



CENTRO UNIVERSITÁRIO UNIVATES
CURSO DE SISTEMAS DE INFORMAÇÃO

**SISTEMA DE VISÃO COMPUTACIONAL PARA CLASSIFICAÇÃO DE
PEDRAS NATURAIS ATRAVÉS DE VÍDEO EM TEMPO REAL**

Lucas Gallon

Lajeado, novembro de 2013

Lucas Gallon

**SISTEMA DE VISÃO COMPUTACIONAL PARA CLASSIFICAÇÃO DE
PEDRAS NATURAIS ATRAVÉS DE VÍDEO EM TEMPO REAL**

Trabalho de Conclusão de Curso II apresentado ao Centro de Tecnologia da Informação do Centro Universitário UNIVATES, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação, na área de concentração Tecnologia da Informação.

Orientador: Prof. Me. Fabrício Pretto

Lajeado, novembro de 2013

Lucas Gallon

SISTEMA DE VISÃO COMPUTACIONAL PARA CLASSIFICAÇÃO DE PEDRAS NATURAIS ATRAVÉS DE VÍDEO EM TEMPO REAL

Este trabalho foi julgado adequado para a obtenção do título de bacharel em Sistemas de Informação do CETEC e aprovado em sua forma final pelo Orientador e pela Banca Examinadora abaixo:

Prof. Fabrício Pretto, UNIVATES, Mestre em Ciência da Computação pela PUCRS, Porto Alegre - Brasil

Prof. Marcelo de Gomensoro Malheiros, UNIVATES, Mestre em Engenharia Elétrica pela UNICAMP, Campinas - Brasil

Prof. Alexandre Stürmer Wolf, UNIVATES, Mestre em Engenharia Elétrica pela PUC-Rio, Rio de Janeiro - Brasil

Lajeado, novembro de 2013

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que me acompanharam durante a minha trajetória acadêmica na UNIVATES e profissional. Em especial aos meus pais, pela dedicação e apoio em todos os momentos difíceis, e a todos os meus amigos e familiares que me incentivaram.

AGRADECIMENTOS

Aos professores da UNIVATES, pela oportunidade de realização deste trabalho que me interessou e me resultou em agregação de conhecimento.

Aos meus pais, colegas de curso e orientador, pela motivação, apoio e auxílio durante a realização deste trabalho durante o período acadêmico.

A todas as pessoas que de uma forma ou de outra contribuíram para realização deste trabalho de conclusão de curso.

RESUMO

As áreas de processamento de imagens digitais e visão computacional estão sendo utilizadas constantemente para criar novas soluções em diversos segmentos, como na medicina, robótica, aeronáutica, multimídia e entretenimento. O reconhecimento de padrões em imagens tem diversas aplicações no setor industrial, principalmente na automação de processos. Dentro deste contexto, foram mapeados os processos de negócio de uma empresa do ramo de pedras naturais, a qual tem a necessidade de classificá-las, com o intuito de separar, empacotar e vender as mesmas. Este projeto tem como objetivo o desenvolvimento de um sistema de visão computacional para classificação de pedras naturais através da captura de vídeo em tempo real realizada por uma *webcam*. Utilizando técnicas de processamento de imagens digitais e visão computacional para reconhecer e interpretar padrões de objetos com o intuito de classificá-los em categorias, automatizando o processo, até hoje realizado manualmente. O sistema foi desenvolvido utilizando a linguagem Java juntamente com a biblioteca de visão computacional OpenCV, com uma interface gráfica para permitir o gerenciamento de cadastros dos produtos, categorias, e realizar o reconhecimento das pedras. Por fim, foram realizadas validações do sistema na empresa e registrado os resultados.

Palavras-chave: Visão Computacional. Processamento de Imagens Digitais. Classificação Automática. OpenCV.

ABSTRACT

The areas of digital image processing and computer vision are constantly being used to create new solutions in various segments, such as in medicine, robotic, aeronautic, multimedia and entertainment. Pattern recognition in images has many applications in industry, especially in process automation. Within this context, the business processes of a company of natural stones was mapped, which needs to classify them, with the intent to separate, package and sell them. This project aims to develop a computer vision system for the classification of natural stones by capturing real-time video made by a webcam. Using techniques of digital image processing and computer vision to recognize and interpret patterns of objects with the intent to classify them into categories, automating the process, until now performed manually. The system was developed using the Java language together with computer vision library OpenCV with a graphical interface to allow management the entries of the products, categories, and perform recognition of the stones. Finally, system validations were performed in the company and the results recorded.

Keywords: Computer Vision. Digital Image Processing. Automatic Classification. OpenCV.

LISTA DE FIGURAS

Figura 1 – Imagem no plano cartesiano (x,y).....	18
Figura 2 - Passos do processamento de imagens digitais.....	18
Figura 3 – Classificação supervisionada de padrões	22
Figura 4 – Classificação não supervisionada de padrões	23
Figura 5 - Componentes de um sistema de visão computacional.....	26
Figura 6 - Exemplo de sistema de visão computacional.	26
Figura 7 - Processo de discretização de uma imagem.....	28
Figura 8 – Histograma em forma gráfica.....	29
Figura 9 – Histograma em forma de barras	31
Figura 10 – Exemplo de limiarização de uma imagem.	32
Figura 11 – Exemplo de detecção de bordas com o algoritmo <i>Canny</i>	34
Figura 12 – Aplicação de erosão em uma imagem.....	35
Figura 13 – Aplicação de dilatação em uma imagem.....	35
Figura 14 - Detecção de linhas com o método <i>cvHoughLine</i>	41
Figura 15 – Método <i>Blob Detection</i>	42
Figura 16 – Pedras naturais com seus respectivos histogramas.	44

Figura 17 – Pedras naturais prontas para venda.	46
Figura 18 – Fluxograma do processo de produção de pedras naturais.	46
Figura 19 – Caixa de luzes	50
Figura 20 – Duto de papel para diminuir o reflexo das luzes.....	51
Figura 21 - Arquitetura ideal do projeto.....	52
Figura 22 – Tela de cadastro de categorias.....	53
Figura 23 – Tela de cadastro de produtos.....	54
Figura 24 – Tela de captura de imagens do produto.....	55
Figura 25 – Tela de cadastro de ordens de classificação.....	56
Figura 26 – Tela de captura de imagens	57
Figura 27- Tela de configurações do sistema.	58
Figura 28 – Tela de busca de registros.	59
Figura 29 – Modelo ER do banco de dados.	60
Figura 30 - Canaleta de PVC.....	64
Figura 31 - Exemplo do processo de comparação com três imagens	65

LISTA DE TABELAS

Tabela 1 – Exemplo de histograma de uma imagem.....	30
Tabela 2 – Índices de porcentagem de semelhança das imagens	44
Tabela 3 – Produtos cadastrados para validação do sistema.	62
Tabela 4 – Produtos classificados da validação do sistema.....	63
Tabela 5 - Produtos classificados da segunda validação do sistema	67

LISTA DE ABREVIATURAS E SIGLAS

- CCD – Charge-Coupled Device - Dispositivo de Carga Acoplada
- CMOS – Complementary Metal-Oxide-Semiconductor - Semicondutor Metal-Óxido Complementar.
- EE – Elemento Estruturante
- Full HD – Full High-Definition
- JavaCV – Java Computer Vision
- KB – Kilobyte
- LED – Light-Emitting Diode - Diodo Emissor de Luz
- OCR – Optical Character Recognition - Reconhecimento ótico de caracteres
- OpenCV – Open Source Computer Vision
- PVC – Policloreto de vinila
- RGB – Red, Green and Blue – Vermelho, Verde e Azul
- USB – Universal Serial Bus

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos.....	14
1.2 Justificativa	14
1.3 Delimitação do trabalho.....	15
1.4 Metodologia	15
1.5 Estrutura do trabalho	16
2 REVISÃO BIBLIOGRÁFICA	17
2.1 Processamento de Imagens	17
2.1.1 Aquisição de imagens	19
2.1.2 Pré-Processamento	19
2.1.3 Segmentação.....	20
2.1.4 Representação e descrição	20
2.1.5 Reconhecimento e interpretação	21
2.1.6 Técnicas de Processamento de Imagens	23
2.2 Visão Computacional	24
2.2.1 Técnicas de Visão Computacional	27
2.3 Resolução Espacial	27
2.4 Histograma	28
2.5 Limiarização	31
2.6 Transformada de Hough.....	32
2.7 Canny.....	33
2.8 Erosão e Dilatação de imagem.....	34
2.9 Blob Detection	36
2.10 OpenCV	36
2.11 JavaCV	37
3 EXPERIMENTAÇÃO	39
3.1 Biblioteca utilizada	39
3.2 Equipamentos utilizados	40
3.3 Testes realizados	40
3.3.1 Detecção de objetos.....	41
3.3.2 Classificação de objetos.....	42

4 IMPLEMENTAÇÃO	45
4.1 Ramo de Negócio	45
4.2 Delimitação do problema	47
4.3 Requisitos do sistema	48
4.3.1 Requisitos funcionais.....	48
4.3.2 Requisitos não funcionais.....	49
4.4 Tecnologias envolvidas	49
4.5 Arquitetura da solução.....	50
4.6 Telas do sistema e modelo do banco de dados	53
5 VALIDAÇÃO	61
6 CONSIDERAÇÕES FINAIS.....	68
REFERÊNCIAS	69

1 INTRODUÇÃO

Com o passar dos anos as áreas de processamento de imagens digitais e visão computacional vêm crescendo, com ênfase na indústria em geral, tendo diversas empresas desenvolvendo, produzindo, comercializando e utilizando, soluções para automatização dos seus processos. Estas soluções estão sendo utilizadas em uma vasta gama de aplicações em áreas como na aeronáutica, para realizar rastreamento de aviões e traçar rotas, através do geoprocessamento de imagens por satélite, na medicina, para auxiliar no diagnóstico de doenças dos pacientes, através da análise de tomografias em 2D e 3D do corpo humano, e na robótica, para realizar tarefas automatizadas através da análise e interpretação de imagens. São utilizadas também em diversas outras áreas como: astronomia, multimídia, entretenimento, e sistemas produtivos (RUDEK; COELHO; CANGIOLIERI, 2001).

A visão computacional e o processamento de imagens digitais estão se tornando peças fundamentais para a rotina de empresas através da realização de tarefas automatizadas em substituição ao trabalho manual, sendo uma ótima solução para redução de gastos e eliminação de erros nos processos, além da criação de um padrão de verificação e qualidade.

Em meio a tantas técnicas para automação de processos, o reconhecimento de padrões por meio do processamento de imagens digitais é parte fundamental nas tarefas automatizadas, tendo sua utilização principalmente na classificação e seleção de produtos. Um exemplo de aplicação acontece na classificação de produtos através da captura de vídeo em tempo real. Em sistemas industriais, os produtos passam por uma esteira e são analisados

através de um software que processa as imagens capturadas pela câmera, para então classificar o item e decidir o caminho que o produto deve seguir.

Dentro deste contexto, este projeto apresenta uma solução para automação na classificação de pedras, para uma empresa do ramo de pedras naturais, através da captura de vídeo em tempo real.

1.1 Objetivos

Este projeto tem como objetivo principal desenvolver um sistema de visão computacional para classificação de pedras naturais, a fim de automatizar a tarefa de classificação realizada até então manualmente, fazendo uso das técnicas de processamento de imagens e visão computacional.

Os objetivos específicos deste projeto são:

- Conhecer/Estudar a área de visão computacional e processamento de imagens digitais;
- Desenvolver um sistema para classificação de pedras naturais;
- Validar a solução desenvolvida em uma empresa do ramo de pedras naturais.

1.2 Justificativa

Existem muitas empresas ao redor do mundo que possuem sistemas de visão computacional para inspecionar e classificar objetos. Dentre as empresas mais conhecidas no Rio Grande do Sul estão:

- Cognex: Possuem sistemas para realizar inspeção, identificação e orientação de peças em indústrias de médio e grande porte, com realização da aquisição de imagens em alta velocidade. Seus sistemas se dividem em várias categorias de câmeras e equipamentos de automação, abrangendo muitos ramos de negócio, como: indústria automotiva, produtos eletrônicos, alimentos e bebidas, farmacêutica, logística, entre outras (COGNEX, 2013);

- Coda: Empresa de Porto Alegre/RS, que possui projetos para qualquer tipo de indústria, customizando projetos com base nos requisitos definidos pelo cliente. A empresa

envolve-se em todas etapas do projeto, com destaque no serviço de pós-venda, dando suporte total ao cliente. Seus sistemas de inspeção, classificação e orientação dos produtos também podem ser customizáveis (CODA, 2013).

Outras grandes empresas do exterior que também desenvolvem sistemas de visão computacional, como: Vision Components (Ettlingen, Alemanha), Eye Vision Technology (Karlsruhe, Alemanha), Asentics (Siegen, Alemanha), AT Automation Technology (Bad Oldesloe, Alemanha), Siemens (Alemanha), SensoPart (Perrysburg, Estados Unidos) , entre outras.

Porém não existem muitos sistemas disponíveis no mercado com a finalidade de classificar pedras naturais, pois devido ao tamanho de variações possíveis das pedras, como em suas cores e listras, acaba se tornando difícil detectar sua categoria. Os poucos sistemas disponíveis no mercado possuem um custo elevadíssimo.

Outra possibilidade de utilização do sistema será com a classificação de outros produtos, não se restringindo a somente as pedras.

A partir deste projeto, visa-se solucionar este problema através da criação de um sistema eficaz para realizar este tipo de classificação.

1.3 Delimitação do trabalho

O presente projeto de sistema de visão computacional para classificação de pedras naturais tem por finalidade desenvolver o sistema para utilização em uma empresa de pedras naturais, localizada na cidade de Teutônia – RS.

O projeto não contempla a esteira rolante para os produtos e a separação mecanizada, mas contempla uma caixa de luzes, onde é fixada a câmera de vídeo para realizar a captura de imagens dos produtos, além do sistema de visão computacional.

1.4 Metodologia

A partir do entendimento e definição do problema e dos objetivos, através de visitas a empresa que realiza os trabalhos com as pedras naturais e classifica as mesmas manualmente,

foi realizada uma pesquisa sobre as ferramentas e bibliotecas de programação das áreas de visão computacional e processamento de imagens digitais que poderiam resolver o problema.

Para isto, foi necessário fazer uma pesquisa bibliográfica em livros e artigos sobre conceitos e técnicas das duas áreas, e nas documentações das bibliotecas de programação relacionadas.

Definida a linguagem, ferramenta e biblioteca de programação a serem utilizadas, foram realizados testes de funcionalidade de métodos presentes na biblioteca de programação para descobrir seu potencial, nesse documento definidos como experimentação no Capítulo 4.

Através disso, foi possível desenvolver uma solução para a classificação das pedras naturais, fazendo uso de técnicas das duas áreas pesquisadas.

Depois de desenvolvida a ferramenta, foi realizada a validação da solução. Aplicando a solução nas pedras da própria empresa e sob a validação do supervisor.

1.5 Estrutura do trabalho

Este trabalho está dividido em capítulos na seguinte estrutura: no Capítulo 2 será apresentado o referencial teórico contendo todo embasamento para realização do projeto, abordando conceitos e técnicas das áreas de processamento de imagens digitais e visão computacional e bibliotecas de programação relacionadas a estas áreas. O Capítulo 3 abrange a experimentação de métodos através de testes realizados com a biblioteca *Open Source Computer Vision* (OpenCV). No Capítulo 4 é apresentada a implementação do projeto, com descrição do ramo de negócio da empresa, a descrição da proposta de implementação, os requisitos do sistema, e as telas do sistema. O Capítulo 5 descreve a validação de testes realizados na empresa. Por fim, no Capítulo 6 são apresentadas as considerações finais do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo será abordado o embasamento teórico necessário para auxiliar o desenvolvimento do projeto, especificamente nas áreas de processamento de imagens digitais e visão computacional, detalhando as especificações e etapas destas áreas, as técnicas utilizadas e bibliotecas de programação relacionadas.

2.1 Processamento de Imagens

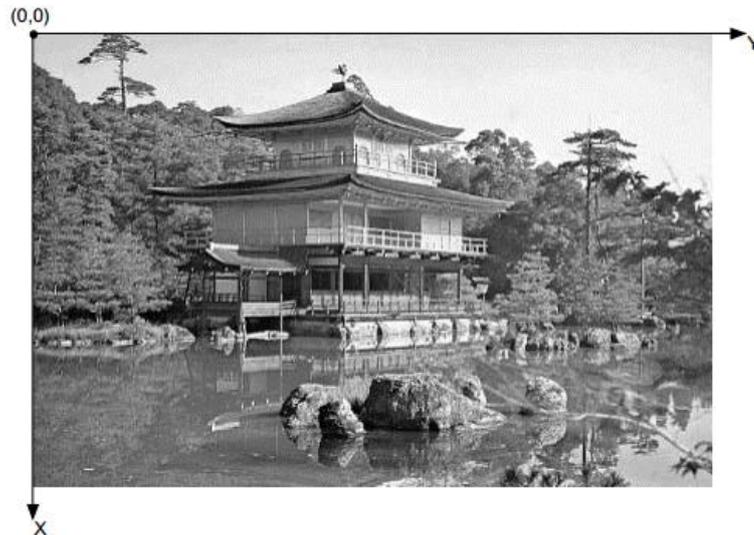
Através de fotos ou vídeos capturados por câmeras digitais, filmadoras, entre outros dispositivos, é obtida a informação necessária para se realizar o processamento de imagens. Segundo Alburquerque e Alburquerque (2000) uma imagem é um conjunto de pontos que se juntam para formar um todo, basicamente a imagem é o suporte para trocarmos informações.

Uma imagem digital é definida como uma função $f(x, y)$ como, por exemplo, na Figura 1, onde x e y são as coordenadas espaciais e o valor de f corresponde à intensidade de brilho (ou níveis de cinza) da imagem nesta coordenada. Gonzalez e Woods (2000) descrevem a seguinte definição:

Uma imagem digital pode ser considerada como sendo uma matriz cujos índices de linhas e de colunas identificam um ponto na imagem, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto. (GONZALEZ; WOODS, 2000, p. 4).

Estes pontos que são identificados na imagem são os pixels, este sendo a menor unidade que representa uma imagem, onde cada um tem o seu valor.

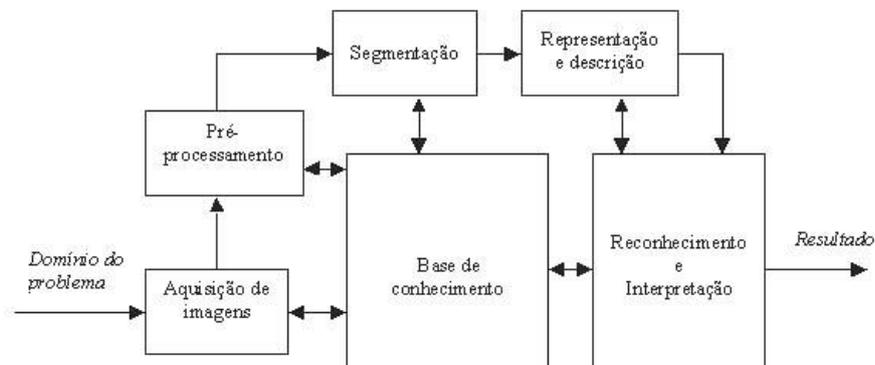
Figura 1 – Imagem no plano cartesiano (x,y).



Fonte: Marques Filho e Vieira Neto (1999).

Ainda segundo Gonzalez e Woods (2000) existem alguns passos que devem ser seguidos para processar as imagens conforme visto na Figura 2.

Figura 2 - Passos do processamento de imagens digitais.



Fonte: Gonzalez e Woods, 2000.

Nas próximas seções serão explicados detalhadamente os passos do processamento de imagens digitais.

2.1.1 Aquisição de imagens

Este é o passo inicial para o processamento de imagens, a partir de uma cena real tridimensional é realizado um processo de conversão para uma imagem analógica. Conforme Marques Filho e Vieira Neto (1999) o processo começa reduzindo a dimensionalidade da cena, a câmera digital ou de vídeo transforma a cena tridimensional em dimensional.

De acordo com Gonzales e Woods (2000) a aquisição de imagens digitais requer somente dois elementos:

- Um dispositivo físico como um sensor de imagens com dispositivo de carga acoplada (CCD), normalmente encontrada em câmeras digitais compactas ou com um semicondutor metal-óxido complementar (CMOS) encontrada em câmeras digitais semi-profissionais e profissionais;
- Um conversor de saída elétrica do dispositivo de sensoriamento físico para o formato digital, ou seja, digitalizar o sinal produzido pelo sensor.

Este passo é fundamental para obter uma imagem de boa qualidade e resolução. Conforme Marques Filho e Vieira Neto (1999) deve-se levar em conta a escolha do tipo do sensor CCD ou CMOS, as lentes a serem utilizadas, condições de iluminação da cena, velocidade da aquisição (tempo de resposta de aquisição de imagem da câmera), resolução da imagem, dentre outros.

2.1.2 Pré-Processamento

Concluída a fase de aquisição de imagem, parte-se então para o pré-processamento, onde segundo Gonzalez e Woods (2000) é realizada a melhoria da imagem, para aumentar as chances de ter sucesso nos processos seguintes. Normalmente nesta fase são feitos ajustes de contraste, redução de ruído entre outras melhorias que podem ser feitas na imagem.

2.1.3 Segmentação

O próximo passo é a segmentação, onde a imagem digital original pode ser dividida em partes ou objetos constituintes. De acordo com Marques Filho e Vieira Neto (1999, p. 11) “A tarefa básica da etapa de segmentação é a de dividir uma imagem em suas unidades significativas, ou seja, nos objetos de interesse que a compõem.”, isto é, selecionar as partes que interessam na imagem.

Segundo Gonzales e Woods (2000) a segmentação automática é um dos processos mais difíceis de fazer durante o processamento de imagens digitais, pois caso os algoritmos de segmentação funcionem de maneira incorreta podem alterar drasticamente o resultado final do processamento. Um exemplo de segmentação automática é a extração de caracteres individuais de uma imagem, como embalagens ou placas. Nesta etapa são utilizadas técnicas como a detecção de bordas, entre outras.

2.1.4 Representação e descrição

Nesta fase é realizada a transformação dos dados obtidos pela fase de segmentação, para que o computador possa processar de forma correta. Conforme Gonzales e Woods (2000) os dados podem ser representados de duas formas. A primeira é a representação por fronteiras, onde, é utilizada quando o interesse se concentra nas características da forma externa, como: pontos ou cantos. A segunda é a representação por região, onde, o interesse se concentra na área interna das formas.

Ainda de acordo com Gonzales e Woods (2000) deve-se especificar um método para descrever os dados, para enfatizar as características de interesse. A descrição extrai as características da imagem que resultem em informações quantitativas para discriminação entre as classes de objetos.

2.1.5 Reconhecimento e interpretação

O último passo no processamento de imagens é o reconhecimento e interpretação. Segundo Gonzales e Woods (2000) pode-se dizer que a análise de imagens é nada mais que um processo de descobrimento, de identificação e de entendimento dos padrões que são importantes ao desempenho de uma tarefa baseada em imagens. Dotar um computador com a capacidade de se aproximar, em um determinado sentido, a capacidade similar dos seres humanos é uma das principais metas da análise de imagens.

O reconhecimento e interpretação de imagens se enquadram no alto nível do processamento de imagens Gonzales e Woods (2000). Pode-se citar como exemplo, um sistema para a leitura automática de placas de veículos automotores, os padrões de interesse são caracteres alfanuméricos, enquanto a meta é atingir uma dada precisão de reconhecimento de caracteres que sejam a mais próxima possível à excelente capacidade exibida por seres humanos na realização de tais tarefas.

Dentro do passo reconhecimento e interpretação existem categorias, tais como o reconhecimento de padrões.

O reconhecimento de padrões tem como objetivo realizar a classificação de objetos conforme sua categoria ou classe a partir da análise de suas características. Conforme Gonzales e Woods (2000, p. 409) citam “um padrão é uma descrição quantitativa ou estrutural de um objeto ou alguma outra entidade de interesse em uma imagem [...]”, com isso o reconhecimento de padrões tem como objetivo reconhecer estas descrições e realizar sua classificação.

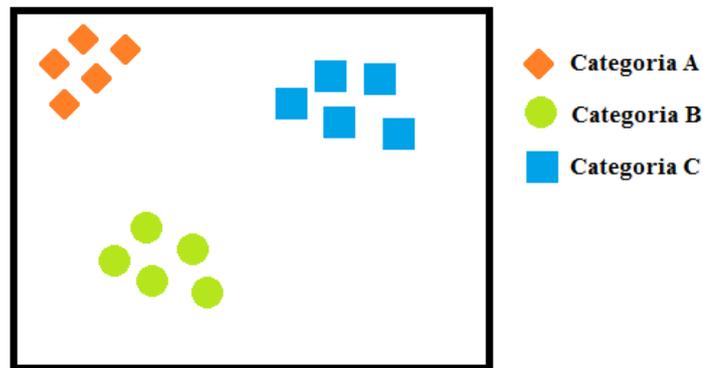
A classificação de objetos observando imagens no campo da visão computacional é uma tomada de decisão difícil de ser realizada, dependendo da quantidade de informações disponíveis para se reconhecer e classificar o objeto (MACHADO, 2008).

Existem diversas aplicações para se classificar, de acordo com Jähne (2002) um dos exemplos mais comuns é o reconhecimento de caracteres em etiquetas e outras embalagens, através do método de *optical character recognition* (OCR). Outro exemplo é a classificação de defeitos, classificando o número de defeitos e quais os defeitos mais comuns encontrados nos objetos.

Podem-se dividir os métodos de classificação em duas categorias, conforme Fonseca (2000, p. 96) “A classificação automática pode ser dividida em dois grupos: Supervisionada e Não Supervisionada”. Estes são os dois grupos principais de métodos de classificação, existe ainda um terceiro grupo chamado de Classificação Híbrida, onde a classificação Supervisionada e Não Supervisionada atuam de forma conjunta, aumentando a probabilidade de sucesso das classificações.

A classificação supervisionada utiliza uma base de dados com padrões previamente cadastrados para definir a categoria de cada objeto. Conforme demonstra Figura 3, existem três categorias cadastradas, uma laranja, outra verde e a terceira azul. Os objetos foram classificados conforme as informações cadastradas de cada categoria na base de dados (FONSECA, 2000).

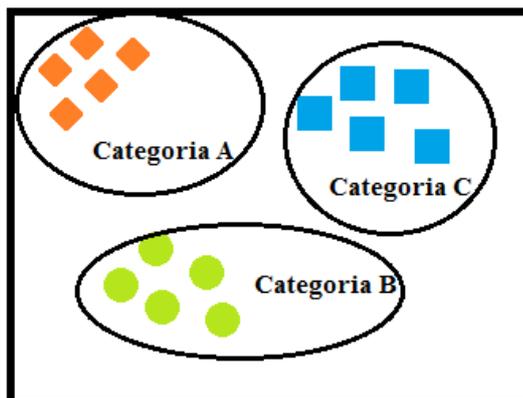
Figura 3 – Classificação supervisionada de padrões



Fonte: Elaborado pelo autor

Na classificação não supervisionada, não existem informações armazenadas sobre as categorias dos padrões reconhecidos na imagem. Através de métodos não supervisionados, são realizados agrupamentos conforme a disposição dos padrões na tela. Fonseca (2000, p. 97) cita “Os algoritmos de classificação não supervisionada são métodos de agrupamento baseados em alguma medida de similaridade”. Como exemplo na Figura 4, onde os objetos são categorizados conforme sua similaridade.

Figura 4 – Classificação não supervisionada de padrões



Fonte: Elaborado pelo autor

2.1.6 Técnicas de Processamento de Imagens

As técnicas de processamento digital de imagens e visão computacional possibilitam analisar uma cena nas várias regiões de uma imagem, integrando vários tipos de dados, de acordo com Fonseca (2000) podem-se dividir em três etapas principais as técnicas de processamento digital de imagem:

- **Técnicas de baixo nível:** Nesta fase é realizada a restauração quantitativa das imagens, com o intuito de corrigir as degradações geométricas e radiométricas inseridas pelo sensor ainda no processo de formação das imagens, e também remover o ruído da imagem;
- **Técnicas de médio nível:** Nesta fase incluem técnicas que acentuam características relevantes da imagem, que transformam a imagem numa forma que as informações possam ser extraídas com melhor resultado, como: manipulação do contraste, isolamento de regiões da imagem, entre outras;
- **Técnicas de alto nível:** Extração de informação de informações das imagens, que incluem várias técnicas, como: segmentação da imagem repartindo a imagem em regiões com características diferentes, classificação da imagem, reconhecendo objetos e padrões. Os resultados destas técnicas resultam em descrições dos resultados esperados.

2.2 Visão Computacional

A visão computacional atua juntamente com o processamento de imagens, analisando imagens e vídeos para obter resultados similares com o que o ser humano enxerga. Conforme Milano e Honorato (2010), a visão computacional é a tecnologia e ciência responsável pela forma como um computador enxerga o meio à sua volta, extraindo informações significativas a partir de imagens capturadas por câmeras de vídeo, sensores, scanners, entre outros dispositivos. Estas informações permitem reconhecer, manipular e pensar sobre os objetos que compõem uma imagem.

Segundo Marengoni e Stringhini (2009) a máquina realiza processos para tentar aproximar-se ao máximo com a realidade da visão humana. A sua existência não é tão antiga, os primeiros artigos mencionam sua origem entre a década de cinquenta e sessenta, mesmo não possuindo computadores capazes de processar imagens, porém somente no final da década de setenta foram realizados os primeiros trabalhos com a visão computacional, juntamente com a inteligência artificial.

Naquela época imaginava-se que no futuro a máquina iria ter o sentido de visão igual ou superior a do ser humano, mas com o desenvolvimento de novas pesquisas e com o avanço da tecnologia nas décadas adiante, percebeu-se que o grau de complexidade para isso acontecer era muito maior do que tinham previsto. Os olhos do ser humano conseguem detectar e interpretar objetos em imagens numa velocidade muito rápida.

Muitos cientistas tentam entender o funcionamento do córtex visual do cérebro, onde são processadas as imagens captadas pelo olho. Esta é uma das partes mais complexas do cérebro (MILANO; HONORATO, 2010). Eles analisam o funcionamento para por em prática novas ideias com o objetivo de evoluir gradativamente a visão computacional até o ponto em que não será possível diferenciar a visão humana com a visão da máquina, de tão perfeita que será sua imagem.

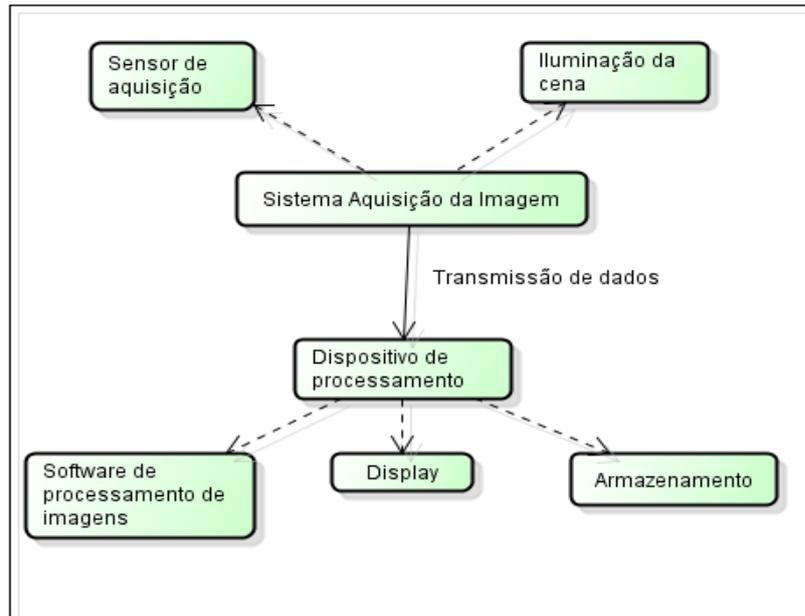
Conforme os anos passam, muitas tecnologias novas na área de visão computacional e processamento de imagens estão surgindo, utilizadas em uma vasta gama de aplicações em áreas como: aeronáutica, astronomia, medicina, multimídia, entretenimento, robótica, sistemas produtivos, entre muitas outras (RUDEK, COELHO, CANGIOLIERI; 2001).

Segundo Milano e Honorato (2010), nos dias atuais a visão computacional oferece uma infinidade de informações precisas através da captura de imagens e vídeos, a fim de tornar possível a máquina à realização de tarefas inteligentes, tornando cada vez mais próxima a inteligência do ser humano.

Um sistema de visão computacional necessita de diversos componentes que estão interligados e dependem um do outro para funcionar. Jähne (2000) definiu uma estrutura geral dos componentes que envolvem um sistema de visão computacional conforme a Figura 5 demonstra. Os componentes e suas respectivas descrições são os seguintes:

- **Sensor de aquisição:** O sensor da câmera fotográfica ou de vídeo é um componente fundamental para a aquisição da imagem digital, sendo o primeiro passo da transformação da imagem real para digital;
- **Iluminação da cena:** Este componente é indispensável para captura de imagens com qualidade, e assim poder trabalhar coerentemente com o objeto que esta na cena;
- **Sistema de aquisição da imagem:** Normalmente é composto por um dispositivo de aquisição de imagens como uma câmera fotográfica ou de vídeo, por onde é realizada a aquisição da imagem real, processada, transformada em digital e transmitida para um computador conectado a este dispositivo;
- **Dispositivo de processamento:** Deve dispor de um hardware de alto desempenho para poder realizar o processamento das imagens digitais. Geralmente são utilizados computadores para realizar o processamento;
- **Software de processamento de imagens digitais:** Responsável por analisar e gerar todas as informações sobre as imagens. Realizando a manipulação das imagens, reconhecimento e classificação de objetos, dentre outros processos;
- **Display:** Monitores ou outros dispositivos de saída de vídeo para exibição das informações geradas pelo software de processamento de imagens;
- **Armazenamento:** Utilizado para armazenar todas as informações coletadas com o software de processamento de imagens.

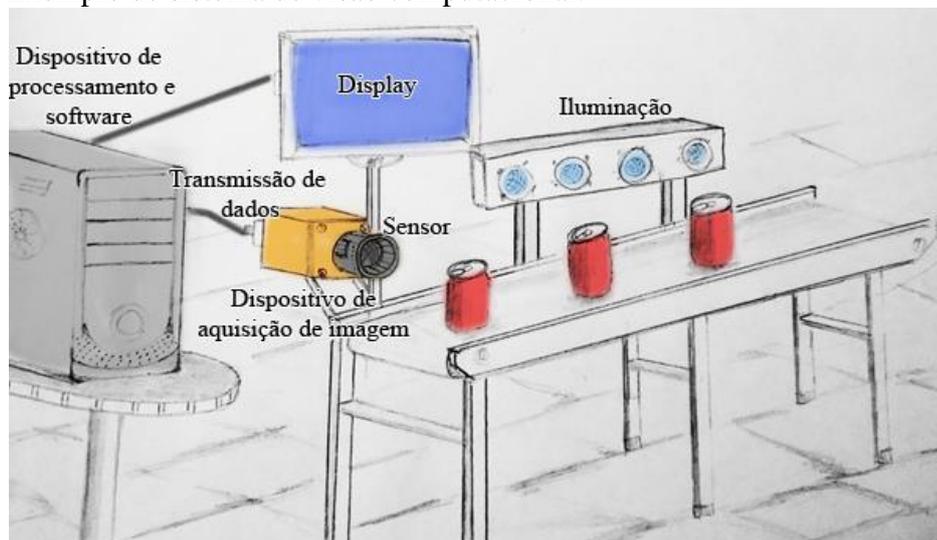
Figura 5 - Componentes de um sistema de visão computacional.



Fonte: Elaborado pelo autor.

Com base no que Jähne (2000) definiu sobre os componentes de um sistema de visão computacional, pode-se observar na Figura 6 um exemplo de aplicação destes componentes, em um processo de reconhecimento e interpretação de rótulos de latas de refrigerante que passam em uma esteira rolante.

Figura 6 - Exemplo de sistema de visão computacional.



Fonte: Elaborado pelo autor.

2.2.1 Técnicas de visão computacional

Segundo Stemmer et al (2005), basicamente todas as técnicas de visão computacional são equivalentes as técnicas de alto nível do processamento de imagens, como o reconhecimento e interpretação de objetos e padrões, classificação de objetos.

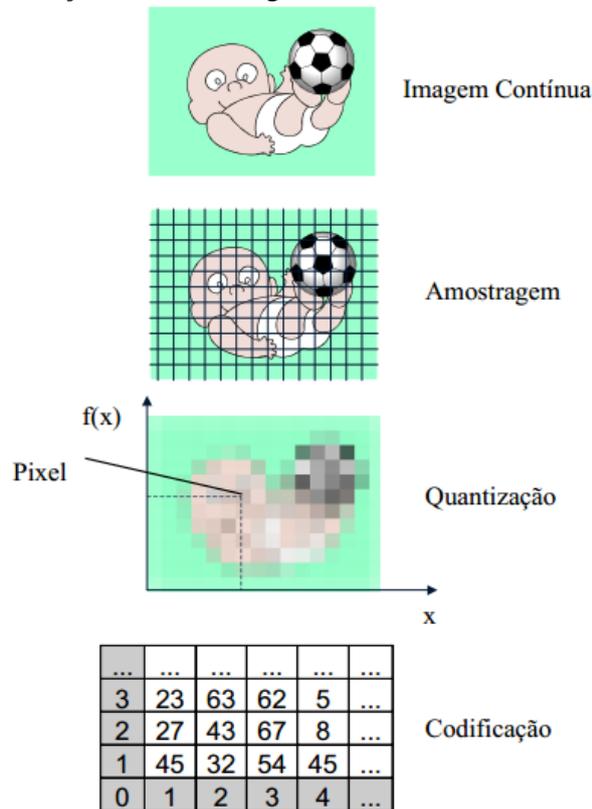
Refere-se ao processamento digital de imagens para a extração de características importantes, que auxiliem na compreensão da imagem e na tomada de decisões inteligentes. Enquadra-se este tipo de operação de processamento como de “nível alto”, sendo comum a utilização de técnicas de inteligência artificial para o processamento dos dados. Um exemplo seria o reconhecimento e a classificação de um objeto da imagem (Stemmer et al, 2005, p. 2).

Marques Filho e Vieira Neto (1999) complementam que o processamento de imagens lida com informações sobre as imagens para interpretação humana e a visão computacional é a análise automática realizada pelo computador de informações obtidas a partir de uma ou várias imagens.

2.3 Resolução Espacial

A partir da captura de uma imagem através do sensor CCD ou CMOS da câmera digital, a imagem passa por um processo de discretização, para que possa se tornar digital (SCURI, 2002). Observa-se na Figura 7 como funciona o processo de discretização da imagem.

Figura 7 - Processo de discretização de uma imagem.



Fonte: Scuri, 2002.

Quando a imagem está digitalizada, ela se torna nada mais do que um conjunto de pixels armazenados em um arquivo, cada pixel com seu valor do brilho naquele ponto. Segundo Moreira (2011, p. 2) “Uma imagem de alta resolução usualmente reproduz mais detalhe e uma mais suave transição de cores, do que uma imagem de mais baixa resolução.”. Isto é, quanto maior for o número de pixels, melhor será a qualidade da imagem.

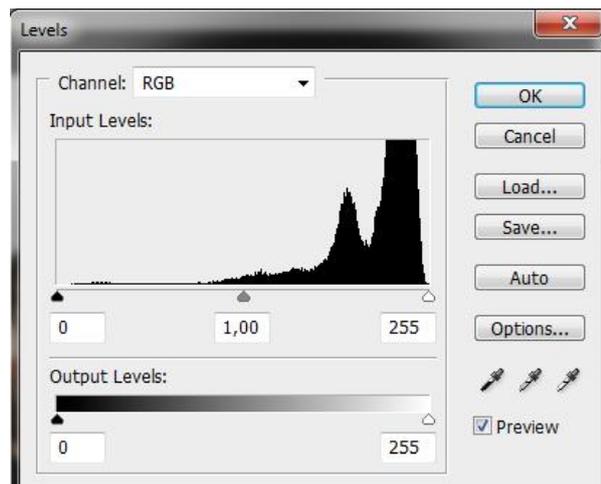
2.4 Histograma

Os histogramas são utilizados para obter indicações da qualidade de imagem quanto ao nível de contraste, ou conforme seu nível médio caso a imagem seja clara ou escura. Conforme Marques Filho e Vieira Neto (1999) o histograma de uma imagem nada mais é do que um conjunto de números que indicam o percentual de pixels na imagem que apresentam

um determinado nível de cinza. Normalmente são demonstrados em gráficos de barras, ou em texto com o número ou percentual de pixels correspondentes na imagem.

Na Figura 8 observa-se no histograma em forma gráfica que a imagem possui mais pixels claros com o índice de cinza mais elevado.

Figura 8 – Histograma em forma gráfica.



Fonte: Elaborado pelo autor.

Através da visualização do histograma de uma imagem obtêm-se uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio.

Pode se aplicar o conceito de histograma também em imagens coloridas. A imagem é decomposta e para cada componente, por exemplo, *Red, Green and Blue*¹ (RGB), é calculado o histograma correspondente.

Cada elemento deste conjunto é calculado segundo a Equação 1:

$$p_r(r_k) = \frac{n_k}{n} \quad (1)$$

¹ RGB: Modelo de três cores primárias, sendo possível gerar qualquer outra cor através de uma combinação entre elas, utilizada em software de processamento de imagens, monitores de vídeo, televisores, câmeras fotográficas e de vídeo, entre outros. (MARQUES FILHO; VIEIRA NETO, 1999).

Onde:

$$0 \leq rk \leq 1$$

$k = 0, 1, \dots, L-1$, onde L é o número de níveis de cinza da imagem digitalizada;

n = número total de pixels na imagem;

$pr(rk)$ = probabilidade do k -ésimo nível de cinza;

n_k = número de pixels cujo nível de cinza corresponde a k .

Para exemplificar a equação de cálculo de um histograma, Marques Filho e Vieira Neto (1999), utilizam uma imagem com oito níveis de cinza no tamanho de 128 x 128 pixels e realizam os cálculos da equação conforme Tabela 1.

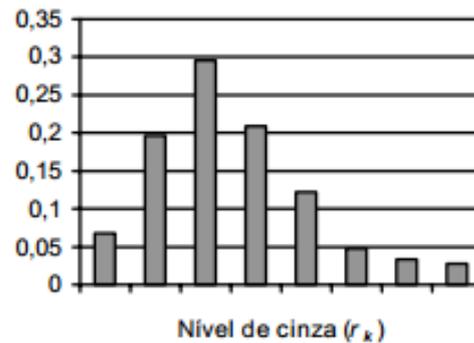
Tabela 1 – Exemplo de histograma de uma imagem.

Nível de cinza (rk)	N_k	$pr(rk)$
0	1120	0,068
1/7	3214	0,196
2/7	4850	0,296
3/7	3425	0,209
4/7	1995	0,122
5/7	784	0,048
6/7	541	0,033
1	455	0,028

Fonte: Marques Filho e Vieira Neto, 1999.

Onde a primeira coluna representa o número do nível de cinza, a segunda coluna representa o número de pixels correspondente ao nível de cinza, e a terceira coluna representa a probabilidade de aparição do nível de cinza na imagem. O gráfico gerado através destes cálculos é demonstrado na Figura 9.

Figura 9 – Histograma em forma de barras



Fonte: Marques Filho e Vieira Neto, 1999.

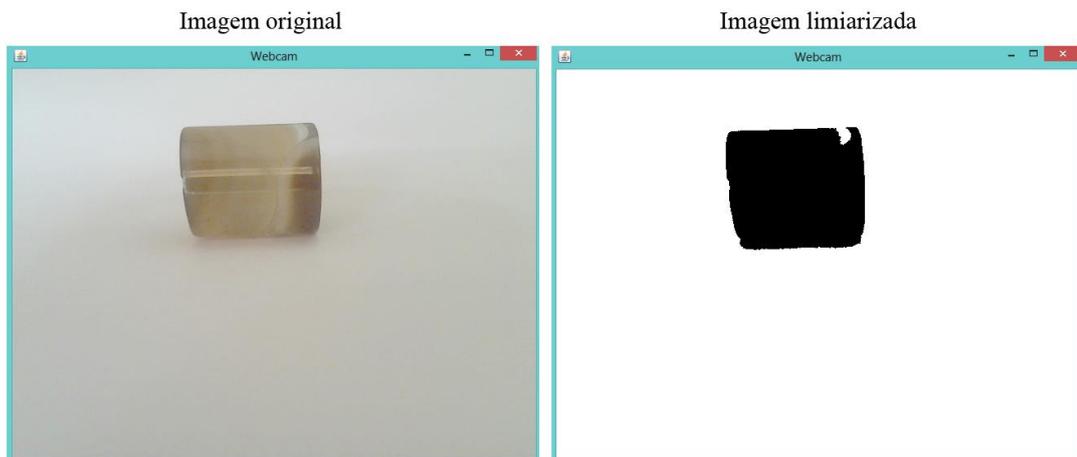
2.5 Limiarização

A limiarização (*thresholding* em inglês) é uma das técnicas principais no processamento de imagens digitais para se realizar o reconhecimento e interpretação de objetos em imagens. É uma técnica de médio nível, onde a imagem é separada em duas visões, o fundo e o objeto.

Segundo Marques Filho e Vieira Neto (1999) a limiarização de imagens é feita basicamente pela conversão dos pixels em somente dois tons: branco ou preto. Transformando os pixels com certo valor de tom de cinza igual ou maior de limiar (T) em pixels brancos e o restante em pretos, resultando em uma imagem de cores binárias.

Na Figura 10 pode-se observar a aplicação de limiarização em uma imagem, onde o objeto é uma pedra natural e o fundo é uma folha branca, o valor de (T) utilizado foi 100. Pode-se perceber que todos os pixels abaixo do valor de (T) foram transformados em pixels pretos, e os que estavam acima deste valor, em brancos.

Figura 10 – Exemplo de limiarização de uma imagem.



Fonte: Elaborado pelo autor.

2.6 Transformada de Hough

São muitas as formas de trabalhar-se com as técnicas de visão computacional, sendo que a Transformada de Hough é uma das principais ferramentas. Originalmente desenvolvida por Paul Hough em 1962 e posteriormente patenteada pela IBM, inicialmente tinha como objetivo realizar a detecção de características analiticamente representáveis em imagens binarizadas, como círculos, eclipses e linhas.

A cada ano que passa a Transformada de Hough vem aprimorando-se e tornando-se uma ferramenta fundamental na visão artificial para o reconhecimento destas características. Atualmente esta técnica pode ser também utilizada para detectar outras formas geométricas. (FISHER et al., 2003).

A Transformada de Hough pode ser utilizada para detectar linhas através do método *hough line*. Através da transformação de gradientes e da limiarização na imagem são obtidas as linhas, detectando primeiramente a maior linha da imagem e depois sucessivamente a maior das linhas restantes (DUDA; HART, 1971).

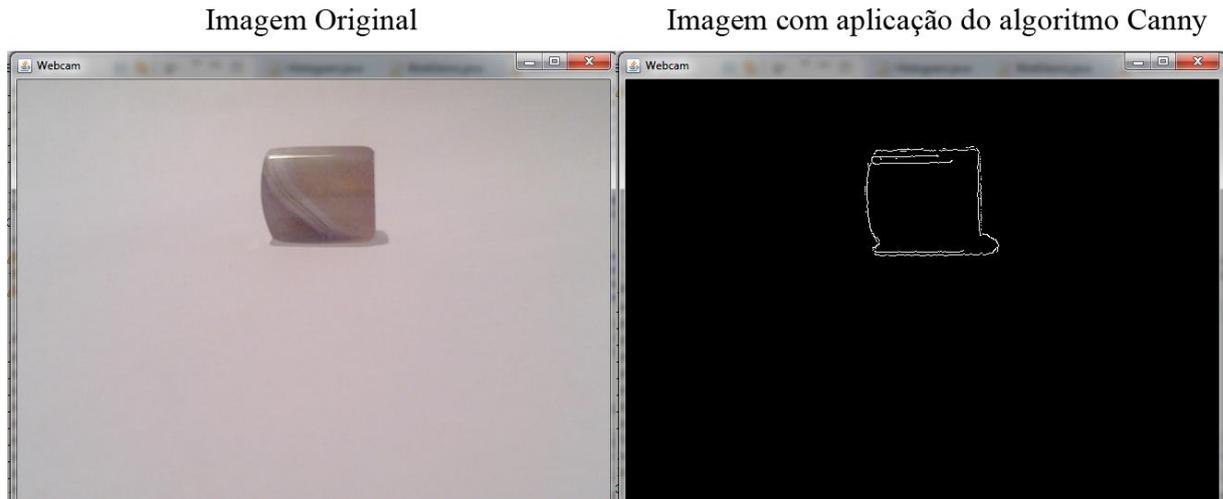
2.7 Canny

A técnica *Canny* é uma técnica de médio nível, sendo utilizada para detectar bordas de objetos dentro de uma imagem, através de um algoritmo desenvolvido em 1986 escrito por John F. Canny no artigo “*Computational Approach To Edge Detection*”. Segundo Canny (1986), neste algoritmo a detecção de bordas começa a partir de três critérios principais que são os seguintes:

- Boa detecção: A imagem deve ter uma qualidade mediana para haver uma baixa probabilidade de não conseguir detectar as bordas ou marcar bordas falsas assim tornando a detecção incorreta;
- Boa localização: As bordas que marcadas devem estar o mais próximo possível da borda verdadeira;
- Uma única resposta para cada borda: Cada borda localizada na imagem deve ser marcada somente uma vez, não criando duas ou mais respostas para cada borda, por isso o ruído da imagem não pode gerar bordas falsas.

Para satisfazer estes critérios o algoritmo de *Canny* realiza inicialmente uma redução de ruído, desfocando minimamente a imagem original, para poder reduzir os pixels ruidosos da imagem. Após isto é realizado um cálculo para determinar quatro máscaras na imagem com ângulos diferentes, a partir da variação brusca da intensidade de cor da imagem, isso é, seu gradiente. Seus resultados são guardados e para cada pixel é marcado o gradiente de maior valor, e o ângulo da máscara que encontrou esse valor. Então são criados dois mapas, um com os gradientes de intensidade de cada ponto na imagem, e outro do sentido do gradiente de cada ponto. Por fim é realizada a detecção das bordas da imagem, sendo os gradientes armazenados com maior índice de intensidade os mais prováveis a serem bordas. Conforme a Figura 11 pode-se ver um exemplo de detecção de bordas de um objeto em uma imagem através deste algoritmo.

Figura 11 – Exemplo de detecção de bordas com o algoritmo *Canny*.



Fonte: Elaborado pelo autor.

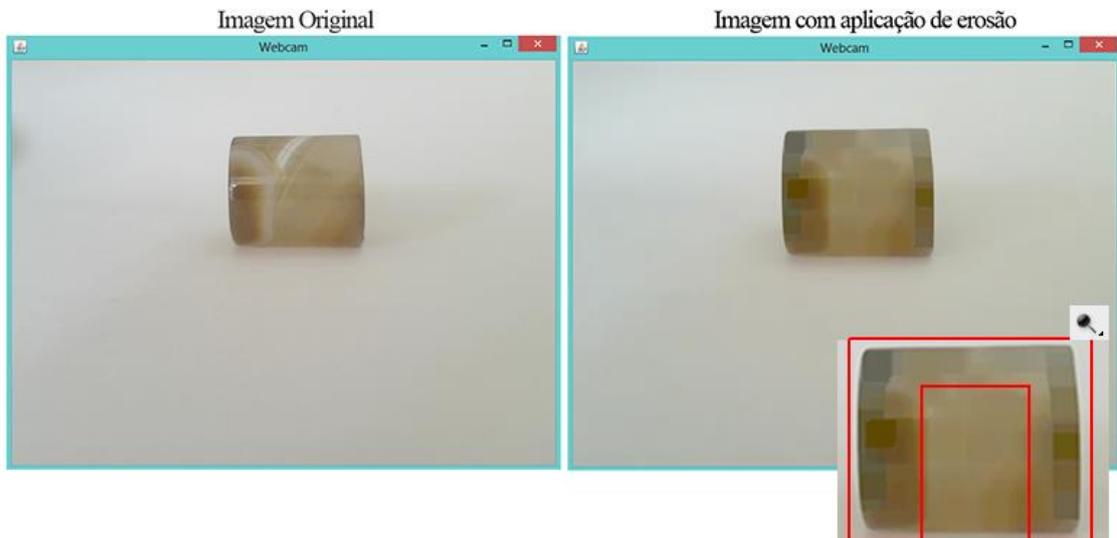
2.8 Erosão e Dilatação de imagem

Dois conceitos importantes de transformação de imagem para as técnicas de processamento de imagem digital são a erosão e dilatação de imagens, utilizados para suavização e redução de ruídos da imagem.

Conforme Júnior (2009) o primeiro conceito destes processos é o elemento estruturante (EE), que indica qual comportamento deve ser seguido. Os objetos que estão presentes na imagem assumem formas pelo EE, como de discos, quadrados, linhas, losangos, entre outros. Influenciando diretamente no resultado final da transformação da imagem.

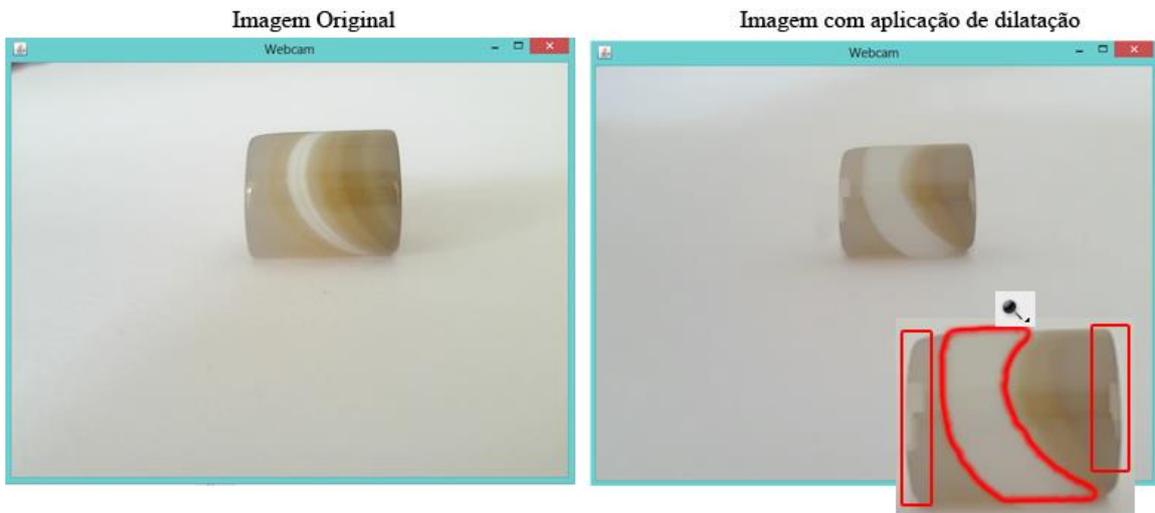
Júnior (2009) afirma que através dos formatos assumidos do EE podem ser dilatadas ou erodidas as bordas de um objeto ou região da imagem. Pode-se observar na Figura 12 um exemplo de erosão na imagem de uma pedra natural, onde se verificou que a erosão provocou uma redução nas cores mais claras do objeto, formando quadrados nas cores mais escuras. Na Figura 13 um exemplo de dilatação de imagem deste mesmo objeto, percebeu-se que as partes mais escuras foram dilatadas, evidenciando as cores mais claras do objeto.

Figura 12 – Aplicação de erosão em uma imagem.



Fonte: Elaborado pelo autor.

Figura 13 – Aplicação de dilatação em uma imagem.



Fonte: Elaborado pelo autor.

2.9 Blob Detection

Blob Detection é um algoritmo que detecta objetos na tela e retorna suas posições de x e y mínimas e máximas na imagem, assim tendo a possibilidade de obter a medida dimensional de objetos a partir de *blobs* conectados um ao outro. De acordo com Rojas (2012, p. 2) “*blobs* são componentes numa imagem binária que estão conectados [...]”.

Para descobrir o que são *blobs* em uma imagem o algoritmo limiariza a imagem, analisa e descobre quais pixels são de segundo plano e quais são de primeiro plano, os de primeiro plano se tornam *blobs*, rotulando os de segundo plano com o valor de zero e os de primeiro plano com o valor de um. Após são detectadas os contornos de cada *blob* para então ser feita a conexão dos *blobs* formando assim um objeto, e então é determinado seu contorno externo para retornar os valores de mínimos e máximos de x e y na imagem. (CHANG; CHUN-JEN; CHIN-JEN L, 2003).

2.10 OpenCV

O OpenCV é uma biblioteca de programação construída na linguagem de programação C, com seu código aberto, desenvolvida originalmente pela Intel Corporation, que implementa uma variedade de ferramentas de interpretação de imagens, partindo desde operações mais simples como um filtro de ruído, até operações complexas, tais como a análise de movimentos, reconhecimento de padrões e reconstrução em 3D (OPENCV, 2013). Segundo Marengoni e Stringhini (2009) atualmente ao todo possui mais de 500 métodos. Conforme Wilson (2001) o OpenCV visa tornar a visão de computador acessível a programadores e usuários na área de interação em tempo real humano-computador e robóticas.

De acordo com Bradski e Kaehler (2008) o OpenCV é compatível com os sistemas operacionais Windows, Linux e Mac OS, podendo ser utilizado o processamento paralelo entre os múltiplos núcleos do processador.

Existem diversas áreas de aplicação do OpenCV, dentre elas estão:

- Reconhecimento facial;
- Realidade aumentada;

- Realidade virtual;
- Reconhecimento de movimentos;
- Gravação e reprodução de vídeos em tempo real;
- Reconstrução de cenas e objetos em 3D.

Segundo Bradski e Kaehler (2008) os métodos da biblioteca estão divididas em cinco grupos:

- Processamento de imagens digitais;
- Análise Estrutural;
- Análise de movimentação e rastreamento de objetos;
- Reconhecimento de padrões;
- Calibração da câmera para reconstrução de cenas 3D.

2.11 JavaCV

A *Java Computer Vision* (JavaCV) é uma interface construída na linguagem de programação Java que fornece uma coleção de classes de outras bibliotecas em outras linguagens normalmente utilizadas por pesquisadores da área de visão computacional. Conforme Audet (2013) ela oferece compatibilidade com as seguintes bibliotecas:

- OpenCV: Oferece centenas de métodos utilizados na área de visão computacional, sua licença é gratuita para qualquer tipo de uso. Atualmente está disponível em C, C++, Python e Java;
- FFMpeg: Software utilizado para capturar, converter e reproduzir vídeos e áudios. Possui uma biblioteca de codecs de áudio e vídeo;
- FlyCapture: Biblioteca utilizada para controle de câmeras digitais;

- libdc1394: Biblioteca para controle de câmeras específicas do tipo IIDC e DCAM;
- OpenKinect: Biblioteca de funções do acessório Kinect do console Xbox 360, podendo ser utilizada em outras plataformas, como em computadores com sistema operacional Windows, Linux ou Mac;
- ArToolKitPlus: Biblioteca utilizada para aplicações de Realidade Aumentada, reconstrói cenas 3D em tempo real;
- videoInput: Biblioteca utilizada para captura de vídeos no Windows.

A JavaCV foi projetado visando utilizar grande parte dos métodos foram escritos em OpenCV, porém executando através de códigos em Java. Outra grande vantagem é sua compatibilidade com o Android, pois também foi adaptada a esta linguagem.

Esta interface é gratuita e pode ser utilizada por qualquer um e para qualquer tipo de uso, seja comercial ou acadêmico, atualmente possui um número maior de exemplos de aplicações do que a versão nativa em Java do OpenCV lançada em fevereiro de 2013, disponível em seu site oficial (JAVACV, 2013).

3 EXPERIMENTAÇÃO

Com a finalidade de conhecer, praticar e descobrir o potencial que os métodos de visão computacional e processamento de imagens digitais têm a oferecer, foram realizados testes com alguns métodos da biblioteca OpenCV para analisar quais se eram adequados ao projeto. Estes testes contemplaram a construção de pequenos exemplos de identificação e classificação de objetos.

3.1 Biblioteca utilizada

Com base nas pesquisas realizadas nas áreas de visão computacional e processamento de imagens, sobre a utilização de suas técnicas de baixo, médio e alto nível, foi encontrada a biblioteca mais utilizada, a OpenCV. Para os recursos desta biblioteca na linguagem de programação Java foi utilizada a interface JavaCV.

Para definir a biblioteca que seria utilizada no projeto e utilização da interface foram levantados os seguintes pontos:

- Afinidade com a linguagem;
- Portabilidade entre sistemas operacionais;
- Orientação a objetos;
- Existência de exemplos.

Devido a estes fatores, a biblioteca escolhida foi a OpenCV, pois atualmente ela possui muitos exemplos para a interface JavaCV de aplicação de seus algoritmos e métodos escritos em C.

Para que fossem realizados os testes com o OpenCV utilizando o sistema operacional Windows XP, foi necessária a instalação da última versão do Java JDK, do Microsoft Visual C++ 2010 Redistributable Package, e foi utilizada a versão 7.3 do IDE NetBeans, posteriormente foram instaladas as bibliotecas OpenCV 2.4.5 e finalmente a interface JavaCV 0.5.

3.2 Equipamentos utilizados

Para realizar os testes foram utilizados dois tipos de câmera, no primeiro momento uma *webcam* de 0,3 *megapixels* de um *notebook* Acer Aspire Timeline X, e no segundo momento uma câmera de maior qualidade na captura de imagens, uma Logitech C525 com capacidade de filmar HD 720p, conectada a porta *Universal Serial Bus* (USB) do *notebook*, que dispôs do display de imagens e do armazenamento de informações.

Todos os experimentos foram realizados em ambiente com luz natural sem a utilização de outras fontes de iluminação artificial.

3.3 Testes realizados

Foram realizados vários testes com diversos métodos através da biblioteca JavaCV, para detectar um objeto em um vídeo em tempo real e para reconhecer o padrão do objeto para realizar sua classificação. A documentação da biblioteca está disponível no site oficial “<https://code.google.com/p/javacv>”, bem como exemplos, e outras dicas de instalação e utilização dos métodos disponíveis.

3.3.1 Detecção de objetos

Para realizar os testes de detecção de objetos foram utilizados os métodos disponíveis na biblioteca OpenCV.

A partir da pesquisa realizada sobre a Transformada de Hough, no primeiro momento foram realizados testes com o método *cvHoughLine* como demonstra a Figura 14, filmando com a *webcam* do *notebook* uma pedra natural em cima de uma folha branca a aproximadamente 15 centímetros de distância da *webcam*, para poder focar corretamente a pedra. Primeiramente a imagem foi transformada em preto e branco, em seguida foi aplicado o algoritmo de *Canny* (*cvCanny*) para detectar as bordas, e por último foi aplicado o método Hough Line para detectar as linhas da imagem.

Figura 14 - Detecção de linhas com o método *cvHoughLine*.



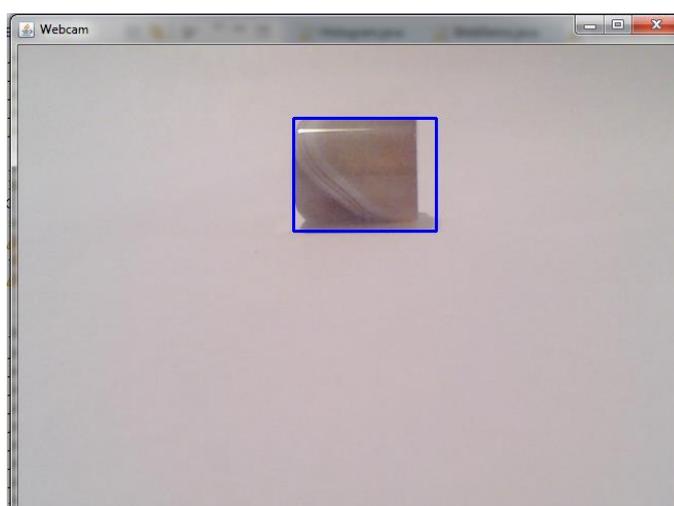
Fonte: Elaborado pelo autor.

A partir deste teste foi possível obter linhas de contorno da pedra natural que estava no vídeo. Devido a iluminação insuficiente e a qualidade da *webcam* utilizada, o resultado não foi adequado, não tendo sucesso na detecção das bordas do objeto e detectando a sombra do objeto como uma borda, porém pode-se notar que o algoritmo funcionou.

O segundo teste a ser realizado foi com o método *Blob Detection* conforme demonstra a Figura 15. Para utilizar este método adequadamente, a imagem passou por várias etapas de

processamento, começando por uma transformação em preto e branco, em seguida foi feita a limiarização (*cvThreshold*) da imagem para realçar o objeto, em seguida, foi aplicado os métodos de erosão (*cvErode*) e dilatação (*cvDilate*) da imagem para reduzir os ruídos, após foi detectado as bordas com o algoritmo de *Canny* (*cvCanny*) e por fim, foi aplicado o método *Blob Detection*. Assim foi possível obter a medida dimensional de um objeto em uma imagem, também pode se utilizar este método para descobrir quando o objeto passa em frente ao vídeo. Comparado com o método *cvHoughLine*, o *Blob Detection* demonstrou ser mais prático e eficiente.

Figura 15 – Método *Blob Detection*.



Fonte: Elaborado pelo autor.

3.3.2 Classificação de objetos

Executando o método *cvHistogram* que utiliza o conceito de histograma, foi possível obter os valores das cores RGB de duas imagens e comparar assim o histograma de cada uma, para poder obter uma porcentagem de semelhança.

No segundo momento utilizou-se uma *webcam* Logitech C525, para realizar os testes de classificação supervisionada com os histogramas das imagens.

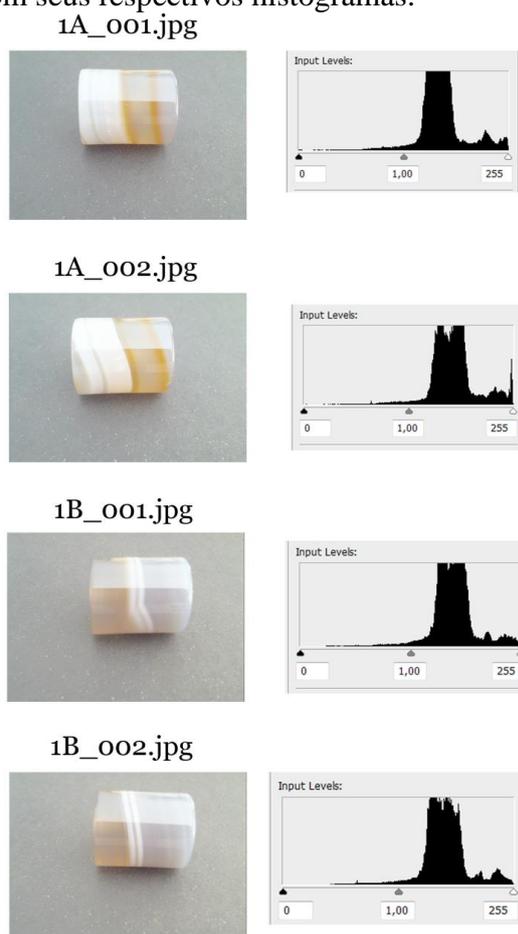
Fixada a *webcam* num tripé, a cerca de 10 centímetros de altura da superfície, percebeu-se que com esta distância a qualidade da imagem estava adequada para realizar a

análise das imagens, pois assim a *webcam* conseguiu focar de maneira correta o objeto e se a *webcam* fosse posicionada a uma distância maior, o ruído aumentaria e se perderiam detalhes importantes da imagem dos objetos. Após posicionar a *webcam*, foi colocado um papel preto na superfície cobrindo todo ângulo de visão da câmera, com o intuito de eliminar outras cores de superfícies, para ter somente o objeto na cena da imagem, tendo seu fundo totalmente de uma cor. A luminosidade foi feita através da luz natural do sol obtida perto de uma janela, porém isto prejudicou em muito os testes, sendo de extrema importância construir uma caixa de luzes do tipo *Light-Emitting Diode* (LED), forrada com folhas e iluminada com lâmpadas fixadas em seu teto.

Assim, foram capturadas imagens de quatro pedras de matriz como demonstra a Figura 16, sendo elas de duas categorias (1A e 1B), estas imagens base para a comparação. Após serem capturadas as imagens das duas matrizes, foi executado o código de comparação de histogramas e assim começado à captura do vídeo, uma pedra foi colocada em cima do papel preto, foi realizada a comparação de seu histograma com os histogramas das matrizes capturadas. Esta pedra posicionada no papel estava entre uma das duas imagens cadastradas como matrizes e pertencia a categoria 1A sendo sua matriz a pedra 002. Mesmo tendo somente uma imagem de um lado da pedra pode-se verificar conforme a Tabela 2 a câmera detectou que o maior índice de porcentagem de semelhança entre os histogramas é o da 1A_002, tornando este teste um sucesso.

Com este teste realizado foi possível prosseguir para a implementação do projeto, priorizando a utilização deste método para comparar as imagens e classificá-las com as categorias cadastradas num banco de dados.

Figura 16 – Pedras naturais com seus respectivos histogramas.



Fonte: Elaborado pelo autor.

Tabela 2 – Índices de porcentagem de semelhança das imagens

Imagem	Índice de semelhança (%)
1A_001.jpg	0.06281173309727274
1A_002.jpg	0.06594992767814256
1B_001.jpg	0.06312368208477892
1B_002.jpg	0.06312549523454436

Fonte: Elaborado pelo autor.

Através deste último teste realizado percebeu-se que todos os aspectos na etapa de aquisição de imagem citados por Marque Filho e Vieira Neto (1999), são de extrema importância para a obtenção de uma imagem de boa qualidade. Então é fundamental ter muito cuidado na hora de capturar as imagens, pois se algo sair errado ou não atingir o esperado, todos os processos seguintes irão ter insucesso em sua realização.

4 IMPLEMENTAÇÃO

Baseado em todos os testes realizados, a implementação do projeto tem como fonte principal a visão computacional, detecção de objetos em tempo real em um vídeo, a análise da imagem, processamento de imagens e a comparação do objeto com imagens pré-cadastradas. Neste capítulo serão abordados todos os detalhes da implementação deste projeto.

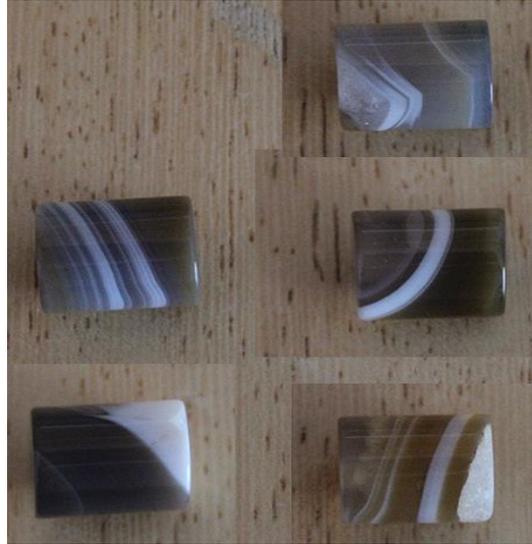
4.1 Ramo de Negócio

Empresa do ramo de pedras naturais localizada na cidade de Teutônia/RS que extrai e transforma pedras encontradas em pedreiras em pedras naturais como, por exemplo, as da Figura 17, com fins comerciais, para venda em diversas áreas como: bijouterias, joias, decoração, esculturas, entre outras.

A empresa trabalha especificamente com as áreas de bijouterias e joias, vendendo seus produtos para consumidor final e também para outras empresas que utilizam as pedras para aplicar em joias folheadas.

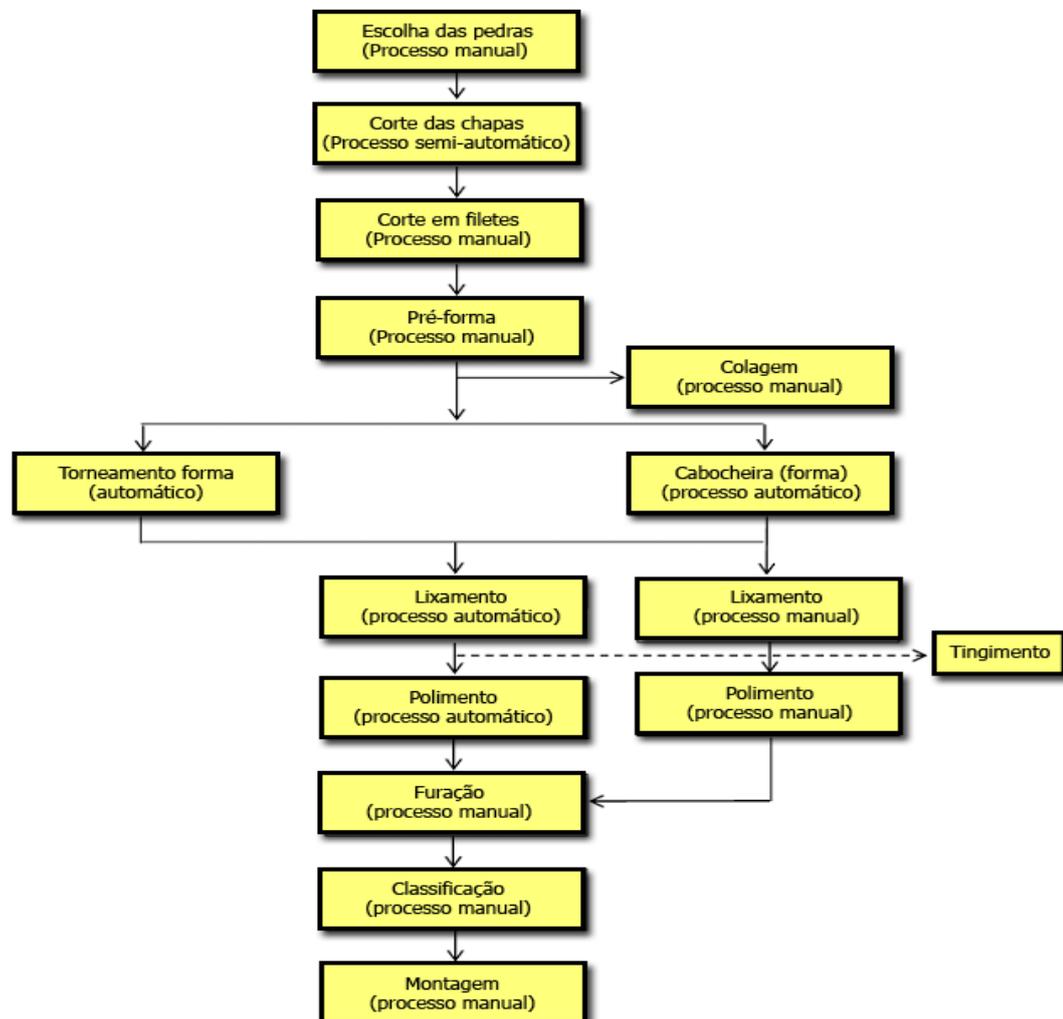
O processo de produção das pedras naturais é composto por diversas etapas, iniciando no corte e finalizando na montagem, conforme pode-se observar na Figura 18.

Figura 17 – Pedras naturais prontas para venda.



Fonte: Elaborado pelo autor.

Figura 18 – Fluxograma do processo de produção de pedras naturais.



Fonte: Elaborado pelo autor.

O processo de produção inicia com a extração de pedaços de pedras em pedreiras de fornecedores. Quando recebida na empresa é realizada a seleção das melhores pedras, de acordo com o produto final que se deseja obter. Esta seleção é feita no depósito de pedras da empresa, onde se encontram diversos tipos de pedras preciosas, tais como: ágatas, quartzos, jaspes, howlitas, dolomitas, sodalitas, olho de tigre, entre outras.

Em seguida é realizado o corte da pedra em forma de chapas, em máquinas denominadas de serra caixão, de acordo com a espessura que se deseja obter, além da remoção das partes da pedra (casca) que não são utilizadas. Este processo usa serras semiautomáticas com controle eletrônico de avanço do corte.

Na etapa seguinte, as chapas são cortadas em filetes e após picadas no tamanho da pré-forma desejada, e estas encaminhadas para máquinas que darão a forma final dos produtos. Dependendo do item, a etapa seguinte envolve tornos pneumáticos manuais ou tornos automáticos ou então máquinas automáticas denominadas cabocheiras onde são realizadas a lapidação das peças de acordo com o tamanho e o formato desejado. Após o processo de lapidação, as peças são encaminhadas para um processo de lixamento e polimento. Em alguns casos, antes do processo de lixamento e polimento, as pedras ainda podem passar por um processo de tingimento.

Por fim, elas são classificadas, separadas e encaminhadas ou para a expedição ou para o setor de montagem, onde se agregam outros acessórios para a produção de colares, brincos, pulseiras e anéis. O prazo médio necessário para a produção das pedras naturais gira em torno de 30 a 60 dias.

4.2 Delimitação do problema

A empresa necessita de um sistema para automatizar seu ambiente de trabalho, reduzindo o tempo gasto com a classificação de seus produtos, neste caso, pedras naturais. Atualmente a classificação é realizada de forma manual através da percepção da pessoa, tendo um nível elevado de dificuldade para separá-las em categorias.

Este projeto visa implementar uma solução que atenda de forma eficaz a necessidade da empresa, a partir da construção de um software com controle feito por vídeo em tempo real utilizando técnicas de processamento de imagens e visão computacional, com a detecção dos produtos que passam por uma esteira rolante em frente a câmera, e então são comparados seus parâmetros com os já cadastrados em um banco de dados para realizar a classificação dos mesmos.

O requisito de parâmetro para a análise de classificação é o seguinte: categoria do produto de acordo com sua cor e quantidade de listras, comparando o nível de semelhança do produto a ser classificado com as imagens dos produtos cadastrados como matrizes das categorias.

Com a realização deste projeto, será possível reduzir o trabalho manual dos colaboradores na classificação de produtos desta empresa, cortando gastos e eliminando erros, nesta etapa de produção.

4.3 Requisitos do sistema

Através de uma reunião realizada com a administração da empresa foram definidos requisitos para o sistema. Eles são divididos em requisitos funcionais e não funcionais. Os funcionais são cadastros do sistema e os não funcionais são a linguagem de programação, banco de dados do sistema e equipamentos do protótipo.

4.3.1 Requisitos funcionais

Os requisitos funcionais do sistema são os seguintes:

- Manter cadastro de usuários do sistema;
- Manter cadastro de produtos do sistema;
- Manter cadastro de fotos dos produtos do sistema: imagens cadastradas como base para a classificação;
- Manter cadastro de categorias dos produtos do sistema;

- Manter cadastro de ordens de classificação do sistema: são utilizadas para iniciar a classificação, possuem dados como a data, total de produtos classificados e total de produtos não classificados;
- Captura de imagens através de vídeo para realizar a classificação dos produtos;
- Monitoramento da classificação: exibir alertas para o usuário dos possíveis erros durante a classificação;
- Manter configurações do sistema: guardar informações de leitura e calibragem da captura de imagens dos produtos e da classificação.

4.3.2 Requisitos não funcionais

Os requisitos não funcionais do sistema são os seguintes:

- Linguagem de programação Java;
- Banco de dados PostgreSQL;
- Caixa de luzes LED: caixa para captura das imagens com iluminação de luzes do tipo LED branca, tendo a quantidade necessária para não ter ruído e sombra na imagem;
- Dispositivo de captura de imagens: *webcam* de alta definição, captura de imagem em *Full High-Definition* (Full HD) com a resolução mínima de 640 x 480 pixels.

A implementação da esteira rolante e da separação mecanizada não fazem parte do escopo deste projeto, pois demandariam maior tempo e conhecimento para construção.

4.4 Tecnologias envolvidas

Para realizar a implementação do sistema foi utilizada a linguagem de programação Java juntamente com a interface JavaCV que utiliza métodos convertidos em Java da biblioteca OpenCV em C. O banco de dados utilizado para armazenamento das informações foi o PostgreSQL 9.2.

4.5 Arquitetura da solução

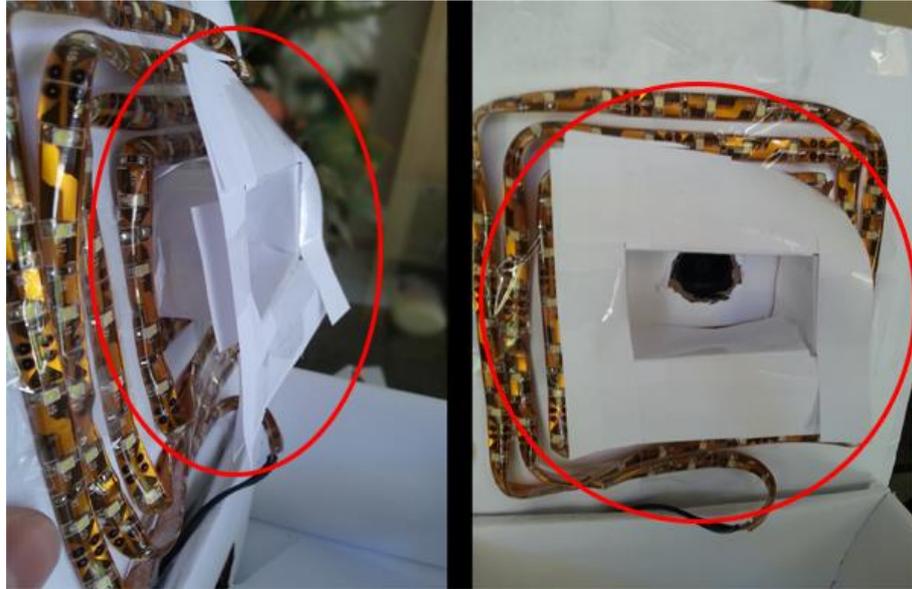
O foco do projeto é a classificação de pedras naturais, fazendo uso de técnicas de reconhecimento de imagens utilizando uma biblioteca de visão computacional. Através de um *notebook*, o mesmo utilizado na experimentação, uma câmera de vídeo (*webcam* Logitech C525) que possui foco automático e uma caixa de luzes LED, como pode se observar na Figura 19, forrada internamente com papel branco. As luzes LED foram posicionadas na parte superior da caixa, em volta do buraco onde foi fixada a *webcam*. Para diminuir os reflexos gerados pelas luzes nas pedras foi construído um duto de papel ao redor da câmera, no tamanho de 2,5 cm como visto na Figura 20.

Figura 19 – Caixa de luzes



Fonte: Elaborado pelo autor.

Figura 20 – Duto de papel para diminuir o reflexo das luzes.

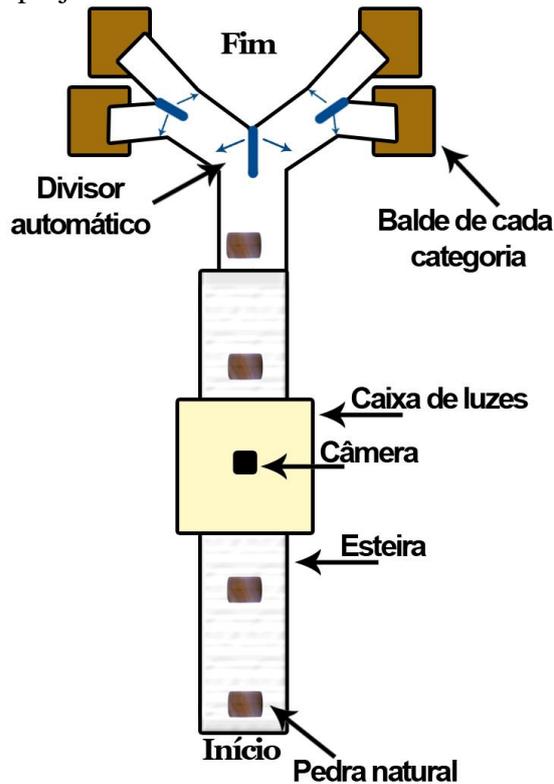


Fonte: Elaborado pelo autor.

A partir disso foi possível criar uma solução de ambiente físico básico necessário para o desenvolvimento do projeto. Porém para obter uma arquitetura ideal para o projeto seria necessário a construção do restante da estrutura física, como demonstra a Figura 21, onde o processo de classificação inicia por uma esteira que recebe as pedras naturais, passando uma pedra por vez pela caixa de luzes LED onde está instalada a *webcam*, assim realizando a captura do vídeo e classificando a pedra em uma categoria. Após a classificação a pedra continua andando pela esteira e então cai em uma canaleta onde existem divisores que são acionados automaticamente através de uma comunicação realizada com o software, despejando a pedra em um balde específico. Caso a pedra não consiga ser classificada ou não atinja o valor mínimo de semelhança pré-determinado nas configurações ela então é despejada em um balde de pedras para futura separação manual.

Esta estrutura ideal descrita não foi contemplada nesse projeto, assim como a comunicação do software com os divisores automáticos.

Figura 21 - Arquitetura ideal do projeto.



Fonte: Elaborado pelo autor.

Para realizar a classificação das pedras naturais foram utilizados dois métodos principais testados na experimentação, *blobDetection* e *cvHistogram* da biblioteca OpenCV. O método *blobDetection* foi utilizado para detectar quando a pedra está presente por completa na câmera (*webcam*), dando sinal verde para o próximo passo. Para que o método funcione perfeitamente, a imagem passou por um pré-processamento, onde foram utilizados os métodos de limiarização, erosão, dilatação e o algoritmo de *Canny*, para detectar os contornos da pedra, para então poder aplicar o método *Blob Detection*.

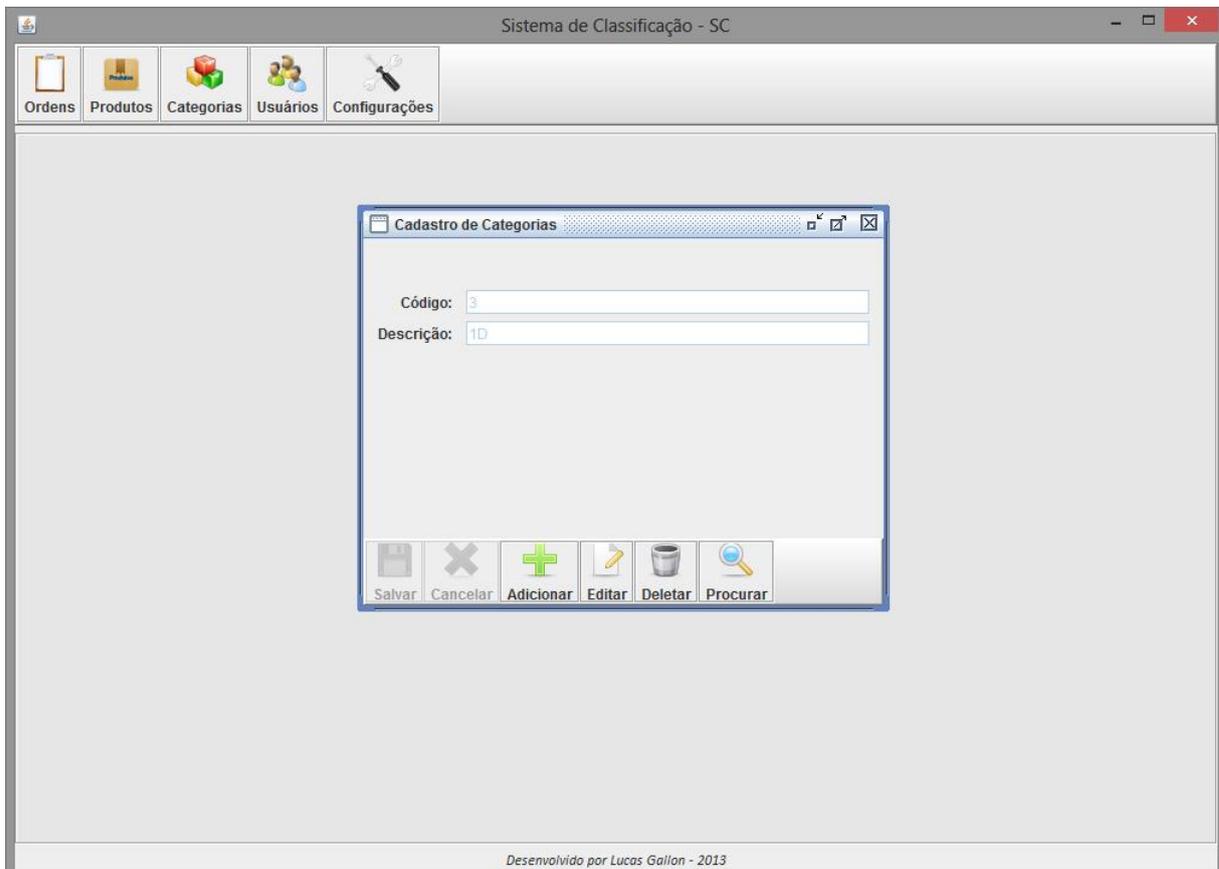
No cadastro de produtos, estes que estão vinculados a uma categoria, a imagem passa por outro pré-processamento, utilizando o método *cvGrabCut* do OpenCV, que remove o fundo da imagem desnecessário para a comparação. Passando assim a ter somente a imagem com uma pequena margem do fundo, para não desperdiçar espaço de armazenamento e também aumentar o índice de acerto nas futuras comparações. Após este pré-processamento a imagem é armazenada no dispositivo físico do computador. Estas imagens salvas no cadastro de produtos servem como matriz de comparação, para que possam ser realizadas as classificações.

Para realizar a classificação da pedra natural a imagem capturada pela câmera é comparada através do método *cvHistogram* com o acervo de imagens de pedras salvas no computador, gerando assim uma porcentagem de semelhança entre cada imagem. Por fim a pedra é classificada com o produto que possui o maior índice de semelhança, tendo assim sua categoria definida.

4.6 Telas do sistema e modelo do banco de dados

A partir dos requisitos definidos anteriormente o sistema e sua interface gráfica foram desenvolvidos, a Figura 22 demonstra a tela do cadastro de categorias do sistema, com um campo para o código e descrição da categoria.

Figura 22 – Tela de cadastro de categorias



Fonte: Elaborado pelo autor.

A Figura 23 demonstra a tela de cadastro de produtos, onde são inseridas as informações do produto, e capturadas as imagens dos produtos, através da *webcam*, armazenando o caminho da imagem salva no computador no formato de *jpeg*, no tamanho máximo de 10 *Kilobyte* (KB). O tamanho da imagem varia de acordo com as dimensões da pedra. Cada produto deve conter no mínimo uma imagem, porém, quanto mais imagens o produto tiver, mais preciso será a classificação, isso porque o sistema compara a imagem da pedra capturada com todas cadastradas.

Figura 23 – Tela de cadastro de produtos

SVCP

Ordens Produtos Categorias Usuários Configurações

Cadastro de Produtos

Código: PEDRA01

Descrição:

Categoria: 10

Situação: Ativo

caminho
C:/SistemaVisao/Imagens/PEDRA01_1.jpg
C:/SistemaVisao/Imagens/PEDRA01_2.jpg
C:/SistemaVisao/Imagens/PEDRA01_3.jpg

Capturar Imagens Deletar Imagem

Salvar Cancelar Adicionar Editar Deletar Procurar

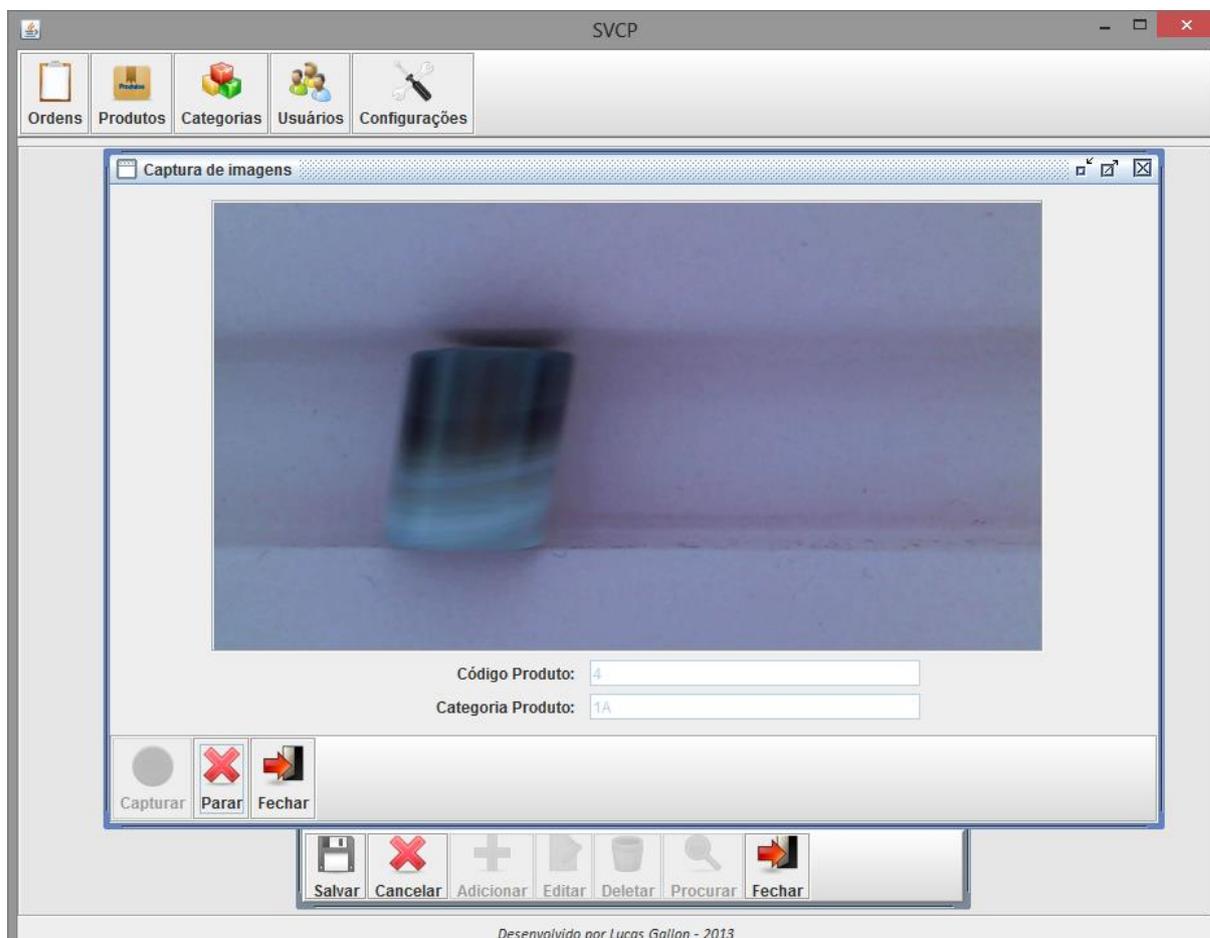
Desenvolvido por Lucas Gallon - 2013

Fonte: Elaborado pelo autor.

A Figura 24 demonstra a tela de cadastro das imagens do produto, onde as imagens são capturadas e salvas. Antes de salvar a imagem o sistema faz a detecção do produto, removendo todo fundo desnecessário, salvando somente o produto em si, contendo somente

uma margem pequena do fundo ao redor do produto para não ter o problema de cortar a sua imagem.

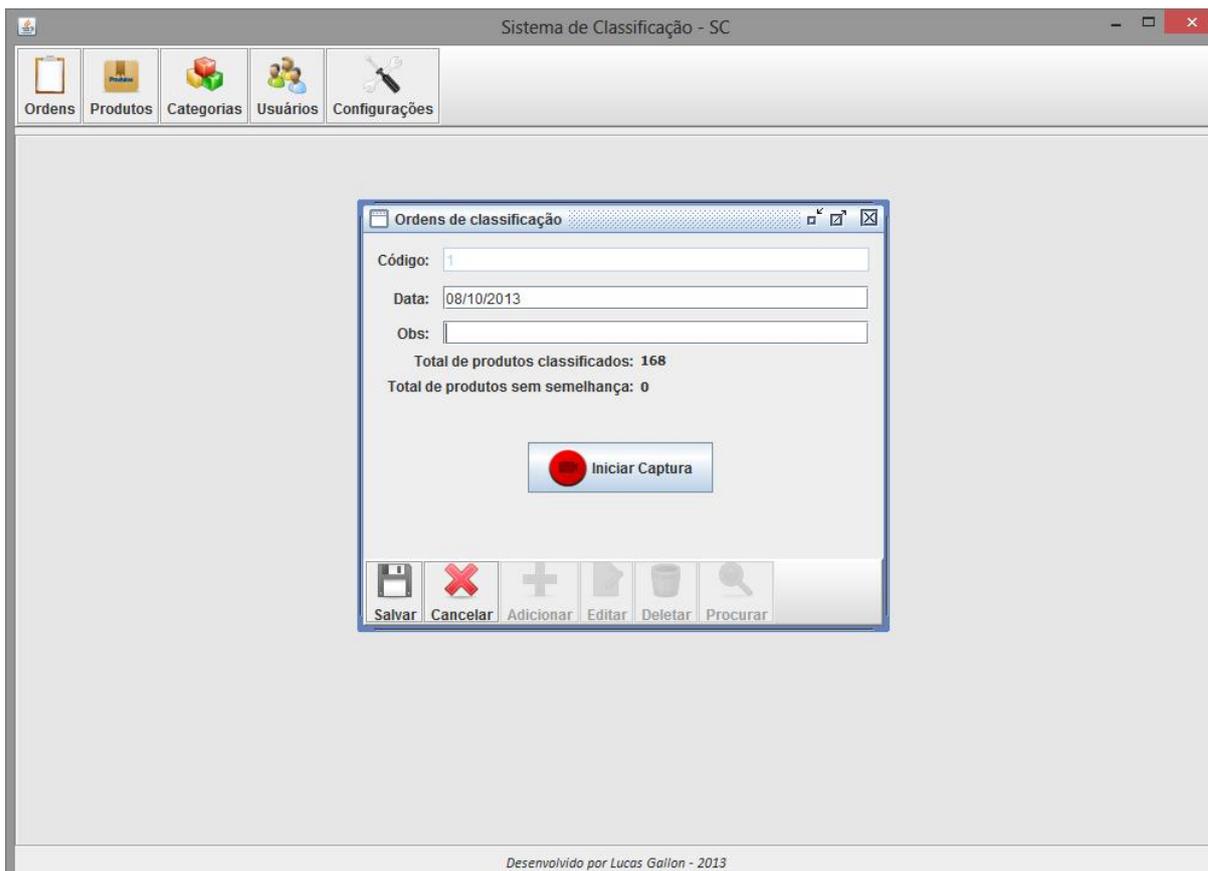
Figura 24 – Tela de captura de imagens do produto



Fonte: Elaborado pelo autor.

A Figura 25 demonstra a tela de cadastro de ordens de classificação, onde a cada início de captura de imagens para classificação é necessário criar uma ordem, com sua data e opcionalmente pode se adicionar alguma observação. A partir desta ordem pode-se iniciar a captura de imagens clicando no botão “Iniciar Captura”, também tendo a possibilidade de visualizar a quantidade total de produtos classificados e produtos sem semelhança nesta ordem de classificação.

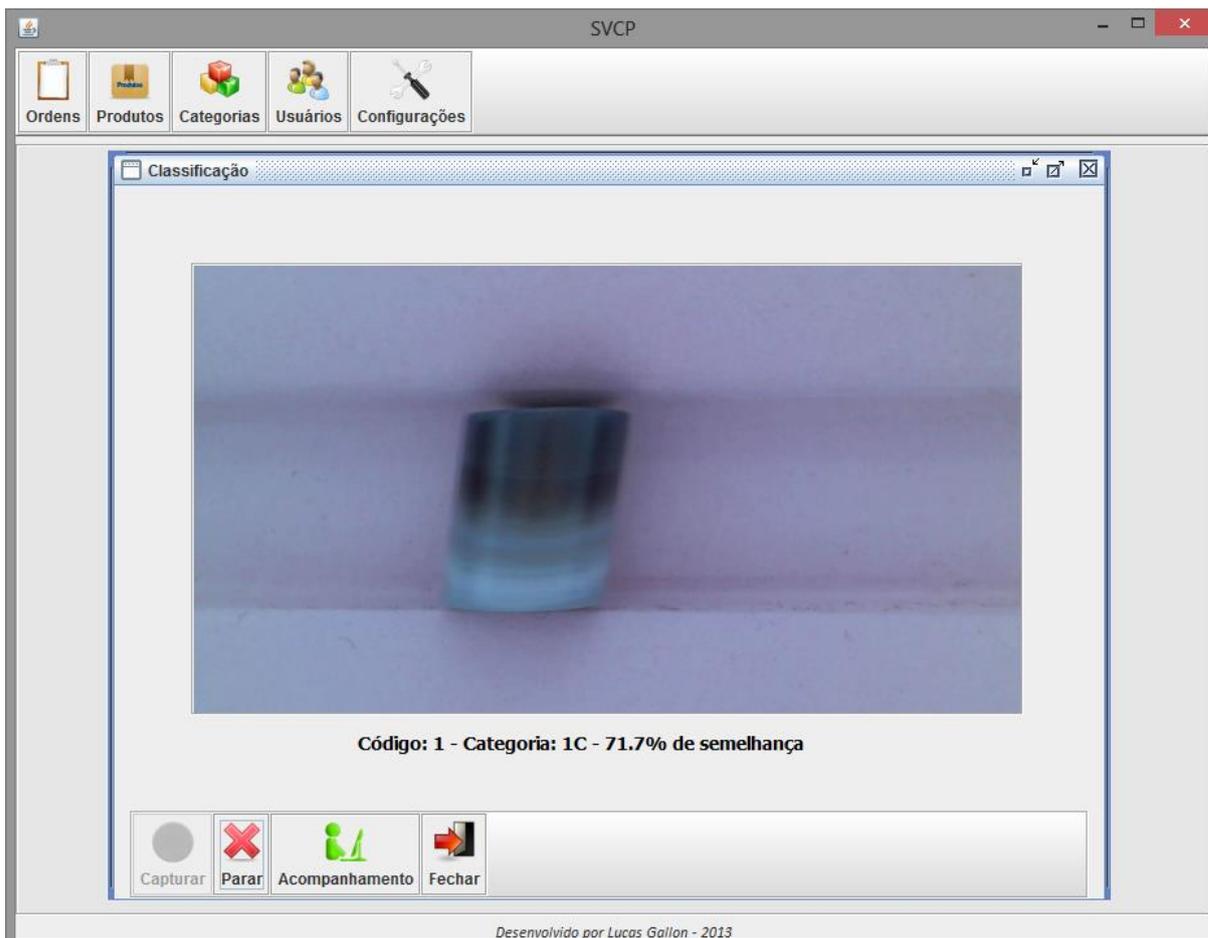
Figura 25 – Tela de cadastro de ordens de classificação



Fonte: Elaborado pelo autor.

Pode-se ver a tela de captura das imagens na Figura 26, onde a imagem da *webcam* é mostrada e quando a pedra aparecer por completa na tela, é informado seu código e sua categoria. Caso aconteça algum erro como o não reconhecimento do produto, é exibida uma tela informando o erro para o usuário.

Figura 26 – Tela de captura de imagens



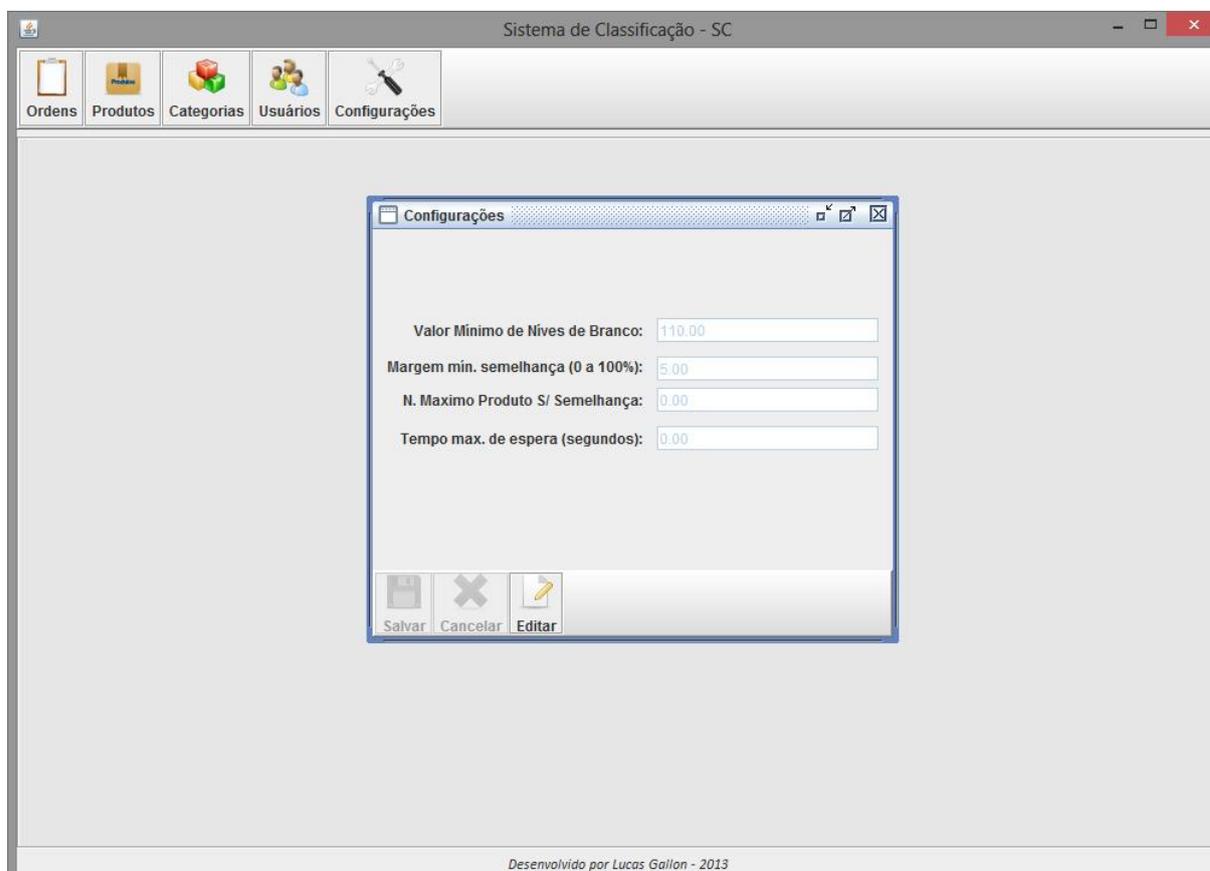
Fonte: Elaborado pelo autor.

A Figura 27 demonstra a tela de configurações, onde são inseridos valores como:

- **Valor mínimo de níveis de branco:** Valor utilizado pelo método de limiarização da imagem para utilização do *Blob Detection*, por padrão o valor definido foi 48;
- **Margem mínima de semelhança:** Valor em percentual do índice mínimo de semelhança na classificação, caso o produto não atinja este valor, exibe uma mensagem informando que o produto não conseguiu ser identificado. Por padrão foi definido o valor de 70%;
- **Número máximo de produtos sem semelhança:** Quantidade máxima de produtos que podem passar na tela e não conseguirem ser classificados. Exibe um alerta ao usuário e pausa a classificação;

- **Tempo máximo de espera:** Aguarda pelo tempo determinado a aparição de novas pedras na tela, caso não apareça nenhuma, exibe um alerta ao usuário e pausa a classificação.

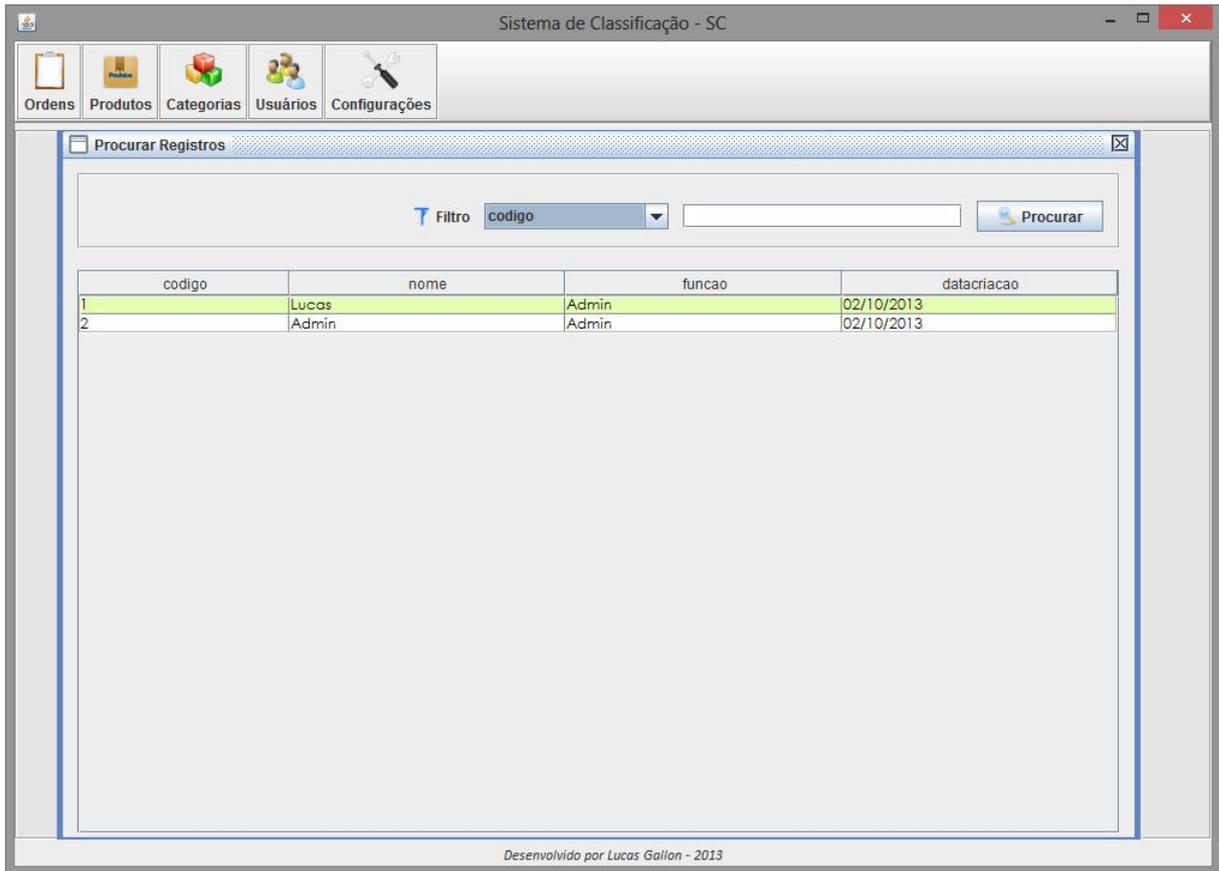
Figura 27- Tela de configurações do sistema.



Fonte: Elaborado pelo autor.

A tela de cadastro de usuários segue a mesma linha das outras, somente mudando a quantidade de campos do cadastro. Todos os cadastros terão uma tela de busca de registros conforme a Figura 28, selecionando um filtro e procurando o registro desejado.

Figura 28 – Tela de busca de registros.

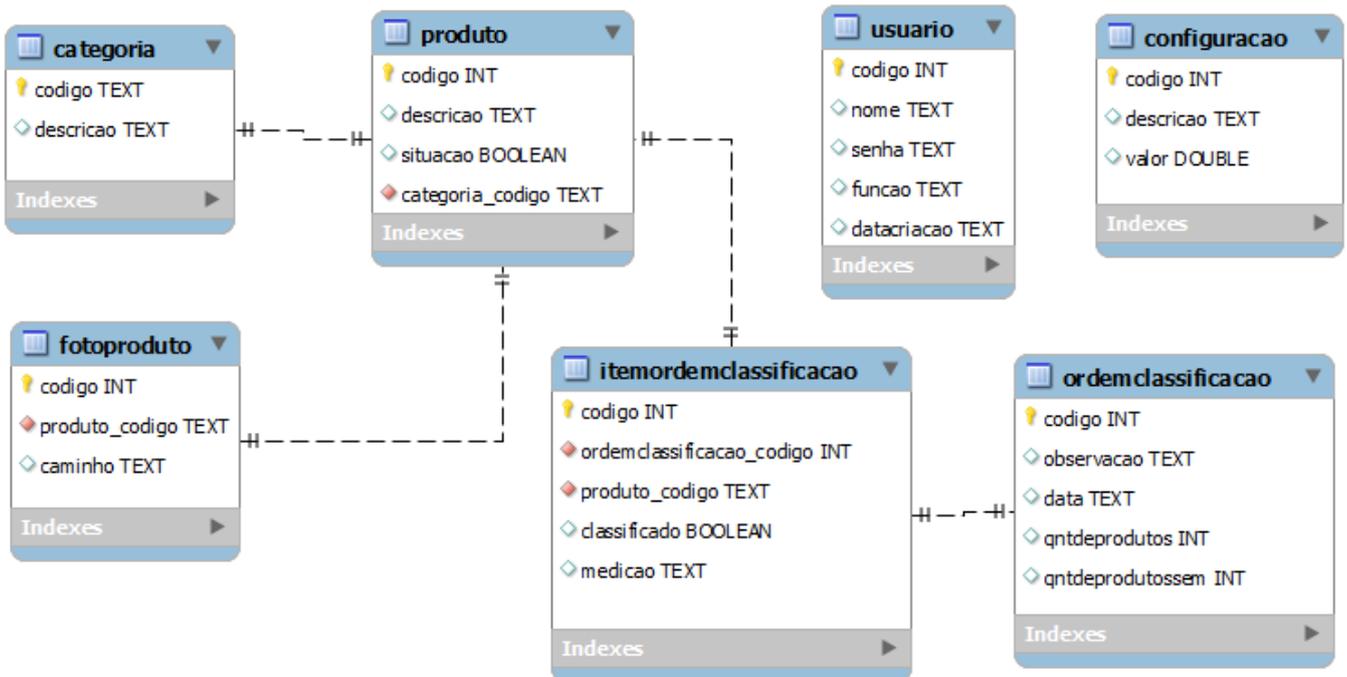


Fonte: Elaborado pelo autor.

Na Figura 29 consta Modelo ER (Entidade Relacionamento) do banco de dados implementado para o sistema. A tabela *produto* contém os dados dos produtos como código, descrição e situação (ativo ou inativo). Ela está vinculada a tabela *categoria* que contém as categorias dos produtos, e a tabela *fotoproduto* onde estão armazenados os caminhos (endereço de onde a foto está armazenada no computador). As tabelas *ordemclassificacao* e *itemordemclassificacao* armazenam as ordens de classificações e os itens classificados respectivamente, juntamente com a data da ordem, a quantidade de produtos classificados, a quantidade de produtos sem semelhança, e cada item guarda a informação se foi classificado ou não e qual a margem de semelhança com a categoria classificada. A tabela *usuário* armazena os usuários que podem acessar o sistema contendo o nome, função e a data de criação do usuário.

Por último a tabela *configuracao* armazena definições do sistema como margem mínima de semelhança, número máximo de peças sem semelhança que passam pela câmera, e tempo máximo de espera por nova classificação, entre outras.

Figura 29 – Modelo ER do banco de dados.



Fonte: Elaborado pelo autor.

5 VALIDAÇÃO

Neste capítulo serão abordados os detalhes da validação do sistema realizada na empresa.

As validações do sistema ocorreram no mês de outubro de 2013 na empresa de pedras naturais em Teutônia/RS, sob a observação de duas pessoas com conhecimento técnico e proprietários da empresa. Os equipamentos utilizados foram um *notebook* (o mesmo da experimentação) e a caixa de luzes.

Para realizar a primeira validação foi selecionada uma amostragem de 32 pedras naturais, prontas para a venda final, somente faltando realizar a classificação de suas categorias.

A partir das 32 pedras selecionadas, foram definidas 4 categorias distintas, sendo compostas de 2 pedras como matriz para cada categoria, cadastrando as 4 categorias no sistema sendo elas: 1A, 1B, 1C e 1D, e cada uma sendo cadastrado dois produtos (pedras), possuindo os códigos de acordo com a Tabela 3.

Tabela 3 – Produtos cadastrados para validação do sistema.

Código Categoria	Código Produto
1A	001
1A	002
1B	003
1B	004
1C	005
1C	006
1D	007
1D	008

Fonte: Elaborado pelo autor.

Após ter cadastrado os produtos, foi definida nas configurações o valor de 70% como margem mínima de semelhança, a seguir foi criada uma nova ordem de classificação e iniciada a captura da câmera. Foram colocados manualmente dentro da caixa de luzes os 24 produtos restantes, um por vez, realizando a classificação de cada um, conforme pode-se observar na Tabela 4, onde o índice médio de semelhança foi de 67,83%. Somente 4 produtos não foram classificados por ficarem abaixo do nível mínimo de semelhança, pois estes 4 produtos deveriam se encaixar a uma nova categoria, estavam muito distintos das categorias cadastradas.

Tabela 4 – Produtos classificados da validação do sistema.

Código do produto	Código da categoria	Índice de semelhança	Classificado
005	1C	75%	S
003	1B	74%	S
008	1D	71%	S
002	1A	76%	S
-	Não classificado	43%	N
006	1C	76%	S
002	1A	73%	S
004	1B	74%	S
-	Não classificado	35%	N
003	1B	78%	S
008	1D	77%	S
001	1A	75%	S
004	1B	73%	S
003	1B	72%	S
-	Não classificado	28%	N
006	1C	74%	S
007	1D	77%	S
001	1A	76%	S
-	Não classificado	37%	S
005	1C	72%	S
007	1D	75%	S
003	1B	71%	S
003	1B	73%	S
004	1B	74%	S

Fonte: Elaborado pelo autor.

A validação foi realizada com sucesso, e os sócios deram as seguintes sugestões de melhorias para o sistema:

1. Criar um modo de acompanhamento nas ordens de classificações de produtos, para cadastrar eventuais produtos que não se enquadram em nenhuma categoria e precisam ser vinculados manualmente;
2. Criar um alerta a partir da quantidade de peças sem semelhanças capturadas em uma ordem de classificação;
3. Criar um alerta quando a tela ficar vazia por um intervalo de tempo;

4. Criar a comunicação com a saída serial do computador, para transmitir informações ao microcontrolador que controlará os divisores automáticos;

5. No cadastro e na classificação dos produtos capturar três imagens do produto em diferentes posições em relação a câmera, agrupando o produto nessas imagens. Aumentando assim a possibilidade de acerto da classificação.

As sugestões foram parcialmente realizadas, faltando somente a sugestão 4, ficando para trabalho futuro, pois necessita de uma demanda de tempo maior e a realização de testes com microcontroladores.

Para realizar a captura de três imagens do produto em diferentes posições, foi necessário a construção de uma canaleta feita de tubo de policloreto de vinila (PVC) como pode-se observar na Figura 30, onde o tubo foi cortado ao meio e colado um filete de madeira para que as pedras deslizem de forma regular para que a câmera possa capturar as imagens necessárias. Foi colada também uma folha de papel cartolina rosa para que o fundo tenha somente uma cor e não que a detecção não se atrapalhe com as cores brancas das pedras. No cadastro de produtos são capturadas as três imagens enquanto a pedra desliza pela canaleta, estas imagens são vinculadas ao produto para a posterior classificação.

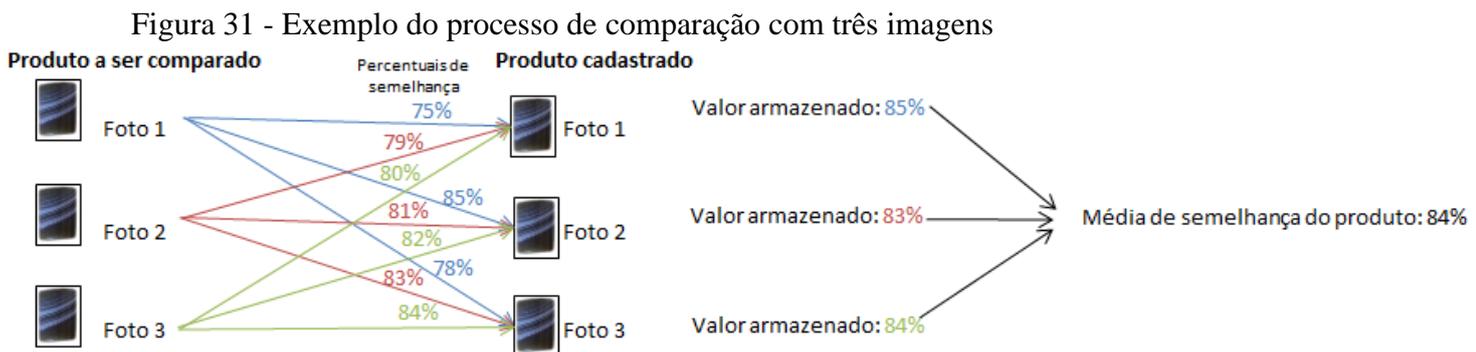
Figura 30 - Canaleta de PVC.



Fonte: Elaborado pelo autor.

Para que as três imagens fossem capturadas de diferentes posições foi medida e definida um valor padrão da distância que a pedra precisava para mudar de lado, sendo que a largura da imagem é de 640 pixels, o valor definido foi de 25 pixels entre cada captura de imagem.

A classificação foi modificada para se adaptar as três imagens, quando a ordem de classificação é iniciada, o produto que passa pela câmera tem as três imagens capturadas, executando o método de comparação de histograma cada uma das imagens é comparada com as três de cada produto cadastrado, recebendo um percentual de semelhança para cada imagem do produto e armazenando o maior valor entre os três, realizando a média dos três valores e vinculando ao produto correspondente. Para melhor exemplificar, a Figura 31 demonstra a comparação de um produto que está sendo feita a leitura, com um produto cadastrado, sendo 84% sua média de semelhança. Este processo é realizado com todos os produtos cadastrados, ao concluir todos os processos é escolhido o produto que tiver a maior média de semelhança.



Fonte: Elaborado pelo autor.

A Figura 26 demonstra a tela da captura de vídeo da classificação, existe um botão chamado “Acompanhamento” que ao ser clicado ativa o modo de acompanhamento, quando um produto passar pela canaleta é feita sua classificação e se tiver o seu índice inferior ao pré-determinado nas configurações o sistema exibe uma mensagem perguntando se o usuário deseja cadastrar como um novo produto, ao aceitar a tela de cadastro de produtos são abertas as 3 imagens e em seguida vinculadas a este produto.

O método *cvBlobDetection* que estava sendo utilizado para detecção dos objetos na imagem estava se demonstrando muito lento para detectar a pedra em três diferentes posições, pois a pedra desliza rápido e o algoritmo demora para processar e capturar a próxima posição do objeto. Buscou-se um método alternativo através da função *cvAbsDiff* do OpenCV onde a leitura é mais rápida e eficiente para a realização deste projeto, o algoritmo funciona através da detecção de movimento na imagem, comparando a imagem anterior com a atual, descobrindo o que se movimentou na cena, retornando a posição e tamanho do objeto que se movimentou na imagem.

Os passos para detectar o objeto com este novo método são os seguintes:

- Aplicação da função *cvAbsDiff* para descoberta de movimentação na cena;
- Limiarização da imagem resultante da função;
- Erosão e dilatação da imagem;
- Aplicação do método *Canny* para descoberta dos contornos da imagem.

A partir dos contornos detectados a imagem pode ser recortada de acordo com a posição e tamanho do objeto, removendo o fundo desnecessário para a comparação e mantendo só o objeto da cena.

Para segunda validação, depois de realizados os ajustes, foi selecionada uma amostragem de dez pedras da produção da empresa, tendo já cadastradas as imagens de 24 pedras nas quatro categorias definidas na primeira validação, para ter uma base de comparação.

Com o mesmo índice mínimo de semelhança definido na primeira validação, foi criada uma nova ordem de classificação e iniciada a captura do vídeo, as pedras foram colocadas na canaleta e desceram rolando passando em frente à câmera, assim foram capturadas as três imagens e realizadas as suas classificações como demonstra a Tabela 5.

Tabela 5 - Produtos classificados da segunda validação do sistema

Código do produto	Código da categoria	Índice de semelhança	Classificado
021	1D	85,3%	S
015	1C	84,7%	S
011	1B	83,2%	S
022	1D	85,4%	S
009	1B	82,5%	S
005	1A	81,3%	S
016	1C	79,4%	S
017	1C	80,9%	S
004	1A	81,2%	S
018	1C	82,4%	S

Fonte: Elaborado pelo autor.

A partir desta segunda validação percebeu-se um acréscimo no índice de semelhança entre as pedras, a média do índice de semelhança foi de 82,6%, pois com a captura de 3 imagens a precisão de acerto ficou maior. O tempo de processamento da classificação foi relativamente baixo nas duas validações, não demorando mais que 1 segundo para classificar o produto. Com essa segunda validação realizada e com as sugestões implantadas, os sócios ficaram satisfeitos com o desempenho do sistema. Verificou-se que é viável a sua implantação na empresa, demonstrando ser um produto potencial.

Utilizando a arquitetura ideal para o funcionamento do sistema, tendo uma amostragem maior de imagens das pedras cadastradas no sistema e realizada a comunicação com a saída serial será possível implantar o sistema para realizar a classificação automática das pedras.

6 CONSIDERAÇÕES FINAIS

A visão computacional tem tudo para ser expandida na indústria em geral, sua aplicabilidade é fundamental para a realização de processos automatizados para analisar, verificar e classificar produtos.

Contudo o intuito do uso de sistemas de visão computacional não é eliminar a operação humana em suas funções, e sim auxiliar na eficiência de processos repetitivos, tornando mais rápidos e com maior precisão, levando em conta a redução de custos e tempo.

A bibliotecas de programação OpenCV é essencial para a construção de ferramentas de visão computacional, possui um grande número de recursos para o processamento e análise das imagens e é constantemente atualizada. Na implementação do sistema esta biblioteca se demonstrou muito útil, tendo resultado bons resultados com a captura de imagens e comparação, também conseguindo capturar imagens de objetos em movimento no vídeo.

Baseado em tudo o que foi pesquisado, nos experimentos realizados, na implementação e validações do sistema, verificou-se que o projeto conseguiu atingir as expectativas com os recursos disponíveis, como a *webcam Full HD* e a caixa de luzes LED. Com a arquitetura ideal para o sistema é possível implantar o sistema dentro da empresa.

REFERÊNCIAS

ALBUQUERQUE, Márcio P.; ALBUQUERQUE, Marcelo P. **Processamento de imagens: Método e análises**. Centro Brasileiro de Pesquisas Físicas – CBPF/MCT Coordenação de Atividades Técnicas – CAT, 2000. Disponível em: <<http://www.cbpf.br/cat/download/publicacoes/pdf/ProcessamentoImagens.PDF>>. Acesso em: 11 abr. 2013.

AUDET, S. **JavaCV**. 2013. Disponível em: <<http://code.google.com/p/javacv/>>. Acesso em: 05 maio 2013.

BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. O'Reilly Media, Inc, 2008.

JÚNIOR, Iális Cavalcante de Paula. **Técnicas de processamento digital de imagens com Java**. Ercemapi, 2009. Parnaíba, Piauí. Disponível em: <<http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/minicurso/mc5.pdf>>. Acesso em: 27 maio 2013.

CANNY, John. **A computational approach to EDGE Detection**. IEEE Transactions on pattern analysis and machine intelligence, 1986.

CHANG, Fu; CHUN-JEN, Chen; CHIN-JEN, Lu. **A linear-time component labeling algorithm using contour tracing technique**. Institute of Information Science, Academia Sinica, 128 Academia Road, Section 2, 7 Nankang, Taipei 115, Taiwan. 2003. Disponível em: <http://www.iis.sinica.edu.tw/papers/fchang/1362-F.pdf>. Acesso em: 27 abr. 2013.

CODA. **Coda Automação Industrial**. Disponível em: <<http://www.codars.com.br/>>. Acesso em: 13 maio 2013.

COGNEX. **Cognex Sistemas de Visão e Sistemas de Identificação Industrial**. Disponível em: <<http://www.cognex.com/>>. Acesso em: 13 maio 2013.

DUDA, Richard O.; HART, Peter E. **Use of the hough tranformtion to detect lines and curves in pictures. Artificial Intelligence Center**, 1971. Disponível em: <<http://www.ai.sri.com/pubs/files/tn036-duda71.pdf>>. Acesso em: 13 maio 2013.

FISHER, Robert; PERKINS, Simons; WALKER, Ashley; WOLFART, Erik. **Hough transform**. Hypermedia Image Processing Reference, 2003. Disponível em: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>>. Acesso: em 11 abr. 2013.

FONSECA, Leila M. G. **Processamento digital de imagens**. INPE, 2010. Disponível em: <http://galileu.iph.ufrgs.br/gis-sr/SR/Apostilas_Sensoriamento_Remoto/Processamento_Digital_Imagens>. Acesso em: 11 abr. 2013.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. Editora Edgard Blucher Ltda, 2000.

JÄHNE, Bernd. **Digital Image Processing**. Springer-Verlag Berlin Heidelberg, 2002.

JAVACV. Site Oficial, 2013. Disponível em <<https://code.google.com/p/javacv/>>. Acesso em: 02 abr. de 2013.

MACHADO, Douglas S. **Reconhecimento de objetos de formas variadas**. Universidade Federal de Minas Gerais, 2008. Disponível em: http://www.verlab.dcc.ufmg.br/_media/cursos/visao/2008-1/grupo18/douglas_silveira_machado_-_ufmg.pdf. Acesso em: 07 maio 2013.

MARENGONI, Maurício; STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV**. UFGRS, 2010. Disponível em <http://seer.ufrgs.br/rita/article/download/rita_v16_n1_p125/7289>. Acesso em: 06 ago. 2013.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999. ISBN 85745200981999.

MILANO, Danilo; HONORATO, Luciano B. **Visão computacional**. Disponível em: <http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010_IA_FT_UNICAMP_visa_oComputacional.pdf>. Unicamp, 2010. Acesso em: 09 abr. 2013.

MOREIRA, Luis M. C. **Tamanho e resolução da imagem**. Escola Secundária Filipa de Vilhena, 2011. Disponível em: <http://paginas.fe.up.pt/~ee03037/tmp/api11/imagem_resol.pdf>. Acesso em: 07 maio 2013.

OPENCV. Site Oficial, 2013. Disponível em: <<http://opencv.org/>>. Acesso em: 02 abr. de 2013.

ROJAS, Oscar E. **Algoritmo de procura de círculos em uma imagem RGB**. Unicamp, 2012. Disponível em: <http://www.sps.fee.unicamp.br/sps2012/proceedings_sps2012/Oscar_RGB_SPS2012.pdf>. Acesso em: 27 abr. 2013.

RUDEK, Marcelo; SANTOS, COELHO, Leandro; CANGIOLIERI JÚNIOR, Osiris. **Visão computacional aplicada a sistemas produtivos: fundamentos e estudo de caso**. In: XXI

Encontro Nacional de Engenharia de Produção, Salvador. 2001. 1 p. Disponível em: <<http://www.las.pucpr.br/rudek/pdf/Enegepxxi.pdf>>. Acesso em: 11 abr. 2013.

SCURI, Rodrigo E. **Fundamentos da imagem digital.** Tecgraf – PUC, Rio de Janeiro, 2002. Disponível em: <<http://www.tecgraf.puc-rio.br/~scuri/download/fid.pdf>>. Acesso em: 07 maio 2013.

STEMMER, Marcelo R.; ORTH, Alexandre; ROLOFF, Márcio L.; DESCHAMPS, Fernando; PAVIM, Alberto X. **Apostila de sistemas de visão.** Disponível em: <<http://s2i.das.ufsc.br/harpia/downloads/apostila-sistemas-visao.pdf>>. Acesso em: 23 maio 2013.

WILSON, Gregory. **Programmer's tool chest – The opencv library.** Dr.Dobbs Journal, 2001. Disponível em: <<http://www.ddj.com/architect/184404319?pgno=1>>. Acesso em: 22 mar. 2013.