

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA ELETRÔNICA

ALESSANDRA IOLANDA PACHECO DOS SANTOS

**IDENTIFICAÇÃO DOS NÍVEIS DE DETERIORAÇÃO EM BARRAS DE AÇO
SUBMETIDAS A PROCESSO DE ACELERAÇÃO DE CORROSÃO EM CORPOS
DE PROVA DE CONCRETO**

TRABALHO DE CONCLUSÃO DE CURSO

TOLEDO
2020

ALESSANDRA IOLANDA PACHECO DOS SANTOS

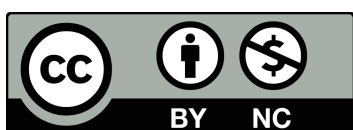
**IDENTIFICAÇÃO DOS NÍVEIS DE DETERIORAÇÃO EM BARRAS DE AÇO
SUBMETIDAS A PROCESSO DE ACELERAÇÃO DE CORROSÃO EM CORPOS
DE PROVA DE CONCRETO**

**IDENTIFICATION OF DETERIORATION LEVELS IN STEEL BARS SUBMITTED TO
ACCELERATED CORROSION PROCESS IN CONCRETE SPECIMENS**

Trabalho de Conclusão de Engenharia Eletrônica, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná – UTFPR Toledo, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Jefferson Gustavo Martins

TOLEDO
2020



4.0 Internacional

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho licenciado para fins não comerciais, com crédito atribuído ao autor. Os usuários não têm que licenciar os trabalhos derivados sob os mesmos termos estabelecidos pelo autor do trabalho original.

Alessandra Iolanda Pacheco dos Santos

**IDENTIFICAÇÃO DOS NÍVEIS DE
DETERIORAÇÃO EM BARRAS DE AÇO
SUBMETIDAS A PROCESSO DE ACELERAÇÃO DE
CORROSÃO EM CORPOS DE PROVA DE
CONCRETO**

Trabalho de Conclusão de Curso apresentado à disciplina de Trabalho de Conclusão de Curso 2 do Curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná - UTFPR Campus Toledo, como requisito parcial para a obtenção do título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Toledo, 15 de maio de 2021:

Prof. Jefferson Gustavo Martins
Orientador

Prof. Felipe Walter Dafico Pfrimer
UTFPR-TD

Prof. Vilson Luiz Dalle Mole
UTFPR-TD

Toledo
2021

A folha de aprovação com as assinaturas encontra-se na coordenação do curso

RESUMO

Este trabalho investigou a aplicação de técnicas de aprendizado de máquinas aplicadas ao problema de corrosão de barras de aço. A degradação de construções ocorre em todo o globo terrestre e afeta a segurança de milhares de pessoas. Tais degradações são ainda mais comprometedoras quando atingem as estruturas de sustentação e impõem grandes gastos para a reparação destas estruturas. O objetivo deste trabalho teve o foco de diagnosticar o nível de deterioração destas estruturas por meio de visão computacional. Foram utilizados os extratores de características GLCM, LBP, e ORB com os classificadores KNN e SVM para avaliar e alcançar as melhores taxas de classificação. Os padrões obtidos por meio dos resultados científicos foram utilizados para a identificação dos diferentes níveis de corrosão.

Palavras-chave: Aprendizado de máquina. Corrosão. Construção Civil. Segurança. Automação.

ABSTRACT

This work investigated the application of machine learning techniques applied to the problem of corrosion in concrete. The degradation of these buildings occurs all over the world and affects the safety of thousands of people, just as there are larger expenses involved to repair these structures. The objective was to diagnose the deterioration level through computer vision. Several techniques and tools, among those selected, were be used to assess and achieve the best classification rates. The standards obtained through scientific results served to identify the different levels of corrosion.

Keywords: Machine Learning. Corrosion. Construction. Safety. Automation.

LISTA DE FIGURAS

Figura 1.1 : Efeitos da corrosão nas estruturas de concreto: (a) manchas e fissuras; (b) armaduras expostas por destacamento do concreto.....	4
Figura 2.1 : Ilustração da pilha de corrosão em um mesmo metal.....	5
Figura 2.2 : Mancha de corrosão.....	5
Figura 2.3 : Sequência de passos para a análise e processamento de imagens.....	7
Figura 2.4 : Geração da GLCM para imagem descrita em tons de cinza com distância $d = 1$ e ângulo $a = 0^\circ$	9
Figura 2.5 : Cálculo LBP: Extração do pixel e vizinhança de uma imagem; decisão binária; multiplicação dos elementos da matriz binária por seus respectivos pesos e resultado LBP apresentado em valor decimal.....	10
Figura 2.6 : LBP Circular com diferentes raios e número de vizinhos.....	11
Figura 2.7 Exemplos de hiperplanos definidos pelo SVM em um caso binário: (a) Margem qualquer; (b) Margem máxima.....	15
Figura 3.1 : Base de imagens: (a) Superfície corpo de prova de concreto sem corrosão (b) Superfície corpo de concreto com corrosão (c) Superfície barra de aço sem corrosão (d) Superfície barra de aço com corrosão.....	18
Figura 3.2 : Etapas para a segmentação das imagens.....	20
Figura 3.3 : Filtro de Niblack: (a) Corpo de concreto com corrosão; (b) Barra de aço com corrosão.....	21
Figura 3.4 : Imagens binárias: (a) Corpo de concreto com corrosão; (b) Barra de aço com corrosão.....	21
Figura 3.5 : Dilatação e Erosão: (a) Corpo de concreto com corrosão; (b) Barra de aço com corrosão.....	22
Figura 3.6 : Segmentação das imagens com corrosão: (a) Corpo de concreto com fundo cinza e sombras; (b) Corpo de concreto segmentado; (c) Barra de aço com fundo cinza e sombras; (d) Barra de aço segmentado.....	22
Figura 3.7 : Ilustração para <i>3-fold cross validation</i>	25

LISTA DE ABREVIATURAS

Sigla	Significado	Tradução
AM	Machine Learning	Aprendizado de Máquina
FN	False Negative	Falso negativo
FP	False Positive	Falso positivo
GLCM	Grey Level Co-Occurrence Matrix	Matriz de co-ocorrência de níveis de cinza
KNN	K Nearest Neighbor	K vizinhos mais próximos
LBP	Local Binary Patterns	Padrão binário local
$LBP_{P,R}^{ri}$	Circular Invariant Local Binary Patterns	Padrão binário local invariante circular
$LBP_{P,R}^{riu2}$	Circular Invariant Uniform Local Binary Patterns	Padrão binário local invariante uniforme circular
$LBP_{P,R}$	Circular Local Binary Patterns	Padrão binário local circular
MC		Matriz de Confusão
SVM	Support Vector Machine	Máquina de vetores de suporte
TN	True Negative	Verdadeiro negativo
TP	True Positive	Verdadeiro positivo
TU	Texture Unit	Unidade de textura

SUMÁRIO

1 INTRODUÇÃO.....	2
1.1 Objetivos.....	3
1.1.1 Objetivos Específicos.....	3
1.2 Justificativa.....	3
2 REFERENCIAL TEÓRICO.....	5
2.1 Corrosão.....	5
2.2 Aprendizado de Máquina.....	6
2.3 Reconhecimento de Padrões.....	6
2.4 Extratores de características.....	9
2.4.1 Matriz de co-ocorrência dos Níveis de Cinza.....	9
2.4.2 Padrão Binário Local.....	10
2.4.3 ORB.....	12
2.5 Classificadores.....	14
2.5.1 K-vizinhos Mais Próximos.....	14
2.5.2 Máquina de Vetores de Suporte.....	15
3 MATERIAIS E MÉTODOS.....	17
3.1 Base de Imagens.....	17
3.2 Pré-processamento e Segmentação.....	20
3.3 Extração de Características.....	23
3.4 Classificação e Avaliação.....	24
4 RESULTADOS E DISCUSSÕES.....	27
4.1 GLCM.....	27
4.2 LBP.....	28
4.3 ORB.....	29
5 CONSIDERAÇÕES FINAIS.....	31
REFERÊNCIAS.....	32

1 INTRODUÇÃO

A degradação das barras de aço em estruturas de concreto armado, devido ao processo de corrosão, é um problema que afeta a construção civil no mundo todo. Nos Estados Unidos da América (EUA), os gastos podem chegar até 5 bilhões de dólares apenas para a realização de reparos em pontes danificadas. Países em desenvolvimento, por sua vez, chegam a perder cerca de 1,25% a 3,5% do PIB destinados à manutenção dessas estruturas. Tais custos podem ainda ser potencializados dependendo do momento que é realizada a intervenção (MEIRA, 2017).

Podem existir diversas razões para ocorrer o processo corrosivo. A falta de cobrimento adequado, o meio ambiente em que se insere a estrutura e a microrregião onde se encontra o componente estrutural são apenas alguns dos possíveis fatores para a causa do fenômeno (CASCUDO, 1997). Nas obras civis, por exemplo, ocorre a corrosão eletroquímica, processo no qual o concreto atua como eletrólito. Este é um problema grave, já que a função do cobrimento do concreto serve para a dupla proteção do aço. A camada física separa a armadura do contato direto com o meio externo, enquanto a proteção química se deve à alcalinidade do meio e à indução de uma camada passivadora no aço que serve como uma película de proteção (FIGUEIREDO; MEIRA, 2012).

A partir do exposto, esta proposta consiste de um experimento científico para identificar graus de corrosão nas barras de aço em estruturas de concreto armado. O estudo se dá com vistas à aplicação de técnicas de Aprendizado de Máquina (AM, do inglês *Machine Learning*) com foco em aprendizado supervisionado. Para isso, serão utilizadas imagens obtidas por meio de um ensaio de aceleração de corrosão em laboratório. Tal procedimento consiste na utilização de 24 barras de aço com características iguais. Assim, seis delas compõem um grupo de controle (referências sem corrosão), e outros três grupos de seis amostras são submetidos a processos de aceleração de corrosão por períodos de 24, 48 e 72 horas. Ao decorrer do processo, adquire-se imagens a partir de corpos de concreto e também diretamente das barras de aço. Desse modo, as imagens obtidas dão origem a dois experimentos: avaliação do grau de corrosão em corpos de concreto (I) e também em barras de aço (II).

1.1 Objetivos

Este trabalho tem como proposta desenvolver experimentos para avaliar o grau de corrosão nas estruturas de aço. Desta forma, uma ferramenta alternativa a partir de tais resultados poderá ser desenvolvida para auxiliar o monitoramento e fiscalização das edificações. A automação para avaliar o grau de corrosão das estruturas de aço pode propiciar vantagens quanto a precisão e rapidez, quando comparadas a seres humanos.

1.1.1 Objetivos Específicos

Para a realização da proposta do trabalho é necessário:

- Realizar estudo de ferramentas para manipulação de imagens e aprendizado de máquina, tal como OpenCV e Scikit-Learn;
- Estudar e avaliar diferentes ferramentas a serem empregadas como descritores de características a serem extraídas das imagens;
- Estudar e avaliar diferentes algoritmos de aprendizado a serem empregados no processo de classificação; e
- Avaliar o desempenho dos descritores e algoritmos de aprendizado nos experimentos para a identificação do nível de corrosão dos corpos de concreto e barras de aço.

1.2 Justificativa

O problema de corrosão eletroquímica é um dos principais problemas de degradação de estruturas nas construções civis. Este pode ser causado por erros de projeto, materiais, execução de obra, fatores ambientais e qualquer combinação destas condições (SOUZA; RIPPER, 2009). Quando há evidências da externalização do processo corrosivo, é comum a armadura já estar comprometida, visto que a manifestação é somente produto deste. A corrosão pode ser de forma generalizada, no qual a deterioração ocorre de forma extensa e uniforme, ou de forma localizada, em que a deterioração se apresenta de forma profunda e em superfície limitada. Além disso, as expressões de corrosão podem ser acompanhadas da formação de pites ou fissuras (FIGUEIREDO; MEIRA, 2013).

A deterioração da ferragem devido ao processo corrosivo pode chegar a causar ruptura estrutural. Um caso que deixou 43 mortos foi o desabamento parcial da ponte Morandi em Gênova (Itália) (G1, 2018 e 2019). O engenheiro responsável apontou a necessidade de tratamento com relação a ferrugem. Outro exemplo que teve grande repercussão nacional, foi o colapso da parcial do viaduto na cidade de Brasília em 2018 (G1, 2018). Diante disso, estudos relacionados aos processos de corrosão em estruturas são importantes no que concerne a longevidade e segurança das estruturas.

Outro ponto importante são as inspeções das estruturas, as quais são realizadas visualmente por seres humanos. Esta consiste em verificar a presença de processos corrosivos, eflorescência, desagregação e fissuras, além de outras patologias que podem estar presentes na estrutura. A Figura 1.1 mostra alguns exemplos. Equipamentos como lupa, máquina fotográfica, trena e paquímetro são possíveis ferramentas a serem utilizadas.



Figura 1.1: Efeitos da corrosão nas estruturas de concreto: (a) manchas e fissuras; (b) armaduras expostas por destacamento do concreto.

Fonte: Tecnosil (2019).

As inspeções visuais realizadas por seres humanos podem ser repetitivas, monótonas e demoradas. Assim, o profissional está sujeito a possíveis distrações e baixas taxas de acerto (CONNERS *et al.*, 1997; PHAM e ALCOCK, 1997; RADOVAN *et al.*, 2001).

2 REFERENCIAL TEÓRICO

2.1 Corrosão

A corrosão pressupõe a formação de uma pilha eletroquímica. Portanto, é composta por reações de oxidação e redução, bem como passagem de íons por meio de um eletrólito. A Figura 2.1 ilustra o processo de corrosão eletroquímica dentro do concreto e mostra as reações parciais que ocorrem nas zonas anódica e catódica (MEIRA, 2017).

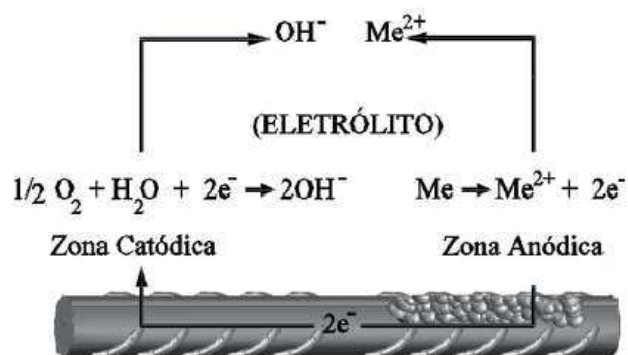


Figura 2.1: Ilustração da pilha de corrosão em um mesmo metal.
Fonte: MEIRA (2017, pg 17).

No ânodo acontece a oxidação do metal, com perda de elétrons e diminuição da seção do material. No cátodo ocorre a redução (ganho de elétrons) do oxigênio (MEIRA, 2017; ANDRADE, 2001). Os elétrons migram do ânodo para o cátodo por contato direto metal-metal e os íons produzidos pelas reações se deslocam por meio do eletrólito. A combinação desses íons origina os produtos de corrosão e tem uma coloração marrom-avermelhada (ANDRADE, 2001).



Figura 2.2: Mancha de corrosão
Fonte: Lipford (2021).

2.2 Aprendizado de Máquina

Aprendizado de Máquina (AM) é um ramo da inteligência artificial no qual algoritmos são aplicados de forma sistemática a um conjunto de dados, os quais são relacionados e provêm algum tipo de informação. Este é um campo de estudo em que os computadores têm a habilidade de aprender sem serem expressamente programados. O aprendizado pode ser observado quando o sistema se modifica em função de suas entradas para obter uma melhora na performance dos valores esperados na saída. Mitchell (1997, p.2) declara que “um sistema é capaz de aprender com experiência E, com respeito a um grupo de tarefas T e índice de performance P, se a performance medida por P em T, aumenta com E”.

Os sistemas de AM são desenvolvidos de forma a categorizar variáveis de um processo e podem se redesenhar à medida que aprendem. Sua característica computacional é generalizar a experiência obtida pelo treino estimando uma função de destino. Esses sistemas são importantes por auxiliar na resolução de problemas de alta dimensão ou de grandes volumes de dados. Muitos modelos computacionais foram desenvolvidos e se comportam de diferentes formas para a classificação dos padrões. O AM pode se dividir em dois grupos, aprendizado supervisionado e aprendizado não supervisionado, sendo que o primeiro é abordado neste trabalho e para o qual é apresentado uma breve descrição do processo na seção seguinte (LORENA; CARVALHO, 2007; AWAD; KHANNA, 2015).

2.3 Reconhecimento de Padrões

A área de reconhecimento de padrões vem evoluindo ao longo dos anos e tem desempenhado um papel importante para a investigação e resolução de inúmeros problemas. Embora possa haver variações quanto à ênfase dada a cada etapa (Figura 2.3), o processo se inicia com a aquisição de sinais que podem ser imagens estáticas, imagens dinâmicas (vídeos), áudios, medidas de sensores diversos relativas a temperatura, umidade, velocidade e pressão, dentre outras (GONZALEZ; WOODS, 2009).

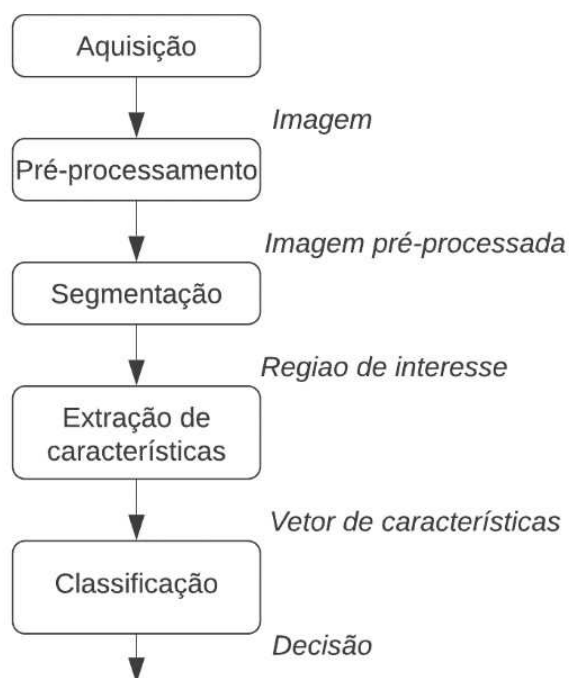


Figura 2.3: Sequência de passos para a análise e processamento de imagens.
Fonte: Autoria Própria (2021).

A aquisição pode ocorrer por meio de um único sensor ou um conjunto destes. No caso de imagens, pode-se ter sensores com movimentos mecânicos para gerar uma imagem 2-D, sensores por varredura de linha, sensores matriciais, entre outros. Na maior parte dos casos, os sensores utilizados geram uma onda de tensão contínua em relação as coordenadas. Para que a imagem possa ser representada de forma adequada para o tratamento computacional, é necessário o processo de amostragem e quantização. A digitalização dos eixos de coordenadas é chamada de *amostragem* e dos valores de amplitude é chamada de *quantização*. (GONZALEZ; WOODS, 2009).

Após a aquisição de imagem, tem-se a etapa de pré-processamento, a qual envolve passos como a filtragem de ruídos. Esta visa melhorar a qualidade da imagem, corrigir imperfeições da etapa de aquisição e realçar pontos de interesse. É comum a utilização de filtros e histogramas. No entanto, esta etapa é caracterizada por ter soluções específicas, cujo bom funcionamento depende do observador. Melhoras no brilho e contraste, redução de ruídos, realce de bordas e correção de iluminação são alguns exemplos de abordagens para solucionar um problema específico. Existem inúmeras formas de processamento de imagens, sendo difícil definir um método particular. As técnicas aplicadas a um problema de interesse podem não se adequar a outro e vice-versa (GONZALEZ; WOODS, 2009).

Em seguida, é necessário separar o(s) objeto(s) de interesse por meio de um processo de segmentação, no qual se identifica características constantes e descontinuidades (GOMES; QUEIROZ, 2006; SONKA; HLAVAC; BOYLE, 2013). A etapa de segmentação subdivide a imagem em regiões ou objetos constituintes (GONZALEZ; WOODS, 2009). No que diz respeito às imagens monocromáticas, os algoritmos de segmentação podem se dividir em dois grupos: similaridade e descontinuidade dos níveis de cinza. As técnicas de descontinuidade consistem no particionamento de regiões caracterizadas pelas transições abruptas dos níveis de cinza. Em geral o foco pode estar na detecção de pontos isolados, linhas ou bordas. Alternativamente, as técnicas por similaridade se baseiam na disposição dos pixels pelas regiões, seja por limiarização ou agrupamento dos pixels por segmento (GOMES; QUEIROZ, 2006).

A partir disso, é possível realizar a extração da informação (ou características) da imagem para análise e classificação. Dois tipos de recursos podem ser extraídos de um conteúdo de imagem: recursos globais ou locais (AWAD; HASSABALAH; 2016). Os recursos globais tratam a imagem como um todo e pode ser interpretada como uma propriedade particular da imagem (que envolve todos os pixels). Por outro lado, os recursos locais identificam pontos-chave na imagem e descrevem as regiões em torno desses pontos. Detalhes dos descritores utilizados são apresentados na Seção 2.4.

Para terminar o processo, tem-se a etapa de classificação. No caso do aprendizado supervisionado, o qual será abordado neste trabalho, há um rótulo y_i que está associado a cada instância x_i e representa a classe à qual x_i pertence. Assim, é possível construir um modelo que leva em consideração o rótulo y_i (também chamado de sinal supervisionado) para a separação das instâncias pertencentes a cada uma das classes. Neste processo, busca-se pela combinação de características com capacidade de diferenciar os elementos pertencentes às diferentes classes inerentes a um problema qualquer e por algoritmos de aprendizado capazes de construir os modelos que consigam mapear os limites entre as classes (LORENA; CARVALHO, 2007; AWAD; KHANNA, 2015).

2.4 Extratores de características

2.4.1 Matriz de co-ocorrência dos Níveis de Cinza

A matriz de co-ocorrência dos níveis de cinza (*Gray Level Co-occurrence Matrix* - GLCM) foi proposta por Haralick *et al.* em 1973. Esta abordagem visa a aquisição de informações sobre textura pela transição dos níveis de cinza entre dois pixels a uma distância d e um ângulo α (HARALICK, 1973). Esta matriz depende apenas da quantidade de níveis de cinza N_g que cada pixel pode assumir e descreve a frequência em que os pares de pixels ocorrem devido à disposição espacial dos pixels na imagem.

Os ângulos propostos inicialmente eram 0° , 45° , 90° e 135° , mas podem assumir qualquer valor. Por sua vez, a distância determinada deverá estar dentro dos limites da imagem (HARALICK, 1973). Seguindo as definições anteriores, a Figura 2.4 ilustra uma GLCM gerada pela ferramenta *Scikit-learn* com $N_g = 4$, distância $d = 1$ e ângulo $\alpha = 0^\circ$. A co-ocorrência dos pares destacados na imagem está ilustrada na matriz menor denominada GLCM.

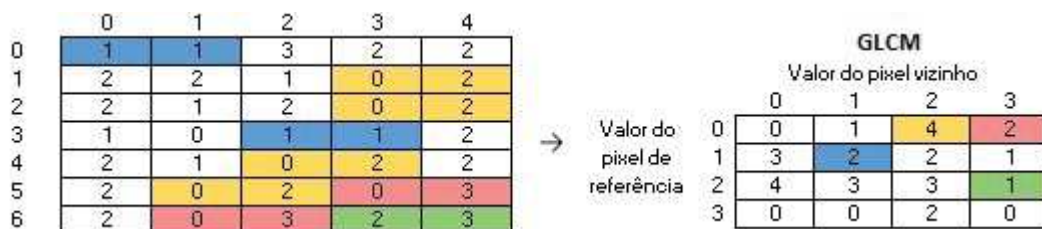


Figura 2.4: Geração da GLCM para imagem descrita em tons de cinza com distância $d = 1$ e ângulo $\alpha = 0^\circ$.

Fonte: Autoria Própria (2021).

Algumas medidas podem ser obtidas a partir da GLCM. Cada elemento da matriz é dividido pela soma total dos seus elementos. Isso permite a utilização de teorias propostas em probabilidade e estatística para o cálculo das características desejadas. Haralick sugere várias medidas em seu trabalho, embora outros autores considerem que existem redundâncias e utilizam apenas um subconjunto delas: energia, contraste, entropia, homogeneidade, probabilidade máxima e momento de terceira ordem. Maiores detalhes podem ser obtidos em Martins (2014).

2.4.2 Padrão Binário Local

O descritor Padrão Binário Local (*Local Binary Pattern* - LBP) foi introduzido por Ojala, Pietikainen e Harwood, em 1996. Para encontrar as texturas que representem a imagem, é necessário realizar a limiarização de uma vizinhança com relação ao pixel central (OJALA; PIETIKAINEN; HARWOOD, 1995). O processo pode acontecer utilizando uma janela retangular ou circular (denominado $LBP_{P,R}$) disposta sobre um pixel central como referência (MAENPAA; PIETIKAINEN, 2004).

A primeira versão do algoritmo realiza a limiarização da vizinhança (a Figura 2.5 mostra uma janela retangular retirada de uma imagem) com relação ao pixel central e, em seguida, a multiplicação pelos respectivos pesos. Além disso, a sequência dos pesos é definida de acordo com a enumeração da vizinhança.

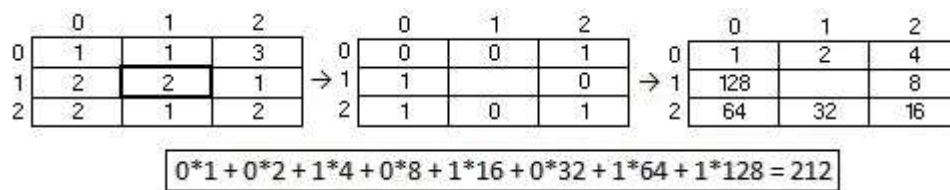


Figura 2.5: Cálculo LBP: Extração do pixel e vizinhança de uma imagem; decisão binária; multiplicação dos elementos da matriz binária por seus respectivos pesos e resultado LBP apresentado em valor decimal.

Fonte: Autoria Própria (2021).

O cálculo é dado pela Equação 2.1, no qual p varia de 0 a $P-1$ dentre os P possíveis pontos da vizinhança. As coordenadas u_c e v_c retornam a intensidade do nível de cinza g_c . A função $s(g_p, g_c)$ retorna 1 ou 0 se a diferença entre as intensidades g_p e g_c for maior ou menor que zero (Equação 2.2) (OJALA; PIETIKAINEN; HARWOOD, 1995).

$$LBP(u_c, v_c) = \sum_{p=0}^{P-1} s(g_p, g_c) \times 2^p \quad \text{Equação 2.1}$$

$$s(g_p, g_c) = \begin{cases} 0, & (g_p - g_c) < 0 \\ 1, & (g_p - g_c) \geq 0 \end{cases} \quad \text{Equação 2.2}$$

Os 256 possíveis valores a serem obtidos com tal cálculo variam no intervalo $[0, 255]$. As bordas da imagem podem ser desconsideradas no algoritmo

(dependendo da aplicação), ou uma operação de *padding* (incluir zeros ao redor da imagem) pode ser realizada. Durante o processo de cálculo dos valores LBPs adquiridos para cada pixel, cria-se um histograma, o qual é utilizado para compor o vetor de características (OJALA; PIETIKAINEN; HARWOOD, 1995).

A Figura 2.6 ilustra a disposição da vizinhança para diferentes valores de P e R. Para isso, é utilizado o operador circular definido na Equação 2.3, P é o número de vizinhos no qual são igualmente espaçados por um raio R ($R > 0$). Os níveis de cinza de pixels vizinhos que não tiverem coordenadas exatas na imagem, poderão ser alcançados a partir de uma interpolação bilinear (MAENPAA; PIETIKAINEN, 2004).

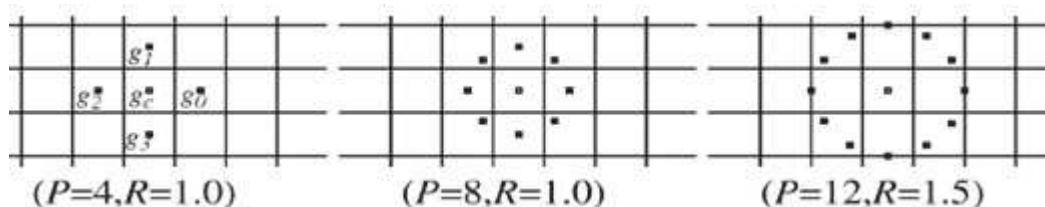


Figura 2.6: LBP Circular com diferentes raios e número de vizinhos.
Fonte: OJALA; PIETIKAINEN; MAENPAA (2002, pg. 3).

$$(u_p, v_p) = \left(u_c + R \cos\left(\frac{2\pi p}{P}\right); v_c - R \sin\left(\frac{2\pi p}{P}\right) \right) \quad \text{Equação 2.3}$$

LBP também pode ser utilizado considerando os padrões na forma de 0's e 1's. Denomina-se $LBP_{P,R}^{ri}$ no qual o subscrito *ri* (*rotation invariance*) denota invariância a rotação. Desta forma, a partir do processo descrito na Figura 2.5, realiza-se uma operação de deslocamento circular a direita nos bits (MAENPAA; PIETIKAINEN; 2004). Isso significa rotacionar o número binário até que seu bit mais significativo seja 0 (operação ROR). No entanto, essa variação do algoritmo não representa bem os padrões locais (MAENPAA; PIETIKAINEN; 2004).

Posteriormente observou-se que 90% dos padrões significativos obedecem ao critério de uniformidade, e podem ser medidos por meio de um valor U. Se as transições dos bits forem menores ou iguais a 2, esse número será considerado como um padrão uniforme. Por exemplo, a sequência de bits 0100 tem um valor de $U = 2$. Portanto, será considerado como um padrão uniforme (OJALA; PIETIKAINEN; MAENPAA, 2002).

2.4.3 ORB

O descritor FAST orientado e BRIEF rotacionado (*Oriented FAST and Rotated BRIEF* - ORB) veio do laboratório OpenCV e foi proposto por Rublee, Rabaud, Konolige e Bradski, em 2011. É uma fusão entre o detector de pontos FAST (*Features from Accelerated and Segments Test*) e o descritor de pontos BRIEF (*Binary Robust Independent Elementary Feature*), com algumas modificações para melhorar o desempenho (RUBLEE; RABAUD; KONOLIGE; BRADSKI, 2011). Este algoritmo é uma alternativa aos descritores de pontos de atenção SIFT e SURF no custo de computação, e tem boa aplicabilidade para processamento em tempo real (RUBLEE; RABAUD; KONOLIGE; BRADSKI, 2011).

Na fase de detecção de *keypoints* do algoritmo ORB, utiliza-se o FAST. Em seguida, aplica-se o detector de cantos Harris para obter os melhores pontos dentre os que foram detectados. Além disso, FAST não é invariante a escala, e por esse motivo ORB usa uma pirâmide para produzir características multiescalares. Outra limitação de FAST, é que este não calcula orientação (RUBLEE; RABAUD; KONOLIGE; BRADSKI, 2011). Para resolver esse problema, Rublee, Rabaud, Konolige e Bradski utilizam a *intensidade do centroide* definida por Rosin (1999).

Segundo Rosin (1999), para encontrar o centroide é preciso encontrar os momentos do *patch* (região definida por alguma janela PxP), dado por:

$$m_{tq} = \sum_{u,v} u^t v^q I(u, v) \quad \text{Equação 2.4}$$

Onde u e v são os valores das coordenadas dos pixels, t e q são as ordens dos momentos e $I(u, v)$ retorna a intensidade do pixel. Desta forma, é possível encontrar o centroide do *patch*:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad \text{Equação 2.5}$$

Sendo assim, a orientação do *patch* é dada pela Equação 2.6 onde atan2 retorna o ângulo respeitando o quadrante dos valores.

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad \text{Equação 2.6}$$

Com relação aos descritores, o algoritmo ORB utiliza o BRIEF com algumas modificações. BRIEF é um descritor binário, que compara intensidades $g_p = (u_p, v_p)$ e $g_c = (u_c, v_c)$ (suavizadas por um filtro) dentro do *patch* p , no qual um par de teste é denotado por (g_p, g_c) . Além disso, define-se uma quantidade de n pares de intensidades e um tipo de distribuição (CALONDER; LEPETIT; STRECHA; FUA, 2011). O teste de intensidades é dado pela Equação 2.7 e retorna 1 se g_p for menor que g_c , caso contrário o valor é 0.

$$\tau(p; u_p, v_p, u_c, v_c) = \begin{cases} 1, & p(u_p, v_p) < p(u_c, v_c) \\ 0, & p(u_p, v_p) \geq p(u_c, v_c) \end{cases} \quad \text{Equação 2.7}$$

Com isso, é possível construir o vetor de características binário de tamanho n . Mais detalhes sobre distribuições de pares são abordados em Calonder *et al.* (2011).

No algoritmo ORB, o tamanho do pixel *patch* é 31x31 por padrão, com um vetor de características fixo de $n = 256$ para representar cada *keypoint*. Desta forma, para armazená-lo é necessário $k = n/8$ que totaliza 32 bytes (RUBLEE; RABAUD; KONOLIGE; BRADSKI, 2011).

Como o descritor BRIEF não é invariante a rotação, ORB faz a correção para o problema. Para isso, com um conjunto de n pares é definida uma matriz de tamanho $S = 2 \times n$. Assim, com a orientação θ obtida na Equação 2.6, e sua matriz de rotação correspondente R_θ , é possível criar uma nova matriz rotacionada S_θ de S :

$$S_\theta = R_\theta S$$

Desta forma, obtém-se alguma invariância a rotação para o ORB (RUBLEE; RABAUD; KONOLIGE; BRADSKI, 2011). Além do mais, ORB discretiza os ângulos em incrementos de 12 graus e constrói uma tabela de consulta pré-computada para BRIEF. Se a orientação θ do *keypoint* for consistente em relação a diferentes perspectivas, o conjunto correto de pontos S_θ será usado para o cálculo do descritor (RUBLEE; RABAUD; KONOLIGE; BRADSKI, 2011).

2.5 Classificadores

2.5.1 K-vizinhos Mais Próximos

O classificador k-vizinhos mais próximos (*k-Nearest-Neighbours* - KNN) é um método de aprendizado baseado em instância que foi inicialmente proposto por Fix e Hodges (1951). Este algoritmo precisa armazenar os dados de treino que são utilizados para a classificação de uma nova instância, não havendo aprendizado ou criação de modelo de classificação (MITCHEL, 1997).

O primeiro passo consiste em calcular as distâncias entre uma nova instância x_j a ser classificada e as instâncias x_i que compõem a vizinhança. O cálculo da distância é dado abaixo:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad \text{Equação 2.8}$$

Na sequência, os vizinhos são ordenados com base na distância e os k vizinhos mais próximos são selecionados. Cada vizinho considerado representa um voto para a classe à qual pertence. A classe predita será aquela com maior representação (frequência) dentre os k vizinhos selecionados (MITCHELL, 1996).

Esse algoritmo é simples quanto a sua implementação e tem bom desempenho em bancos de dados pequenos ou médios (OLIVEIRA, 2016). No entanto, pode apresentar problemas de classificação quando os dados são de alta dimensão ou em grande quantidade (MITCHELL, 1996).

Outra variante de KNN considera a multiplicação de pesos para os k - vizinhos. Nesse caso a votação será influenciada por meio do inverso do quadrado da distância. As instâncias que forem mais próximas terão maior peso no somatório (MITCHELL, 1996). Este é altamente eficaz para muitos problemas práticos. É robusto para dados de treino com ruídos e eficiente para grande volume de dados. Ao ponderar os vizinhos mais próximos da nova instância, pode-se suavizar o impacto de dados isolados ruidosos (MITCHELL, 1996).

2.5.2 Máquina de Vetores de Suporte

A proposta de máquina de vetores de suporte (*Support Vector Machines - SVM*) é estimar o melhor hiperplano de separação em um conjunto de possíveis hiperplanos que solucionam o problema de classificação. Isto se refere como o hiperplano ótimo (ou hiperplano de margem máxima). Na Figura 2.7 é possível observar dois hiperplanos, sendo um com uma margem qualquer na Figura 2.7(a) e outro com maximização das margens na Figura 2.7(b).

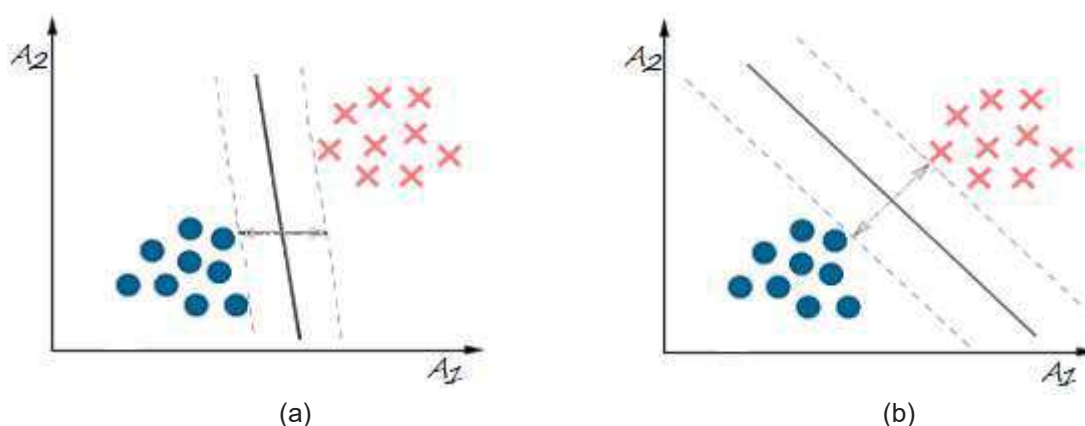


Figura 2.7 Exemplos de hiperplanos definidos pelo SVM em um caso binário: (a) Margem qualquer; (b) Margem máxima.
Fonte: Autoria Própria (2021).

A formulação do algoritmo pode ser caracterizada como um problema genérico de otimização convexa apresentado abaixo:

$$\text{minimize}_{w,b} \quad \frac{1}{2} \|w\|^2 \quad \text{Equação 2.9}$$

$$\text{sujeito a} \quad y_i (wx_i + b) \geq 1, \quad i = 1, \dots, N \quad \text{Equação 2.10}$$

Em que $w = \{w_1, w_2, w_3, \dots, w_n\}$ é o vetor normal ao hiperplano e $\|w\|$ sua norma. A variável y_i é o rótulo de classe sendo $\{-1, 1\}$, x_i é a instância de treinamento e b é o deslocamento do hiperplano em relação a origem. Mais detalhes sobre a formulação do problema podem ser encontrados em (CORTES; VAPNIK, 1995) e (LORENA; CARVALHO, 2007).

Uma forma de resolver o problema descrito é com a utilização dos multiplicadores de Lagrange α_i , $i = 1, \dots, N$ (LORENA; CARVALHO, 2007). Sendo assim, ao encontrar a solução ótima observa-se que para valores $\alpha_i^* > 0$, os vetores

de treinamento x_i são chamados de vetores de suporte (CORTES; VAPNIK, 1995). Se os valores de $\alpha_i = 0$, os vetores de treinamento x_i não interferem na obtenção do hiperplano ótimo. Se esses vetores forem retirados do treinamento, não haverá impacto na solução (LORENA; CARVALHO, 2007).

Para classificar uma nova instância x_n , basta substituir os valores na equação de decisão 2.11. Se o cálculo do valor for positivo, a função sinal sgn retornará 1 de tal forma que a nova instância pertencerá a classe positiva (LORENA; CARVALHO, 2007). Do contrário, a função retorna -1 e será classificada como uma instância negativa.

$$f(x) = sgn \left(\sum_{i=1}^N \alpha_i * y_i x_i x_n + b^* \right) \quad \text{Equação 2.11}$$

Este algoritmo é conhecido como SVM de margens rígidas, pois é eficaz apenas para dados linearmente separáveis. Em problemas práticos torna-se difícil a aplicabilidade desse algoritmo, uma vez que os dados podem ser não-lineares ou haver a presença de ruídos. Para tornar o algoritmo adequado a problemas reais, introduziu-se variáveis de folga ε_i para todo $i = 1, \dots, n$. Essas variáveis relaxam as restrições do problema apresentado, permitindo que algumas instâncias violem a mesma. Tal proposta refere-se ao algoritmo SVM de margens suaves (LORENA; CARVALHO, 2007).

O SVM é robusto para dados de alta dimensão, e em geral apresenta melhor desempenho do que outras técnicas de aprendizado que podem obter classificadores super ou sub ajustados. Além disso, suas variantes podem ser utilizadas juntamente com as funções de Kernel para problemas não lineares (LORENA; CARVALHO, 2007). Tal combinação permite a construção de simples hiperplanos em grande dimensão de forma a reduzir o custo computacional. Algumas de suas limitações referem-se a sensibilidade dos valores de parâmetros, e também dificuldade em interpretar os modelos gerados (LORENA; CARVALHO, 2007).

3 MATERIAIS E MÉTODOS

A metodologia empregada neste trabalho está inserida na Figura 2.3, na qual mostra a visão geral de um sistema para reconhecimento de padrões. A etapa de aquisição teve sua construção concluída em um projeto de pesquisa em parceria de docentes do curso de Engenharia Civil e Tecnologia em Sistemas para Internet, do câmpus Toledo da UTFPR. O processo para obtenção das imagens está descrito na Seção 3.1.

Os experimentos se dividem na avaliação do grau de corrosão de barras de aço por meio de imagens dos corpos de concreto (I) e também das próprias barras (II). Para isso, será utilizada a linguagem Python juntamente com bibliotecas OpenCV e Scikit-Learn. Essa escolha se deve ao fato da facilidade da linguagem e também das bibliotecas que fornecem diversos módulos. Mais detalhes da metodologia utilizada para os experimentos são descritos no decorrer deste capítulo.

3.1 Base de Imagens

As imagens foram obtidas por meio de um experimento de aceleração de corrosão das barras de aço normalmente empregadas em concreto armado. Tal procedimento consistiu na utilização de 24 barras de aço, sendo que seis compõem um grupo de controle (referências sem corrosão), e outros três grupos de seis amostras foram submetidos a processos de aceleração de corrosão por períodos de 24, 48 e 72 horas. Cada amostra é caracterizada por um diâmetro de 12,5 mm e comprimento de aproximadamente 25 cm. As amostras são da marca Gerdau e pertencem ao mesmo lote de fabricação.

A construção da base de imagens começou com a captura de 5 imagens abrangendo diferentes perspectivas para as amostras de aço. Essas imagens servem como referências das barras de aço sem corrosão para o experimento II (Figura 3.1(c)). Em seguida, colocou-se um fio elétrico conectado a parte central de cada barra de aço. Depois, as amostras de aço foram recobertas com concreto 1: 1,49: 2,77: 0,65 (Cimento: Areia: Brita: Água). A espessura do cobrimento é de 2 cm para simular o contexto real das estruturas nas edificações. Para o processo de hidratação do cimento, as amostras permaneceram sob cura por 28 dias. Seguiu-se

com a captura de 5 fotos para servirem como referências sem corrosão para o experimento I (Figura 3.1(a)).

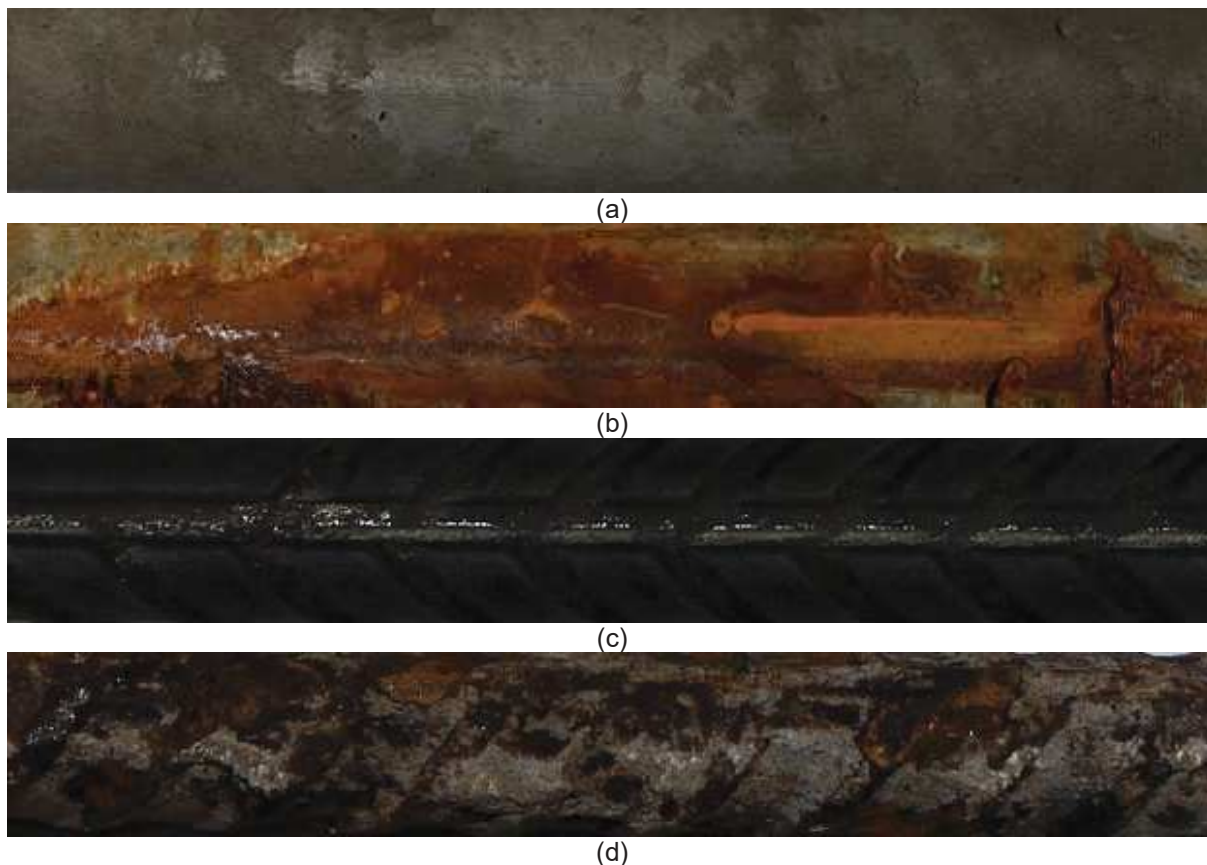





Figura 3.1: Base de imagens: (a) Superfície corpo de prova de concreto sem corrosão (b) Superfície corpo de concreto com corrosão (c) Superfície barra de aço sem corrosão (d) Superfície barra de aço com corrosão.

Fonte: Autoria Própria (2021)

Na sequência, os corpos de concreto foram submetidos ao processo de aceleração de corrosão. Nesta etapa, as amostras estavam parcialmente imersas em um tanque de água com 3.5% de Cloreto de Sódio (NaCl), no qual houve condução de energia elétrica através do fio anteriormente conectado a sua parte central. Aqui, os três grupos de corpos de concreto ficaram por tempos diferentes no tanque, sendo estes 24, 48 e 72 horas para gerar os níveis de corrosão. Paralelamente ocorreu a coleta de imagens (exemplo de corrosão na Figura 3.1(b)), as quais seriam utilizadas em conjunto com as imagens dos corpos de concreto sem corrosão. Por fim, as barras de aço foram extraídas do concreto e aconteceu novamente a captura das imagens das amostras corroídas (exemplo de corrosão na Figura 3.1(d)).




A partir da base construída, foram avaliados os graus de deterioração utilizando imagens das superfícies dos corpos de prova de concreto (Quadro 3.1) e também imagens das próprias barras de aço (Quadro 3.2). Em ambos os casos, a base foi dividida de acordo com o tempo em que o corpo de concreto foi submetido ao processo corrosão. Desta forma, os tempos 24, 48 e 72 horas representam as classes a serem trabalhadas. Cada classe contém 30 amostras para constituir os respectivos graus de deterioração.

Quadro 3.1: Experimento I.

Classes	Amostras	Imagem (exemplo)
0 (24 horas)	30	
1 (48 horas)	30	
2 (72 horas)	30	

Fonte: Autoria própria (2021).

Quadro 3.2: Experimento II.

Classes	Amostras	Imagem (exemplo)
0 (24 horas)	30	
1 (48 horas)	30	
2 (72 horas)	30	

Fonte: Autoria própria (2021).

3.2 Pré-processamento e Segmentação

Após a aquisição da base de imagens, para a fase de pré-processamento as imagens foram convertidas para níveis de cinza. Na sequência, conforme ilustrado na Figura 3.2, ocorreu a segmentação das imagens. Os parâmetros utilizados em cada etapa foram os mesmos para as imagens dos Experimentos I e II, com exceção da filtragem. Após realizar o processo de segmentação, com a definição dos objetos de interesse (corpo de concreto ou barra de aço), foram retomadas e recortadas as imagens em tons de cinza para a fase de extração de características.

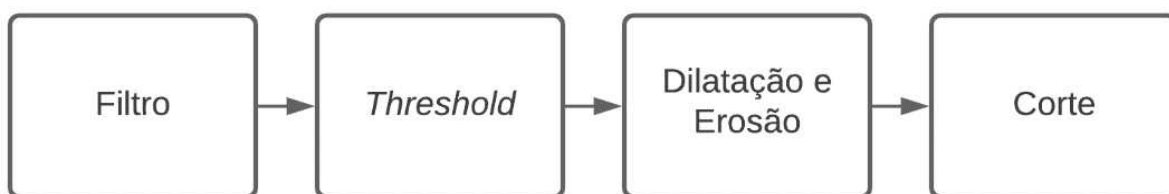


Figura 3.2: Etapas para a segmentação das imagens
Fonte: Autoria própria (2021)

Na primeira parte do fluxograma utilizou-se o filtro de Niblack, o qual realiza um *threshold* na imagem. Inicialmente a tentativa de utilização desse filtro se deu para tal função. No entanto, ao aplicá-lo na imagem de dimensões 5184×3456 tornou-se um filtro passa banda. O comportamento observado de acordo com os parâmetros experimentais, promoveu a remoção de ruídos, borramento da imagem e também o realce das bordas.

O cálculo para o filtro de Niblack é apresentado na Equação 3.1, para a qual T representa o *threshold* calculado para cada pixel da imagem, $m(x, y)$ e $s(x, y)$ se referem à média e ao desvio padrão local da imagem com o mesmo tamanho da imagem analisada. As matrizes m e s são obtidas ao calcular a média e desvio padrão para cada pixel utilizando uma janela com valor especificado. A variável h pondera o desvio padrão, adaptando o *threshold* com h desejado.

$$T = m(x, y) - h * s(x, y) \quad \text{Equação 3.1}$$

Para os corpos de concreto com corrosão (Experimento I) foram utilizados os parâmetros de janela 45×45 e $h = 4$ (imagem de saída Figura 3.3 (a)).

Para as barras de aço com corrosão (Experimento II) os valores foram de 55×55 e $h = 1$ (imagem de saída Figura 3.3(b)).

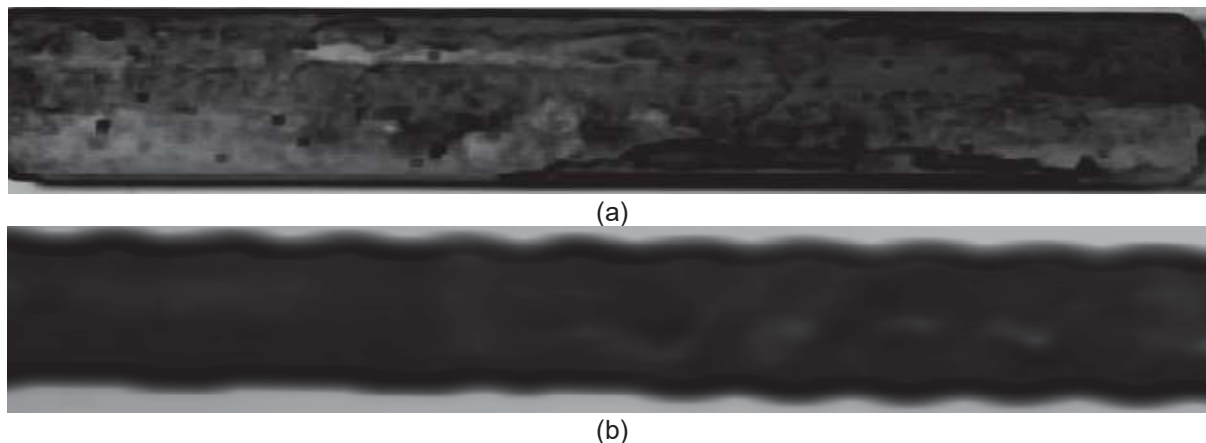


Figura 3.3: Filtro de Niblack: (a) Corpo de concreto com corrosão; (b) Barra de aço com corrosão
Fonte: Autoria Própria (2021)

Após a filtragem, análises por meio de histogramas foram realizadas para encontrar o limiar para binarizar as imagens, sendo que os melhores resultados foram obtidos com por meio da seguinte condição: $(x, y) \geq 60$ se tornaram 1 (255) e $(x, y) < 60$, se tornaram 0 (zero), tal como ilustram as Figuras 3.4 (a) e (b).

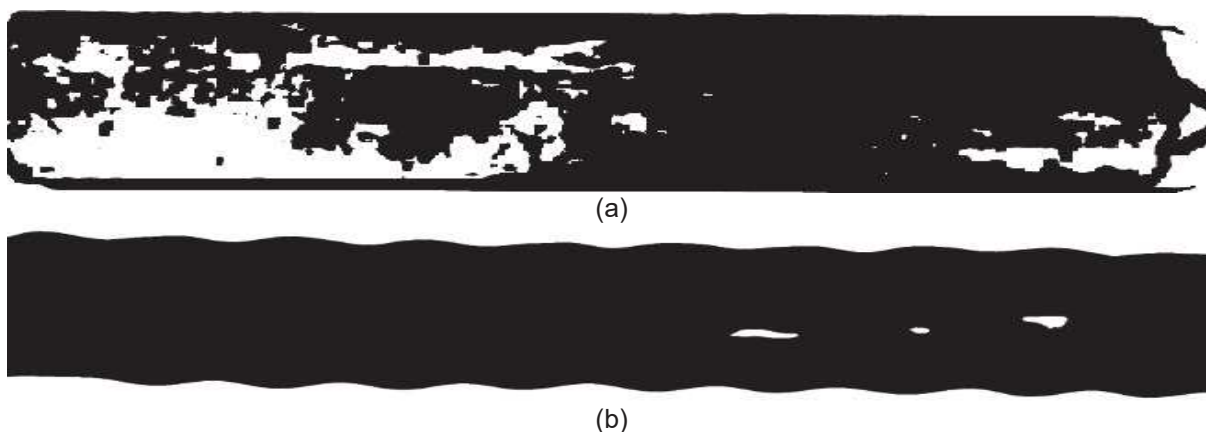


Figura 3.4: Imagens binárias: (a) Corpo de concreto com corrosão; (b) Barra de aço com corrosão
Fonte: Autoria própria (2021)

Na terceira etapa foram realizadas as operações de dilatação e erosão, tendo sido utilizados diferentes formatos de máscaras e quantidades de iterações. Optou-se pela utilização das mesmas para remover possíveis borramentos e reforçar bordas fracas da imagem (Figura 3.5 (a) e (b)).

Os melhores resultados foram gerados com a aplicação de dilatação e uma máscara em formato de cruz de dimensão 5×5 (sendo a linha e coluna central

de 1's e o restante 0's), seguida com erosão e uma máscara de dimensão 5x5 com os todos elementos iguais a 1's.



Figura 3.5: Dilatação e Erosão: (a) Corpo de concreto com corrosão; (b) Barra de aço com corrosão
Fonte: Autoria Própria (2021)

As imagens da Figura 3.6 apresentam imagens originais e os recortes finais obtidos com a execução do processo descrito. Para alcançar tal resultado, realizou-se a detecção de bordas com o algoritmo *Canny* e a função *findContours*.



Figura 3.6: Segmentação das imagens com corrosão: (a) Corpo de concreto com fundo cinza e sombras; (b) Corpo de concreto segmentado; (c) Barra de aço com fundo cinza e sombras; (d) Barra de aço segmentado.
Fonte: Autoria Própria (2021)

O algoritmo *Canny* utiliza da definição de derivada para encontrar grandes variações entre os pixels vizinhos. A imagem de saída de *Canny* contém os principais traços relevantes na imagem. Assim, a imagem produzida por esse algoritmo é utilizada como entrada para a função *findContours*. Esta função, tem

como objetivo retornar os contornos dos objetos detectados na imagem, sendo que cada contorno é definido por uma lista de pixels (e suas coordenadas).

Considerou-se que o tamanho (área) de cada possível objeto presente na imagem era dado pela quantidade de pixels que compunham seu contorno. Com isso, selecionou-se o maior contorno em cada imagem, o qual representava o objeto de interesse (corpo de concreto ou barra de aço) para definir as coordenadas adequadas para realizar o recorte na imagem original.

3.3 Extração de Características

Após obter as imagens de interesse, seguiu-se para a extração de características. Para os experimentos com o descritor LBP, foram consideradas as variações LBP uniforme *variante* à rotação $LBP_{P,R}^{u2}$ e LBP uniforme *invariante* à rotação $LBP_{P,R}^{riu2}$. Os parâmetros de $LBP_{P,R}^{u2}$ e $LBP_{P,R}^{riu2}$ variaram através de dois loops *for* combinando a vizinhança $P = \{6,8,10,12,14,16,18,20\}$ e raio $R = \{1,2,3,4,5\}$.

Cada variação LBP obteve uma matriz com as mesmas dimensões da imagem original por meio da função *local_binary_patterns* da *Scikit-Learn*, contendo os padrões locais organizados de acordo com seu funcionamento. Desta maneira, calculou-se os histogramas de cada LBP por meio da biblioteca *numpy* e por fim se acrescentou a classe correspondente ao final do vetor de características. Assim, foram gerados 80 arquivos contendo os histogramas resultantes das variações de P e R . Sendo 40 arquivos advindos de $LBP_{P,R}^{u2}$ e os outros 40 arquivos de $LBP_{P,R}^{riu2}$.

A GLCM foi calculada considerando 256 níveis de cinza, sendo que os parâmetros variaram com distâncias $d = \{1,2,3, \dots, 10\}$ e ângulos $a = \{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$. As funções utilizadas para implementação foram *greycomatrix* e *greycoprops*, que aceitam parâmetros individuais de distância e ângulo, ou lista de parâmetros com várias distâncias e ângulos por vez.

Foram geradas e avaliadas diferentes combinações entre distâncias e ângulos. Por exemplo gerou-se uma matriz de co-ocorrência para $d = 1$ e $a = 0^\circ$, outra para $d = 1$ e $a = 45^\circ$, e assim para todos os outros valores de d e a . Os parâmetros passados para as funções variaram primeiro com os elementos individuais, e posteriormente com a lista de parâmetros com várias distâncias e

ângulos. Assim, resultou 285 combinações para GLCM. Após a obtenção das matrizes de co-ocorrência, calculou-se as características contraste, dissimilaridade, homogeneidade, ASM, energia e correlação para cada uma.

Para o detector e descritor ORB, utilizou-se 500 *keypoints* sendo cada um descrito por um vetor de 32 bytes. Após a obtenção dos descritores de cada imagem, foi usada a técnica de *Bag of Visual Words* para representar melhor essas características. O algoritmo utilizado para encontrar os centroides foi *KMeans* com valores de *clusters* = {10, 20, 30, 40, 60, 80, 100}.

Para criar as palavras de identificação, foi usada a função *vq* que recebe a descrição dos centroides encontrados e o vetor de características analisado. Esta função retorna um vetor de 500 dimensões que contém as palavras mais frequentes para descrever a instância. Depois disso, foi feito um histograma, sendo que a quantidade de rótulos foi igual ao tamanho da quantidade de *clusters* escolhida. A construção das palavras de identificação foi feita para cada conjunto de treino e teste independentemente.

3.4 Classificação e Avaliação

Para SVM foi utilizado GridSearch (biblioteca Scikit-Learn) com as listas de parâmetros $C = \{0.1, 1, 10, 100, 1000\}$ e $\text{gamma} = \{1, 0.1, 0.01, 0.001, 0.0001\}$ para o Kernel RBF. Para KNN foram utilizados os valores de vizinhos $k = \{1, 2, 3, \dots, 30\}$.

Em cada combinação de conjunto de características e algoritmo de aprendizado, as instâncias foram divididas usando K-fold cross validation com valor $K = 3$. Este consistiu em dividir todo o conjunto em 3-folds, no qual um conjunto foi separado como conjunto de teste enquanto os outros 2 folds foram utilizados como conjunto de treino. A Figura 3.7 exibe o processo que foi utilizado para os dois experimentos.

Para avaliar a performance do classificador foi utilizada a matriz de confusão (MC) e as seguintes métricas dela derivadas: acurácia, precisão, sensibilidade e F-score. Embora a matriz apresentada no Quadro 3.3 seja 2x2 e o problema tratado possua três classes, ela é suficiente para explicar sua função. As variáveis da MC estão descritas abaixo:

- **Verdadeiro Positivo (TP, do inglês *True Positive*):** Refere-se ao número de vezes que o classificador prevê corretamente a classe positiva como positiva.
- **Verdadeiro Negativo (TN, do inglês *True Negative*):** Refere-se ao número de vezes que o classificador prevê corretamente a classe negativa como negativa
- **Falso Positivo (FP, do inglês *False Positive*):** Refere-se ao número de vezes que o classificador prevê incorretamente a classe negativa como positiva.

Falso Negativo (FN, do inglês *False Negative*): Refere-se ao número de vezes que o classificador prevê incorretamente a classe positiva como negativa.

Iteração 1	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	Métricas 1
Iteração 2	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	Métricas 2
Iteração 3	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	Métricas 3



Figura 3.7: Ilustração para 3-fold cross validation
Fonte: Autoria Própria (2021)

Quadro 3.3: Matriz de Confusão

		Rótulo Predito	
		Positivo	Negativo
Rótulo da Classe	Positivo	TP	FN
	Negativo	FP	TN

Fonte: Autoria Própria (2020).

A acurácia representa a taxa de instâncias classificadas corretamente dentre as instâncias que compõem o conjunto de teste, ou seja, cujo rótulo predito é o rótulo real (Equação 3.2).

$$acurácia = \frac{TP + TN}{TP + TN + FN + FP} \quad \text{Equação 3.2}$$

A precisão (Equação 3.3) representa a taxa de acertos das instâncias pertencentes à classe positiva e todas as instâncias preditas para esta classe (AWAD; KHANNA, 2015; FAWCETT, 2005; HAKER, 2005; KHREICH, 2011; KHREICH, 2010).

$$precisão = \frac{TP}{TP + FP} \quad \text{Equação 3.3}$$

Já a sensibilidade (Equação 3.4) mede a taxa de acerto das instâncias da classe positiva sobre o que deveria ter sido classificado como da classe positiva.

$$sensibilidade = \frac{TP}{TP + FN} \quad \text{Equação 3.4}$$

Outra medida é F-Score (Equação 3.4), uma média harmônica que relaciona as taxas precisão e sensibilidade (AWAD; KHANNA, 2015).

$$F - score = \frac{2 * precisão * sensibilidade}{precisão + sensibilidade} \quad \text{Equação 3.5}$$

4 RESULTADOS E DISCUSSÕES

Nesta Seção são apresentados os resultados obtidos para os Experimentos I e II. Foram utilizadas combinações entre os extratores de características GLCM, LBPs e ORB, e os algoritmos de aprendizado SVM e KNN, considerando as variações apresentadas para cada um no capítulo de materiais e métodos. Neste capítulo são destacados os experimentos com as combinações que alcançaram as melhores taxas de classificação.

4.1 GLCM

Os resultados apresentados no Quadro 4.1 procedem das métricas obtidas na MC para o Experimento I (corpos de concreto). São mostrados os diferentes classificadores com as variações nos parâmetros para ângulos α e distâncias d . Cada métrica da MC é a média dos resultados de *3-fold cross validation* e está exibida junto com os respectivos desvios padrões (Dp).

Quadro 4.1: Resultados para GLCM do Experimento I.

Classificador	GLCM		Acurácia	Precisão	Sensibilidade	F-Score
	Distâncias d	Ângulos α				
SVM C = 100 G = 1	1,2,3,4,5,6 ,7,8,9,10	180°	0,52 Dp = 0,08	0,54 Dp = 0,10	0,55 Dp = 0,07	0,51 Dp = 0,07
SVM C = 1000 G = 0,1	3	0°	0,48 Dp = 0,08	0,43 Dp = 0,13	0,50 Dp = 0,06	0,45 Dp = 0,10
KNN $k = 11$	8	180°	0,48 Dp = 0,08	0,51 Dp = 0,10	0,47 Dp = 0,09	0,47 Dp = 0,10
KNN $k = 1$	1,2,3	0°	0,47 Dp = 0,03	0,47 Dp = 0,03	0,46 Dp = 0,03	0,45 Dp = 0,03

Fonte: Autoria própria (2021)

Observa-se para cada métrica da MC que os resultados não diferem tanto entre si, mesmo com a extensa busca pelos parâmetros ideais para o descritor GLCM e classificadores. As taxas de acurácia, precisão, sensibilidade e F-Score não

foram promissoras neste experimento. O descritor não conseguiu fornecer boas características para representar o grau de deterioração dos corpos de concreto.

No Quadro 4.2 tem-se os valores da MC para o Experimento II (barras de aço), seguindo as mesmas regras já descritas para o Quadro 4.1.

Quadro 4.2: Resultados para GLCM do Experimento II.

Classificador	GLCM		Acurácia	Precisão	Sensibilidade	F-Score
	Distâncias d	Ângulos α				
SVM C = 10 G = 1	1	0°,45°,135°,180°,225°, 270°	0,61 Dp = 0,05	0,64 Dp = 0,05	0,63 Dp = 0,06	0,60 Dp = 0,08
SVM C = 100 G = 0,1	7	270°	0,60 Dp = 0,06	0,64 Dp = 0,01	0,63 Dp = 0,01	0,59 Dp = 0,05
KNN $k = 7$	3	0°,45°,135°,180°,225°, 270°	0,70 Dp = 0,09	0,72 Dp = 0,07	0,72 Dp = 0,07	0,69 Dp = 0,09
KNN $k = 1$	7	0°,45°,135°,180°,225°	0,70 Dp = 0,13	0,74 Dp = 0,09	0,71 Dp = 0,14	0,70 Dp = 0,13

Fonte: Autoria própria (2021)

Nota-se que para estes resultados, as métricas da MC apresentam valores superiores àqueles alcançados no Experimento I. Aqui, o algoritmo GLCM gera características mais adequadas para representar os graus de deterioração nas barras de aço.

Os melhores resultados do Quadro 4.2 se mostram com a junção de KNN e GLCM. Sendo que apresentam valores razoavelmente bons quanto a precisão da separação das classes, além da boa taxa de recuperação das instâncias de cada classe.

4.2 LBP

No Quadro 4.3 é possível observar alguns diferentes parâmetros para KNN e SVM, sendo que estes representam os melhores resultados dentre os obtidos para a classificação do Experimento I (corpos de concreto). Também são mostradas as variantes LBPs com as métricas acurácia, precisão, sensibilidade e F-Score providas da MC.

Quadro 4.3: Resultados para $LBP_{P,R}^{u2}$ e $LBP_{P,R}^{riu2}$ do Experimento I.

Classificador	LBP	Acurácia	Precisão	Sensibilidade	F-Score
SVM C = 1000, G = 0,01	$LBP_{8,4}^{u2}$	0,77 Dp = 0,07	0,77 Dp = 0,06	0,78 Dp = 0,07	0,77 Dp = 0,07
SVM C = 1000, G = 1	$LBP_{14,5}^{riu2}$	0,69 Dp = 0,05	0,71 Dp = 0,07	0,69 Dp = 0,06	0,68 Dp = 0,07
KNN k = 1	$LBP_{12,4}^{u2}$	0,63 Dp = 0,12	0,64 Dp = 0,12	0,65 Dp = 0,12	0,63 Dp = 0,12
KNN k = 6	$LBP_{14,4}^{riu2}$	0,57 Dp = 0,06	0,63 Dp = 0,01	0,60 Dp = 0,02	0,57 Dp = 0,06

Fonte: Autoria própria (2021)

O classificador SVM tem as melhores taxas de acurácia no Quadro 4.3. Quando associado ao $LBP_{8,4}^{u2}$, este também apresentou a taxa de classificação mais alta, com acurácia 0,77 e desvio padrão de 0,07, além de bons valores para precisão, sensibilidade e F-score.

As métricas da MC, juntamente com as variantes LBPs e classificadores estão apresentados no Quadro 4.4 para o Experimento II (barras de aço).

Quadro 4.4: Resultados para $LBP_{P,R}^{u2}$ e $LBP_{P,R}^{riu2}$ do Experimento II

Classificador	LBP	Acurácia	Precisão	Sensibilidade	F-Score
SVM C = 1000, G = 1	$LBP_{20,4}^{riu2}$	0,70 Dp = 0,08	0,63 Dp = 0,22	0,72 Dp = 0,09	0,66 Dp = 0,17
SVM C = 1000, G = 0,01	$LBP_{16,2}^{u2}$	0,66 Dp = 0,05	0,67 Dp = 0,06	0,67 Dp = 0,05	0,64 Dp = 0,03
KNN k = 16	$LBP_{16,1}^{riu2}$	0,64 Dp = 0,03	0,63 Dp = 0,02	0,60 Dp = 0,03	0,58 Dp = 0,05
KNN k = 9	$LBP_{20,1}^{u2}$	0,69 Dp = 0,05	0,72 Dp = 0,04	0,71 Dp = 0,08	0,68 Dp = 0,07

Fonte: Autoria própria (2021)

A associação entre $LBP_{20,4}^{riu2}$ e SVM gera a melhor taxa de acurácia do Quadro 4.4 com valor 0,70 e desvio padrão de 0,08. Apesar de apresentarem bons resultados para as métricas, ainda é necessário elevar estes valores para garantir um ótimo modelo de classificação.

4.3 ORB

Os resultados para o detector e descritor ORB está apresentado no Quadro 4.5 para o Experimento I.

Quadro 4.5: Resultados para ORB do Experimento I

Classificador	N° Clusters	Acurácia	Precisão	Sensibilidade	F-Score
SVM C = 10, G = 1	20	0,40 Dp = 0,06	0,51 Dp = 0,04	0,40 Dp = 0,06	0,40 Dp = 0,06
KNN k = 1	30	0,43 Dp = 0,13	0,49 Dp = 0,17	0,43 Dp = 0,13	0,43 Dp = 0,13

Fonte: Autoria própria (2021)

Embora tenham sido realizados vários testes para SVM e KNN com diferentes números de *clusters* para caracterizar os descritores obtidos com ORB, os melhores resultados tem baixos valores para as métricas da MC. Os *keypoints* detectados e seus descritores não fornecem características dissimilares entre as classes para gerar boas classificações no Experimento I.

O Quadro 4.6 mostra os resultados para o Experimento II.

Quadro 4.6: Resultados para ORB do Experimento II

Classificador	N° Clusters	Acurácia	Precisão	Sensibilidade	F-Score
SVM C = 0,1, G = 1	80	0,36 Dp = 0,05	0,35 Dp = 0,16	0,36 Dp = 0,05	0,36 Dp = 0,05
KNN k = 25	60	0,44 Dp = 0,10	0,45 Dp = 0,08	0,44 Dp = 0,10	0,44 Dp = 0,10

Fonte: Autoria própria (2021)

Observa-se que os resultados apresentados tem baixos valores assim como no Experimento I. Mostram que o algoritmo ORB também não fornece boas características para representar as barras de aço com corrosão.

5 CONSIDERAÇÕES FINAIS

Foram armazenadas cerca de 10 mil classificações em arquivos para cada experimento, considerando a combinação entre os descritores GLCM, LBPs e ORB, com os classificadores KNN e SVM. Os melhores resultados apresentados consideraram até 5 itens para cada combinação, utilizando a métrica acurácia para seleção. A realização da pesquisa foi feita por meio da linguagem Python com diferentes bibliotecas.

Com relação ao **Experimento I**, os melhores resultados para LBP foram obtidos para $LBP_{3,4}^{u2}$ e SVM, com acurácia 0,77 e desvio padrão de 0,07. A taxa de classificação mais alta para GLCM foi com SVM, apresentando acurácia de 0,52 e desvio padrão de 0,08. Por fim, ORB e KNN resultaram em acurácia de 0,43 e desvio padrão 0,13.

Para o **Experimento II**, a pontuação mais alta foi obtida com GLCM e KNN com acurácia 0,70 e desvio padrão 0,13. Para LBP a taxa de classificação mais alta foi com $LBP_{20,4}^{riu2}$ e KNN com acurácia 0,70 e desvio padrão de 0,08. Para ORB o melhor resultado foi com KNN, sendo a acurácia 0,44 e desvio padrão de 0,10.

Apesar da combinação desses algoritmos terem resultado em boas métricas (das melhores apresentadas para os Experimentos I e II), é preciso investigar novos descritores e classificadores para aperfeiçoar ainda mais os resultados encontrados. Esses podem ser KAZE, AKAZE, BRISK, PCA, LDA, Redes neurais/convolucionais e outros. Além disso, a base de imagens poderia ser aumentada para fornecer mais informações.

REFERÊNCIAS

- AHANONEN, T.; MATAS, J.; PIETIKAINEN, M. **Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features**. 2009.
- AWAD, A. I.; HASSABALAH, M. **Image Feature Detectors and Descriptors: Foundations and Applications**. 2016.
- CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. **BRIEF: Binary Robust Independent Elementary Features**. 2011.
- CANGELOSI, D.; MUSELLI, M.; PARODI, S.; BLENGIO, F.; BECHERINI, P.; VERSTEEG, R.; CONTE, M.; VARESIO, L.; **Use of Attribute Driven Incremental Discretization and Logic Learning Machine to build a prognostic classifier for neuroblastoma patients**. 2013.
- CASCUDO, O. **O controle da corrosão de armaduras em concreto: inspeção e técnicas eletroquímicas**. São Paulo: Pini; Goiania: Editora UFMG, 1997.
- CONNERS, R. W.; KLINE, D. E.; ARAMAN, P. A.; DRAYER, T. H. **Machine vision technology for the forest products industry**. Computer, v. 30, n. 7, p. 43-48, 1997.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2001.
- CORTES, C.; VAPNIK, V.; **Support-Vector Networks**. 1995.
- FIGUEIREDO, E. P.; MEIRA, G. **Corrosão de armadura de estruturas de concreto**. 2013. p.5.
- G1. **Ponte Morandi, que desabou parcialmente em Gênova, começa a ser desmantelada**. 2019. Disponível em: <<https://g1.globo.com/mundo/noticia/2019/02/08/ponte-morandi-que-desabou-parcialmente-em-genova-comeca-a-ser-desmantelada.ghtml>>
- G1. **Projetista da ponte que desabou na Itália alertou sobre risco de corrosão em 1979**. 2018. Disponível em: <<https://g1.globo.com/mundo/noticia/2018/08/19/projetista-da-ponte-que-desabou-na-italia-alertou-sobre-risco-de-corrosao-em-1979.ghtml>>
- G1. **Queda parcial de ponte deixa 37 mortos em Gênova, na Itália**. 2018. Disponível em: <<https://g1.globo.com/mundo/noticia/2018/08/14/ponte-cai-na-italia.ghtml>>
- GONZALEZ, R. C.; WOODS, R. E. **Processamento Digital de Imagens**. 2009.
- GOU, J. ZHANG, L. XIONG, T. **A new distance-weighted k-nearest neighbor classifier**. 2012.
- HARALICK, R. M. **Statistical and structural approaches to texture**. 1979.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. **Textural features for image classification**. IEEE Transactions on Systems, Man, and Cybernetics, 1973.

HEIKILA, M.; PIETIKAINEN, M.; SCHIMID, C. **Description of interest regions with local binary patterns**. 2008.

HELENE, P. **Manual para reparo, reforço e proteção de estruturas de concreto**. 2. ed. São Paulo: PINI, 1992.

HELENE, P. R. L. **Vida útil das estruturas de concreto**. In: CONGRESSO IBEROAMERICANO DE PATOLOGIA DAS CONSTRUÇÕES, 4., CONGRESSO DE CONTROLE DE QUALIDADE, 6., 1997, Porto Alegre. Anais... Porto Alegre, 1997. p. 1-30.

LI, P.; GOU, J.; YANG, H. **The Distance-Weighted K-nearest Centroid Neighbor Classification**. 2017.

LORENA, A. C.; CARVALHO, A. C. P. L. F. **Uma introdução às Support Vector Machines**. 2007.

Marketing Tecnosil. **Corrosão de armadura: o que causa e como amenizar esse dano?** 2020? Disponível em: <<https://www.tecnosilbr.com.br/corrosao-de-armadura-o-que-causa-e-como-amenizar-esse-dano/>>. Acessado em: 28/10/2020.

MARTINS, J. F. **Identificação de espécies florestais utilizando seleção dinâmica de classificadores no espaço de dissimilaridade**. Curitiba, 2014.

MEHTA, P.K.; MONTEIRO, P. **Concreto: Microestrutura, Propriedades e Materiais**. 3.ed. São Paulo: IBRACON, 2008.

MEIRA, G.R. **Corrosão de armaduras em estruturas de concreto: fundamentos, diagnósticos e prevenção**. João Pessoa: IFPB, 2017.

MITCHEL, T. **Machine Learning**. 1997.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. **A comparative study of texture measures with classification based on feature distributions**. 1996.

OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**. Pattern Analysis and Machine Intelligence, IEEE Transactions, 2002.

OJALA, T.; PIETIKAINEN, M.; XU, Z. **Rotation-invariant texture classification using feature distributions**. 2000.L. Wang, D.C. He. A new statistical approach for texture analysis. 1990.

OLIVEIRA, A. C. **Máquina de Aprendizagem Mínima com Opção de Rejeição**. Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de PósGraduação em Ciência da Computação, Fortaleza, 2016.

PHAM, D. T.; ALCOCK, R. J. **Automated visual inspection of birch wood boards.** IEE Colloquium on Artificial Intelligence in Manufacturing, p. 1-4, 1997.

RADOVAN, S.; GEORGE, P.; PANAGIOTIS, M.; MANOS, G.; ROBERT, A.; IGOR, D. **An approach for automated inspection of wood boards.** International Conference on Image Processing, 1, p. 798-801, 2001.

ROSIN, P. L. **Measuring Corner Properties.** Computer vision and Image Understanding, p. 1-3. 1999.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. **ORB: an efficient alternative to SIFT or SURF.** IEEE International Conference on Computer Vision. 2011.

SCIKIT-LEARN. **Cross-validation: evaluating estimator performance.** <https://scikit-learn.org/stable/modules/cross_validation.html>

SONKA, M.; HLAVACK, V.; BOYLE, R. **Image Processing, Analysis and Machine Vision.** 2013.

VALENTE, A. P. V.; SILVA, A. P.; CALIXTO, J. M. F. **Análise dos processos de recuperação de patologias: trincas e impermeabilização.** Construindo, Belo Horizonte, v.1, n.2, p. 7-11, jul. / dez. 2009.

WANG, L.; HE, D.C. **A new statistical approach for texture analysis.** 1990.