

## TECHNICAL REPORT

Aluno: LUIS SAVIO GOMES ROSA

### 1. Introdução

O propósito deste projeto consiste em categorizar imagens conforme a presença de fogo, fumaça ou a ausência de ambos. As imagens foram examinadas nos modelos de cores RGB, HSV e YCrCb. A avaliação dos modelos ocorreu por meio de três abordagens distintas: a inclusão dos pixels das imagens, a inclusão do histograma das imagens, e a utilização de métricas relacionadas à dispersão e centralidade tanto dos pixels quanto dos histogramas, sendo elas média, mediana e desvio padrão. Durante a análise, constatou-se que o modelo de Regressão Logística, alimentado com as métricas dos canais das imagens em formato RGB, alcançou uma precisão superior.

### 2. Dataset

*O objetivo deste projeto foi desenvolver um modelo de machine learning para classificar imagens em quatro categorias: "water," "green\_area," "desert," e "cloudy." O dataset utilizado contém um número desigual de imagens por classe, com 1500 imagens para cada uma das classes "water," "green\_area," e "cloudy," e 1131 imagens para a classe "desert." Este desbalanceamento pode impactar negativamente o desempenho do modelo de classificação. O foco principal foi abordar esse desbalanceamento e avaliar o impacto da amostragem em uma quantidade igual de 500 imagens por classe.*

### 3. Descrição do Dataset

- *Número de Classes: 4 (water, green\_area, desert, cloudy)*
- *Número de Imagens por Classe: Aproximadamente 1500, mas o código carrega uma amostra de 500 imagens de cada classe para a construção do modelo.*
- *Dimensões das Imagens: Redimensionadas para 128x128 pixels.*
- *Formato das Imagens: Todas as imagens são convertidas para o espaço de cores RGB.*

#### 2.1. eitura do Dataset:

*O dataset foi carregado em uma estrutura de dados com imagens (X) e rótulos correspondentes (y). Cada imagem foi lida e convertida para um array NumPy após o redimensionamento para 128x128 pixels.*

- *Separação de Treino e Teste:*

*Os dados foram divididos em conjuntos de treino e teste utilizando uma divisão de 80% para treino e 20% para teste, com estratificação para manter a proporção das classes.*

*Mineração dos Atributos Necessários:*

*A extração de atributos foi realizada utilizando histogramas de cores das imagens. Cada imagem foi representada por um vetor de características baseado na distribuição de cores.*

- *Armazenamento dos Atributos em CSV:*

*Os atributos extraídos das imagens de treino e teste foram salvos em arquivos CSV (train\_data.csv e test\_data.csv) para posterior análise e treinamento do modelo.*

- *Classificação dos Atributos com o Modelo Escolhido:*

*O modelo SVM com kernel linear foi selecionado para a classificação. A escolha deste modelo foi baseada em sua capacidade de lidar com dados de alta dimensionalidade e sua eficácia em problemas de classificação binária e multi-classe. O parâmetro class\_weight='balanced' foi utilizado para compensar o desbalanceamento das classes.*

- *Geração de Métricas de Desempenho do Algoritmo:*

*As métricas de desempenho foram avaliadas utilizando o relatório de classificação e a acurácia. As métricas incluem precisão, recall e F1-score para cada classe, além de médias ponderadas e macro.*

## **2. Observações**

- *Desbalanceamento das Classes:*

*As classes "water," "green\_area," e "cloudy" possuem um número maior de imagens comparado à classe "desert," que tem 1131 imagens. Isso pode causar um viés no modelo, favorecendo as classes mais representadas.*

- *Amostragem:*

*Para lidar com o desbalanceamento, foi realizada uma amostragem de 500 imagens de cada classe. Essa abordagem buscou criar um conjunto de dados mais equilibrado para treinamento e avaliação.*

- **Erros de Carregamento:**

*Durante o carregamento das imagens, alguns erros foram encontrados e tratados. Essas falhas não impactaram significativamente o resultado final, mas foram importantes para garantir a integridade dos dados processados.*

### 3. Resultados e Discussão

<b>Classe</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<i>Cloudy</i>	<i>0.87</i>	<i>0.84</i>	<i>0.85</i>	<i>100</i>
<i>Desert</i>	<i>0.89</i>	<i>0.89</i>	<i>0.89</i>	<i>100</i>
<i>Green Area</i>	<i>0.75</i>	<i>0.94</i>	<i>0.83</i>	<i>100</i>
<i>Water</i>	<i>0.88</i>	<i>0.68</i>	<i>0.77</i>	<i>100</i>
<b>Acurácia</b>			<b>0.84</b>	<b>400</b>
<b>Macro Avg</b>	<i>0.85</i>	<i>0.84</i>	<i>0.84</i>	<i>400</i>
<b>Média Ponderada</b>	<i>0.85</i>	<i>0.84</i>	<i>0.84</i>	<i>400</i>

- **Amostragem e Pré-processamento:**

*A amostragem uniformizada garantiu que cada classe fosse igualmente representada no conjunto de dados de treino e teste, mitigando o impacto do desbalanceamento. O pré-processamento incluiu o redimensionamento das imagens para 128x128 pixels e a conversão para arrays NumPy, seguido da extração de histogramas de cores como características.*

- **Treinamento do Modelo:**

*Um modelo SVM com kernel linear foi treinado utilizando os dados amostrados. O parâmetro `class_weight='balanced'` foi configurado para ajudar a lidar com o desbalanceamento das classes. A acurácia do modelo foi de 83.75%, indicando um desempenho geral razoável.*

- *Análise das Métricas de Desempenho:*

*As métricas variaram entre as classes. A classe "desert" apresentou valores mais baixos de precisão e recall em comparação com as outras classes, refletindo o desafio de aprender a partir de um número menor de exemplos. Apesar da acurácia geral, o modelo teve dificuldades específicas com a classe "desert," o que pode ser atribuído ao menor número de amostras dessa classe.*

- *Impacto do Desbalanceamento:*

*O desbalanceamento de imagens impactou o desempenho do modelo, especialmente para a classe com menos imagens. A amostragem ajudou a equilibrar a representação das classes, mas ainda assim houve variação nas métricas de desempenho.*

#### **4. Conclusões**

*A análise do modelo de classificação de imagens demonstrou que, embora a amostragem de 500 imagens de cada classe tenha ajudado a criar um conjunto de dados mais equilibrado, o desbalanceamento original das classes ainda teve impacto nas métricas de desempenho. A acurácia geral do modelo foi de 83.75%, refletindo uma performance sólida, mas com variações significativas entre as classes.*

*A classe "desert," com menos imagens no dataset original, apresentou menor precisão e recall, o que destaca o desafio de treinar modelos eficazes em datasets desbalanceados. A amostragem uniformizada foi eficaz para melhorar a representação das classes durante o treinamento, mas ainda há espaço para melhorar a performance do modelo, especialmente para as classes com menor número de exemplos.*

*Os resultados indicam que, enquanto a estratégia de amostragem ajudou a mitigar o impacto do desbalanceamento, ajustes adicionais são necessários para alcançar uma melhor precisão e consistência nas previsões para todas as classes. O modelo mostrou*

*um bom desempenho geral, mas a identificação precisa e equitativa de todas as classes continua sendo um desafio importante.*

- **Próximos Passos**
- *Coleta de Mais Dados:*

*Objetivo: Obter mais imagens para a classe "desert" para melhorar a representatividade dessa classe no dataset.*

*Benefícios: Reduzirá o impacto do desbalanceamento e poderá melhorar a precisão e recall da classe "desert."*

- *Aprimoramento do Modelo:*

*Experimentação com Diferentes Modelos: Testar outros algoritmos de machine learning, como Random Forest e Redes Neurais, para comparar o desempenho.*

*Ajuste de Hiperparâmetros: Realizar uma busca de hiperparâmetros para otimizar o desempenho do modelo SVM ou outros algoritmos escolhidos.*

- *Análise Adicional e Validação Cruzada:*

*Validação Cruzada: Implementar validação cruzada para obter uma avaliação mais robusta do modelo e garantir que o desempenho seja consistente em diferentes subconjuntos dos dados.*

*Análise de Erros: Realizar uma análise detalhada dos erros de classificação para entender melhor onde o modelo falha e como isso pode ser corrigido.*