Nome: Raphael Baptista da Silva Fontenelle

Relatório de Segmentação Semântica de Imagens

Objetivo

O objetivo deste projeto foi desenvolver um modelo de segmentação semântica capaz de classificar pixels em uma imagem de acordo com classes pré-determinadas. O modelo foi treinado e validado em um conjunto de dados específico com a intenção de automatizar e acelerar o processo de anotação de imagens para aplicações de mapeamento urbano e análise geográfica.

Arquitetura do Modelo

A arquitetura do modelo consiste em uma Rede Neural Convolucional (CNN) baseada no modelo densenet161, que é conhecido por sua eficácia em tarefas de segmentação semântica. A arquitetura DenseNet-161, que é um tipo de rede neural convolucional (CNN), DenseNet, ou Dense Convolutional Network, é uma arquitetura de CNN conhecida por sua eficiência em conectar cada camada a todas as outras camadas de uma maneira feed-forward. Essa conexão densa facilita o reaproveitamento de recursos na rede, o que pode levar a uma maior eficiência em termos de desempenho e uso de memória.

A PSPNet, ou Pyramid Scene Parsing Network, é uma arquitetura avançada de rede neural convolucional (CNN) projetada especificamente para tarefas de segmentação semântica em imagens. O objetivo principal da PSPNet é identificar a categoria a que cada pixel de uma imagem pertence, permitindo assim a segmentação detalhada de cenas complexas. Essa capacidade faz da PSPNet uma ferramenta valiosa para diversas aplicações, como visão computacional em veículos autônomos, análise de imagens médicas e monitoramento ambiental por imagens de satélite.

Eu utilizei uma taxa de overlape de 0.5.

Componentes chave da PSPNet:

1. **Backbone Convolutional Network**: No coração da PSPNet está uma rede convolucional profunda que atua como a espinha dorsal (backbone) para a extração de características. Essa rede pode ser uma ResNet ou

qualquer outra arquitetura CNN eficaz em extrair recursos ricos e discriminativos das imagens.
2. **Pyramid Pooling Module**: Após a extração de características pelo backbone, a PSPNet emprega um módulo inovador de agrupamento piramidal (pyramid pooling module). Este módulo divide o mapa de características final em diferentes regiões, aplicando operações de pooling (agrupamento) em escalas variadas. Ao fazer isso, a rede consegue capturar contextos em diferentes níveis, desde o nível global até detalhes locais finos, garantindo que informações contextuais essenciais não sejam perdidas.
3. **Estratégia de Upsampling e Concatenação**: Após o pooling piramidal, os mapas de características resultantes são upsampled (aumentados) para o tamanho original da imagem e concatenados. Essa abordagem assegura que as características em diferentes escalas sejam efetivamente integradas, otimizando a capacidade da rede de interpretar e segmentar a cena com precisão.
4. **Classificação de Pixels e Decodificação**: Finalmente, um classificador é aplicado aos mapas de características concatenados para predizer a categoria de cada pixel. Isso é geralmente seguido por uma camada de decodificação, que ajusta as previsões para mais perto da segmentação final desejada.

Benefícios da PSPNet:

- **Alto Desempenho em Segmentação Semântica**: Graças à sua habilidade de capturar informações contextuais em múltiplas escalas, a PSPNet demonstra excelentes resultados em segmentação semântica, superando muitas outras arquiteturas em benchmarks padrão.
- **Versatilidade**: Pode ser aplicada a uma ampla gama de tarefas de visão computacional, adaptando-se bem a diferentes tipos de cenas e desafios de segmentação.
- **Eficiência em Capturar Contexto**: O módulo de pooling piramidal é eficaz em reunir informações contextuais essenciais, o que é crucial para entender cenas complexas e realizar segmentações precisas.

Protocolo Experimental

O treinamento do modelo foi realizado seguindo estas etapas:

- **Preparação dos Dados**: As imagens foram pré-processadas conforme descrito abaixo:

- **Extração de patches de imagens**: As imagens são inicialmente processadas para extrair pequenas seções ou "patches". Essa técnica é útil para focar em partes específicas da imagem e facilitar o processamento por redes neurais, especialmente em casos de imagens muito grandes.
- **Divisão em treino e validação**: Após a extração dos patches, o conjunto de dados é dividido em dois grupos: um para treinamento e outro para validação. Essa separação é essencial para treinar o modelo em um conjunto de dados e validar seu desempenho em outro, garantindo que o modelo generalize bem para novos dados.
- **Salvar patches de imagens na pasta do Colab como PNG ou NPY**: Os patches extraídos são salvos em uma pasta no Google Colab no formato PNG ou NPY. O formato PNG é usado para imagens e o NPY é um formato de arquivo que salva arrays de forma eficiente, comum no ecossistema Python para dados numéricos.
- **Dataloader carrega as amostras de treino e validação**: Um Dataloader é utilizado para carregar automaticamente os dados de treino e validação. Ele facilita o processo de iterar sobre os dados durante o treinamento do modelo, fornecendo um meio eficiente de alimentar os dados para a rede neural.
- **Definir o modelo usando segmentation_models_pytorch (SMP)**: O modelo de rede neural é definido usando a biblioteca `segmentation_models_pytorch`, que oferece uma variedade de modelos pré-configurados e otimizados para tarefas de segmentação de imagens. Essa biblioteca facilita a implementação e o uso de modelos de segmentação avançados.
- **Treinamento com parada antecipada (early stopping)**: Durante o treinamento, a técnica de parada antecipada é usada para interromper o treinamento se o desempenho do modelo na validação não melhorar após um determinado número de épocas. Isso ajuda a prevenir o overfitting, ou seja, quando o modelo se ajusta demais aos dados de treino e performa mal em dados novos.
- **Inferência na imagem de teste é feita patch-wise**: A inferência é realizada nas imagens de teste usando a abordagem patch-wise, ou seja, aplicando o modelo treinado a cada patch individual da imagem de teste. Essa técnica permite obter previsões detalhadas e locais para cada seção da imagem.

- **Configuração do Treinamento**: O modelo foi treinado usando a função de perda de Entropia Cruzada, otimizada pelo algoritmo Adam com uma taxa de aprendizagem inicial de 0.001. O treinamento foi conduzido por 50 épocas com early stopping implementado para prevenir sobreajuste.

- **Validação**: O modelo foi validado em um conjunto de dados separado para garantir a generalização do modelo para novos dados.
- **Avaliação**: O modelo foi avaliado com base na acurácia geral, F1-score por classe, F1-score médio e mIoU no conjunto de testes.

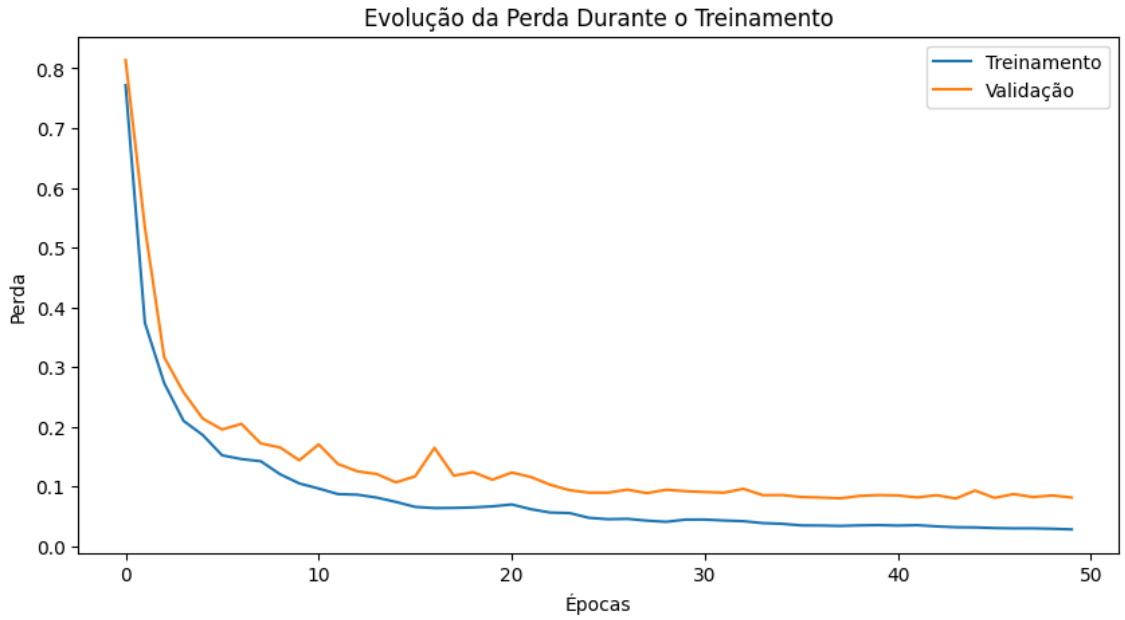
Resultados de Acurácia

O modelo alcançou os seguintes resultados no conjunto de testes:

- **Acurácia Geral**: 76.44%
- **F1-score por Classe**: [0.779, 0.889, 0.668, 0.777, 0.413]
- **F1-score Médio (Macro)**: 70.54%
- **mIoU**: 56.73%

Estes resultados indicam que o modelo possui uma capacidade substancial de distinguir entre as classes relevantes, embora ainda haja espaço para melhorias, especialmente nas classes com F1-scores mais baixos.

Evolução da Perda Durante o Treinamento

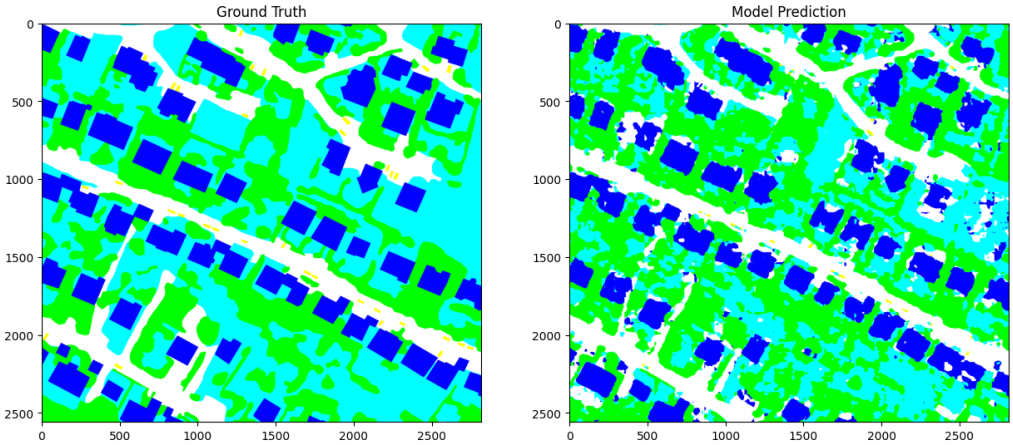


A figura acima ilustra a evolução das perdas de treino e validação ao longo das 50 épocas.

A figura ilustra a evolução da função de perda durante o processo de treinamento de uma rede neural, por um período de 50 épocas. Observa-se uma rápida diminuição na perda do conjunto de treinamento, refletindo a capacidade de aprendizado do modelo. Simultaneamente, a perda no conjunto

de validação diminui em um ritmo semelhante, indicando uma boa generalização. Após as primeiras 10 épocas, ambas as curvas tendem a se estabilizar, com a perda de validação mantendo-se próxima à perda de treinamento, o que sinaliza que o modelo alcançou uma fase de convergência, sem evidências de superajuste. Essa estabilidade na fase posterior das épocas é um indicativo de que o treinamento foi bem-sucedido e o modelo está apto a realizar previsões consistentes tanto nos dados de treinamento quanto nos de validação.

Resultado Visual da Segmentação



Conclusão

Os experimentos conduzidos demonstram que a arquitetura densenet161e a PSPNet é eficaz para a tarefa de segmentação semântica no conjunto de dados utilizado. As métricas obtidas fornecem um ponto de partida sólido para futuras iterações e melhorias do modelo.