;
- 16 Visão Computacional:
<https://pixforce.com.br/visao-computacional-os-olhos-das-maquinas-suuperando-a-visao-humana/>, 15 de fevereiro de 2022;
- 17 Visão Computacional:
<https://developer.nvidia.com/opency>, 23 de fevereiro de 2022

ANEXOS

**UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA**

ACTA DE ENCONTROS

Data:	10/11/21
-------	----------

1. AGENDA:

Apresentação da proposta do tema

2. PRESENCAS

Supervisor	MSc, Roxan Cadir, Eng.
Estudante	Junga, Abel Mateus

3. RESUMO DO ENCONTRO:

Apresentação do tema a supervisora presencialmente e posterior aceitação do tema por parte da supervisora.
--

4. RECOMENDAÇÕES:

Continuar a pesquisar sobre o tema e apresentar o termo de atribuição

5. OBSERVAÇÕES

--	--

6. DATA DO PRÓXIMO ENCONTRO	24/11/21
-----------------------------	----------

Estudante

Supervisor

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ACTA DE ENCONTROS

Data:	24/11/21
-------	----------

1. AGENDA:

Apresentação do termo de atribuição

2. PRESENCAS

Supervisor	MSc, Roxan Cadir, Eng.
Estudante	Junga, Abel Mateus

3. RESUMO DO ENCONTRO:

Definição dos objectivos e metodologia a usar

4. RECOMENDAÇÕES:

Continuar a pesquisa para a resolução do problema

5. OBSERVAÇÕES

--	--

6. DATA DO PRÓXIMO ENCONTRO

26/01/22

Estudante

Supervisor

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ACTA DE ENCONTROS

Data:	26/01/22
-------	----------

1. AGENDA:

Apresentação da revisão literária

2. PRESENÇAS

Supervisor	MSc, Roxan Cadir, Eng.
Estudante	Junga, Abel Mateus

3. RESUMO DO ENCONTRO:

Análise da revisão literária completa, com os pontos relacionados com os objectivos do trabalho

4. RECOMENDAÇÕES:

Avançar com os testes do sistema proposto

5. OBSERVAÇÕES

--	--

6. DATA DO PRÓXIMO ENCONTRO

09/02/22

Estudante

Supervisor

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ACTA DE ENCONTROS

Data:	09/03/22
-------	----------

1. AGENDA:

Apresentação dos testes e resultados

2. PRESENÇAS

Supervisor	MSc, Roxan Cadir, Eng.
Estudante	Junga, Abel Mateus

3. RESUMO DO ENCONTRO:

Análise dos testes e resultados

4. RECOMENDAÇÕES:

Avançar com a conclusão

5. OBSERVAÇÕES

--	--

6. DATA DO PRÓXIMO ENCONTRO

23/02/22

Estudante

Supervisor

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ACTA DE ENCONTROS

Data:	23/02/22
-------	----------

1. AGENDA:

Apresentação da conclusão

2. PRESENÇAS

Supervisor	MSc, Roxan Cadir, Eng.
Estudante	Junga, Abel Mateus

3. RESUMO DO ENCONTRO:

Observância do cumprimento dos objectivos

4. RECOMENDAÇÕES:

Seguir com as sugestões e correções do relatório

5. OBSERVAÇÕES

--	--

6. DATA DO PRÓXIMO ENCONTRO

09/03/22

Estudante

Supervisor

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ACTA DE ENCONTROS

Data:	09/03/22
-------	----------

1. AGENDA:

Ponto de situação do relatório

2. PRESENÇAS

Supervisor	MSc, Roxan Cadir, Eng.
Estudante	Junga, Abel Mateus

3. RESUMO DO ENCONTRO:

Apresentação do relatório final e aprovação da supervisora
--

4. RECOMENDAÇÕES:

Seguir com as sugestões e correções do relatório, e fazer a impressão do relatório
--

5. OBSERVAÇÕES

--	--

6. DATA DO PRÓXIMO ENCONTRO	
-----------------------------	--

Estudante

Supervisor

APÊNDICE

```
import numpy as np

import imutils

import time

from scipy import spatial

import cv2

list_of_vehicles = ["bicycle", "car", "motorbike", "bus", "truck", "train"]

FRAMES_BEFORE_CURRENT = 10

inputWidth, inputHeight = 416, 416

LABELS = open('coco.names').read().strip().split("\n")

configPath = 'yolov3.cfg'

weightsPath = 'yolov3.weights'

preDefinedConfidence = 0.5

preDefinedThreshold = 0.3

outputVideoPath = '/Abel/Desktop/Semaforo_Inteligente'

np.random.seed(42)

COLORS = np.random.randint(0, 255, size=(len(LABELS), 3), dtype="uint8")

def displayVehicleCount(frame, vehicle_count):

    cv2.putText(frame, 'Detected Vehicles: ' + str(vehicle_count), (20, 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0xFF, 0), 2,
cv2.FONT_HERSHEY_COMPLEX_SMALL,)
```

```

def boxAndLineOverlap(x_mid_point, y_mid_point, line_coordinates):
    x1_line, y1_line, x2_line, y2_line = line_coordinates

    if (x_mid_point >= x1_line and x_mid_point <= x2_line+5) and (y_mid_point >=
y1_line and y_mid_point <= y2_line+5):
        return True

    return False

def displayFPS(start_time, num_frames):
    current_time = int(time.time())

    if(current_time > start_time):
        num_frames = 0

        start_time = current_time

    return start_time, num_frames

def drawDetectionBoxes(idxs, boxes, classIDs, confidences, frame):
    if len(idxs) > 0:
        for i in idxs.flatten():
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])

            color = [int(c) for c in COLORS[classIDs[i]]]
            cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
            text = "{}: {:.4f}".format(LABELS[classIDs[i]], confidences[i])
            cv2.putText(frame, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
color, 2)

            cv2.circle(frame, (x + (w//2), y + (h//2)), 2, (0, 0xFF, 0), thickness=2)

```

```

def initializeVideoWriter(video_width, video_height, videoStream):
    sourceVideofps = videoStream.get(cv2.CAP_PROP_FPS)
    fourcc = cv2.VideoWriter_fourcc(*"MJPG")
    return cv2.VideoWriter(outputVideoPath, fourcc, sourceVideofps, (video_width,
video_height), True)

def boxInPreviousFrames(previous_frame_detections, current_box,
current_detections):
    centerX, centerY, width, height = current_box
    dist = np.inf

    for i in range(FRAMES_BEFORE_CURRENT):
        coordinate_list = list(previous_frame_detections[i].keys())
        if len(coordinate_list) == 0:
            continue

        temp_dist, index = spatial.KDTree(coordinate_list).query([(centerX, centerY)])
        if (temp_dist < dist):
            dist = temp_dist
            frame_num = i
            coord = coordinate_list[index[0]]

    if (dist > (max(width, height)/2)):
        return False

    current_detections[(centerX, centerY)] =
previous_frame_detections[frame_num][coord]

```

```

return True

def count_vehicles(idxs, boxes, classIDs, vehicle_count, previous_frame_detections,
frame):
    current_detections = {}
    if len(idxs) > 0:
        for i in idxs.flatten():
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])

            centerX = x + (w//2)
            centerY = y+ (h//2)

            if (LABELS[classIDs[i]] in list_of_vehicles):
                current_detections[(centerX, centerY)] = vehicle_count
                if (not boxInPreviousFrames(previous_frame_detections, (centerX,
centerY, w, h), current_detections)):
                    vehicle_count += 1

                ID = current_detections.get((centerX, centerY))

                if (list(current_detections.values()).count(ID) > 1):
                    current_detections[(centerX, centerY)] = vehicle_count
                    vehicle_count += 1

            cv2.putText(frame, str(ID), (centerX, centerY),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, [0,0,255], 2)

```

```

return vehicle_count, current_detections

net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
net.setPreferableBackend(cv2.dnn.DNN_TARGET_CPU)

ln = net.getLayerNames()
ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]

videoStream = cv2.VideoCapture('bridge.mp4')
video_width = int(videoStream.get(cv2.CAP_PROP_FRAME_WIDTH))
video_height = int(videoStream.get(cv2.CAP_PROP_FRAME_HEIGHT))

x1_line = 0
y1_line = video_height//2
x2_line = video_width
y2_line = video_height//2

previous_frame_detections = [(0,0):0] for i in
range(FRAMES_BEFORE_CURRENT)]

num_frames, vehicle_count = 0, 0
writer = initializeVideoWriter(video_width, video_height, videoStream)
start_time = int(time.time())

```

```

while True:

    num_frames+= 1

    boxes, confidences, classIDs = [], [], []

    vehicle_crossed_line_flag = False

    start_time, num_frames = displayFPS(start_time, num_frames)

    (grabbed, frame) = videoStream.read()

    if not grabbed:

        break

    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (inputWidth, inputHeight),
swapRB=True, crop=False)

    net.setInput(blob)

    start = time.time()

    layerOutputs = net.forward(ln)

    end = time.time()

    for output in layerOutputs:

        for i, detection in enumerate(output):

            scores = detection[5:]

            classID = np.argmax(scores)

            confidence = scores[classID]

            if confidence > preDefinedConfidence:

                box = detection[0:4] * np.array([video_width, video_height, video_width,
video_height])

                (centerX, centerY, width, height) = box.astype("int")

```

```

x = int(centerX - (width / 2))
y = int(centerY - (height / 2))

boxes.append([x, y, int(width), int(height)])
confidences.append(float(confidence))
classIDs.append(classID)

if vehicle_crossed_line_flag:
    cv2.line(frame, (x1_line, y1_line), (x2_line, y2_line), (0, 0xFF, 0), 2)
else:
    cv2.line(frame, (x1_line, y1_line), (x2_line, y2_line), (0, 0, 0xFF), 2)

idxs = cv2.dnn.NMSBoxes(boxes, confidences, preDefinedConfidence,
preDefinedThreshold)

drawDetectionBoxes(idxs, boxes, classIDs, confidences, frame)

vehicle_count, current_detections = count_vehicles(idxs, boxes, classIDs,
vehicle_count, previous_frame_detections, frame)

displayVehicleCount(frame, vehicle_count)

writer.write(frame)

cv2.imshow('Frame', frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

```

```
previous_frame_detections.pop(0) #Removing the first frame from the list
previous_frame_detections.append(current_detections)
```

```
writer.release()
```

```
videoStream.release()
```

```
cv2.destroyAllWindows()
```

```
L = 9
```

```
S=525*L
```

```
if L < 5.5:
```

```
    if L > 5.2 or L == 5.2:
```

```
        S = 2700
```

```
    else:
```

```
        if L > 4.8 or L == 4.8:
```

```
            S = 2475
```

```
        else:
```

```
            if L > 4.5 or L == 4.5:
```

```
                S = 2250
```

```
            else:
```

```
                if L > 4.2 or L == 4.2:
```

```
                    S = 2075
```

```
                else:
```

```
                    if L > 3.9 or L == 3.9:
```

```
                        S = 1950
```

```
                    else:
```

if $L > 3.6$ or $L == 3.6$:

$$S = 1900$$

else:

if $L > 3.3$ or $L == 3.3$:

$$S = 1875$$

else:

if $L > 3$ or $L == 3$:

$$S = 1850$$

$$t = (24/3600)$$

$$q = (\text{vehicle_count}/t)$$

$$Y = q/S$$

$$T_p = 4$$

$$T_a = 3$$

$$C_o = ((1.5 * T_p + 5)/(1 - Y))$$

if $C_o < 25$:

$$C_o = 25$$

if $C_o > 120$:

$$C_o = 120$$

$$G_{ef} = Y * ((C_o - T_p)/Y)$$

$$g = G_{ef} + T_p - T_a$$

```
print("o tempo de verde real eh: "+str(g)+" segundos")
```