



Gabrielle Brandenburg dos Anjos

**Segmentação Automática de Breakouts em
Perfis de Imagem com Aprendizado Profundo**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio.

Orientador: Prof. Marcelo Gattass

Rio de Janeiro
Março de 2023

Gabrielle Brandenburg dos Anjos

**Segmentação Automática de Breakouts em
Perfis de Imagem com Aprendizado Profundo**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Prof. Marcelo Gattass

Orientador

Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo

PUC-Rio

Dra. Cristina Nader Vasconcelos

Google Inc

Dr. Jan Jose Hurtado Jauregui

PUC-Rio

Dr. Rodrigo Surmas

CENPES/Petrobras

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

Gabrielle Brandenburg dos Anjos

Graduou-se em Engenharia da Computação pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), onde trabalha em projetos no grupo de Geofísica Computacional do Instituto Tecgraf.

Ficha Catalográfica

Brandenburg dos Anjos, Gabrielle

Segmentação Automática de Breakouts em Perfis de Imagem com Aprendizado Profundo / Gabrielle Brandenburg dos Anjos; orientador: Marcelo Gattass. – 2023.

51 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Informática – Teses. 2. Aprendizado Profundo. 3. Redes Neurais. 4. Segmentação. 5. Visão Computacional. I. Gattass, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Aos meus pais, irmãos e família
pelo apoio e encorajamento.

Agradecimentos

Aos meus colegas da equipe de pesquisa do grupo de Geofísica Computacional. Sem vocês o resultado deste trabalho não teria sido o mesmo. E aos demais colegas do grupo que acompanharam tudo do início ao fim.

Ao meu orientador Professor Marcelo Gattass, pela parceria desde a graduação. E aos demais colegas orientandos que enriqueceram este trabalho com sugestões e conhecimento.

Ao Bruno, por enfrentar comigo diariamente os altos e baixos dessa jornada. E também ao Nankin e Mipha. Obrigada por tudo.

À minha família e amigos por me apoiarem e acreditarem mais em mim do que eu mesma.

À minha psicóloga e psiquiatra por cuidarem da minha saúde mental durante este período.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Brandenburg dos Anjos, Gabrielle; Gattass, Marcelo. **Segmentação Automática de Breakouts em Perfis de Imagem com Aprendizado Profundo**. Rio de Janeiro, 2023. 51p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Breakouts são zonas colapsadas nas paredes de poços causadas por falhas de compressão. A identificação desses artefatos é fundamental para estimar a estabilidade das perfurações e para obter a orientação e magnitude da tensão horizontal máxima presente na formação rochosa. Tradicionalmente, os intérpretes caracterizam os *breakouts* manualmente em perfis de imagem, o que pode ser considerado uma tarefa muito demorada e trabalhosa por conta do tamanho massivo dos dados. Outros aspectos que dificultam a interpretação estão associados à complexidade das estruturas e a presença de diversos artefatos ruidosos nos dados de perfil. Sendo assim, métodos tradicionais de processamento de imagem tornam-se ineficientes para solucionar essa tarefa de detecção. Nos últimos anos, soluções baseadas em aprendizado profundo tem se tornado cada vez mais promissoras para problemas de visão computacional, tais como, detecção e segmentação de objetos em imagens. O presente trabalho tem como objetivo a classificação pixel a pixel das regiões de breakouts em dados de perfil de imagem. Para isso foi empregado a rede neural convolucional DC-UNet de forma supervisionada. Essa arquitetura é uma variação do modelo clássico U-Net, a qual é uma rede consagrada na segmentação de dados médicos. A metodologia proposta atingiu uma média de 72.3% de *F1-Score* e, em alguns casos, os resultados qualitativos mostraram-se melhores que a interpretação de referência. Após avaliação dos resultados junto a especialistas da área, o método pode ser considerado promissor na caracterização e segmentação automática de estruturas em perfis de imagem de poços.

Palavras-chave

Aprendizado Profundo; Redes Neurais; Segmentação; Visão Computacional.

Abstract

Brandenburg dos Anjos, Gabrielle; Gattass, Marcelo (Advisor). **Automatic Segmentation of Breakouts in Image Logs with Deep Learning**. Rio de Janeiro, 2023. 51p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Breakouts are collapsed zones on wellbore walls caused by compressive failure. Their identification is fundamental for estimating the borehole's stability and obtaining the direction and magnitude of the maximum horizontal stress in the rock formation. Traditionally, professional interpreters identify and characterize breakouts manually in image logs, which can be considered a very laborious and time-consuming task due to the massive size of the wellbore data. Other aspects that make the interpretation difficult are the complexity of the structures of interest and several noisy artifacts in the image log data. Therefore, more than traditional image processing methods are required to solve this detection task. In recent years, solutions based on deep learning have become increasingly promising for computer vision problems, such as object detection and image segmentation. This work explores using a convolutional neural network to create a pixel-by-pixel classification of the breakout regions in the image log data. The architecture model used in this work for the supervised training was the DC-UNet. This architecture is a variation of the classical U-Net, an acknowledged network for medical image segmentation. The proposed method reached an average F-Score of 72.3% and qualitative results with some prediction cases even better than ground truth. After evaluating the results, the work can be considered promising for automatically characterizing and segmenting borehole structures in well image logs.

Keywords

Deep Learning; Neural Networks; Segmentation; Computer Vision.

Sumário

1	Introdução	14
2	Trabalhos relacionados	16
3	Fundamentação Teórica	18
3.1	Estabilidade de poços	18
3.2	Formação de <i>breakouts</i>	18
3.3	Perfis de imagem	20
3.4	Redes Neurais Artificiais	22
3.5	Redes Neurais Convolucionais	23
3.6	Arquiteturas <i>Encoder-Decoder</i>	25
3.7	U-Net	26
4	Metodologia	30
4.1	Obtenção do dado rotulado	30
4.2	Pré-processamento	30
4.3	Estratégia para extração de amostras	32
4.4	Validação Cruzada	32
4.5	Métricas de avaliação	34
5	Experimentos e Discussão	36
5.1	Descrição dos testes	36
5.2	Experimento 1: Base 1 com filtro	38
5.3	Experimento 2: Base 1 sem filtro	39
5.4	Experimento 3: Base 2 sem filtro	40
5.5	Experimento 4: U-Net e MultiResUNet	43
6	Conclusão e trabalhos futuros	46
7	Referências bibliográficas	48
A	Lista de hiperparâmetros de treino	51

Lista de figuras

Figura 3.1	Falhas de <i>breakouts</i> ilustradas em 3 visões diferentes. Fonte: (ZOBACK, 2007)	19
(a)	Diâmetro do poço deformado em lados opostos. W_{BO} é o ângulo de abertura do <i>breakout</i> .	19
(b)	Esquema em 3D evidenciando a posição do desmoronamento na mesma direção da tensão horizontal mínima.	19
(c)	Nos perfis de imagem, os <i>breakouts</i> aparecem em pares, afastados de 180 graus.	19
Figura 3.2	Esquema ilustra processo de planificação do dado de perfil imagem. Fonte: (ASQUITH et al., 2004)	21
Figura 3.3	Perfis AMP e TT de uma seção de poço com cavidades de <i>breakout</i> . Os pontos em branco no perfil TT são as anomalias causadas pela falha na captura em regiões profundas.	22
Figura 3.4	Esquema de uma rede <i>Perceptron</i> .	22
Figura 3.5	Exemplo de uma arquitetura MLP com uma camada escondida. Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)	23
Figura 3.6	Exemplo de uma CNN básica para reconhecimento de dígitos (LeNet). Disponível em (LECUN et al., 1998).	24
Figura 3.7	Esquema de uma arquitetura <i>encoder-decoder</i> . Fonte: (MI-NAEE et al., 2022)	25
Figura 3.8	Arquitetura da U-Net. Fonte: (RONNEBERGER; FISCHER; BROX, 2015)	27
Figura 3.9	Arquitetura dos blocos em a) e b) são equivalentes. Em c) bloco MultiRes adiciona mais informações espaciais ao adicionar a entrada com a saída dos mapas concatenados. Fonte: (IBTEHAZ; RAHMAN, 2020).	27
Figura 3.10	Cadeia de convoluções que conecta uma saída do <i>encoder</i> com seu respectivo nível no <i>decoder</i> . Fonte: (IBTEHAZ; RAHMAN, 2020).	28
Figura 3.11	Bloco <i>dual-channel</i> da DC-UNet.	29
Figura 4.1	Geração do perfil segmentado a partir partição do histograma em intervalos. As estruturas de <i>breakouts</i> estão inclusas no intervalo de menor amplitude, na cor preta, junto com outras cavidades. Fonte: (JESUS; COMPAN; SURMAS, 2016)	31
(a)	Duas seções perfil imagem com dado de amplitude à esquerda e o dado segmentado à direita.	31
(b)	Histograma do perfil de amplitude segmentado em 4 intervalos	31
Figura 4.2	Exemplo de uma seção do perfil segmentado antes e depois dos preprocessamentos aplicados.	31
(a)	Perfil de amplitude	31
(b)	Perfil segmentado	31
(c)	Perfil segmentado após remoção de pequenas áreas e operação de dilatação	31

Figura 4.3	Exemplo da extração de <i>patches</i> de um perfil imagem com 200 pixels de largura. Os <i>patches</i> da segunda coluna na Figura (b) tem uma rotação de 230 graus em relação aos da primeira coluna.	33
(a)	Seção do perfil com extensão à esquerda para completar o tamanho do <i>patch</i> e extensão à direita com metade do tamanho do <i>patch</i> .	33
(b)	<i>Patches</i> extraídos da seção do perfil ao lado utilizando o método de janela deslizante e passo de metade do <i>patch</i> .	33
Figura 4.4	Exemplo do método de validação cruzada <i>k-fold</i> com $k=5$. Fonte: GitHub (Disponível em: < http://ethen8181.github.io/ > Acesso em: 12 mar 2023)	33
Figura 4.5	Representação da métrica de IoU como razão entre a sobreposição e a união das máscaras de predição e <i>ground truth</i> .	35
Figura 5.1	Comparação da segmentação de <i>breakouts</i> para os modelos gerados no experimento 1 e 2. Predição do experimento 1 cobriu mais área no limite superior da região do <i>breakout</i> , em comparação com o experimento 2.	40
(a)	Amplitude	40
(b)	<i>Ground truth</i>	40
(c)	Predição do experimento 1	40
(d)	Predição do experimento 2	40
Figura 5.2	Comparação da detecção de falsos positivos para os modelos gerados no experimento 1 e 2.	41
(a)	Amplitude	41
(b)	<i>Ground truth</i>	41
(c)	Predição do experimento 1	41
(d)	Predição do experimento 2	41
Figura 5.3	Predição do modelo gerado no experimento 3 com nenhuma falha na segmentação.	42
(a)	Amplitude	42
(b)	<i>Ground truth</i>	42
(c)	Predição do experimento 3	42
Figura 5.4	Exemplos cuja predição gerada pelo modelo ficaram melhores que o <i>ground truth</i> .	43
(a)	Amplitude	43
(b)	<i>Ground truth</i>	43
(c)	Predição do experimento 3	43
(d)	Amplitude	43
(e)	<i>Ground truth</i>	43
(f)	Predição do experimento 3	43
Figura 5.5	Exemplo de uma predição de menor qualidade cuja segmentação ficou fragmentada. Para esse dado não foi provido um <i>ground truth</i> .	43
(a)	Amplitude	43
(b)	Predição do experimento 3	43
Figura 5.6	Comparação da segmentação obtida pelos três modelos da família U-Net.	45
(a)	Amplitude	45
(b)	<i>Ground truth</i>	45

(c)	U-Net	45
(d)	MultiResUNet	45
(e)	DC-UNet	45

Lista de tabelas

Tabela 5.1	Total de <i>patches</i> e percentual das classes da Base 1	36
Tabela 5.2	Total de <i>patches</i> e percentual das classes da Base 2	37
Tabela 5.3	K-fold da Base 1	37
Tabela 5.4	K-fold da Base 2	37
Tabela 5.5	Resultado para a partição de teste da Base 1 com filtro	38
Tabela 5.6	Resultado para a partição de teste da Base 1 sem filtro	38
Tabela 5.7	Resultado para a partição de teste da Base 1 com filtro	39
Tabela 5.8	Resultado para a partição de teste da Base 1 sem filtro	39
Tabela 5.9	Resultado da Base 2 no Modo 2	42
Tabela 5.10	Resultado da U-Net na Base 1 sem filtro	44
Tabela 5.11	Resultado da MultiResUNet na Base 1 sem filtro	44
Tabela 5.12	Comparação entre os modelos U-Net, MultiResUNet e DC-UNet na Base 1 sem filtro	44

Lista de Abreviaturas

AMP – Amplitude

TT – Tempo de Trânsito

MLP – *Multilayer Perceptron* CNN – *Convolutional Neural Network*

RCNN – *Region-based Convolutional Neural Network*

1

Introdução

Na indústria de óleo e gás, a análise de tensões geomecânicas é uma etapa imprescindível na modelagem de reservatórios. A perfuração de um poço pode modificar o estado de tensão local de uma formação rochosa, levando a potenciais danos ou falhas no buraco. As falhas de compressão são zonas colapsadas na parede do poço chamadas de *breakouts*. Essas estruturas aparecem dos dois lados do poço e distam de 90 graus do azimute de máxima tensão horizontal. Por isso, sua compreensão e análise é utilizada para criar perfis de orientação de tensão e magnitude ao longo da trajetória (ZOBACK, 2007).

O melhor meio de visualizar estruturas na parede do poço como os *breakouts* é no dado de perfil imagem acústico. A ferramenta de captura desse dado emite um pulso sonoro na formação e registra a amplitude e o tempo de trânsito da onda refletida pela rocha. Em registros de imagem, eles são reconhecidos como cavidades em pares, verticais e irregulares, separadas por aproximadamente 180°. Devido à enorme extensão dos dados do poço, a identificação manual das falhas e outras estruturas da parede por intérpretes especializados geralmente é uma operação que consome muito tempo.

Embora a busca por métodos para detectar e segmentar automaticamente essas estruturas pareça muito importante, ainda não há soluções efetivas propostas para esse problema. Destaca-se o trabalho recente de Dias (DIAS et al., 2020) que propôs um método para detecção de breakout usando a rede neural de convolução *fast-RCNN*. Embora a solução proposta alcance alta precisão em imagens sintéticas, ela não funciona tão bem em dados reais. Ademais, o método fornece apenas detecção e não segmentação do objeto. As características complexas das estruturas dos poços e os artefatos ruidosos nos perfis de imagem de poços reais são alguns dos desafios a serem superados.

O presente trabalho visa a criação de uma metodologia para segmentação automática de estruturas de *breakouts* em perfis de imagens de poços. De forma geral, o método consiste no pré-processamento dos dados seguidos da aplicação do modelo convolucional. Para realizar essa tarefa, foi empregado uma variação da arquitetura clássica U-Net, denominada DC-UNet (LOU; GUAN; LOEW, 2020). Além disso, foi utilizado um conjunto de dados reais fornecidos pela empresa Petrobras para treino e validação do modelo.

Esta dissertação está organizada em 6 capítulos. No capítulo 1 foi feita uma breve introdução. O capítulo 2 apresenta trabalhos relacionados, que

aplicam segmentação à dados perfil de imagem de poços, e uma menção ao artigo da arquitetura DC-UNet. O capítulo 3 consiste na fundamentação teórica do trabalho: a descrição sobre o tipo de dado utilizado e sobre as arquiteturas da rede neural aplicada. O capítulo 4 detalha a metodologia do trabalho, definindo os *datasets* utilizados e o *pipeline* para geração dos resultados. Já o capítulo 5 discute os resultados obtidos. Por fim, no capítulo 6 serão apresentadas as conclusões e sugestões para trabalhos futuros.

2

Trabalhos relacionados

Apesar de escassos, alguns trabalhos já exploraram técnicas para detecção e segmentação de *breakouts* em dados de imagem. No campo de aprendizado de máquina, o artigo de (DIAS et al., 2020) propõe o uso da arquitetura *fast-RCNN* para detecção de regiões de *breakouts* em perfis de imagem, utilizando dados sintéticos para treinamento supervisionado da rede. O método reportou 90% de AUC para os dados sintéticos e um resultado qualitativo em uma seção de dado real. Porém, não é informada a performance do modelo em relação a capacidade de discernir *breakouts* de outras estruturas porosas presentes no perfil. Além disso, o método não fornece a segmentação da região de interesse, somente a detecção da caixa envolvente. A segmentação é importante pois as características geométricas são informações relevantes no processo de interpretação dessas estruturas.

Outra abordagem sugere o uso de técnicas de processamento de imagem para segmentação de *breakouts* nos perfis de tempo de trânsito (VALENTIN, 2018). O autor afirma que regiões de *breakouts* apresentam alta taxa de ruído e heterogeneidade no perfil TT. Portanto, ele utiliza um filtro de desvio padrão de tamanho 3x3 para acentuar as regiões da imagem com uma distribuição local de grande variação. O resultado é então binarizado e pós-processado com um filtro de dilatação. As regiões conexas candidatas são testadas para ver se satisfazem as propriedades morfológicas característica dos *breakouts*. Entretanto, o autor não informa no texto se as condições são testadas utilizando algum método computacional ou se é feito de forma manual.

Além de *breakouts*, há diversos trabalhos voltados para a detecção de fratura em perfis de imagem. Uma boa parte busca detectar o formato senoidal da estrutura utilizando transformada de Hough (HALL et al., 1996) (TAIEBI; AKBARIZADEH; FARSHIDI, 2017) (SHAFIABADI; KAMKAR-ROUHANI; SAJADI, 2021). A desvantagem dessa técnica é o alto índice de falsos positivos detectados, por isso foi proposto em (CRUZ et al., 2017) utilizar uma rede neural para filtrar os candidatos à fratura detectados por Hough. Também foram publicados métodos para segmentação de fraturas utilizando operadores morfológicos em (LI et al., 2018) e (XAVIER; GUERRA; ANDRADE, 2015).

O objetivo neste trabalho é realizar a segmentação semântica das estruturas de *breakout* nos perfis de imagem. Para isso, foi escolhida a rede de segmentação DC-UNet (LOU; GUAN; LOEW, 2020), originalmente proposta para segmentação de imagens médicas. Redes para segmentação de imagens

médicas são projetadas para gerar predições com alto nível de detalhe na borda das regiões, o que por sua vez também é um requisito necessário na segmentação de cavidades em perfis de poços. O modelo foi avaliado em três datasets médicos e apresentou pontuações superiores em relação aos antecessores, U-Net e MultiResNet. As arquiteturas serão descritas na seção 3.7.

3

Fundamentação Teórica

Neste capítulo são apresentados alguns conceitos relacionados à estabilidade de poços e à formação dos *breakouts*. Em seguida são introduzidos fundamentos básicos sobre redes neurais convolucionais. Por fim, é feita uma descrição da arquitetura empregada neste trabalho.

3.1

Estabilidade de poços

Esta seção e a próxima se baseiam fortemente no livro Projeto de Poços de Petróleo, de Luiz Rocha e Cecilia Azevedo (ROCHA; AZEVEDO, 2007) e Geomecânica de Reservatórios, de Mark Zoback (ZOBACK, 2007).

O projeto da construção de um poço de petróleo se inicia com um estudo da área, feito a partir do levantamento de dados de sísmica e também de dados de poços já perfurados na região. Quanto maior a quantidade de informações obtidas, menores o risco e maiores as chances de sucesso na realização do projeto.

Em (ROCHA; AZEVEDO, 2007), o termo geopressões engloba todas as pressões e tensões existentes no subsolo. Seu estudo na trajetória do poço é uma etapa fundamental, visto que estimativas incorretas ou desconhecidas sobre as pressões existentes na região a ser perfurada podem desencadear complicações operacionais que resultam em prejuízos econômicos, danos ambientais e até mesmo perdas humanas. A determinação das geopressões consiste no cálculo das pressões e tensões existentes no subsolo e também daquelas que serão impostas durante a perfuração e podem levar à falhas nas rochas do entorno.

3.2

Formação de *breakouts*

Tensões *in situ* é o estado das tensões que a formação está submetida antes de qualquer atividade artificial, como perfuração, ser realizada. Em condições normais, em qualquer ponto no solo existem três eixos de tensão perpendiculares. A tensão vertical σ_v é dada pelo peso das camadas de formação sobrejacentes, também chamada de tensão de sobrecarga. O elemento de rocha submetido a essa tensão tende a se deformar lateralmente e se expandir, porém é restringido pelos elementos adjacentes, causando o aparecimento das tensões laterais horizontais σ_H e σ_h , respectivamente tensão horizontal máxima e mínima.

A perfuração do poço leva a uma alteração no estado de tensão original da região, visto que ocorre a remoção do material original e substituição por um fluido cuja principal função é manter a parede do poço estável. O peso da lama de perfuração precisa estar dentro do limite de uma janela operacional, a fim de não exercer uma pressão superior ou inferior aos limites da rocha. Saber o limite dessa janela operacional ao longo da trajetória para garantir a estabilidade do poço é uma tarefa muito difícil e requer o máximo de informações possíveis das propriedades geomecânicas da formação ao redor.

Quando a concentração de tensão na parede do poço ultrapassa a resistência da rocha, pode ocorrer ruptura. Na situação de uma concentração de tensão compressiva, que ocorre principalmente quando a lama de perfuração não exerce pressão o suficiente, a ruptura se dá pelo desmoronamento da parede do poço, levando a uma deformação do seu diâmetro (Figura 3.1a). Para o caso de poços verticais, esse desmoronamento ocorre na direção de tensão horizontal mínima σ_h , como ilustrado na Figura 3.1b.

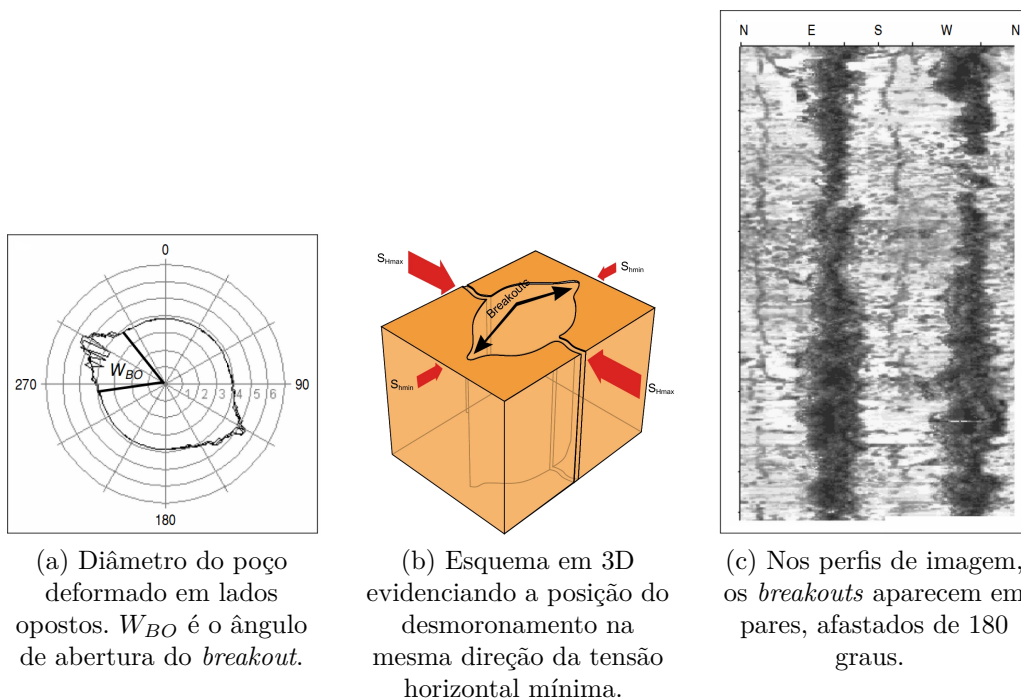


Figura 3.1: Falhas de *breakouts* ilustradas em 3 visões diferentes. Fonte: (ZOBACK, 2007)

A falha por compressão recebe o nome de *breakout* e é um fenômeno comum em muitos poços. Uma análise abrangente dessas estruturas pode render milhares de observações, o que permite a construção de perfis de orientação de tensão (e algumas vezes de magnitude) ao longo do comprimento de um poço. *Breakouts* com ângulos de abertura pequenos (menores que 90 graus de azimuth) não representam um risco significativo para a estabilidade

do poço, entretanto as falhas mais alargadas podem levar à problemas de instabilidade potencialmente catastróficos. A observação dessas estruturas também é indispensável para o monitoramento da estabilidade do poço durante a perfuração, visto que o início da formação dos *breakouts* deve ser considerado um indicativo para aumentar o peso do fluido de perfuração e evitar colapsos maiores.

A forma mais confiável de observar os *breakouts* é através dos perfil de imagem, no qual pode-se determinar a orientação e o ângulo de abertura das cavidades. No perfil planificado, as estruturas aparecem em cores mais escuras como pares aproximadamente iguais e distam de 180 graus uma da outra (Figura 3.1c).

3.3

Perfis de imagem

O perfil de imagem é um tipo de dado que captura informação da superfície da parede do poço, como uma fotografia. Existem dois tipos de dados de imagem de poço: o perfil elétrico e o perfil acústico. O perfil elétrico não será utilizado neste trabalho, mas em linhas gerais esse dado representa a resistividade da rocha à passagem de corrente elétrica, que é medida através da emissão de corrente na formação por um conjunto de eletrodos.

O perfil acústico é um dado obtido através da emissão de pulsos sonoros na parede do poço. A ferramenta de perfilagem possui um transdutor que rotaciona, permitindo a emissão das ondas em toda circunferência do poço. A quantidade de pulsos emitidos por revolução define a resolução azimutal da imagem acústica e varia conforme o fabricante da ferramenta. Os perfis utilizados nesta dissertação, por exemplo, variam entre 180 e 250 amostras de azimuth. No eixo vertical, as ferramentas acústicas são capazes de capturar o dado em uma resolução muito alta, pois a dimensão vertical dos *pixels* são menores do que 1 centímetro.

A topologia do perfil imagem é, assim como do poço, cilíndrica. Entretanto, a interpretação do dado é feita no domínio bidimensional, portanto é necessário planificar o cilindro para se obter a imagem 2D. A posição de corte para planificação costuma ser feita na orientação norte para poços verticais. A Figura 3.2 ilustra esse processo de planificação, evidenciando como estruturas planares que intersectam o poço adquirem uma geometria senoidal na imagem planificada.

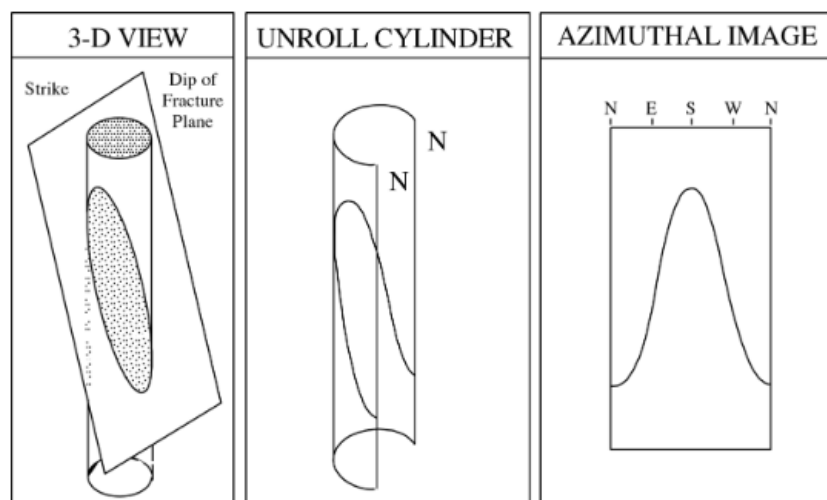


Figura 3.2: Esquema ilustra processo de planificação do dado de perfil imagem.
Fonte: (ASQUITH et al., 2004)

Para cada amostra a ser gerada no dado acústico, a ferramenta de perfilagem emite um pulso e registra informações da amplitude e do tempo de trânsito da onda refletida pela rocha. Para formações mais maciças, a amplitude refletida tem valores maiores; enquanto que para rochas menos compactas, esses valores são menores. Portanto, existe uma forte correlação entre os valores de amplitude captados e o arranjo de poros e cavidades da formação rochosa (JESUS; COMPAN; SURMAS, 2016). O tempo de trânsito é a informação do tempo que leva para emissão, reflexão e registro da onda, logo, é proporcional à distância entre a ferramenta e a superfície da rocha. Em regiões cuja superfície é muito profunda, a ferramenta falha em capturar corretamente a informação de tempo e acaba atribuindo valores anormalmente curtos para essas amostras. A tabela de cor convencional para visualização dos perfis de imagem é composta por tons quentes. No dado de amplitude, os valores mais baixos estão associados aos tons mais escuros e os mais altos, aos tons mais claros. No tempo de trânsito a tabela se inverte, a fim de manter uma correlação de cor com a amplitude.

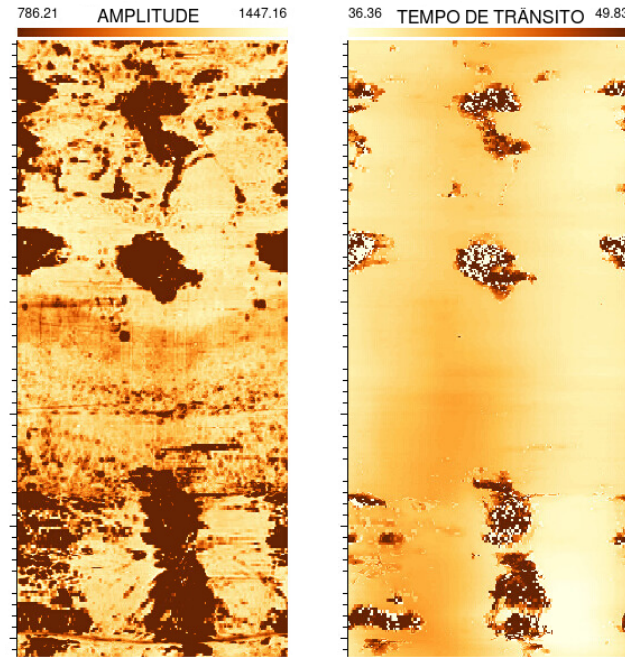


Figura 3.3: Perfis AMP e TT de uma seção de poço com cavidades de *breakout*. Os pontos em branco no perfil TT são as anomalias causadas pela falha na captura em regiões profundas.

3.4

Redes Neurais Artificiais

Redes Neurais Artificiais são modelos computacionais inspirados na estrutura neural de seres que adquirem conhecimento pela experiência. Uma rede neural é formada por uma coleção de nós interconectados chamados de neurônios. Assim como a célula biológica, o neurônio artificial recebe uma série de sinais captados na entrada, processa a informação e propaga um sinal de saída.

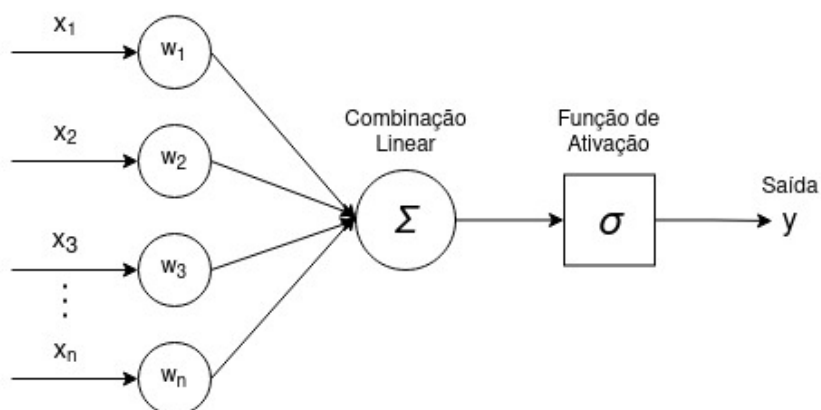


Figura 3.4: Esquema de uma rede *Perceptron*.

A arquitetura de rede neural mais simples, chamada de *Perceptron*, é composta por apenas um neurônio e está ilustrada na Figura 3.4. Ela recebe

como entrada um vetor $x = x_1, x_2, \dots, x_n$ e computa uma combinação linear com os pesos $w = w_1, w_2, \dots, w_n$. O resultado dessa combinação, somado a um valor de *bias* β , é passado para uma função diferenciável e em geral não linear que computa a saída final do neurônio. Em síntese, o *Perceptron* é descrito pela expressão algébrica

$$y = \sigma(w^T x + \beta) \quad (3-1)$$

sendo $w^T x$ a combinação linear entre os sinais de entrada e os pesos. A função de ativação σ define a forma de como um neurônio é ativado e introduz não-linearidade ao critério de ativação, possibilitando o aprendizado de padrões complexos, o que não seria possível utilizando somente funções lineares.

Um *Multilayer Perceptron* (MLP) é uma arquitetura de rede neural composta por vários neurônios organizados em camadas. As camadas de um MLP são chamadas de camadas totalmente conectadas, visto que a saída de um neurônio alimenta todos os neurônios da camada seguinte. Assim como os *perceptrons*, cada camada aplica uma transformação linear somada a um *bias* e seguido por uma função de ativação. A primeira e última camada são chamadas de camada de entrada e saída, e as demais intermediárias são chamadas de camadas escondidas. A figura 3.5 ilustra um MLP com duas camadas escondidas e a saída com somente um neurônio.

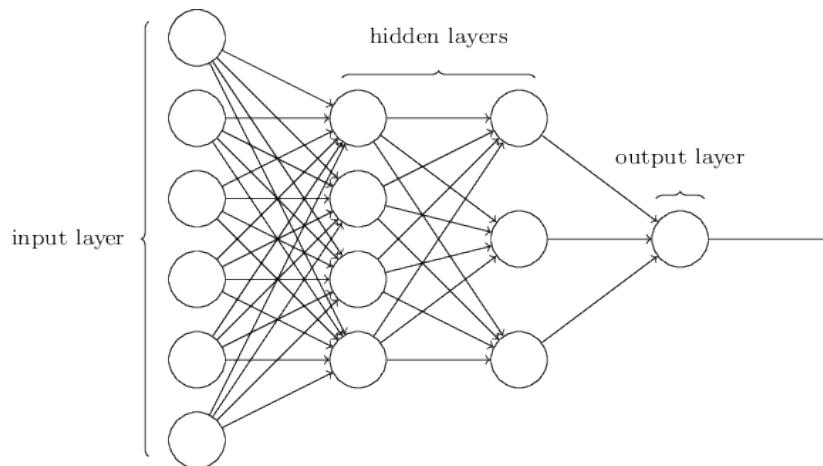


Figura 3.5: Exemplo de uma arquitetura MLP com uma camada escondida. Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

3.5

Redes Neurais Convolucionais

As redes neurais de convolução (CNN, do inglês *convolutional neural network*) são um tipo especial de rede neural usada principalmente para

reconhecimento de padrões em imagens. A convolução é uma operação linear entre duas funções definida como:

$$s(t) = (x * w)(t) = \int x(a)w(t - a)da \quad (3-2)$$

No domínio discreto bidimensional, a formulação matemática da convolução é descrita como:

$$S[i, j] = (I * K)[i, j] = \sum_m \sum_n I[m, n]K[i - m, j - n] \quad (3-3)$$

No contexto de redes neurais, I é a imagem ou mapa de entrada do operador, K é o *kernel*, ou filtro, de convolução de dimensões (m, n) e S o mapa de características extraídas pela operação de convolução.

A figura 3.6 ilustra um exemplo de arquitetura básica de uma rede convolucional, que é composta por uma camada de entrada, camadas de extração de características e camada de classificação. A entrada e saída de cada camada são volumes 3D de tamanhos que variam de acordo com as transformações aplicadas a cada etapa.

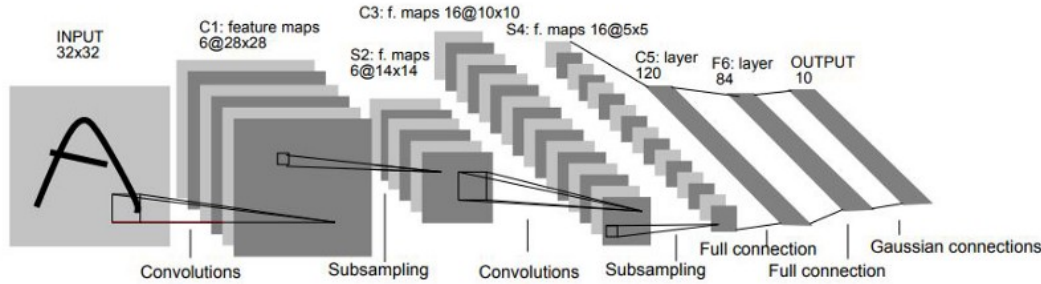


Figura 3.6: Exemplo de uma CNN básica para reconhecimento de dígitos (LeNet). Disponível em (LECUN et al., 1998).

- Camada de entrada: carrega os valores de pixel de entrada. Possui dimensões de altura, largura e quantidade de canais igual à imagem de entrada.
- Camadas de extração de características: essas camadas geralmente repetem uma sequência composta por camada convolucional, camada de ativação e camada de *pooling*.

A camada de convolução determina a saída dos neurônios que estão conectados com a entrada através do cálculo do produto escalar dos seus pesos com a região do volume no qual eles se conectam. A resolução do volume saída possui largura e altura que dependem da resolução de entrada e do filtro de convolução. A profundidade é determinada pela quantidade de filtros que serão aplicados no volume e cada filtro extrai

uma característica diferente desse volume. Os pesos de cada filtro são atualizados iterativamente no processo de aprendizado da rede.

A camada de ativação é equivalente à função de ativação de uma MLP e é aplicada para cada neurônio da entrada. Normalmente é utilizado a função ReLU (abreviação para *Rectified Linear Unit*) .

A camada de *pooling* realiza uma subamostragem que diminui a resolução do mapa de características, resumindo as informações em uma determinada janela. A operação é feita de forma similar ao processo de janela deslizante que acontece na convolução. O mais comum é o uso da função de *max pooling*, que seleciona o neurônio de maior valor da janela como saída da operação, ou a *average pooling* que calcula a média dos neurônios para a saída de cada janela.

- Camada de classificação: é uma rede MLP totalmente conectada que computa o vetor de saída com a probabilidade para cada classe.

3.6

Arquiteturas *Encoder-Decoder*

Redes neurais de deconvolução realizam o processo reverso de uma rede convolucional. As redes convolucionais comprimem a informação do sinal de entrada por sucessivas convoluções e subamostragens. Esse tipo de arquitetura é chamado de *encoder*, pois elas codificam a informação complexa do sinal de entrada para um espaço latente de menor dimensão.

As redes de deconvolução buscam reconstruir o sinal a partir da operação de convolução transposta¹. São arquiteturas do tipo *decoder* por realizar a decodificação das características codificadas de volta ao espaço original.

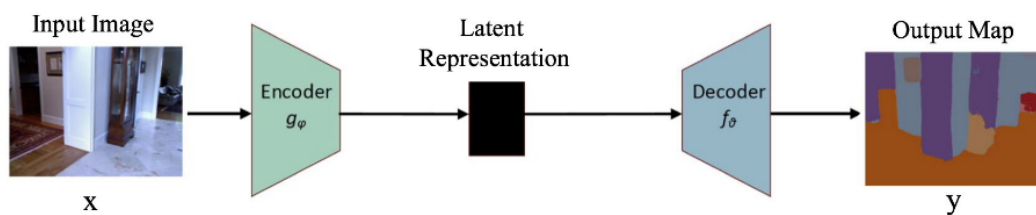


Figura 3.7: Esquema de uma arquitetura *encoder-decoder*. Fonte: (MINAEE et al., 2022)

As arquiteturas *encoder-decoder* são formadas por esses dois estágios e compõe a base da maior parte das redes de segmentação semântica, por possibilitar uma classificação pixel a pixel ao retornar a predição com a

¹Convoluções transpostas foram introduzidas com a arquitetura FCN em (LONG; SHELHAMER; DARRELL, 2015)

mesma dimensão da entrada. Esses modelos são treinados pela minimização de uma função de perda $L(y, \hat{y})$ que mede a diferença do *ground truth* y com a reconstrução \hat{y} . Uma das arquiteturas *encoder-decoder* mais populares para segmentação semântica é a U-Net, que é descrita na próxima seção.

3.7

U-Net

U-Net é uma arquitetura de rede neural projetada para fazer a segmentação semântica de imagens médicas. O artigo sobre a rede foi publicado em 2015 em (RONNEBERGER; FISCHER; BROX, 2015) e se tornou uma das principais referências para problemas de segmentação semântica por ter obtido performances acima da média para segmentação médica. A U-Net ganhou muitas adaptações propostas por outros autores para melhorar sua performance em tarefas específicas, como a MultiResUNet e a DC-UNet que serão descritas nas próximas seções.

A arquitetura original da U-Net, ilustrada na figura 3.8 segue o padrão *encoder-decoder*. O *encoder* é formado por 5 níveis de subamostragem do sinal. A cada nível é realizado em sequência duas convoluções 3x3 com função de ativação ReLU, seguidos por uma operação de *max pooling* que reduz a resolução dos mapas de características pela metade. A quantidade total de canais duplica a cada nível, indo de 64 a 1024 canais. O caminho do *decoder* é simétrico, com as operações de *pooling* substituídas por convoluções transpostas 2x2 que duplicam a resolução dos mapas a cada nível e divide a quantidade de canais pela metade. A saída de cada convolução transposta é concatenada com os mapas de características do encoder no nível correspondente, a fim de combinar as características de alta resolução com as de baixa

para gerar um resultado mais preciso espacialmente. Uma convolução 1x1 é adicionada ao final para gerar a saída do mapa de segmentação.

3.7.1

MultiResUNet

A MultiResUNet é uma arquitetura que reformulou a clássica da U-Net para melhorar a análise dos mapas de características em escalas diferentes e mitigar um possível problema de *gap* semântico que ocorre na conexão do *encoder* com o *decoder* (IBTEHAZ; RAHMAN, 2020).

O primeiro ponto de mudança proposto é a substituição das duas convoluções 3x3 pelo "bloco MultiRes". Esse bloco incorpora a ideia dos blocos *Inception* (SZEGEDY et al., 2014) de computar em paralelo convoluções com tamanhos variados de kernel a fim de extrair características em escalas diferentes

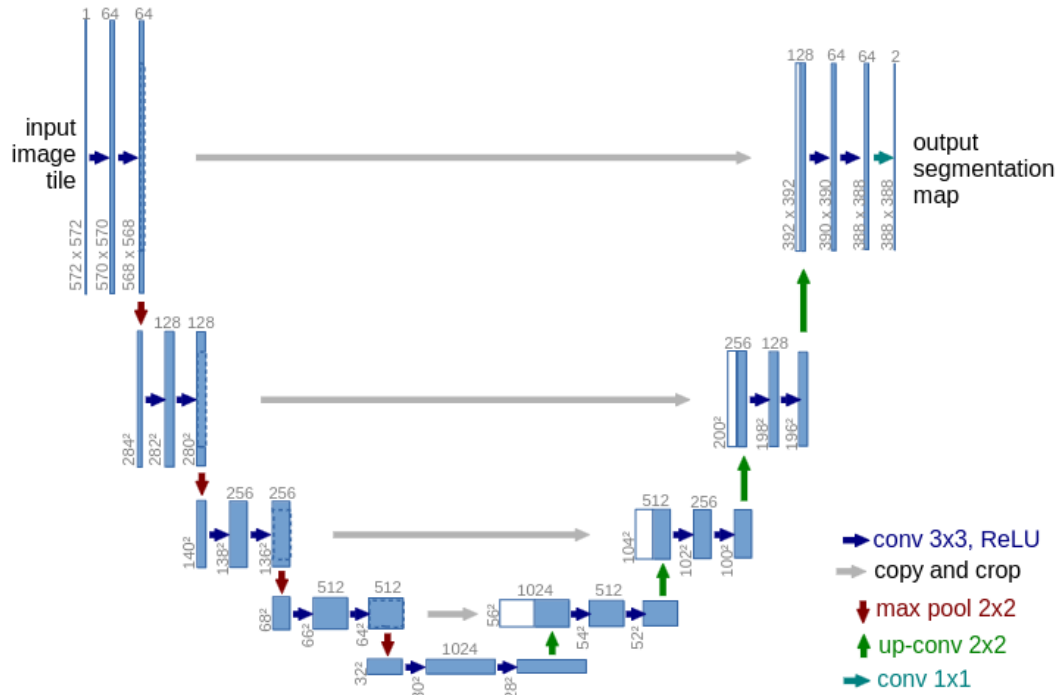


Figura 3.8: Arquitetura da U-Net. Fonte: (RONNEBERGER; FISCHER; BROX, 2015)

de um mesmo mapa. De acordo com (SZEGEDY et al., 2015), duas convoluções sucessivas com um kernel 3x3 se aproximam de uma convolução com um kernel 5x5, mas com menor custo computacional. O mesmo vale para uma convolução de kernel 7x7 que pode ser fatorada em três convoluções 3x3. A arquitetura do bloco MultiRes computa as três convoluções de forma otimizada como mostrado na figura 3.9 c). A três saídas de cada operação são concatenadas e somadas a uma conexão residual² para incluir mais informação espacial do mapa de entrada.

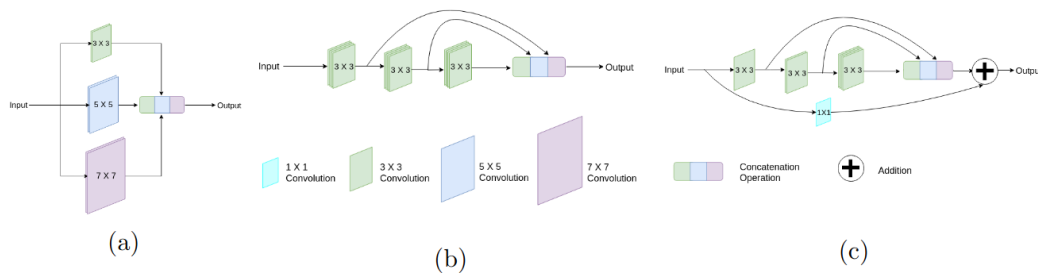


Figura 3.9: Arquitetura dos blocos em a) e b) são equivalentes. Em c) bloco MultiRes adiciona mais informações espaciais ao adicionar a entrada com a saída dos mapas concatenados. Fonte: (IBTEHAZ; RAHMAN, 2020).

²É mostrado empiricamente em (BALDUZZI et al., 2017) que conexões residuais ajudam na estabilização do aprendizado em redes profundas ao adicionar informação espacial em gradientes ruidosos.

O segundo ponto de mudança é na conexão das camadas de *encoder* com *decoder*. Na U-Net, os mapas de saída do encoder são diretamente concatenados nos mapas do decoder. As características provenientes dos primeiros estágios do *encoder* são de alto nível de detalhe e pouco processadas, visto que foram computadas nas camadas iniciais da rede; em contraste com as características dos mapas do *decoder*, que são menos detalhadas e mais processadas, por serem gerados nas camadas mais profundas da rede. Os autores da MultiResUNet apontam que a concatenação direta desses mapas pode causar uma discrepância no processo de aprendizado e afetar negativamente a predição. A solução proposta para mitigar esse problema é substituir essa conexão direta por uma cadeia de convoluções se baseando na hipótese de que essas transformações não-lineares adicionais devem compensar os mapas de saída do *encoder* em relação aos mapas mais processados do *decoder*. Cada convolução 3x3 é somada a uma conexão residual para estabilizar os gradientes, como sugerido em (SZEGEDY et al., 2016).

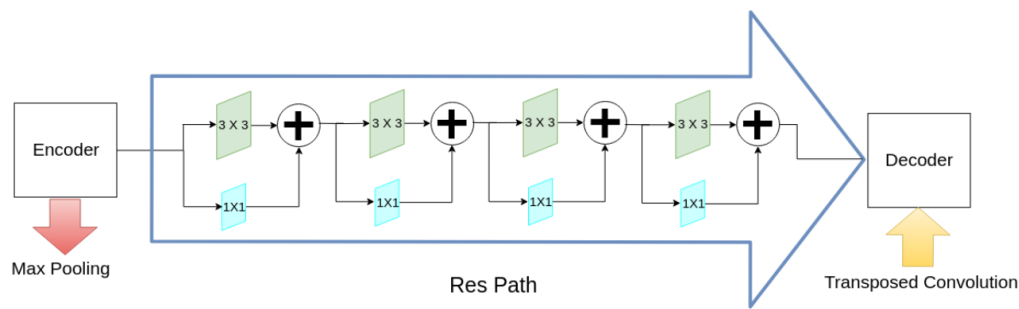


Figura 3.10: Cadeia de convoluções que conecta uma saída do *encoder* com seu respectivo nível no *decoder*. Fonte: (IBTEHAZ; RAHMAN, 2020).

O esquema apresentado na figura 3.10 mostra a cadeia de convoluções, que foi nomeada, em inglês, de *Res Path*. A primeira cadeia, que conecta o primeiro estágio do *encoder* com o último estágio do *decoder*, é formada por 4 operações de convolução 3x3. A quantidade de operações diminui a cada nível, com o último nível possuindo apenas uma convolução 3x3 com conexão residual. O motivo do encurtamento da cadeia se deve ao fato de que a discrepância entre os mapas também diminui conforme os níveis avançam, pois a quantidade de processamento nos mapas do *encoder* aumentam e no *decoder* diminuem.

3.7.2 DC-UNet

DC-UNet é uma variação da U-Net que segue basicamente a mesma arquitetura da MultiResUNet, porém com uma modificação no bloco MultiRes.

Os autores desse novo modelo relatam a imprecisão da MultiResUNet na segmentação de contornos em objetos menores. Por isso, eles decidem trocar a conexão residual do bloco MultiRes por uma segunda sequência de convolução 3×3 , a fim de captar mais características em diferentes escalas. O bloco passa a se chamar *Dual-Channel* por ter dois canais para extração de *features* em múltiplas escalas, como ilustrado na Figura 3.11.

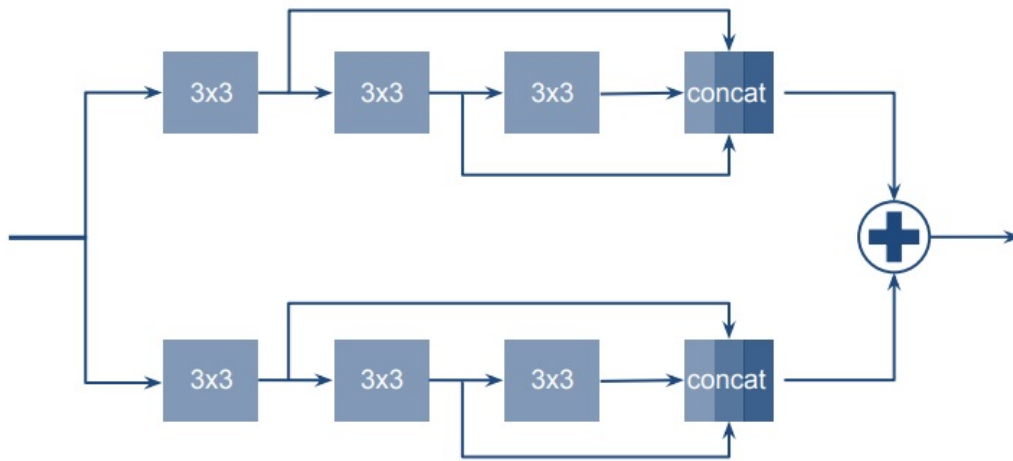


Figura 3.11: Bloco *dual-channel* da DC-UNet.

4

Metodologia

Neste capítulo serão detalhados a base de dados e o procedimento para obtenção do *ground truth*. Assim como os processamentos aplicados para correção dos rótulos e geração de *patches*. Por fim, é feita uma descrição das métricas e da técnica de *cross validation*, empregadas para avaliação quantitativa dos experimentos.

4.1

Obtenção do dado rotulado

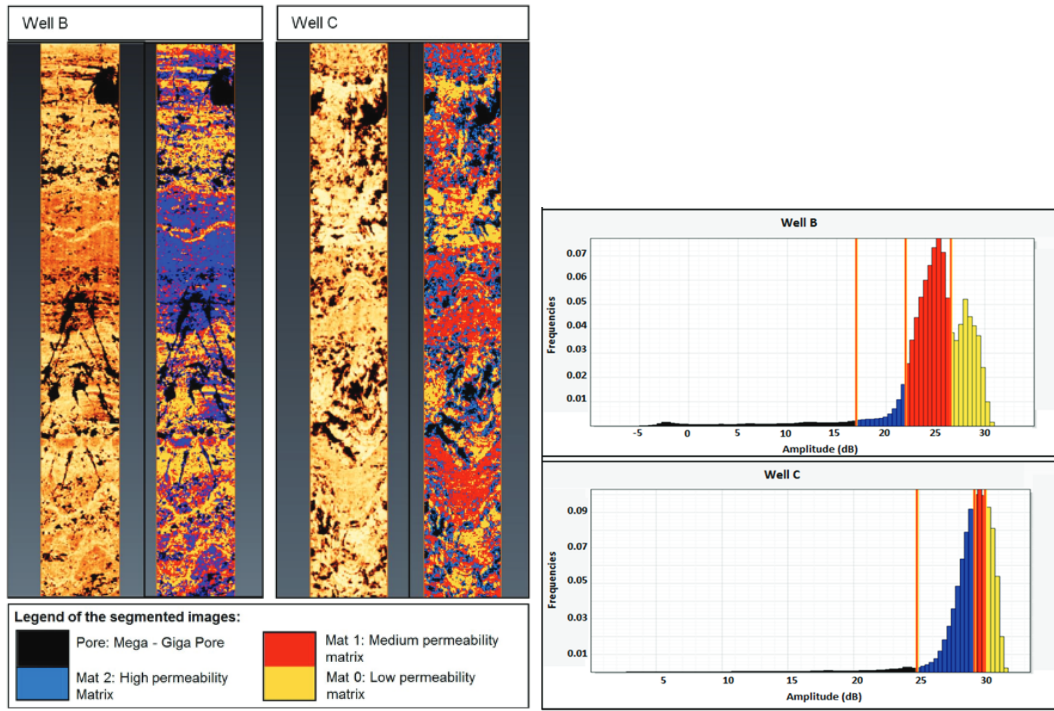
Para realização deste trabalho foram utilizados perfis de imagem de amplitude providos pela Petrobras. Os perfis de tempo de trânsito também foram disponibilizados, porém por conta das anomalias relatadas na seção 3.3 e testes iniciais com resultados insatisfatórios, foi optado por não utilizar este dado nos experimentos. O conjunto de dados é composto de 32 poços de 4 bacias distintas e de tamanhos variados, com cada perfil de poço variando entre $2 \cdot 10^4$ à $1.5 \cdot 10^5$ pixels de altura e 150 à 250 pixels de largura.

A geração dos rótulos (máscara de *ground-truth*) para o treinamento, validação e teste dos modelos supervisionados foi feita empregando a técnica de segmentação por histograma para a imagem de amplitude (JESUS; COMPAN; SURMAS, 2016). De forma geral, as regiões de *breakout* e outras estruturas geomecânicas e geológicas estão localizadas em faixas de valores mais baixos de amplitude. Assim, com uma ferramenta de segmentação de perfis, um intérprete capacitado seleciona manualmente um *threshold* ideal para separação dos *breakouts* e demais estruturas do restante do perfil, como pode ser visto na Figura 4.1. Em uma segunda etapa, o intérprete utiliza ferramentas básicas de edição para separar essas estruturas que estão na mesma faixa de valores em diferentes rótulos, entre um deles, a classe dos *breakouts*. O resultado final gerado por esse tratamento é o perfil segmentado.

4.2

Pré-processamento

O método semi-automático de anotação das estruturas acaba gerando regiões imprecisas. Isso acontece pois a segmentação por histograma é muito sensível a ruídos do dado (Figura 4.2b). A fim de melhorar a qualidade do *ground truth*, as regiões com menos de 10 pixels de área são removidas reduzindo, assim, a quantidade de artefatos ruidosos na máscara de segmentação.

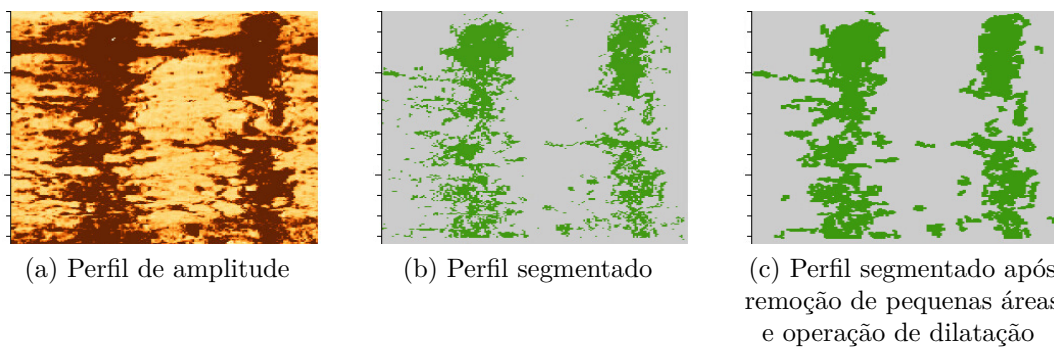


(a) Duas seções perfil imagem com dado de amplitude à esquerda e o dado segmentado à direita.

(b) Histograma do perfil de amplitude segmentado em 4 intervalos

Figura 4.1: Geração do perfil segmentado a partir partição do histograma em intervalos. As estruturas de *breakouts* estão inclusas no intervalo de menor amplitude, na cor preta, junto com outras cavidades. Fonte: (JESUS; COMPAN; SURMAS, 2016)

Esse processo é seguido por uma operação de dilatação para expandir as regiões de *breakout* e melhor delimitar as bordas, além de conectar regiões desconexas (Figura 4.2c).



(a) Perfil de amplitude

(b) Perfil segmentado

(c) Perfil segmentado após remoção de pequenas áreas e operação de dilatação

Figura 4.2: Exemplo de uma seção do perfil segmentado antes e depois dos preprocessamentos aplicados.

4.3

Estratégia para extração de amostras

A altura dos perfis de imagem variam entre dezenas a centenas de milhares de pixels. Sendo assim, é necessário amostrar o perfil de poço em pequenos *patches* para enviar o dado ao modelo. A extração dos *patches* foi feita utilizando o método de janela deslizante, que consiste na definição de um retângulo que 'desliza' horizontal e verticalmente na imagem, com um determinado tamanho de passo, extraindo uma imagem menor da região que o retângulo está localizado. Para o conjunto de dados em questão, como dito anteriormente, a largura do poço varia entre 180 a 250 amostras de azimuth. Sendo assim, foi escolhido um tamanho de *patch* de 256x256 pixels para que ele tenha a parede do poço inteira horizontalmente. Como a geometria do poço é cilíndrica direcional, o lado esquerdo do perfil imagem foi preenchido com as últimas colunas do lado direito (ver Figura 4.3a). Concatenando as primeiras 128 colunas ao lado direito do perfil e definindo um passo do mesmo tamanho, foi possível extrair um segundo patch na mesma faixa de profundidade com uma rotação diferente que varia de 184 a 256 graus, como ilustrado na Figura 4.3b. O passo vertical escolhido da janela deslizante também foi de metade do tamanho do patch, o que gerou sobreposições. Esse método pode ser considerado um tipo de técnica de aumento de dados (*data augmentation*), por empregar as características do domínio.

4.4

Validação Cruzada

A validação cruzada é uma técnica amplamente utilizada para avaliar a capacidade de generalização de modelos de predição. Existem vários métodos de validação cruzada na literatura, mas o conceito principal deles é a partição do conjunto de dados em subconjuntos mutuamente exclusivos, utilizando uma parte desses subconjuntos para treinamento e a outra parte, não conhecida pelo modelo, para avaliar o desempenho.

A Figura 4.4 ilustra a técnica de **Validação Cruzada *K-Fold***. Nessa técnica o dado é particionado em k conjuntos de tamanhos aproximadamente iguais, chamados de *fold*, e o processo de treino e validação é feito k vezes. Para cada iteração um *fold* diferente é separado para validação e os demais $k-1$ *folds* são utilizados para o treino do modelo. A medida de desempenho do modelo é obtida pela média do resultado de todas as iterações. O objetivo do *k-fold* é possibilitar que cada uma das partições seja usada para a validação em alguma das iterações a fim de evitar uma avaliação enviesada do modelo (REFAEILZADEH; TANG; LIU, 2009).

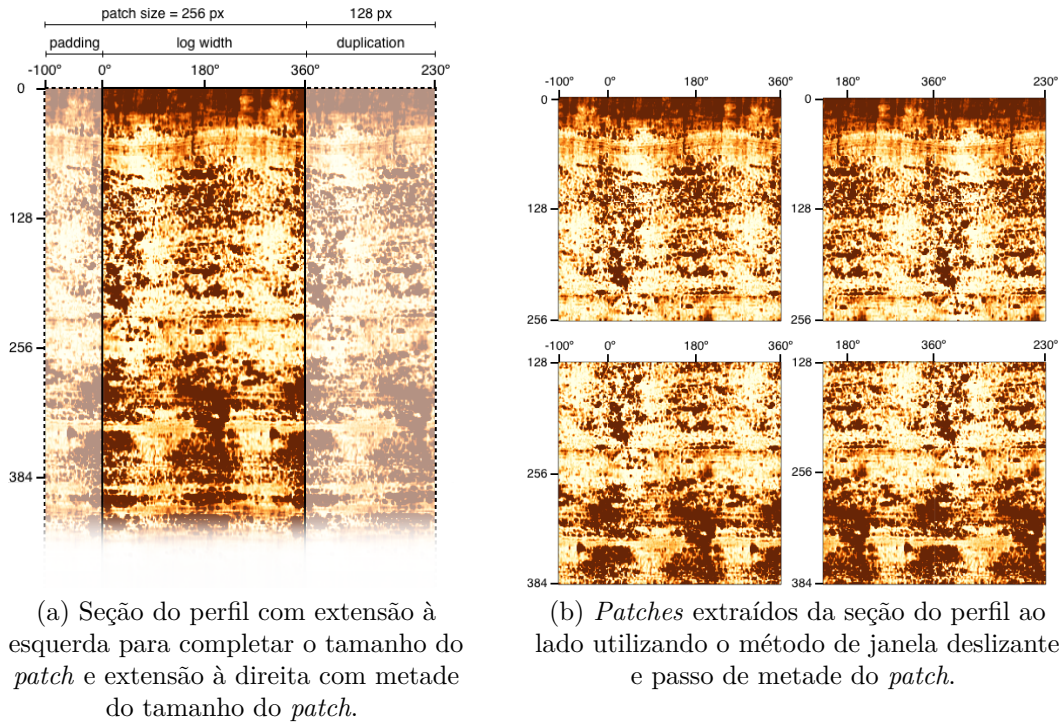


Figura 4.3: Exemplo da extração de *patches* de um perfil imagem com 200 pixels de largura. Os *patches* da segunda coluna na Figura (b) tem uma rotação de 230 graus em relação aos da primeira coluna.

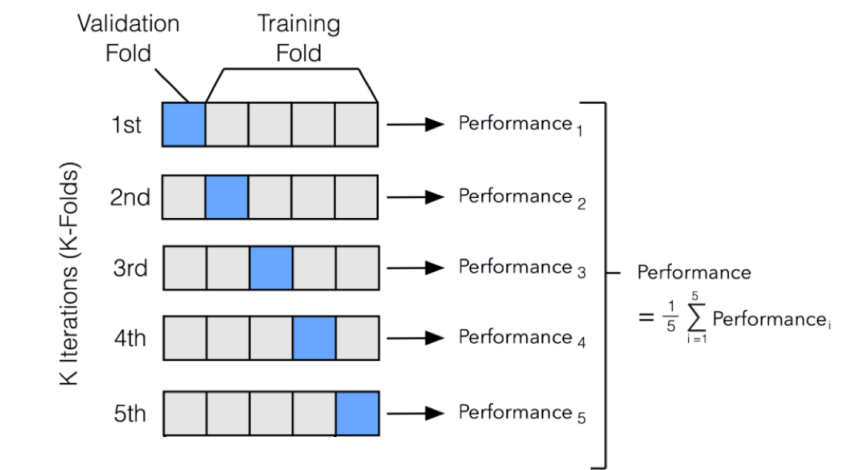


Figura 4.4: Exemplo do método de validação cruzada *k-fold* com $k=5$. Fonte: GitHub (Disponível em: <<http://ethen8181.github.io/>> Acesso em: 12 mar 2023)

No contexto do problema abordado neste trabalho não seria possível criar *folds* de tamanhos iguais, pois os poços da base de dados possuem tamanhos variados. Assim, dividir os *patches* de um mesmo poço em dois ou mais *folds* violaria a regra de subconjuntos mutuamente exclusivos. Sendo assim, optou-se por utilizar o K-Fold para avaliação do modelo sujeito a algumas condições. Para a construção das diferentes partições de treino, validação e teste, a variação do conjunto de poços para cada combinação estão sujeitas às regras de: 1. Ter ao menos um poço de cada bacia por partição; 2. Manter uma proporção mais ou menos igual de *patches* de *breakouts* para cada combinação. As combinações criadas para realização dos experimentos são descritas na seção 5.1 do capítulo 5.

4.5

Métricas de avaliação

Para avaliação quantitativa do desempenho do modelo proposto, utilizamos quatro métricas convencionais para problemas de classificação: *Precision*, *Recall*, *F1-Score* e *Intersection over Union (IoU)*.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4-1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4-2)$$

$$\text{F1-Score} = \frac{2 | \text{Recall} * \text{Precision} |}{| \text{Recall} | + | \text{Precision} |} \quad (4-3)$$

$$\text{IoU} = \frac{| \text{Recall} * \text{Precision} |}{| \text{Recall} | + | \text{Precision} | - | \text{Recall} * \text{Precision} |} \quad (4-4)$$

Sendo:

- **TP**: Verdadeiro Positivo (*True Positive*), quando o valor de predição e de referência é igual da classe de interesse (também chamada de classe positiva);
- **FN**: Falso Negativo (*False Negative*), no caso do pixel de predição ser da classe de fundo e o da referência da classe de interesse.
- **FP**: Falso Positivo (*False Positive*) quando o valor da predição é igual da classe positiva, mas o valor da referência é negativa;

Em termos probabilísticos¹ e no contexto dos *Breakouts* como classe de interesse, **Recall** é a probabilidade do modelo classificar uma amostra de pixel como *breakout* dado que ela é de fato um breakout. **Precision** é

¹Para uma explicação sobre a interpretação probabilística das métricas de avaliação consultar (GOUTTE; GAUSSIER, 2005)

a probabilidade da predição de uma amostra ser correta dado que ela foi classificada como *breakout* pelo modelo. Já ***F1-Score*** corresponde à média harmônica entre a precisão e *recall*.

Intersection Over Union (IoU) é a área de sobreposição da máscara de segmentação predita e a segmentação alvo (*ground truth*) dividido pela área da união entre a predição e o alvo. Uma pontuação de IoU de 0 corresponde a nenhuma sobreposição entre a predição e o alvo e uma pontuação de IoU de 1 corresponde a uma sobreposição perfeita da segmentação de predição e *ground truth*. Esta métrica pode ser obtida a partir dos valores de precisão e *recall*, como descrito na equação (4-4) e interpretada visualmente pela Figura 4.5


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figura 4.5: Representação da métrica de IoU como razão entre a sobreposição e a união das máscaras de predição e *ground truth*.

5

Experimentos e Discussão

Nesta seção, são descritos os cenários de testes para a realização dos experimentos. Em seguida, são discutidos os resultados obtidos para a classificação pixel a pixel das regiões de breakout.

5.1

Descrição dos testes

Os experimentos descritos neste capítulo foram feitos em duas bases de teste. A primeira base, denominada de **Base 1**, é composta por poços de duas bacias, com o total de **18 poços**. A segunda base, **Base 2**, engloba os poços da Base 1 e mais 14 poços provenientes de outras duas bacias, totalizando **32 poços**. Além disso, os experimentos foram realizados em duas configurações distintas. No primeiro caso, após extração dos patches como descrito na seção 4.3, foi feita uma filtragem para manter somente aqueles que continham algum pixel rotulado como *breakout*. No segundo cenário, foram mantidos após a filtragem os patches que tinham algum pixel rotulado como *breakout*, *vug* ou fratura. A base de dados gerada no segundo é portanto significativamente maior que o caso 1. O primeiro e segundo cenário serão referidos como base **com filtro** e **sem filtro**, respectivamente.

As tabelas 5.1 e 5.2 mostram o total de patches para cada um dos cenários e o percentual de *pixels* para cada uma das classes. O percentual de *vugs* e fraturas são apresentados com intuito de passar uma noção sobre a distribuição das cavidades em cada base, mas eles são tratados como classe de fundo ao passar os dados para o modelo.

Tabela 5.1: Total de *patches* e percentual das classes da Base 1

Base 1	Com filtro	Sem filtro
Total de <i>patches</i>	3300	14132
% Classe Fundo	91.70%	94.84%
% Classe <i>Breakout</i>	7.88%	1.24%
% Classe Fratura	0.10%	0.71%
% Classe <i>Vug</i>	0.32%	3.21%

Tabela 5.2: Total de *patches* e percentual das classes da Base 2

Base 2	Com filtro	Sem filtro
Total de patches	5832	26740
% Classe Fundo	92.10%	96.50%
% Classe Breakout	7.63%	1.08%
% Classe Fratura	0.09%	0.42%
% Classe Vug	0.18%	2.00%

Na fase de teste, as métricas de avaliação do modelo não foram calculadas para os perfis inteiros (ou seja, sem nenhum tipo de filtragem) por conta das falhas de anotação nos dados. Muitas regiões de cavidade nos perfis não receberam nenhum rótulo, inclusive regiões de *breakout*. Isso poderia prejudicar o levantamento de falsos positivos, caso o modelo fizesse uma classificação correta de uma região de *breakout* que está rotulada como fundo no dado *ground truth*. Portanto, avaliação quantitativa pelas métricas foi feita em cima das bases filtradas com somente *breakouts* (com filtro) ou com *breakouts*, vugues e fraturas (sem filtro). A predição do poço inteiro foi utilizada somente para uma avaliação qualitativa dos resultados.

Utilizamos a técnica de validação cruzada mencionada na seção 4.4 com 5-*folds* para avaliar o modelo na Base 1 e 3-*folds* para avaliar o modelo na Base 2. As tabelas 5.3 e 5.4 descrevem a configuração do *k-fold* para cada base, com o total de poços para cada partição e o percentual de *patches* referente a cada uma.

Tabela 5.3: K-*fold* da Base 1

Partição	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Treino	7 poços (73%)	7 poços (73%)	6 poços (56%)	10 poços (72%)	11 poços (72%)
Validação	3 poços (16%)	4 poços (12%)	2 poços (26%)	5 poços (12%)	4 poços (12%)
Teste	8 poços (11%)	7 poços (15%)	10 poços (18%)	3 poços (16%)	3 poços (16%)

Tabela 5.4: K-*fold* da Base 2

Partição	Fold 1	Fold 2	Fold 3
Treino	20 poços (59.1%)	22 poços (58.9%)	21 poços (60.5%)
Validação	7 poços (20.0%)	7 poços (19.9%)	4 poços (21.3%)
Teste	5 poços (20.9%)	4 poços (21.2%)	6 poços (18.2%)

5.2

Experimento 1: Base 1 com filtro

Neste experimento treinamos o modelo da DC-UNet utilizando a Base 1 com filtro, ou seja, apresentamos à rede somente exemplos de *breakout*. O resultado das métricas para a base de teste filtrada e não-filtrada são apresentados na tabela 5.5 e 5.6, respectivamente.

Tabela 5.5: Resultado para a partição de teste da Base 1 com filtro

Métricas	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Precisão	60.1%	86.5%	84.5%	95.5%	88.8%	83.1%
Recall	39.9%	76.2%	74.0%	68.9%	74.4%	66.7%
F1-Score	47.9%	81.0%	78.9%	80.0%	80.9%	73.8%
IoU	31.5%	68.1%	65.1%	66.7%	68.0%	59.9%

Tabela 5.6: Resultado para a partição de teste da Base 1 sem filtro

Métricas	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Precisão	6.7%	54.1%	39.2%	74.6%	75.4%	50.0%
Recall	39.9%	76.2%	74.1%	68.9%	74.4%	66.7%
F1-Score	11.5%	63.3%	51.2%	71.6%	74.9%	54.5%
IoU	6.1%	46.3%	34.4%	55.8%	59.9%	40.5%

A tabela 5.5 mostra o desempenho do modelo para segmentação das estruturas de interesse, atingindo uma alta pontuação para a métrica de precisão, porém em uma partição de teste que só englobava estruturas de *breakout*. Na tabela 5.6 as métricas foram calculadas para a partição de teste com exemplos de *breakouts* e outras estruturas porosas (fraturas e *vugs*). Analisando o resultado das duas tabelas, podemos concluir que o modelo treinado somente com exemplos de *breakout* não é capaz de discernir a estrutura de interesse das demais estruturas porosas, visto que ocorre uma queda significativa de 33% na precisão para a segunda tabela, causada pelo aumento de falsos positivos preditos pelo modelo neste teste.

5.3

Experimento 2: Base 1 sem filtro

O experimento anterior foi repetido porém utilizando a partição de treino e validação da Base 1 sem filtro, que continham exemplos de *breakout*, fratura e *vug*. O resultado das métricas para a partição de teste com e sem filtro se encontram nas tabelas 5.7 e 5.8, respectivamente.

Tabela 5.7: Resultado para a partição de teste da Base 1 com filtro

Métricas	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Precisão	68.1%	88.6%	88.8%	96.7%	88.9%	86.2%
Recall	30.9%	68.8%	63.9%	60.3%	67.0%	58.2%
F1-Score	42.5%	77.4%	74.3%	74.3%	76.4%	69.0%
IoU	27.0%	63.2%	59.1%	59.1%	61.8%	54.0%

Tabela 5.8: Resultado para a partição de teste da Base 1 sem filtro

Métricas	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Precisão	21.9%	74.3%	62.4%	92.5%	88.0%	67.8%
Recall	30.9%	68.8%	64.0%	60.3%	67.1%	58.2%
F1-Score	25.6%	71.5%	63.1%	73.0%	76.1%	61.9%
IoU	14.7%	55.6%	46.1%	57.5%	61.4%	47.1%

Em comparação com o experimento anterior houve uma queda na taxa de *recall* pois a introdução de *patches* com outras cavidades que são parecidas com *breakouts*, mas pertencem à classe de fundo torna a separação das duas classes menos clara. Entretanto, o aumento da taxa da precisão na segunda tabela mostra uma melhora significativa do modelo para diferenciar os *breakouts* das estruturas de *vug* e fratura, em comparação com o resultado da tabela 5.6. O resultado geral de *F1-Score* e *IoU* do neste experimento foram maiores do que no experimento 1, com um aumento de 7.4% e 6.6%, respectivamente.

As figuras 5.1 e 5.2 ilustram dois resultados comparativos dos experimentos 1 e 2. A primeira mostra a pouca variação na predição de uma estrutura de *breakout*, sendo a segmentação do experimento 1 ligeiramente melhor do que o experimento 2. A segunda figura mostra uma seção de poço com estruturas de fratura (em azul) e *vugues* (em vermelho) no qual o modelo do experimento 1 classificou erroneamente como *breakout* (verde). Em contraste,

o modelo do experimento 2 detectou poucas regiões de falso positivo, o que mostra a capacidade do segundo modelo de discernir melhor os *breakouts* das demais cavidades.

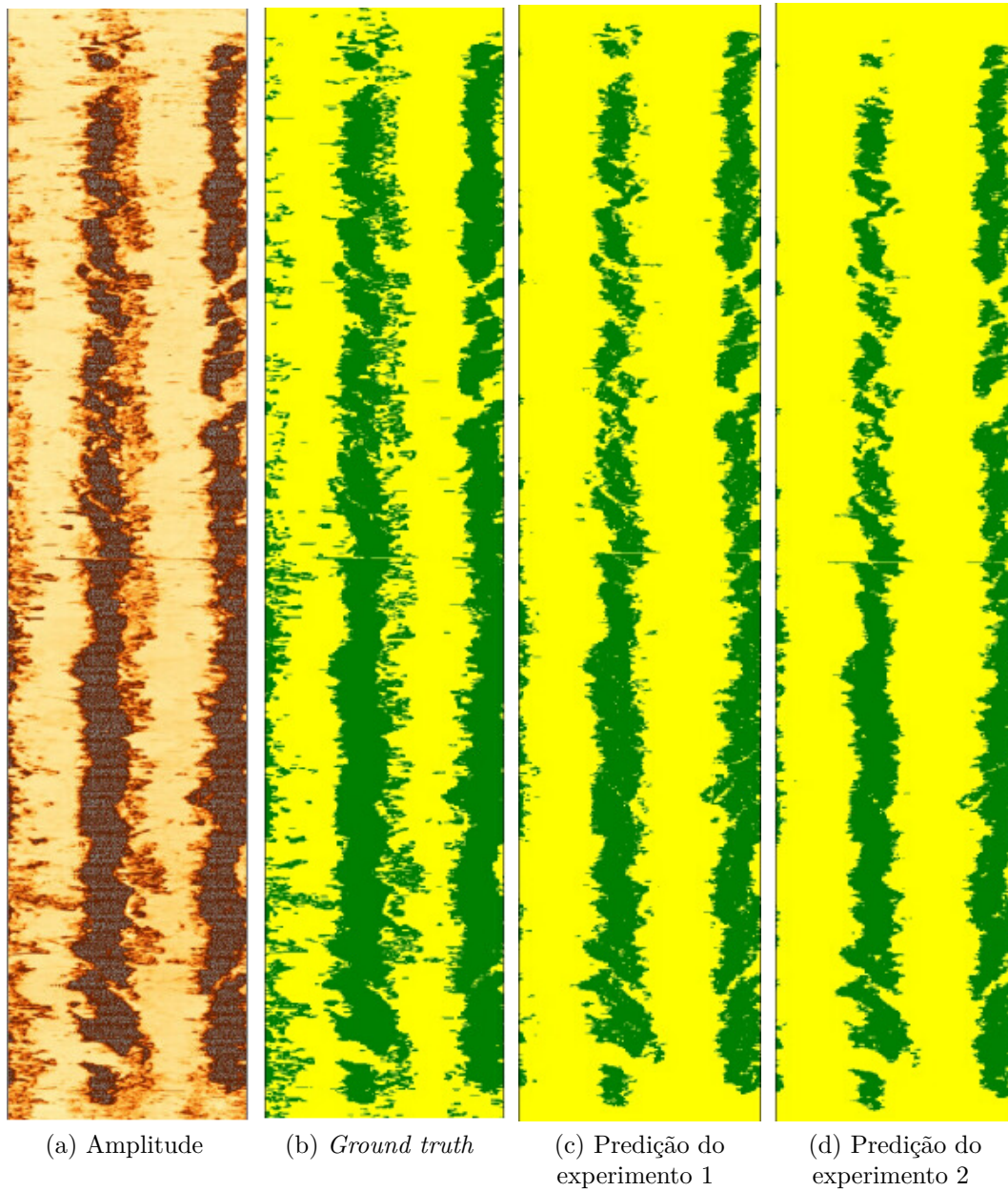


Figura 5.1: Comparação da segmentação de *breakouts* para os modelos gerados no experimento 1 e 2. Predição do experimento 1 cobriu mais área no limite superior da região do *breakout*, em comparação com o experimento 2.

5.4

Experimento 3: Base 2 sem filtro

Neste experimento treinamos a rede da DC-UNet com a Base 2 sem filtro. O modelo foi avaliado utilizando 3 *folds* e as mesmas configurações de

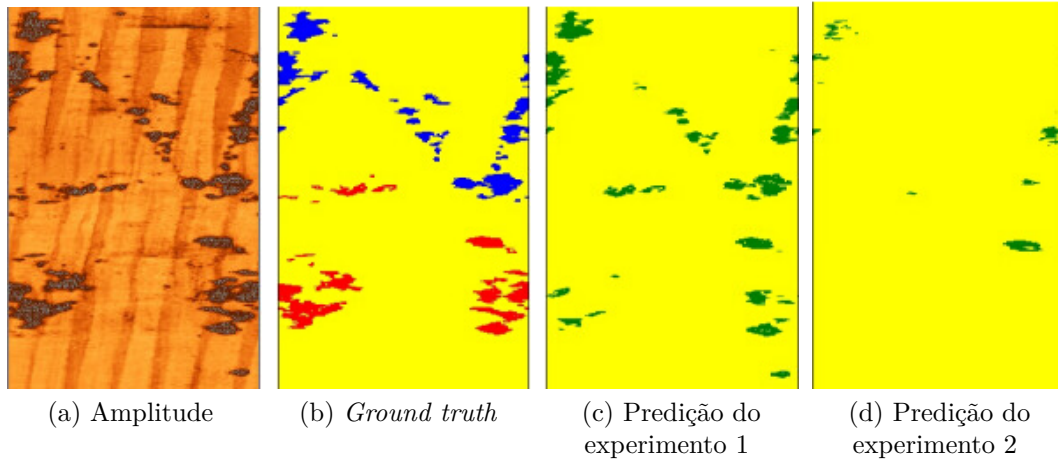


Figura 5.2: Comparação da detecção de falsos positivos para os modelos gerados no experimento 1 e 2.

treino do Experimento 1. Os resultados de teste no Modo 2 são apresentados na tabela 5.9. Esse modelo alcançou, em média, pontuações 10% maiores em comparação às pontuações obtidas na tabela 5.6, relativa ao modelo treinado com a Base 1, que é formada por poços de apenas 2 bacias. A Figura 5.3 mostra uma seção de *breakout* com um resultado de predição sem falhas. Como mencionado nas seções 4.2 e 4.1, a máscara de *ground truth* utilizada não tem uma anotação perfeita, por isso. Em alguns exemplos nota-se um resultado de predição do que a anotação original. A Figura 5.4 mostra duas seções do perfil cuja segmentação da predição ficou com qualidade superior ao *ground truth*. Na Figura 5.5 foi realizada a predição de um perfil de teste cujo *ground truth* não foi disponibilizado. Apesar da região de *breakout* ter sido detectada, a segmentação ficou muito fragmentada. Uma possível justificativa é que esse perfil de teste não faz parte de nenhuma das quatro bacias utilizadas no treino, e a rede não foi capaz de generalizar o suficiente para gerar uma segmentação de melhor qualidade para poços de outras bacias. Futuramente este problema pode ser amenizado com a aplicação de alguma técnica de pós-processamento que preencha os espaços não segmentados, como por exemplo uma simples dilatação ou uma técnica de *floodfill* mais elaborada.

Tabela 5.9: Resultado da Base 2 no Modo 2

Métricas	Fold 1	Fold 2	Fold 3	Média
Precisão	75.75%	82.0%	79.6%	79.1%
Recall	70.97%	54.9%	76.0%	67.3%
F1-Score	73.28%	65.8%	77.8%	72.3%
IoU	57.83%	49.0%	63.6%	56.8%

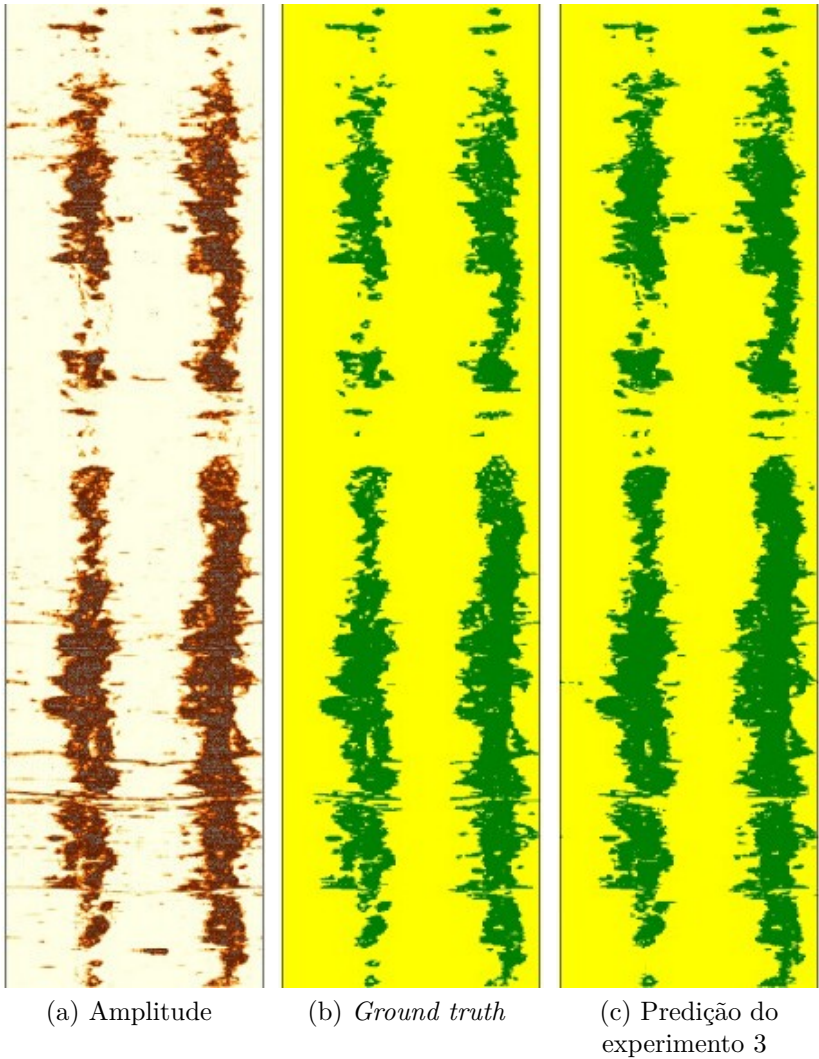


Figura 5.3: Predição do modelo gerado no experimento 3 com nenhuma falha na segmentação.

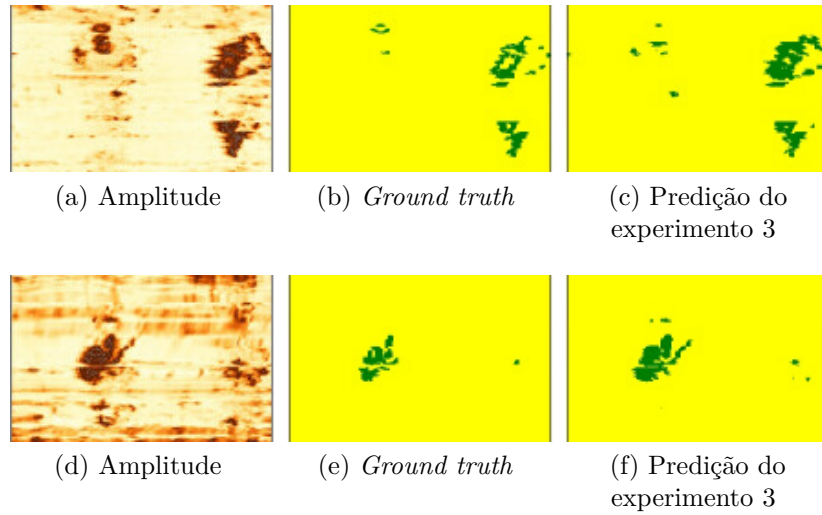


Figura 5.4: Exemplos cuja predição gerada pelo modelo ficaram melhores que o *ground truth*.

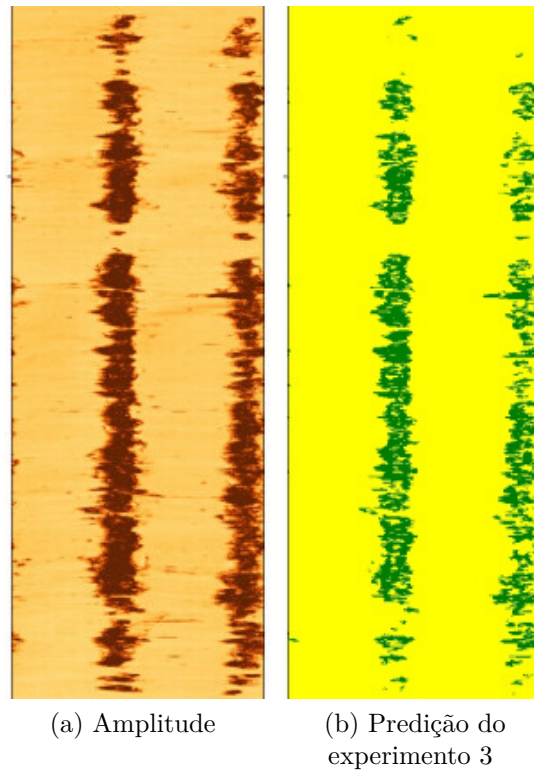


Figura 5.5: Exemplo de uma predição de menor qualidade cuja segmentação ficou fragmentada. Para esse dado não foi provido um *ground truth*.

5.5

Experimento 4: U-Net e MultiResUNet

O objetivo deste experimento foi comparar a performance da DC-UNet com as suas redes precursoras, a MultiResUNet e a U-Net. O treinamento foi feito com a Base 1 sem filtro. Na tabela 5.12 é mostrado a comparação

das pontuações obtidas por cada modelo. O modelo da DC-UNet supera a U-Net e a MultiResUNet, o que pode confirmar a melhora de performance anunciada pelos autores em relação às arquiteturas antecessoras. Entretanto, os hiperparâmetros utilizados para os treinos são os mesmos do Experimento 1 e foram otimizados para treinamento da rede DC-UNet. Não foi feita uma otimização específica para treino da U-Net e da MultiResUNet, o que pode justificar a diferença expressiva de performance da DC-UNet em relação às demais. A figura 5.6 mostra um resultado comparativo dos 3 modelos para a mesma seção de perfil que corrobora o resultado quantitativo da tabela.

Tabela 5.10: Resultado da U-Net na Base 1 sem filtro

Métricas	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Precisão	11.93%	49.0%	20.5%	68.4%	50.1%	40.0%
Recall	31.94%	58.5%	60.8%	23.9%	58.5%	46.7%
F1-Score	17.37%	53.4%	30.7%	35.4%	54.0%	38.2%
IoU	9.51%	36.4%	18.1%	21.5%	37.0%	24.5%

Tabela 5.11: Resultado da MultiResUNet na Base 1 sem filtro

Métricas	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Precisão	19.0%	39.5%	38.5%	75.9%	20.7%	38.7%
Recall	27.1%	71.0%	64.2%	62.1%	64.7%	57.8%
F1-Score	22.6%	50.8%	48.1%	68.3%	31.4%	44.2%
IoU	12.7%	34.0%	31.7%	51.9%	18.6%	29.8%

Tabela 5.12: Comparação entre os modelos U-Net, MultiResUNet e DC-UNet na Base 1 sem filtro

Modelo	Precisão	Recall	F1-Score	IoU
U-Net	40.01%	46.73%	38.16%	24.50%
MultiResUNet	38.72%	57.82%	44.24%	29.79%
DC-UNet	67.80%	58.21%	61.86%	47.06%

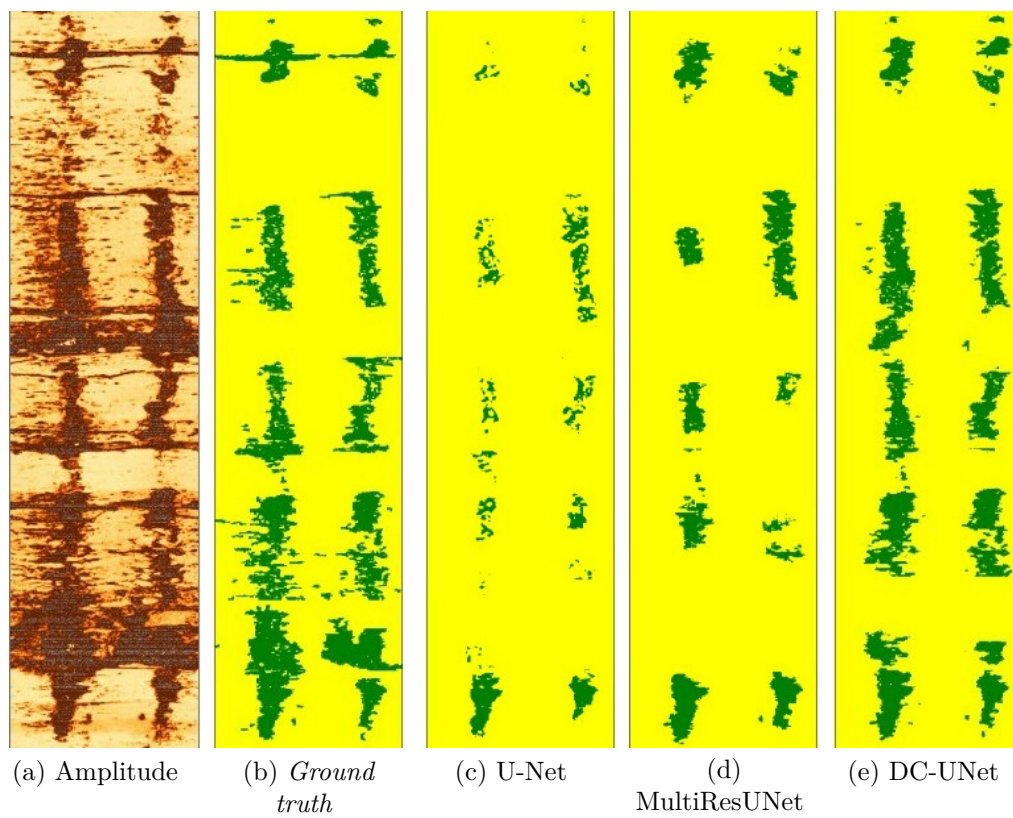


Figura 5.6: Comparação da segmentação obtida pelos três modelos da família U-Net.

6

Conclusão e trabalhos futuros

O presente trabalho consistiu no desenvolvimento de uma metodologia automática para detecção de estruturas de *breakouts* em perfis de imagem de poços. Para validar e avaliar o método, foi empregado um conjunto de dados reais fornecidos pela empresa Petrobras. O *dataset* era composto por 32 poços e suas interpretações de 4 diferentes bacias, com a presença, ou não, do artefato estudado.

A metodologia consistiu em duas etapas principais: a de pré-processamento, seguida pela aplicação do modelo de deep learning. O pré-processamento consistiu na normalização dos dados de amplitude entre $[0,1]$, e nos dados segmentados a aplicação de um filtro de remoção de pequenas áreas e uma operação de dilatação. Na etapa final de pré-processamento foi feita a extração de *patches* dos dados da imagem. Após esse tratamento o modelo de rede neural é treinado e aplicado em novos dados. Para avaliação quantitativa dos resultados, foram utilizadas métricas clássicas de aprendizado de máquina *Precision*, *Recall*, *F1-Score* e *IoU*.

Mesmo com dados de treinamento escassos, a metodologia desenvolvida produziu resultados promissores. O método alcançou um *F1-Score* médio de 72,3% e resultados qualitativos validados por especialistas, com algumas previsões de casos com melhor qualidade do que os *ground truth* disponíveis. A partir disso, pode-se concluir que o método oferecido pode auxiliar especialistas a identificar possíveis *breakouts* em dados de perfis imagem de poços. Assim, o procedimento de interpretação pode então ser otimizado para reduzir os custos e a duração da fase de interpretação.

Para trabalhos futuros, é interessante estudar melhor como introduzir o dado de tempo de trânsito como um segundo canal na entrada do modelo. Além disso, outras funções de perda que focam na melhoria na detecção das bordas, como, por exemplo, a *Focal Loss* apresentada por Lin (LIN et al., 2017), devem ser testadas. A técnica de aumento de dados apresentada neste trabalho em conjunto com outras técnicas clássicas devem ser mais explorados para aprimorar os resultados. Ainda no pré-processamento, outra sugestão de trabalho seria avaliar o uso de um recorte aleatório para extração dos patches considerando o domínio cilíndrico para o aumento da generalização dos dados de entrada. Uma outra contribuição seria a adição de métodos sofisticados de treinamento, tais como o aprendizado contrastivo (ZHAO et al., 2020), para suavizar o fato de que rótulos de treinamento são produzidos

usando métodos básicos de processamento de imagem.

O método de segmentação desenvolvido com esta dissertação será incorporado como uma ferramenta de um software a fim de ser utilizado pelos intérpretes para segmentar as cavidades. Utilizando essa ferramenta e realizado as devidas correções na predição pelo usuário especializado, a segmentação corrigida pode ser utilizada novamente para criação de modelos cada vez mais robustos.

Referências bibliográficas

Asquith et al. 2004ASQUITH, G. et al. **Basic well log analysis**. American Association of Petroleum Geologists, 2004. ISBN 9781629810492. Disponível em: <<https://doi.org/10.1306/Mth16823>>.

Balduzzi et al. 2017BALDUZZI, D. et al. The shattered gradients problem: If resnets are the answer, then what is the question? **CoRR**, abs/1702.08591, 2017. Disponível em: <<http://arxiv.org/abs/1702.08591>>.

Cruz et al. 2017CRUZ, R. A. Q. et al. Improving Accuracy of Automatic Fracture Detection in Borehole Images with Deep Learning and GPUs. In: **2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. Niteroi: IEEE, 2017. p. 345–350. ISBN 9781538622193. Disponível em: <<https://ieeexplore.ieee.org/document/8097332/>>.

Dias et al. 2020DIAS, L. O. et al. Automatic detection of fractures and breakouts patterns in acoustic borehole image logs using fast-region convolutional neural networks. **Journal of Petroleum Science and Engineering**, v. 191, 2020.

Goodfellow, Bengio e Courville 2016GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

Goutte e Gaussier 2005GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. **Lecture Notes in Computer Science**, p. 345–359, 2005.

Hall et al. 1996**Automatic Extraction And Characterisation Of Geological Features And Textures Front Borehole Images And Core Photographs**, All Days de **SPWLA Annual Logging Symposium**, (SPWLA Annual Logging Symposium, All Days). SPWLA-1996-CCC.

Ibtehaz e Rahman 2020IBTEHAZ, N.; RAHMAN, M. S. MultiResUNet : Rethinking the u-net architecture for multimodal biomedical image segmentation. **Neural Networks**, Elsevier BV, v. 121, p. 74–87, jan 2020. Disponível em: <<https://doi.org/10.1016/j.neunet.2019.08.025>>.

Jesus, Compan e Surmas 2016JESUS, C. de; COMPAN, A.; SURMAS, R. Permeability estimation using ultrasonic borehole image logs in dual-porosity carbonate reservoirs. **PETROPHYSICS**, v. 57, p. 620–637, 12 2016.

LeCun et al. 1998LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

Li et al. 2018LI, X.-N. et al. Automatic fracture–vug identification and extraction from electric imaging logging data based on path morphology. **Petroleum Science**, v. 16, n. 1, p. 58–76, 2018.

Lin et al. 2017LIN, T.-Y. et al. **Focal Loss for Dense Object Detection**. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1708.02002>>.

Long, Shelhamer e Darrell 2015LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2015. p. 3431–3440.

Lou, Guan e Loew 2020LOU, A.; GUAN, S.; LOEW, M. **DC-UNet: Rethinking the U-Net Architecture with Dual Channel Efficient CNN for Medical Images Segmentation**. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2006.00414>>.

Minaee et al. 2022MINAEE, S. et al. Image segmentation using deep learning: A survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 7, p. 3523–3542, 2022.

Refaeilzadeh, Tang e Liu 2009REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. In: _____. **Encyclopedia of Database Systems**. Boston, MA: Springer US, 2009. p. 532–538. ISBN 978-0-387-39940-9. Disponível em: <https://doi.org/10.1007/978-0-387-39940-9_565>.

Rocha e Azevedo 2007ROCHA, L. A.; AZEVEDO, C. **Projetos de poços de petróleo: geopressões e assentamento de colunas de revestimentos**. [S.l.]: Interciência: Petrobras, 2007.

Ronneberger, Fischer e Brox 2015RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. **CoRR**, abs/1505.04597, 2015. Disponível em: <<http://arxiv.org/abs/1505.04597>>.

Shafiabadi, Kamkar-Rouhani e Sajadi 2021SHAFIABADI, M.; KAMKAR-ROUHANI, A.; SAJADI, S. M. Identification of the fractures of carbonate reservoirs and determination of their dips from FMI image logs using Hough transform algorithm. **Oil & Gas Science and Technology - Revue d'IFP Energies nouvelles**, Institut Français du Pétrole (IFP), v. 76, p. 37, 2021. Disponível em: <<https://hal.science/hal-03252848>>.

Szegedy et al. 2016SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. **AAAI Conference on Artificial Intelligence**, v. 31, 02 2016.

Szegedy et al. 2014SZEGEDY, C. et al. **Going Deeper with Convolutions**. arXiv, 2014. Disponível em: <<https://arxiv.org/abs/1409.4842>>.

Szegedy et al. 2015SZEGEDY, C. et al. **Rethinking the Inception Architecture for Computer Vision**. arXiv, 2015. Disponível em: <<https://arxiv.org/abs/1512.00567>>.

Taiebi, Akbarizadeh e Farshidi 2017TAIEBI, F.; AKBARIZADEH, G.; FARSHIDI, E. Detection of reservoir fractures in imaging logs using directional filtering. In: . [S.l.: s.n.], 2017.

Valentin 2018VALENTIN, M. B. **Deep learning methods on geological reservoir borehole log images and applications**. Dissertação (Mestrado), 03 2018.

Xavier, Guerra e Andrade 2015XAVIER, A.; GUERRA, C. E.; ANDRADE, A. Fracture analysis in borehole acoustic images using mathematical morphology. **Journal of Geophysics and Engineering**, v. 12, n. 3, p. 492–501, 05 2015. ISSN 1742-2132. Disponível em: <<https://doi.org/10.1088/1742-2132/12/3/492>>.

Zhao et al. 2020ZHAO, X. et al. **Contrastive Learning for Label-Efficient Semantic Segmentation**. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2012.06985>>.

Zoback 2007ZOBACK, M. D. Compressive and tensile failures in vertical wells. In: _____. **Reservoir Geomechanics**. [S.l.]: Cambridge University Press, 2007. p. 167–205.

A

Lista de hiperparâmetros de treino

Todos os experimentos utilizaram a mesma parametrização de treino.

<i>Framework</i>	TensorFlow 2.1	
Tamanho do <i>batch</i>	16	
Função de perda	BinaryCrossentropy	
Otimizador	Adam	
	Epsilon	1.00E-07
	Taxa de aprendizado	1.00E-04
Monitor de <i>Early Stopping</i>	Erro de validação	
	Delta mínimo	0
	Épocas paciência	Exp 1, 2, 4: 50
		Exp 3: 20