PROVA DE CONCEITO (PoC) PARA DETECÇÃO DE OBJETOS USANDO VISÃO COMPUTACIONAL EM UMA COLETA SELETIVA

Fabricio Fantin de Oliveira[1], Bruno Bastos Stoll[2]

1- Acadêmico do curso de Engenharia de Computação

2- Mestre – Professor Multivix - Vitória

RESUMO

Um dos grandes problemas da reciclagem é a coleta seletiva de lixo. No entanto, existem iniciativas de seleção manual de materiais recicláveis. Nesse contexto, este trabalho apresenta uma Prova de Conceito focado em visão computacional, na finalidade de detectar de objetos recicláveis, usando técnica de reconhecimento de imagem para detectar materiais recicláveis de cada objeto em uma coleta seletiva. Para o reconhecimento, realizou-se o treinamento de um modelo preditivo utilizando uma base de imagens de materiais recicláveis. E para avaliar o trabalho, foram realizadas 10 fotos distintas, no qual as garrafas plásticas e as latas foram detectadas distintamente com possibilidade de coleta seletiva automatizada. A abordagem proposta apresenta preliminarmente bons resultados, tendo possibilidades de demonstrar um software de classificação que poderá ser utilizado em uma coleta seletiva, com fins de colaboração do desenvolvimento da sustentabilidade.

Palavras-chave: Visão Computacional, OpenCV, Python, Coleta Seletiva.

1. INTRODUÇÃO

Um estudo realizado pela WWF-Brasil, revela que o Brasil é o 4º país do mundo que mais gera lixo plástico, com 11,3 milhões de toneladas, ficando apenas atrás dos Estados Unidos, China e Índia. Desse total, foram coletadas mais de 10,3 milhões de toneladas (91%), mas desse total, apenas 145 mil toneladas (1,28%) são realmente recicladas. Segundo a pesquisa, com novo estudo, “Solucionar a Poluição Plástica: Transparência e Responsabilização”, a menos que todos os atores da cadeia de valores do plástico se responsabilizem pelo custo real para a natureza e para as pessoas, se não, a crise mundial do plástico só irá piorar. Também, segundo o estudo, a quantidade de plástico que vaza para os oceanos todos os anos é de aproximadamente 10 milhões de toneladas, o que equivale a 23 mil aviões Boeing 747 pousando nos mares e oceanos todos os anos. Por fim, a pesquisa enfatiza um sistema de separação que envolva as empresas produtoras do plástico ajuda a viabilizar a uniformidade e volume, ampliando a possibilidade de reuso (WWF, 2021).

Com tantos avanços tecnológicos, a sociedade está em uma constante evolução. Por outro lado, o meio ambiente vem sofrendo as consequências ao grande acúmulo de lixo, acarretando a criação de meios sustentáveis e efetivos resolução desses problemas (PINA, MARTINS, 2018).

A coleta seletiva é uma das alternativas para redução do excesso de lixo, fazendo com que ele retorne ao consumo através de sua reciclagem. Coleta seletiva é a coleta diferenciada de resíduos que foram previamente separados segundo a sua constituição ou composição. Ou seja, resíduos com características similares são selecionados pelo gerador (que pode ser o cidadão, uma empresa ou outra instituição) e disponibilizados para a coleta separadamente (MMA, 2021).

O presente trabalho estuda a viabilidade do desenvolvimento em software, utilizando o OpenCV para realização de reconhecimento empregando a Visão Computacional em uma coleta seletiva. Dessa forma, foi feita a Prova de Conceito (PoC), que é uma abordagem na construção de software que verifica determinada ideia, a fim de confirmar sua viabilidade e aplicação (MCADAM, BROWN, 2009).

Diante do exposto, este trabalho objetiva demonstrar através da visão computacional que a tecnologia pode colaborar para a automatização da coleta seletiva, reduzindo os riscos de exposição às doenças que os trabalhadores estão sujeitos e tornando o processo mais efetivo (PINA, MARTINS, 2018).

1.1. JUSTIFICATIVA DO TEMA

Atualmente, com grande produção de lixo, nota-se a existência de muitos materiais que levam anos para a decomposição, fazendo-se necessária a utilização de uma coleta seletiva para reciclagem e reutilização desses materiais.

No Brasil, jogam-se em média 76 milhões de toneladas de lixo ao mês. Dentre estes, 30% poderiam ser reaproveitados, mas apenas 3% são reciclados. Com o intuito de evitar a danos ao meio ambiente, muitas cidades brasileiras estão aderindo à reciclagem, tendo como resultado, um menor desperdício (ROSA, 2005).

A melhor forma de realizar o reaproveitamento da matéria-prima, é através da reciclagem. A partir de materiais como: plástico, alumínio, aço, vidro e outros, é possível produzir uma nova quantidade de matérias, ocasionando grandes economias em energia e matéria prima (ROSA, 2005).

1.2. DELIMITAÇÃO DO TEMA

Com a detecção de objetos, utilizando a visão computacional, tem um processo mais rápido, automatizado e reduz erros para realização de uma coleta seletiva. Assim, fazendo uma reciclagem mais rápida e eficaz.

1.3. PROBLEMA DA PESQUISA

Uma das preocupações que o mundo vem tendo, é com a grande produção de lixo que está sendo descartado no nosso planeta. Diante disso, o que a visão computacional é capaz de fazer para melhorar o meio ambiente hoje em dia?

1.4. HIPÓTESE

A tecnologia se mostra extraordinária, estando presente em todas as áreas. Através dela é possível a criação de um software para automatização de uma casa, automatizar um processo complexo em uma indústria e até reconhecimento facial ou de objetos para detecção de resíduos recicláveis.

O presente projeto identifica e comprova a possibilidade de se realizar uma coleta seletiva para resíduos recicláveis ou orgânicos utilizando a Visão Computacional, a partir da biblioteca usada em Python, o OpenCV.

1.5. OBJETIVOS

1.5.1 Objetivo Geral

Detectar objetos recicláveis, usando técnica de reconhecimento de imagem para detectar materiais recicláveis de cada objeto em uma coleta seletiva

1.5.2 Objetivos Específicos

- Compreender as técnicas de treinamento de imagem
- Utilizar o Cascade Trainer GUI para treinamento das imagens
- Utilizar a biblioteca OpenCV no desenvolvimento do software
- Apresentar os resultados obtidos

2. FUNDAMENTAÇÃO TEÓRICA

A seguir são apresentados os fundamentos teóricos que foram essenciais para o desenvolvimento deste projeto.

2.1. Coleta Seletiva

O Relatório Preliminar da Política Nacional de Resíduos Sólidos no Capítulo I, Art. 2º define a coleta seletiva como o recolhimento diferenciado de resíduos sólidos previamente selecionados nas fontes geradoras, com o intuito de encaminhá-los para reciclagem, compostagem, reuso, tratamento e outras destinações alternativas, como aterros, coprocessamento e incineração (BRAVO, 2018).

Reduzir os milhões de toneladas de lixo que nossa civilização vem gerando diariamente, é um dos maiores desafios do século XXI (RIBEIRO, BESEN, 2007). Para

reduzir a carga abundante de lixo nos depósitos, é a implantação da coleta seletiva de resíduos recicláveis (BRAVO, 2018).

2.2. Visão Computacional

A visão computacional é a ciência e tecnologia das máquinas que enxergam. Ela é responsável pela forma como um computador consegue enxergar o que está a sua volta, extraindo assim, informações relevantes a partir de uma captura de uma imagem ou de um vídeo. Com essas informações é possível manipular, reconhecer e pensar sobre os objetos que compõem a imagem (DE MILANO, HONORATO, 2014).

Para facilitar e suprir limitações humanas, a visão computacional ajuda no processamento e na identificação de imagens. A visão computacional pode ser estabelecida, como sendo a união de técnicas computacionais utilizadas para explicar ou estimar propriedades dinâmicas e geométricas da realidade tridimensional por meio de imagens. Câmeras, ao invés de olhos, são empregadas para se obter as imagens digitais. Como essas informações sofrem variações de natureza, podem ser aplicadas em diferentes ações e áreas. A visão computacional busca obter dados de cenas através de imagens digitais capturadas previamente. Para que isso seja possível, quase sempre se faz uso de metodologias de processamento de imagens. Para concluir, a visão computacional tem por objetivo deduzir e descrever automaticamente propriedades e estruturas do mundo 3D, que pode ser dinâmico, com base em várias imagens do mundo em 2D, as quais podem ser monocromáticas ou coloridas, sendo capturadas por apenas um ou diversos sensores, que podem ser móveis ou estacionários (FERNANDES NETO, 2020).

Na visão computacional, procura-se simular a visão humana, então possui uma entrada como uma imagem, mas a saída como uma compreensão de imagem ao todo, ou parcialmente. Podemos dizer que um processamento de imagem é quando a entrada de um sistema é uma imagem e um conjunto de valores numéricos é a saída (MARENGONI, STRINGHINI, 2009).

Para os sistemas de visão computacional existem duas técnicas de Processamento de Imagem. A primeira é o reconhecimento e a segunda é a inspeção automatizada (Fig. 1). Para um sistema de reconhecimento, é retirado características dos objetos da imagem e usar certo tipo de inteligência computacional para poder interpretar os objetos. Nesse caso não é necessário a coleta de valores exatos com essas características, mas em sistemas de inspeção a exatidão é fundamental. No processo automatizado de inspeção é utilizado uma medição sem contato que utiliza algoritmos computacionais (FELICIANO, SOUZA, LETA, 2005).

Segundo Fernandes Neto (2020), visão computacional seguem um conjunto predefinido de etapas, aquisição, pré-processamento, segmentação, representação, reconhecimento e interpretação.

No processo de aquisição é feita a captura da imagem através de um sensor ou um dispositivo, no caso deste projeto, em um celular. Nesta etapa, a configuração do dispositivo de captura, é denominada como o formato da imagem digital, resolução, as configurações de luminosidade, níveis de cinza ou cores da imagem digital. Na etapa do pré-processamento, tem o objetivo de realizar a melhoria da qualidade de imagem vinda da aquisição, fazendo uso das técnicas de atenuação de ruídos, correção de contraste e brilho e equalização de histograma. A etapa responsável pelo processo de separar a imagem em regiões de pixels similares é a de segmentação, onde os destaques são selecionados com base na extração de características que possibilitam a diferenciar classes de objetos. Esse processo de extração e conhecido como descrição e representação é a estrutura que possibilita o armazenamento e manipulação dessa sessão. Como última etapa, reconhecimento ou classificação, é o processo de identificar os objetos da imagem, conforme as características presentes descritas. Na interpretação, é o resultado para a situação descoberta, determinada tal classificação (FERNANDES NETO, 2020).

2.2.1 Inteligência Artificial

Com o chamado “aprendizado de máquina”, do inglês M*achine Learning*, uma área da inteligência artificial restrita que trata dos reconhecimentos de padrões através utilizando uma base de dados (SILVA, 2019).

2.3. OpenCV

O OpenCV (Open Source Computer Vision) é uma linguagem de código aberto, incrementa uma variedade de ferramentas de interpretação de imagem. O pacote está disponível gratuitamente na Internet e seu manual de referência. A biblioteca de programação usada está dividia em cinco grupos de funções: Processamento de Imagem; Análise estrutural; Análise de movimento e rastreamento de objetos; Reconhecimento de padrões e Calibração de câmara e reconstrução 3D (MARENGONI, STRINGHINI, 2009).

O OpenCV é uma biblioteca que incluem um conjunto amplo de algoritmos de visão computacional e aprendizado de máquina, com mais de 2.500 algoritmos otimizados. Com isso, é possível identificar objetos, reconhecer rostos, rastrear movimentos de câmeras, classificar ações humanas em vídeo, rastrear objetos em movimento etc. (FERNANDES NETO, 2020).

A biblioteca foi desenvolvida com o objetivo de tornar a visão computacional abordável a programadores e usuários em áreas como interação do homem com a máquina em tempo real (MARENGONI, STRINGHINI, 2009).

3. TRABALHOS CORRELATOS

Este capítulo apresenta uma análise de algumas propostas sobre detecção de objetos usando visão computacional para uma coleta seletiva, base deste trabalho.

Dentro deste trabalho, tem-se como proposta, utilizando Inteligência Artificial através da Visão Computacional a detecção de objetos para realizar uma coleta seletiva. Portanto, tive a necessidade de pesquisar novos trabalhos que utilizam esse tipo de detecção e os mecanismos que foram usados para chegar até um resultado.

3.1 Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

Um trabalho com caracterizado por um estudo exploratório e que realizou pesquisa do tipo qualitativa, realizando a captura de imagens e processamento, com objetivo de capturar as cores primárias e secundárias em tempo real. Para tal projeto foi acoplado uma câmera ao computador para desenvolvimento do algoritmo, conforme Imagem 1.

Imagem 1 – Topologia do Projeto

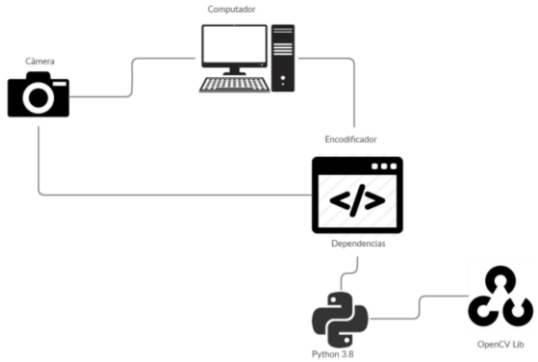
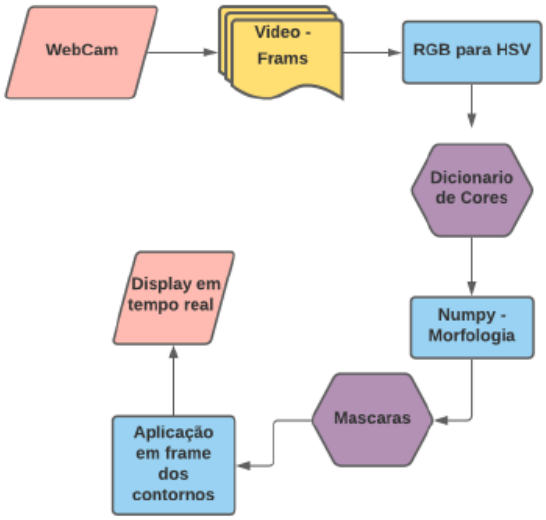


Imagem 2 – Fluxo do Script

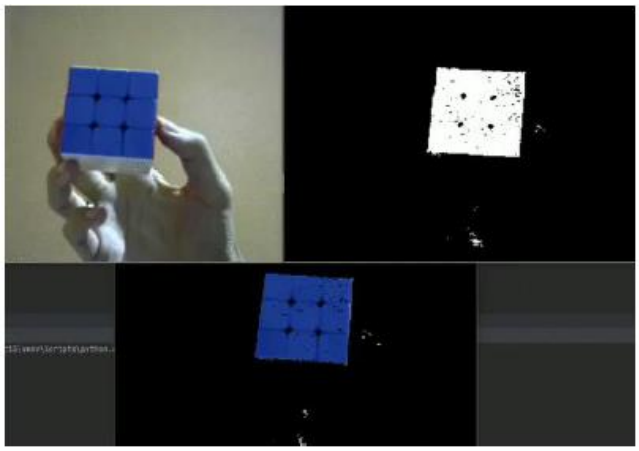


Fonte: Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

Com as devidas instalações de sistemas preparadas, foi dado início ao desenvolvimento da estrutura do código. Dando início a partir da captura das imagens, conforme Imagem 2.

O tratamento das imagens foi realizado utilizando o Imutilis, biblioteca nativa do Python, para amostragem no display. Na etapa seguinte foi realizado a conversão da imagem utilizando a função cv2.COLOR_BGR2HSV, disponível nas bibliotecas do OpenCV. Aplicando suas respectivas funções, obteve-se uma imagem descentralizada e nos padrões de identificação de cores por ‘range’, com isso, é possível a aplicação de um dicionário de cores a serem identificadas pelo algoritmo. Conforme Imagem 3, foi realizada a configuração para reconhecimento da cor azul, a fim de exemplificar o processo de transformação entre espaços de cores.

Imagem 3 – Conversão RGB para HSV



Fonte: Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

Com a utilização de um plano cartesiano e realizados alguns testes para a validação de cores selecionadas, sendo definido um seguinte dicionário de cores.

Segundo o autor:

> A proposta do uso do HSV é a possibilidade de se definir um intervalo entre as cores e nomear qualquer cor contida naquela variação uma determinada cor, neste caso, o valor mínimo de cada cor está contido na variável ‘lower’ e seu valor máximo na variável ‘upper’. Assim, qualquer valor contido entre estas variáveis é catalogado como uma cor, por exemplo, o amarelo, que tem o seu valor mínimo de (23 H,70 S, 120 V), e o valor máximo de (30 H, 255 S, 255 V), qualquer valor dentre o intervalo destas matrizes é identificado como amarelo.

Identificado as definições do aplicativo, iniciou-se o tratamento da imagem através do conceito de morfologia, que são utilizadas técnicas de erosão e dilatação, e aplicação de máscaras. O OpenCV possui funções onde ele realiza a tratativa de imagem com a combinação técnicas de erosão e dilatação. A função cv2.MORPH_OPEN realiza uma erosão seguida de uma dilatação, removendo os ruídos nas imagens (Imagem 4). Com os ruídos tratados, utilizou-se a função cv2.MORPH_CLOSE, onde realiza primeiro a dilatação

e depois a erosão, realizando a remoção de pequenos pontos dentro de objetos (Imagem 5).

Imagem 4 – Técnica de abertura da imagem

Imagem 5 – Técnica de fechamento da imagem

Fonte: Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

O próximo passo, foi de delimitação do contorno do objeto detectado e criação da máscara de detecção a ser aplicada no objeto. Para isso, foi utilizado a função para realizar a detecção de borda do objeto com a cor identificada no dicionário:

Imagem 6 – Detecção de borda

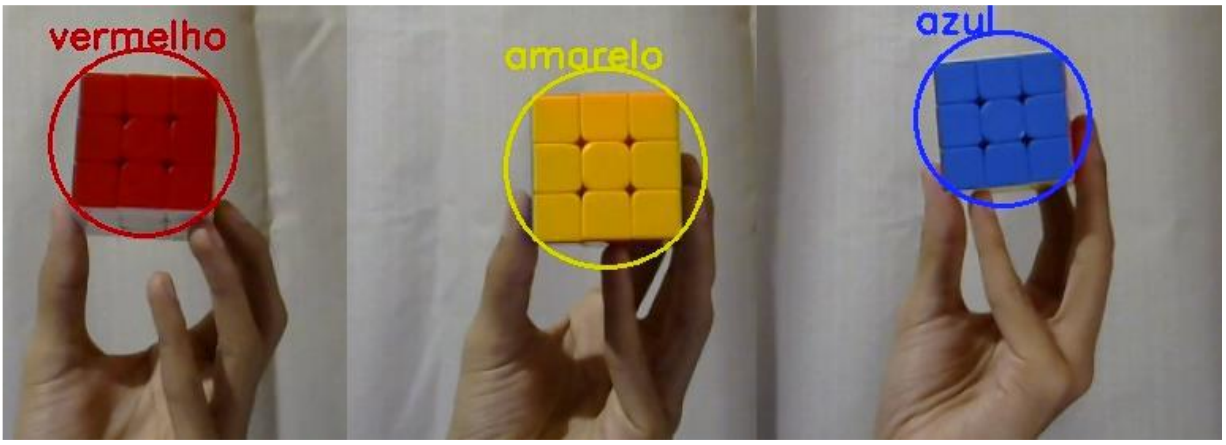
```
# encontra contornos na máscara e inicializa o atual
# (x, y) centro da esfera
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
                        cv2.CHAIN_APPROX_SIMPLE)[-2]
center = None
```

Fonte: Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

A função é acionada para que se possa determinar os contornos da imagem, quando ela corresponde ao dicionário, é gerado uma máscara que detecta esse objeto. Para que esse contorno seja detectado na imagem, foi atribuído a ele uma cor sólida para cada chave encontrada. Essas chaves foram determinadas pelo dicionário de cores, com a junção de seu valor máximo e mínimo.

A partir deste ponto, iniciou-se a etapa de testes e análises finais. Foram realizados diversos testes com cubo mágico, como vamos observar nas próximas imagens, o primeiro teste foi realizado individualmente das cores primárias e secundárias.

Imagem 7 – Teste de reconhecimento de cores primárias



Fonte: Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

No último teste, foram inseridos mais 3 objetos azuis, mas o algoritmo consegue reconhecer apenas o maior objeto. Isso é devido a iluminação do ambiente que pode influenciar na captação da imagem interferindo o reconhecimento das cores.

Imagem 8 - Situação adversa

Fonte: Visão Computacional para Identificação de corem em tempo real com OpenCV e Python (UniCEUB- GO)

Conclui-se deste projeto, que a Visão Computacional juntamente com a programação, foi possível realizar uma detecção das cores através de imagens. Isso pode ser um aliado dos portadores de daltonismo, já que a implementação do algoritmo pode realizada em aplicativos para smartfones.

3.2 Cenário Brasileiro da Gestão dos Resíduos Sólidos Urbanos e Coleta Seletiva (Anais do VI SINGEP–SP)

Este estudo tem o objetivo analisar o cenário brasileiro de gestão dos resíduos sólidos e da coleta seletiva dos Estados. Para isso, desenvolveu-se a fundamentação teórica sobre resíduos sólidos e coleta seletiva e caracterizaram-se os estudos nacionais sobre o assunto. Trata-se de uma pesquisa descritiva, em que se buscou observar o panorama nacional dos resíduos sólidos urbanos, por meio de análise documental, com abordagem qualitativa e quantitativa, em que o universo pesquisado envolveu a análise dos 26 Estados brasileiros mais o Distrito Federal.

O universo de investigação envolveu a análise do cenário do Brasil, bem como dos seus respectivos Estados, que foram segregados por região do país: Norte, Nordeste, CentroOeste, Sudeste e Sul.

Realizou-se uma análise dos dados, utilizando uma pesquisa pela Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais (ABRELPE). Com a análise do documento, foi possível esboçar um cenário nacional da Gestão dos Resíduos Urbanos e Coleta Seletiva, que permitiu a comparação dos resultados por meio de testes de hipótese. Para facilitar a compreensão da análise dos dados coletados, a Tabela 1 elenca as principais categorias, unidades de análise e autores utilizados para tal.

Tabela 1 – Principais categorias, unidades de análise e autores utilizados na pesquisa

Categoria de análise	Unidade de análise	Autores
Resíduos sólidos Urbanos	Geração de RSU	ABRELPE (2014); ABRELPE (2015); IBGE (2016); BRASIL (2016); Galbiati (2005); Souza, Paula e Pinto (2011); Rocha (2012); Bense (2006).
	Coleta de RSU	
	Disposição final de RSU	
	Recursos aplicados no setor	
	Geração de empregos no setor	
Coleta seletiva	Reciclagem	ABRELPE (2015); ABIPLAST (2015); Vellani e Nakao (2009); Bortoli (2013); Grimberg (2004); Bensen (2006).

Fonte: SANTOS, Tabatha et al. Cenário brasileiro da gestão dos resíduos sólidos urbanos e coleta seletiva. Anais do VI SINGEP–São Paulo–SP–Brasil–13 e, v. 14, n. 11, 2017.

Este trabalho relata a importância da coleta, onde fica evidente que a necessidade de realizar melhorias no processo de geração, coleta e destinação dos Resíduos Sólidos do Brasil. Pois, a população e as taxas de lixo vem aumentando mais a cada ano, em uma proporção ainda maior.

Segundo a pesquisa realizada pela ABRELPE, 10% dos resíduos coletados no país são dispostos de maneira irregular. Num cenário, onde mais de 3.000 (três mil) municípios ainda não retêm corretamente os resíduos, demonstrando uma situação ambienta preocupante.

3.3 Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (metal e vidro) Aplicado a Coleta Seletiva (UniEVANGÉLICA – GO)

Com auxílio da Visão Computacional, o devido trabalho, foi desenvolvido para detecção de materiais, metal e vidro, em uma coleta seletiva com o uso das tecnologias: OpenCV, TensorFlow, Python e LabelImg.

Sua premissa, baseia-se que com a constante evolução da sociedade contemporânea, há também uma crescente no desenvolvimento das tecnologias. Mas, em contrapartida a alta geração de lixo vem destruindo o meio ambiente, fazendo assim, a criação de meios sustentáveis e efetivos para solucionar esses problemas.

Em seu desenvolvimento, partiu-se de um treinamento de imagens utilizando o TensorFlow[1] utilizando pelo menos 100 imagens de cada classe de itens, para conseguir obter bons resultados. Foram obtidas as seguintes imagens, através do Google Imagens:

Tabela 2 - Base de dados para treinamento.

Classe do objeto	Especificação	Número de Imagens por classe
Metal	• Latinhas • Tampinhas • Lacres	117
Vidro	• Garrafas • Copos • Vidro quebrado	101

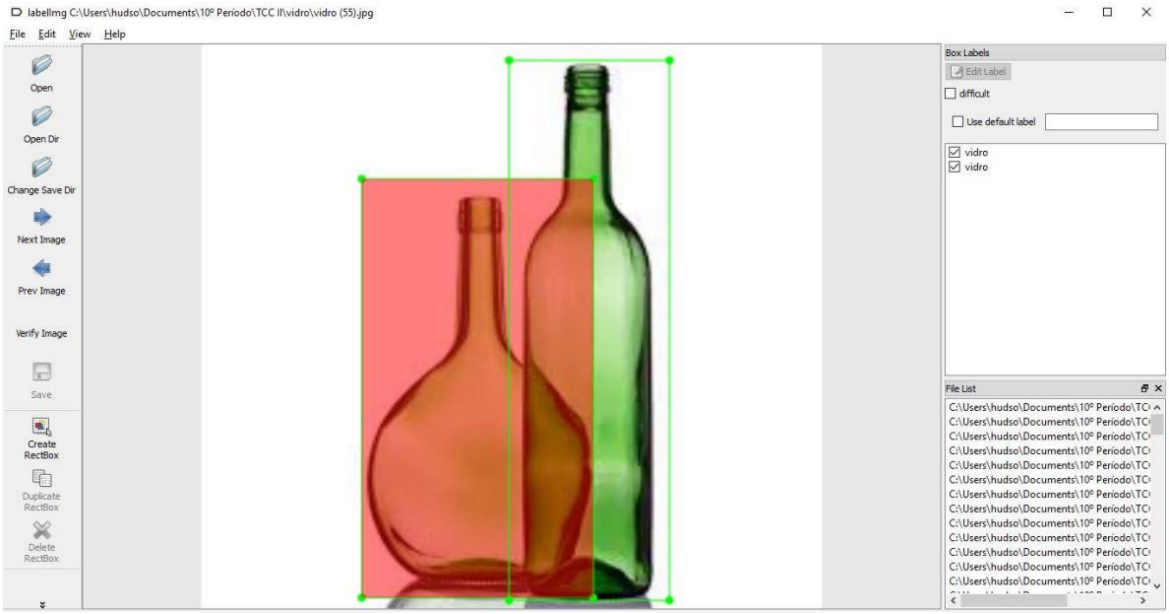
Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

Foi utilizado a técnica supervisionada de aprendizado de máquina, devido sua baixa complexidade a respeito de sua implementação, sendo assim, necessitou-se que um agente externo indicasse qual a classe do objeto em determinada imagem. Os objetos de interesse, foram mapeados em cada uma das imagens, conforme sua classe, um processo

[1] https://www.tensorflow.org/tutorials/image_retraining

que demorou cerca de 24 horas. A Imagem 9, demostra o mapeamento realizado com o software LabelImg.

Imagem 9 – Mapeamento de imagem com LabelImg.



Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

Para cada imagem de entrada, o software LabelImg gera um arquivo de saída no formato XML, que contém informações sobre as delimitações em pixels dos objetos, suas dimensões e caminho da imagem.

Para utilização do TensorFlow, realizaram a geração e conversão de arquivos, para transformar os arquivos no formato XML para o formato CSV, utilizando a um conversor genérico em Python. Depois, foi criado um arquivo “label_map.pbtxt”, que era quem rotulava as classes dos materiais (vidro e metal) e seus identificadores, esse processo foi feito para a criação de um aruivo binário no formato “.record”, que é entendível pela API do TensorFlow e contém informações das imagens, seus caminhos e segmentação dos objetos.

Ao dar-se início ao treinamento, o processo foi realizado através da execução do arquivo “train.py”, disponível na API do TensorFlow. A condição de parada para o procedimento é que a perda em cada step do treinamento esteja constantemente abaixo de 0.05, garantindo um índice de acertos aceitáveis. A quantidade de perda pode ser notada, através da Imagem 10 ou pelo gráfico de perdas gerado pela própria biblioteca, Imagem 11.

Imagem 10 – Perda por step vista no terminal

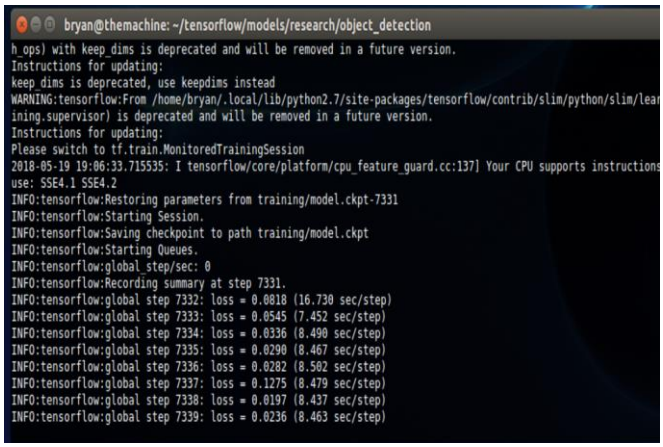
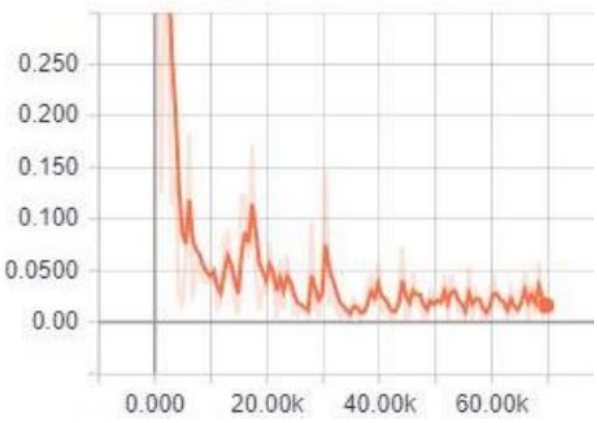


Imagem 11 – Gráfico de perda



Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

A partir do último checkpoint gerado foi exportado os dados de treinamento em único arquivo no formato “.pb”, arquivo classificador. O projeto foi implementado voltado em recebimento de vídeos através de uma webcam.

Para a codificação do sistema foi utilizado a linguagem Python na versão 2.7, que foram aplicadas as bibliotecas TensorFlow na versão 1.5 para implementação da rede neural convolucional e OpenCV para tarefas relacionadas a processamento de imagens.

Alguns impedimentos foram encontrados ao longo da implementação do sistema, apesar dos impedimentos, foram realizados testes obedecendo a seguinte ordem:

1. Teste com um objeto de metal.
2. Teste com um objeto de vidro.
3. Teste com múltiplos objetos.

A Tabela 3, contém um detalhamento dos testes com um único objetivo por tentativa e a Tabela 4, contém o detalhamento dos testes com múltiplos objetivos simultaneamente.

Tabela 3 - Testes com um objeto.

Teste	Classe	Quantidade de objetos por tentativa	Quantidade de tentativas (cada tentativa utilizou um tipo de objeto diferente)	Acertos	Erros	Porcentagem de acertos
1	Metal	1	6	4	2	66,66%
2	Vidro	1	4	4	0	100%

Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

Tabela 4 - Testes com múltiplos objeto.

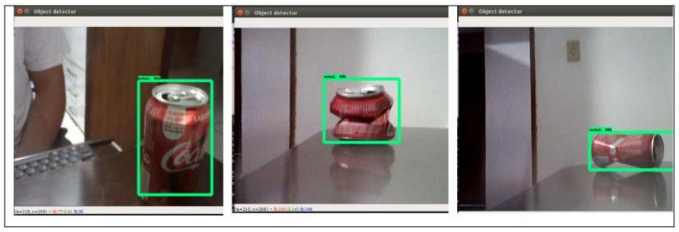
Teste	Quantidade de objetos de metal	Quantidade de objetos de vidro	Quantidade de tentativas	Acertos	Erros	Porcentagem de acertos
1	1	1	1	2	0	100%
2	1	2	1	3	0	100%
3	1	2	1	2	1	75%
4	2	2	1	4	0	100%

Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

De acordo com as tabelas acima, os resultados foram satisfatórios e o software desenvolvido conseguiu alcançar o objetivo proposto. Pois, de seis casos de testes executados, apenas um obteve taxa abaixo de 75%. A Imagem 10, demonstra um caso de teste com múltiplos objetos, onde houve uma taxa de acerto de 100%. Algumas pequenas falhas foram detectadas, pois ao obter uma imagem, a rede neural tenta realizar uma previsão a respeito do objeto presente no frame.

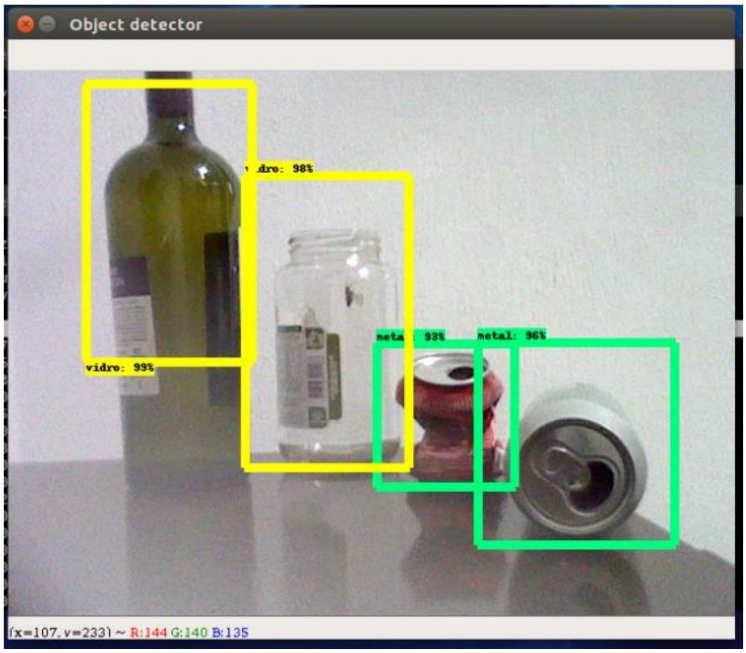
O sistema conseguiu realizar testes com a capacidade de identificação e classificação de um objeto mesmo em diferentes perspectivas, ou com alto grau de deformação. Como é possível observar na Imagem 12. Isso foi possível devido às características de absorção de padrões de dados de uma rede neural convolucional.

Imagem 12 – Identificação de lata de metal com diferentes deformações



Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

Imagem 13 – Teste com múltiplos objetos.



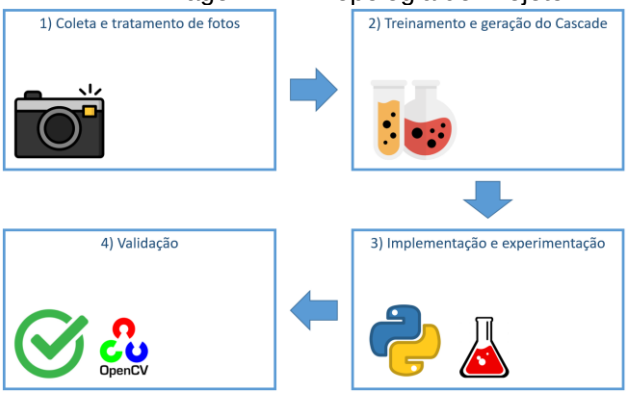
Fonte: PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

Com o estudo deste trabalho é possível observar que a biblioteca OpenCV se comportou muito bem no desenvolvimento de seu sistema. Assim, com o projeto é identificamos que o reconhecimento de imagem para materiais, nesse caso vidro e metal, funcionaram muito bem para o reconhecimento de imagens, por vídeo, para realizar uma coleta seletiva e assim, contribuindo para o desenvolvimento sustentável.

4. METODOLOGIA

Para realizar uma coleta seletiva, foi definido alguns materiais. Tais como: copo plástico, lata de alumínio, garrafa pet, pote de sorvete, casca de banana etc.

Imagem 14 – Topologia do Projeto



Fonte: Produzida pelo autor

Reconhecidos tais objetos, que serão utilizados no projeto, foi iniciado a primeira etapa do projeto que foi a coleta de imagens, no mínimo 20 imagens de cada objeto, para realizar o treinamento.

Para melhor entendimento dos processos realizados neste projeto, foi criada a ilustração da topologia do projeto, na Imagem 14. Onde as imagens foram capturadas de um celular, processadas em um notebook que estava instalado o Python versão 3.8.5 e o Pycharm, que é uma IDE de desenvolvimento gratuita.

4.1 Etapa 1 – Coleta e Tratamento de fotos

As imagens positivas, foram coletadas através de um smartfone, capturando 20 imagens do objeto a ser identificado, em diferentes posições, para um melhor reconhecimento do objeto através do treinamento.

Já as imagens negativas, como foi necessária uma quantidade maior de imagens, foram obtidas através de um banco de dados na internet, disponível para quem quiser baixar no link https://www.kaggle.com/muhammadkhalid/negative-images.

4.2 Etapa 2 – Treinamento e Geração Cascade

Para realizar a detecção de imagens, na maioria das vezes há o uso do Classificador HAAR-Cascade pré-treinado, cujo desempenho é bastante atingível com todos os objetos. Mas quando é usado o classificador pré-treinado, não conseguimos saber se o treinamento desse classificador pode ser feito. Quando questionado esse tipo de assunto, é apresentado um classificador muito simples para usar a ferramenta GUI chamada “Cascade Trainer GUI[2] (uma ferramenta projetada por Amin Ahamid)”. O Cascade Trainer GUI é um programa para realizar treinamento, testar e melhorar os modelos classificadores. Nele é usado uma interface gráfica para definir os parâmetros e tornar mais fácil a usabilidade das ferramentas OpenCV para treinamento e testes classificadores (PATIL, 2019).

Com os devidos materiais definidos, foi dado início ao treinamento de imagens no OpenCascadeGUI. Para o treinamento, foi usado 1000 imagens negativas. As imagens negativas podem ser qualquer imagem que seja a imagem positiva, na teoria, Imagem 15.

Na etapa de implementação, foram usadas 20 imagens positivas no treinamento, configurando as dimensões da imagem para 20x30 pixels e realizando 10 Stages de treinamento. As imagens positivas são os objetos que se deseja treinar e detectar seu classificador, Imagem 16.

[2] https://amin-ahmadi.com/cascade-trainer-gui/

Imagem 15 – Cascade Trainer GUI, entrada de dados (imagens negativas)

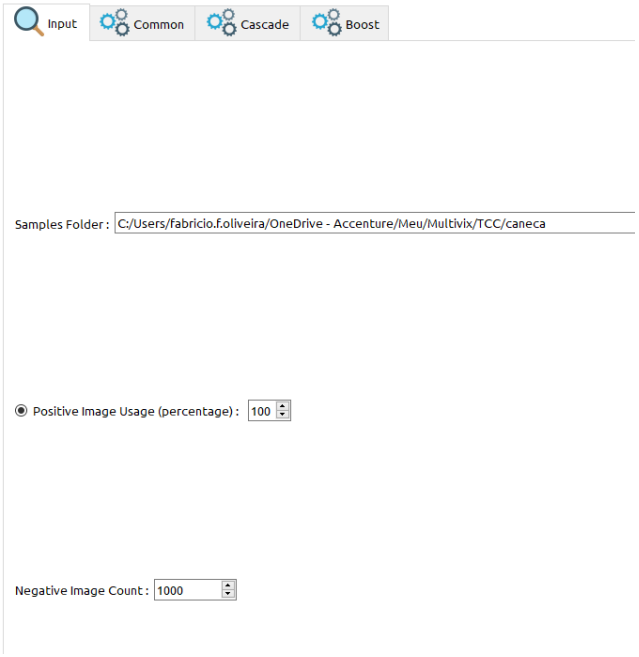
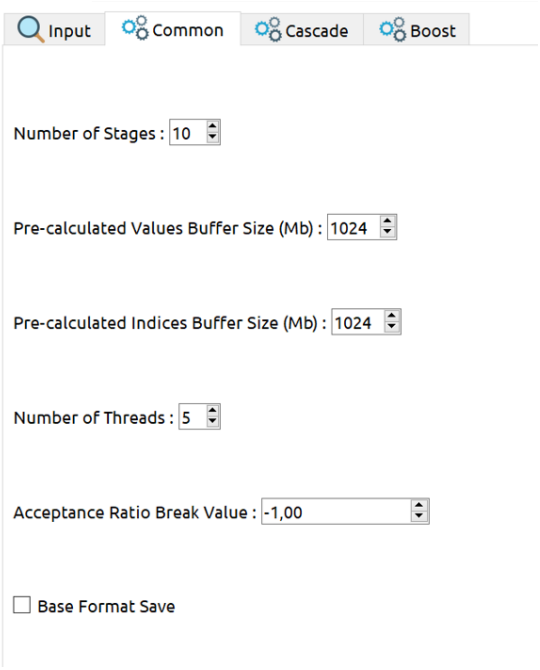


Imagem 16 – Cascade Trainer GUI, treinamento do classificador (imagens positivas)



Fonte: Produzido pelo autor

4.3 Etapa 3 – Implementação e experimentação

Com o treinamento finalizado, deu-se início a etapa de experimentação, com ajustes da imagem e sua codificação. Gerado alguns arquivos após o treinamento, foi utilizado a função cv2.CascadeClassifier para buscar arquivo “Cascade.xml”, que é o classificador em cascata real, para realizar a codificação. A codificação foi realizada em Python no aplicativo PyCharme, usando a biblioteca do OpenCV (import cv2) para realizar o reconhecimento de determinada imagem.

Imagem 17 – Codificação do projeto em Python

```python
#-*- coding:utf-8 -*-
import cv2

carregaAlgoritmo = cv2.CascadeClassifier('C:/Users/fabricio.f.oliveira/Desktop/cascade.xml')

imagem = cv2.imread('canecas.jpeg')
imagemCinza = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
objetos = carregaAlgoritmo.detectMultiScale(imagemCinza)
print(objetos)

for(x, y, l, a) in objetos:
    cv2.rectangle(imagem, (x, y), (x + l, y + a), (255, 0, 255), 2)

cv2.imshow("Objetos", imagem)
cv2.waitKey()
```

Fonte: Produzido pelo autor

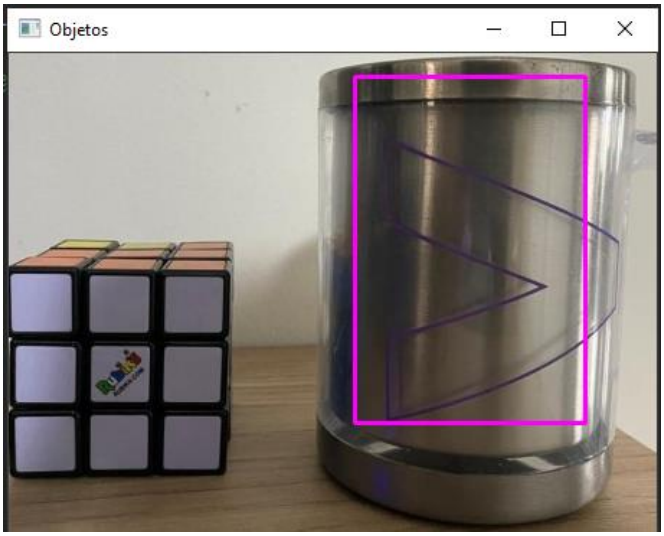
4.4 Etapa 4 – Validação

Para validar os dados, o objetivo do código era realizar a detecção do objeto, assim, validando o princípio da POC e poder pôr em prática o objetivo de realizar uma coleta seletiva de objetos recicláveis.

Em uma primeira etapa de validação, foi realizado a detecção apenas do objeto que foi treinado, na Imagem 18, é possível identificar que o código conseguiu uma ótima resposta a respeito do que ele foi desenvolvido. Na segunda validação do projeto, foi realizado a detecção do objeto que foi treinado junto a um outro objeto não treinado. O código, também obteve uma boa resposta, conseguindo identificar o objeto que foi treinado, conforme mostrado Imagem 19.

Imagem18 – Primeira Validação

Imagem 19 – Segunda Validação



Fonte: Produzido pelo autor

De acordo com as informações anteriores, tem-se como resultados de acurácia de precisão conforme descrito abaixo:

Acurácia: mostra o quanto o modelo acertou das possíveis previsões (SILLVA,2020).

$$Acurácia = \frac{Verdadeiros\ Positivos(VP) + Verdadeiros\ Negativos(VN)}{Total}$$

Precisão: dentre todas as classificações de classe positivo que o modelo fez, quantas estão corretas (SILVA, 2020).

$$Precisão = \frac{Verdadeiros\ Positivos(VP)}{Verdadeiros\ Positivos(VP) + Falsos\ Positivos(FP)}$$

Para ser calculado a acurácia e a precisão, toma-se em conta que o Verdadeiro Positivo (VP), é contabilizado quando o método previu corretamente a classe a ser prevista, neste caso, a quantidade de treinamentos que ocorreram corretamente. O Falso Positivo (FP), ocorre quando no conjunto real, a classe a ser prevista previu incorretamente, neste caso, é quando é realizado o treinamento, mas nada foi treinado. Já no Verdadeiro Negativo (VN), ocorre quando o modelo previu corretamente o que realmente não era de interesse, neste caso, quando não treina nenhum objeto onde não existe treinamento (SILVA, 2020). Obtemos então o seguinte resultado:

Tabela 5 - Resultado de amostras

Amostras	Resultado
VP	8
FP	2
VN	0
Total de Treinamentos	10

Tabela 6 – Resultado geral

	Resultado (%)
Acurácia	80%
Precisão	80%

Fonte: Produzido pelo autor

Conforme descrito nesta seção e apresentado nas imagens 18 e 19, a ferramenta Trainer Cascade GUI, de treinamento de imagens, junto com o desenvolvimento do código em Python com a biblioteca OpenCV, foi submetido a experimentos, na finalidade de avaliar se as funcionalidades estão de acordo com o objetivo proposto. Desta forma, a arquitetura tecnológica proposta atende aos requisitos descritos na solução.

5. CONCLUSÃO

O presente estudo possibilitou o desenvolvimento de uma PoC para identificação de objetos para realizar uma coleta seletiva, através da aplicação de visão computacional contida na biblioteca OpenCV. Com o auxílio da Visão Computacional e utilização da biblioteca OpenCV, é possível observar que o treinamento da imagem se comportou muito bem ao realizar a codificação para realizar a detecção do objeto por imagem. Esse projeto teve por objetivo, demonstrar a importância de se realizar a coleta seletiva para um ambiente mais sustentável e com menos poluição. A partir da implementação desse projeto, foi possível demonstrar uma possível solução para classificação de materiais recicláveis, mas para isso seria necessário o mapeamento de uma grande quantidade de objetos. Dessa forma, os impactos advindos da sua utilização é a possibilidade de realização de uma coleta seletiva mais rápida e eficaz, aumentando o número de materiais a serem reciclados e melhorando o meio ambiente.

REFERÊNCIAS

BRAVO, Thamara Lins et al. Educação ambiental e percepção da implantação de coleta seletiva de lixo urbano em de alegre, es. **Revista Gestão & Sustentabilidade Ambiental**, v. 7, n. 1, p. 375-396, 2018.

DE MILANO, Danilo; HONORATO, Luciano Barrozo. Visao computacional. 2014.

SILVA, Tarcizio. Visão Computacional e Vieses Racializados: branquitude como padrão no aprendizado de máquina. II COPENE Nordeste: Epistemologias Negras e Lutas Antirracistas, p. 29-31, 2019.

FELICIANO, Flávio F.; DE SOUZA, Igor L.; LETA, Fabiana Rodrigues. Visão computacional aplicacada à metrologia dimensional automatizada: considerações sobre sua exatidão. **Engevista**, 2005.

FERNANDES NETO, Euripedes Purcinio. Visão computacional para identificação de cores em tempo real com opencv e python. 2020

MARENGONI, Maurício; STRINGHINI, Stringhini. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, p. 125-160, 2009.

MCADAM, Rodney; MCADAM, Maura; BROWN, Valerie. Proof of concept processes in UK university technology transfer: an absorptive capacity perspective. **R&d Management**, v. 39, n. 2, p. 192-210, 2009.

MMA. Ministério do Meio Ambiente. < https://www.gov.br/mma/pt-br >, acessado em maio de 2021.

PINA, Hudson Murilo Leal; MARTINS, Michael Bryan Miranda. Utilização de Bibliotecas de Visão Computacional e Aprendizado de Máquina na Identificação de Resíduos Sólidos (Metal e Vidro) Aplicado a Coleta Seletiva. 2018.

PATIL, Suhas S. Building Custom HAAR-Cascade Classifier for face Detection. 2019.

RIBEIRO, Helena; BESEN, Gina Rizpah. Panorama da coleta seletiva no Brasil: desafios e perspectivas a partir de três estudos de caso. **InterfacEHS**, v. 2, n. 4, p. 1-18, 2007.

ROSA, Bruna Nogueira et al. A importância da reciclagem do papel na melhoria da qualidade do meio ambiente. **ENEGEP, nº XXV**, p. 6, 2005.

SILVA, LUCAS MOREIRA GUERRA. DETECÇÃO DE COLÔNIAS DE BACTÉRIAS UTILIZANDO VISÃO COMPUTACIONAL. 2020.

WWF. World Wildlife Fund. < https://www.wwf.org.br/?70222/Brasil-e-o-4-pais-do-mundo-que-mais-gera-lixo-plastico >, acessado em agosto de 2021.