

**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS
FATEC PROFESSOR JESSEN VIDAL**

CAUANE EMANUELA ANDRADE DOS SANTOS

**PROCESSAMENTO DE IMAGEM APLICADA À DETECÇÃO
DE MUDANÇA NO USO DA TERRA**

Orientador: Prof. Dr. Fabrício Galende Marques de Carvalho

São José dos Campos
2021

SUMÁRIO

1	Introdução	3
1.1	Motivação	3
1.2	Definição do problema	4
1.3	Objetivo	4
2	Desenvolvimento	5
2.1	Fundamentação básica de PDI	5
2.1	Tecnologias Utilizadas	6
2.2	Arquitetura da Aplicação	7
2.3	Descrição Geral das Operações do Sistema	8
2.3.1	Aplicação web	9
2.3.2	Detalhamento	11
3	Resultados e Discussão	16
3.1	Sugestões para trabalhos futuros	16

1 INTRODUÇÃO

Esse capítulo apresenta o problema a ser resolvido bem como o objetivo deste Trabalho de Graduação.

1.1 Motivação

De acordo com o relatório do Painel Intergovernamental sobre Mudanças Climáticas (IPCC) de 2019, a humanidade utiliza mais de 70% da Terra sem gelo, e as atividades primárias, como agricultura, pecuária, extrativismo vegetal e mineral, são responsáveis por quase um quarto das emissões de gases de efeito estufa. Ainda, a utilização da superfície terrestre não coberta por gelo para fins de extração de madeira em florestas manejadas representa 20% do uso total.

Dessa forma, é possível observar a influência do ser humano nas constantes mudanças no solo e seu impacto socioeconômico. As diversas formas de uso da terra e a crescente necessidade de melhoria da sua eficiência são respostas à evolução natural e às atividades do homem. A detecção dessas modificações na paisagem permite a compreensão do espaço como um sistema ambiental, físico e socioeconômico. Essa compreensão pode servir como instrumento de apoio à tomada de decisão, mostrando-se urgente em frente as demandas sociais e geracionais.

A detecção de mudanças em sensoriamento remoto é a ação de identificar alterações em uma superfície terrestre através do processamento de imagens de uma cena capturada em datas diferentes. Um sensor remoto produz imagens digitais, que são uma função bidimensional $f(x,y)$, o qual x e y são coordenadas espaciais e o valor de f para algum ponto representa a intensidade da imagem no par de coordenadas.

Os problemas que dependem de classificação de imagens estão dentro da área de Reconhecimento de Padrões, que se objetiva de identificar, descrever e classificar padrões em imagens. Os seres humanos são ótimos em reconhecer padrões em diversas situações. Apesar de não se ter conhecimento total deste processo, busca-se aperfeiçoar o reconhecimento realizado pelo computador à habilidade do homem.

No mundo contemporâneo, o reconhecimento de padrões é essencial. Dada a crescente necessidade de sistemas automáticos que lidem com a vasta quantidade de dados disponíveis. Essa área está associada ao Processamento de Imagens e ao sensoriamento remoto, além de outras áreas como Aprendizado de Máquina, Estatística Aplicada etc.

1.2 Definição do problema

As mudanças no uso do solo são causadas e acentuadas pelo interesse das atividades socioeconômicas, como os processos de urbanização e industrialização, que utilizam recursos naturais e geram impactos sobre o meio ambiente.

As técnicas de análise de dados multidimensionais recebem o nome de processamento digital de imagens, onde a entrada e a saída do processo são imagens. Essas técnicas podem ser aplicadas no desenvolvimento de produtos, que podem inclusive ser submetidos a outros processamentos.

O uso de imagens multiespectrais registradas por satélites é uma técnica para a extração dos dados presentes nela e são destinados à várias aplicações. A obtenção das informações espectrais registradas pelos sistemas nas diferentes partes do espectro eletromagnético, visando a identificação e discriminação dos objetos de interesse, depende da qualidade da representação dos dados contidos nas imagens.

As técnicas de processamento digital de imagens para detecção de mudança no uso do solo, permitem analisar uma cena nas várias localidades do espectro eletromagnético. Essas técnicas são essenciais na compreensão das dinâmicas dos processos de mudanças no uso e na cobertura da terra. Tais resultados podem servir como subsídio, como por exemplo, ao poder público e privado para o gerenciamento, planejamento e a execução de projetos no âmbito global e nacional.

Por fim, salienta-se que este Trabalho de Graduação atende aos critérios estabelecidos na disciplina de Engenharia de Software III, que compreendem: organização de repositório do código em um formato especificado, desenvolvimento de um sistema para a internet com persistência de dados e automatizar aplicações de métodos de comparação de imagens.

1.3 Objetivo

Visando fornecer uma ferramenta para a detecção automatizada da mudança do uso do solo, neste trabalho foram utilizadas técnicas de processamento de imagens.

O processo de detecção de alterações da cobertura e uso do solo consiste em localizar e qualificar as transformações que ocorreram na área de interesse através da avaliação das respostas espectrais registradas nos pares de pixels correspondentes da imagem, que possibilita a distinção entre a mudança e a não alteração de uma localidade.

Assim, o objetivo deste trabalho consiste em desenvolver um software para automatizar o processo de detecção de mudança no uso da terra, buscando facilitar o processamento e análise de identificação de alteração na cobertura e no solo. Dentre as funcionalidades estão:

- Filtragem espacial
- Registro
- Comparação ponto a ponto de imagens.

2 DESENVOLVIMENTO

Este capítulo apresenta detalhes sobre o desenvolvimento da aplicação.

2.1 Fundamentação básica de PDI

A área de Processamento Digital de Imagens abrange operações realizadas sobre imagens matriciais e que resultam em novas imagens. Já a área de Visão Computacional inclui operações de alto nível sobre imagens e que resultam em significação ou tomada de decisão similar à que ocorre no processo cognitivo humano. A aplicação e escolha dessas técnicas depende dos objetivos a serem atingidos pelo sistema.

Uma imagem digital é, tipicamente, uma representação bidimensional e planar de objetos em uma matriz, sendo cada elemento denominado pixel, e corresponde a uma determinada intensidade da luz refletida pelo objeto. O pixel é o menor elemento de uma imagem, seu valor depende da resolução espacial do sensor usado na cena. Do ponto de vista físico, a cor se refere à luz, que percorre o espaço, ora se comportando com onda, ora como partícula.

Assim, matematicamente uma imagem é uma função da forma:

$$I = f(x,y)$$

sendo I a intensidade e x e y as coordenadas do plano. Em uma imagem digital I , x e y assumem valores finitos e discretos.

O processo de digitalização de uma imagem envolve a discretização dos elementos contidos nesta. Esta etapa é realizada para a aplicar o plano da imagem sob forma de uma malha regular finita, cujos elementos têm um tamanho constante, os pixels. Para realizar a discretização da imagem, aplica-se funções matemáticas, que quando aplicadas no eixo do domínio é chamada de Amostragem, quando no contradomínio é chamada de Quantização.

A representação matricial considera que o espaço pode ser observado como uma superfície plana, o qual cada célula está associada a uma porção da superfície. A resolução de

uma imagem é dada pela correlação entre o tamanho da célula e a área por ela coberta no terreno, assim, basta saber a razão entre o número de pixels e o tamanho real da imagem.

Define-se estrutura geométrica matricial como um conjunto de espaços localizados em coordenadas adjacentes, implementadas como uma matriz de dimensão 2. Cada célula é referenciada por índices de linha e coluna e contém um número representando o valor do atributo mapeado. O valor de cada pixel está limitado a um intervalo numérico, como por exemplo de 0 a 255 para imagens em 8 bits, ou valores associados a classes em caso de imagens temáticas.

- Para a aquisição da imagem pode-se utilizar sistemas de captura, para este estudo utilizou-se imagens de satélites. Os satélites de observação terrestre possuem sensores capazes de produzir imagens da superfície da Terra em várias bandas simultâneas. Após este processo, aplica-se métodos de realce e filtragem para facilitar as demais etapas de processamento da imagem. Exemplos de filtragem e realce incluem a limiarização e a aplicação de filtros derivativos.
- A divisão de uma imagem em termos de partes constituintes é chamada de segmentação, que nada mais é do que agrupamento de pixels em regiões. Feita a segmentação pode-se aplicar técnicas para classificação e reconhecimento de objetos contidos na imagem.
- Para comparar duas imagens e, conseqüentemente, haver a possibilidade de detecção de mudanças, é necessária a etapa de registro. O registro, por sua vez, é baseado na identificação de padrões ou elementos característicos de uma certa imagem. Na operação de registro busca-se alinhar as imagens que serão comparadas delimitando-se uma área de comparação. Os algoritmos que detectam tais padrões com frequência fazem uso de medidas de erro de casamento do padrão que é utilizado como referência em ambas as imagens (esses padrões contém os chamados pontos de controle – tie points).

2.1 Tecnologias Utilizadas

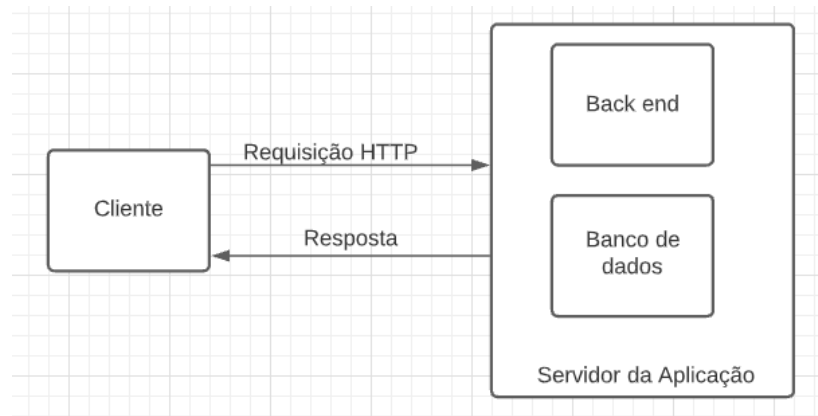
O software desenvolvido neste trabalho é um sistema para internet com persistência de dados. Para atingir este objetivo, no front end da aplicação utilizou-se a linguagem de marcação HTML (HyperText Markup Language), a linguagem de estilização CSS (Cascading Style Sheets) e a linguagem de programação JavaScript.

Para o back end utilizou-se o framework Flask que utiliza a linguagem de programação Python para o desenvolvimento das funcionalidades da aplicação e o banco de dados não relacional MongoDB para armazenar as imagens e o resultado da comparação.

2.2 Arquitetura da Aplicação

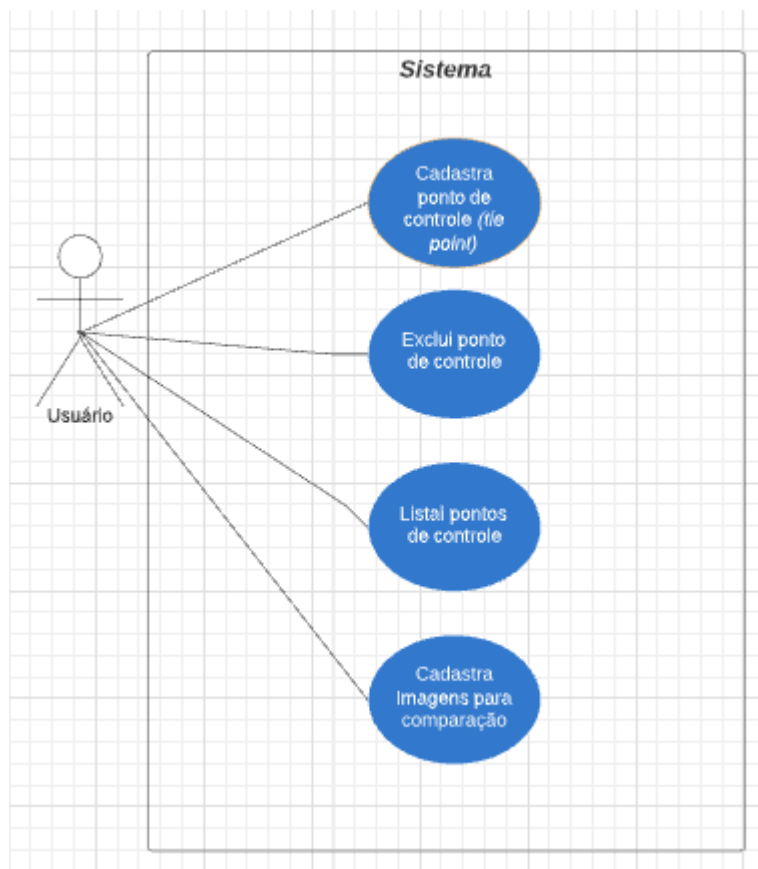
A figura 1 ilustra o funcionamento em alto nível da aplicação.

Figura 1. Diagrama de Implantação alto nível.



A figura 2 apresenta o diagrama de caso de uso do sistema, para demonstrar as formas que o usuário pode interagir com o sistema.

Figura 2. Diagrama de Caso de Uso



2.3 Descrição Geral das Operações do Sistema

A partir da imagem matricial é possível fazer operações para extrair dados e informações sobre uma área de específica. Para a detecção de mudança no uso e cobertura da terra, busca-se identificar as possíveis alterações em duas imagens.

O usuário pode submeter as seguintes entradas no sistema:

- *Template*: usado para o registro das imagens da comparação;
- Imagens multitemporais para a detecção de alteração da área de interesse.

No *back end* visualiza-se os módulos de Escala de Cinza (*Gray Scale*), *Thresholding* (técnica de limiarização, filtragem espacial), Registro das imagens (*Template Matching*) e Detecção de Mudança (*Change Detection*).

Neste processo, define-se um ponto de interesse e imagens multitemporais para realizar a comparação e gerar uma nova imagem que represente a diferença detectada. Dessa forma, o sistema desenvolvido realiza as operações de identificação de modificações nas imagens, este recebe 3 imagens como entrada. O primeiro parâmetro é a imagem da área em que se busca realizar a análise, a referência, uma pequena cobertura que deve inclusa nas imagens a serem processadas. A seguir o usuário seleciona as cenas em períodos distintos, o qual devem abranger a imagem de referência.

Após, um algoritmo para tornar as imagens temporais em escala de cinza é aplicado, a fim de tornar a imagem binária, este processo é feito pela ponderação da soma dos valores dos canais do pixel, o RGB (red, green, blue). Assim, realizar a médias das intensidades em cada faixa de frequência para transformá-la em monocromática. Ainda, na segmentação definiu-se níveis para alterar a cor do pixel, para cores abaixo de 0.3 torna-se preto e acima branco.

Em seguida, é realizado a etapa de identificação da área de interesse nas imagens da comparação, em um processo chamado de Casamento de Modelo (*Template Matching*). Para esta operação, assume-se que não há distorções nas imagens. O cálculo com base no número de linhas e colunas da imagem matricial retorna as coordenadas no plano cartesiano do deslocamento, que é usado para alinhá-las.

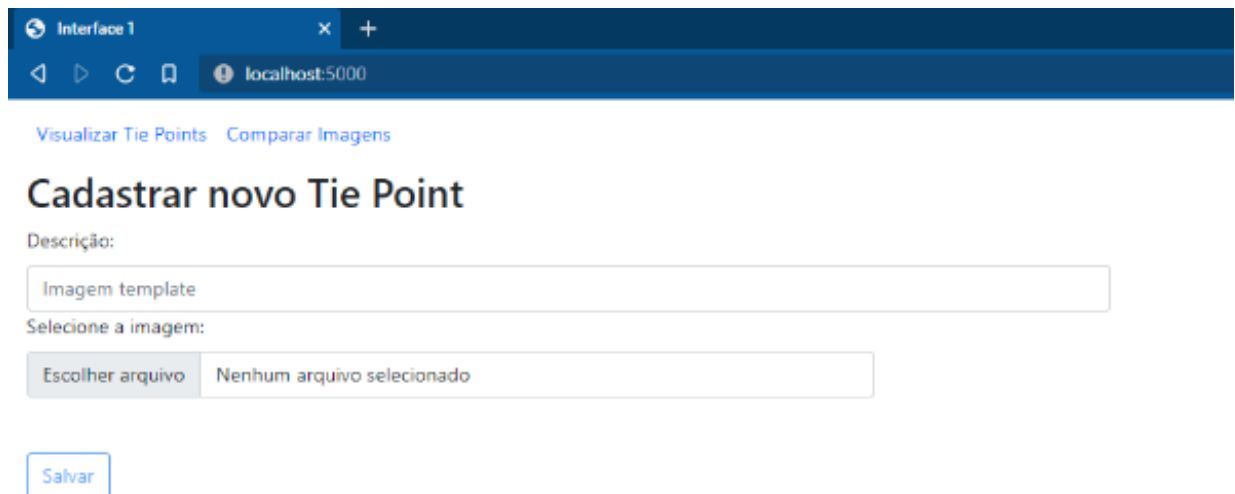
Por fim, com o alinhamento das imagens com a área de referência, realiza-se a comparação pixel a pixel das duas imagens temporais, destacando as diferenças entre os pixels e gerando uma nova imagem binária a partir disso. O resultado da comparação é uma imagem preto e branco referenciada em um plano, cujo preto destaca a área modificada identificada.

2.3.1 Aplicação web

A seguir a apresentação do produto desenvolvido, um sistema para *internet*. Essa aplicação resolve o problema de detecção de mudança em uma superfície terrestre de forma automatizada.

A figura 6 representa a interface de usuário para a submissão de imagem da área a ser comparada (Tie Point). Tal imagem é salva em um banco de dados e posteriormente usada na comparação.

Figura 6. Interface para cadastro do Tie Point, a área de interesse a ser identificada nas imagens para comparação.



The screenshot shows a web browser window with the title 'Interface 1' and the address bar displaying 'localhost:5000'. The page has a dark blue header with navigation links 'Visualizar Tie Points' and 'Comparar Imagens'. The main heading is 'Cadastrar novo Tie Point'. Below this, there is a 'Descrição:' label followed by a text input field containing 'Imagem template'. Underneath is a 'Selecione a imagem:' label followed by a file selection interface. This interface includes a button labeled 'Escolher arquivo' and a text area showing 'Nenhum arquivo selecionado'. At the bottom of the form is a blue 'Salvar' button.

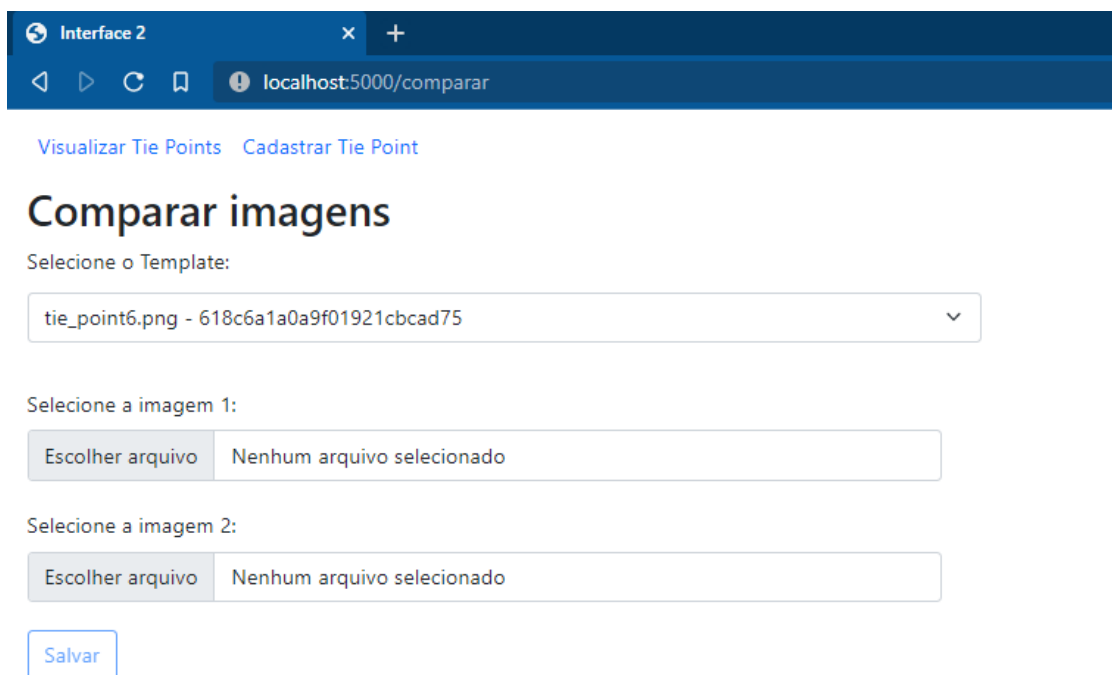
A figura 7 apresenta a interface de usuário para listar, visualizar e excluir um tie point cadastrado. Nesta tela é exibido um link para visualizar a figura, uma descrição cadastrada previamente e botão para ação de excluir a imagem.

Figura 7. Interface para visualização e alteração de um Tie Point cadastrado.



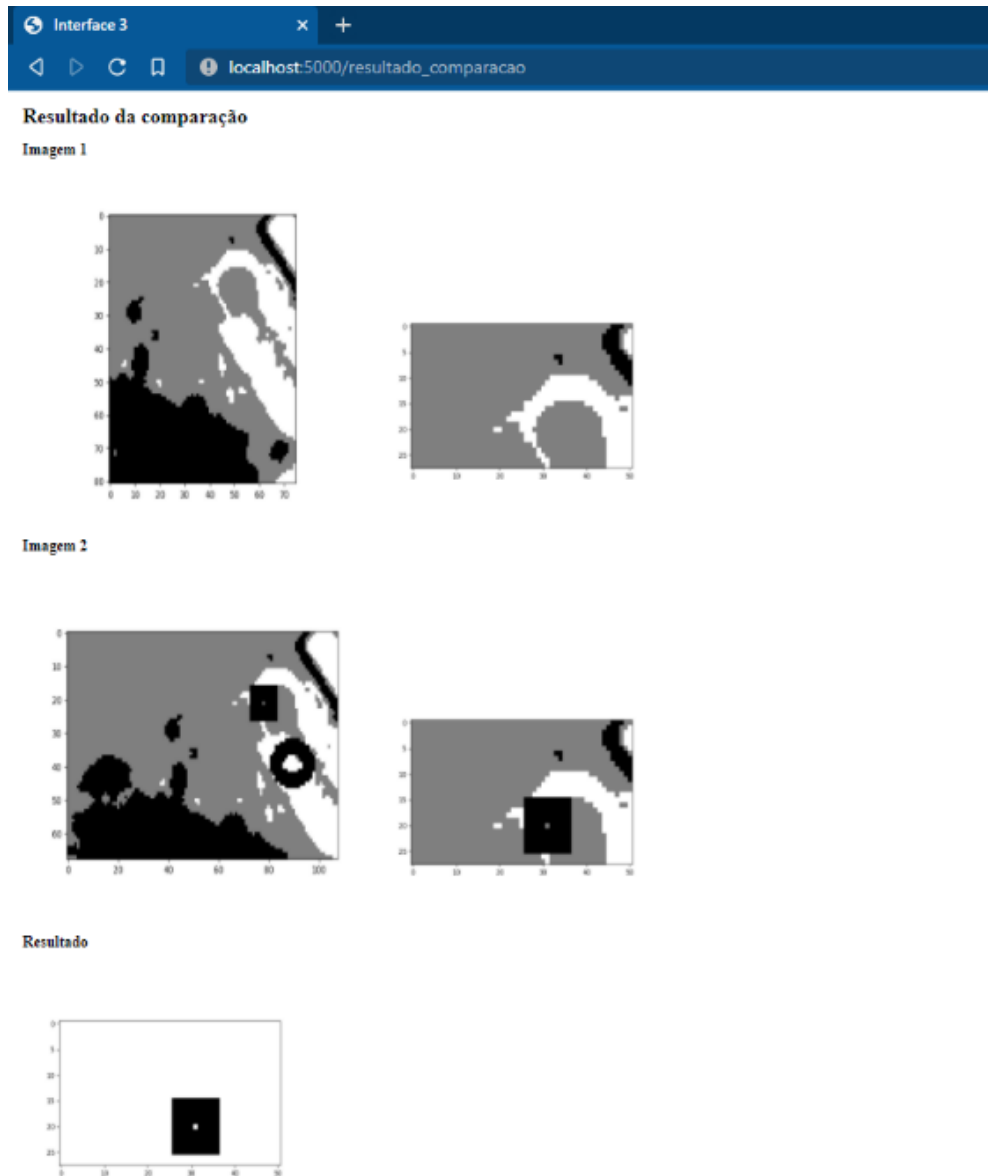
A figura 8 apresenta a interface principal do sistema web, o qual o usuário pode enviar as imagens para processamento da detecção de alteração. O usuário seleciona o template da área que será identificada nas duas imagens de comparação e as imagens 1 e 2 que serão aplicados os métodos de identificação de mudança na cobertura do solo.

Figura 8. Interface para selecionar imagens da série temporal e Tie Point para a comparação.



Como última etapa, o programa exibe uma nova imagem representada no plano cartesiano, com a extração das áreas comparáveis e a marcação dos locais da imagem com as alterações encontradas, processo representado na figura 9.

Figura 9. Saída do programa com resultado da comparação.



2.3.2 Detalhamento

Segue uma breve explicação sobre a aplicação web construída e detalhamento dos algoritmos de processamento de imagens desenvolvidos.

Figura 10. Função desenvolvida em Python para transformação da imagem original para escala de cinza.

```
1. from .utils import qtd_row_col
2.
3. def gray_scale(img):
4.     n_rows, n_cols = qtd_row_col(img)
5.
6.     for row in range(n_rows):
7.         for col in range(n_cols):
8.             v = img[row][col]
9.             new_gray_level = (v[0] + v[1] + v[2]) / 3
10.            for index in range(3):
11.                img[row][col][index] = new_gray_level
12.    return new_gray_level
```

A figura 10 ilustra o código desenvolvido para converter uma imagem colorida para níveis de cinzento:

- A função definida na linha 3 transforma a imagem matricial em escala de cinza, monocromática;
- Na linha 5 e 6 é feita uma iteração sobre a matriz (linhas e colunas) e, então calcula-se, na linha 8, a média dos canais de cores RGB. A média atribui pesos iguais aos canais para facilitar a tarefa de comparação das imagens, como é feito na linha 10;
- A média calculada é por fim reatribuída na matriz. Após, na linha 11, retorna-se a matriz da imagem modificada.

Figura 11. Função desenvolvida em *Python* para aplicação filtragem espacial na imagem em escala de cinza.

```
1. from .utils import qtd_row_col
2. BRANCO = 1
3. PRETO = 0
4. def thresholding(img):
5.     n_rows, n_cols = qtd_row_col(img)
6.     level_1 = 0.3
7.     values = range(3)
8.     for row in range(n_rows):
9.         for col in range(n_cols):
10.            if img[row][col][0] < level_1:
11.                for index in values:
12.                    img[row][col][index] = PRETO
13.            else:
14.                for index in values:
15.                    img[row][col][index] = BRANCO
16.     return img
```

A figura 11 apresenta o código para aplicação do método de filtragem espacial, o qual se classifica os pixels da imagem:

- Na linha 6 define-se o tom de corte para a filtragem dos elementos.
- Nas linhas 8 e 9 é feito uma iteração sobre as linhas e colunas da matriz para fazer a comparação dos valores dos pixels;
- Após a comparação dos pixels na linha 10 e 13, é feito uma iteração sobre os canais do pixel e atribui-se um novo valor, preto ou branco. Aplicou-se a técnica com 1 único limiar de separação (binarização), para a separação entre as classes “claro” e “escuro”;
- Após, na linha 16 é retornada a matriz da imagem filtrada.

Figura 12. Função desenvolvida em Python para aplicação de segmentação. O código da imagem é baseado no código fornecido pelo Professor Fabrício Carvalho.

```
1. def template_matching(src_img, template_img):
2.     src_xy = {'rows': len(src_img), 'cols': len(src_img[0])}
3.     template_xy = {'rows': len(template_img), 'cols':
4.         len(template_img[0])}
5.     max_x_disp = src_xy['rows'] - template_xy['rows']
6.     max_y_disp = src_xy['cols'] - template_xy['cols']
7.
8.     sum_abs_diff = [] # será uma array bidimensional
9.     for row in range(max_x_disp):
10.        sum_abs_diff.append([])
11.        for col in range(max_y_disp):
12.            sum_abs_diff[row].append(0) # erro 0, ainda não calculado
13.            for x in range(template_xy['rows']):
14.                for y in range(template_xy['cols']):
15.                    sum_abs_diff[row][col] += abs(template_img[x][y][0]
16. - src_img[x + row][y + col][0])
17.     min_error = sum_abs_diff[0][0] # SAD: sum of average difference
18.     delta_x = 0
19.     delta_y = 0
20.
21.     for row in range(max_x_disp):
22.         for col in range(max_y_disp):
23.             if sum_abs_diff[row][col] < min_error:
24.                 # nem sempre o sad vai dar tem que ser o menor
25.                 min_error = sum_abs_diff[row][col]
26.                 delta_x, delta_y = row, col
27.
28.     return (delta_x, delta_y)
```

A figura 12 mostra o código desenvolvido para identificação do *template* na imagem. Este processo ocorre com ambas as imagens da comparação:

- O código da função *template_matching*, na linha 1, calcula o delta entre a imagem da área de interesse e a imagem multitemporal;
- Define-se nas linhas 4 e 5 a máxima área da comparação, para isto calcula-se a diferença das linhas e colunas na imagem template, área de interesse, com a imagem da comparação e obtém-se o deslocamento máximo;
- Na linha 7 e 9 itera-se sobre o display máximo que a área está contida, e na linha 8 cria-se uma array bidimensional;
- Com base nos valores das linhas e colunas da imagem template, é feito um loop percorrendo a imagem maior na ordem esquerda para a direita e de cima para baixo, como na linha 13. Ainda, calcula-se a soma da diferença absoluta da comparação com

o template, atribuindo o valor ao array `sum_abs_diff`. O resultado obtido identifica em qual posição há maiores chances de o padrão ter sido identificado (casamento);

- Na linha 17 e 18 itera-se sobre deslocamento máximo no eixo x e y, linha e coluna, e na linha 19 verifica-se qual coordenada tem o menor erro absoluto;
- Por fim, tem-se o resultado com as coordenadas X e Y com o menor erro de cálculo. O resultado obtido identifica em qual posição há maiores chances de o padrão ter sido identificado (casamento).

Figura 13. Função desenvolvida em Python para aplicação de segmentação

```
1. import numpy as np
2. BRANCO = 1
3. PRETO = 0
4. def change_detection(img1, img2):
5.     img1_rows = len(img1)
6.     img1_cols = len(img1[0])
7.     new_image = np.zeros((img1_rows, img1_cols, 3), dtype=np.float32)
8.     for col in range(img1_cols):
9.         for row in range(img1_rows):
10.            for index in range(3):
11.                if img1[row][col][index] == img2[row][col][index]:
12.                    new_image[row][col][index] = BRANCO
13.                else:
14.                    new_image[row][col][index] = PRETO
15.     return new_image
```

A figura 13 mostra o código para aplicação do algoritmo de detecção de mudança em imagens:

- No método *change_detection* aplica-se operações para detectar as alterações nas imagens;
- Na linha 7 cria-se a imagem que será alterada para destacar a mudança;
- Itera-se sobre a matriz, nas linhas 8 e 9, e verifica-se os valores dos pixels correspondentes em ambas as imagens, na linha 11 e 13;
- Caso os valores dos pixels das imagens sejam iguais, na linha 12 altera-se o ponto para cor branca, simbolizando que são pontos que permaneceram inalterados; o contrário, na linha 14, ocorre se forem pixels diferentes, adiciona-se preto;
- Na linha 15 é retorno a matriz da imagem binária da alteração detectada.

3 RESULTADOS E DISCUSSÃO

A partir dos resultados obtidos, evidencia-se que a automatização do processo de detecção de alteração em imagens de uso do solo simplifica a detecção e visualização da mudança em uma superfície terrestre. Os resultados apontam viabilidade da proposta.

Cabe ressaltar que os resultados dependem da qualidade da imagem obtida, em dois principais aspectos: a resolução da imagem e o seu tamanho.

Ademais, dentro das limitações encontradas destaca-se que o programa não considera a possível rotação ou distorções de uma imagem para a comparação. O que se mostra de grande relevância para aplicações que utilizem diferentes fontes de imagens para a detecção de alteração.

3.1 Sugestões para trabalhos futuros

Como citado anteriormente, o sistema desenvolvido não contempla imagens com distorções geométricas. Essas distorções podem ser inerentes a plataforma e o instrumento utilizado para a captura da imagem real, ao modelo da Terra como os relevos, rotação e esfericidade.

Em função das limitações de tempo para a conclusão desse trabalho, sugere-se para pesquisas futuras a adição de métodos de identificação e correção de deformidade em uma das imagens de entradas para comparação. É necessário que se realize pré-processamento para a correção geométrica de imagens, o que permitiria ao usuário da aplicação maior flexibilidade quanto a fonte da imagem.